PhD Thesis

Ahmad Muklason

Wednesday 17$^{\text{th}}$ May, 2017



The University of Nottingham

UNITED KINGDOM · CHINA · MALAYSIA

The University of Nottingham

Hyper-heuristics and Fairness in Examination Timetabling Problems

Ahmad Muklason

Thesis submitted to the University of Nottingham for the degree of Doctor of Philosophy

May, 2017

Abstract

Examination timetabling is a challenging optimisation problem in operations research and artificial intelligence. The main aim is to spread exams evenly throughout the overall time period to facilitate student comfort and success; however, existing examination timetabling solvers neglect fairness by optimising the sum or average of the objective function value without considering its distribution among students or other stakeholders. The balance between quality of the overall timetable and fairness (global fairness and within a cohort) is a major concern, thus the latter is added as a new objective function and quality indicator of examination timetables. The objective function is also considered from the perspectives of multiple stakeholders of examination timetabling (i.e. students, invigilators, markers and estates), as opposed to viewing the objective function as an aggregate function. These notions make the problem become a multi-objective optimisation problem.

We study sum of power rather than linear summation to enforce fairness and concurrently minimise the objective function, using some perturbation-based hyper- heuristics approaches to optimise the standard objective function. Secondly, multi-stage approach is studied (generating initial feasible solution, improving the standard quality of solution and then improving fairness), to improve the fairness objective function. Given that the standard objective function and fairness objective function conflict, we then studied several multi-objective algorithms employed within the framework of hyper-heuristics. The proposed hyper-heuristic algorithms mainly can be divided into two approaches: classical scalarisation technique-based weighted sum and Tchebyceff; and population-based non-dominated sorting memetic algorithm II (NSMA-II) and artificial bee colony and strength pareto evolutionary 2 (SPEA2) hybrid (ABC-SPEA2).

The experiments were conducted over two multi-objective examination timetabling problem formulations (i.e. with fairness and with multiple stakeholder perspectives), tested

over problem instances from four different datasets: Carter, Nottingham, Yeditepe and ITC 2007. The experimental results over multi-objective examination timetabling problem with fairness showed that in terms of the standard objective function the proposed approach could produce results comparable with the best known solutions reported in the literature, whilst in the same time could be forced to be fairer that does or does not compensate on worsening the standard objective function. Fairness within a cohort could be improved much better than global fairness and treating as multi-objective problem could help the search for near-optimal standard objective function escape from local optima trap. The scalarisation technique based hyper-heuristics outperforms the population-based hyper-heuristic.

The advantage of treating examination timetabling problem as multi-objective problem is that approximations of the Pareto optimal solutions give the optimal trade-off between standard objective function, fairness among all students, and fairness within a cohort. In addition, the decision maker also can view the solution from multiple stakeholders view. We believe that by giving this more detailed information, the decision maker of examination timetable could make better decisions.

Acknowledgement

I thank to my supervisor Dr Andrew J Parkes, and my co-supervisor Dr Ender Ozcan for their endless support and fruitful discussion in my PhD studies. I also deeply thank to Dr. Jingpeng Li and Dr Dario Landa Silva as my examiners who already gave very valuable feedbacks to improve the thesis. I also owe the debt to DIKTI, Higher Education Directorate Scholarship, the ministry of education and culture of the Republic of Indonesia, who sponsored the Ph.D. studentship. Last, but not least I thank to all members of ASAP group who already made very nice working environment during my PhD study. Overall, I deeply thank to Allah and my family: Lilik, Ilyas, and Alisa who always give me restless support all day and night.

Contents

Abstract i

Acknowledgement iii

List of Figures viii

List of Tables xi

1 Introduction 1

1.1 Research Background and Motivation 1

1.2 Aim and Objectives 2

1.2.1 Aim 2

1.2.2 Objectives 2

1.3 Research Contribution 3

1.4 List of Publications 3

1.5 Thesis Outline 4

2 Literature Review 6

2.1 Examination Timetabling Problem 6

2.1.1 Carter Dataset 8

2.1.2 ITC 2007 Dataset 10

2.1.3 Yeditepe and Other Datasets 13

2.2 Examination Timetabling Problem Approaches: state-of-the-art 14

2.3 Hyper-heuristics 18

2.4 Hyper-heuristics for Examination Timetabling Problems 22

2.5 Multi-Objective Optimisation Algorithm 27

2.6 Multi-Objective Examination Timetabling Problems 30

2.7 The Performance Assessment of Algorithms over Multi-Objective Problems 33

2.8 Fairness in General 34

2.9 Fairness in Timetabling Problems 36

3 Solving Benchmark Examination Timetabling Problems With a Hyper-heuristic 41

3.1 Introduction . . . 41
3.2 Solving The Uncapacitated Examination Timetabling Problems . . . 42
3.2.1 Method . . . 42
3.2.2 Experimental Environment . . . 46
3.2.3 Data Structure and Algorithm Implementation . . . 49
3.2.4 Parameter Setting . . . 49
3.2.5 Experimental Scenario . . . 49
3.2.6 Preliminary Experiment . . . 50
3.2.7 Experimental Result and Discussion . . . 51
3.3 Solving The Capacitated Examination Timetabling Problem . . . 53
3.3.1 Method . . . 54
3.3.2 Experimental Results and Discussion . . . 69
3.4 Conclusion . . . 80

4 The Extended Examination Timetabling Problem Formulation and 'Sum of Power' Approach 83

4.1 Introduction . . . 83
4.2 Extended Problem Formulation . . . 84
4.2.1 An Extension to Consider The Hardness of Examinations . . . 84
4.2.2 An Extension to Take Fairness Issues into Account . . . 85
4.2.3 Viewing the objective function of examination timetabling problem from multiple stakeholders . . . 89
4.3 'Sum of Powers' Approach to Enforce Fairness . . . 92
4.3.1 'Sum of Powers' Method . . . 92
4.3.2 Experimental Results . . . 94
4.4 Conclusion . . . 102

5 Multi-Stage Approach to Enforcing Fairness in Examination Timetabling Problems 104

5.1 Introduction . . . 104
5.2 Method: Two-Stage Approach . . . 105
5.3 Experimental Results . . . 106
5.3.1 Evaluating The Objective Functions Values of The Initial Solutions 106
5.3.2 Enforcing Fairness . . . 108
5.4 Conclusion . . . 118

6 Multi-objective Approach with Scalarisation Techniques 122

6.1 Introduction 122

6.2 Method: Multi-objective Hyper-heuristic With Classical Scalarisation Techniques 123

6.2.1 Extension of HyFlex 124

6.2.2 Hyper-heuristic Strategy 126

6.3 Preliminary Experiments 127

6.3.1 Experimental Results with The Best Parameter Setting over All Problem Instances 129

6.3.2 Comparing Reinforcement Learning and Self Adaptive learning as Low-level heuristic Selection Strategies 135

6.3.3 Comparing Two Different Scalarisation Techniques : Weighted Tchebycheff and Weighted Summation 136

6.3.4 Comparing Reinforcement Learning and Simple Random as Low-level Selection Strategy 136

6.4 Experimental Results Over The Multi-objective Optimisation Examination Timetabling Problem with Different Notions of Fairness 138

6.4.1 Experimental results over multi-objective examination timetabling problem with bi-objective functions: standard objective function and global unfairness objective function 141

6.4.2 Experimental results over multi-objective examination timetabling problem with bi-objective functions: standard objective function and 'specified cohort unfairness' objective function 143

6.4.3 Experimental results over multi-objective examination timetabling problem with bi-objective functions: standard objective function and 'exam-count-based cohort' unfairness objective function . . . 145

6.4.4 Experimental results over multi-objective examination timetabling problem with three objective functions: standard objective function, global unfairness objective function and 'specified cohort unfairness' objective function 147

6.4.5 Experimental results over multi-objective examination timetabling problem with three objective functions: standard objective function, global unfairness objective function and 'exam-count-based cohort unfairness' objective function 154

6.5 Experimental Results Over The Multi-objective Optimisation Problem with Multiple Stakeholder Perspectives 159

6.6 The Effect of Multi-objectivisation On The Benchmark Objective function 167

6.7 Deeper Understanding of the Multi-objective Nature of Examination Timetabling Problems . . . 171
6.7.1 Multi-objective Examination Timetabling Problems With Fairness 171
6.7.2 Multi-Objective Examination Timetabling Problems with Multiple Stakeholder Perspectives . . . 175
6.8 Conclusion . . . 176

7 Multi-objective Population-based Algorithm Hyper-heuristic Approach 179
7.1 Introduction . . . 179
7.2 Method: Population-based Hyper-heuristics for Multi-objective Optimisation Problem . . . 180
7.2.1 NSGA-II based Algorithm Hyper-heuristic . . . 180
7.2.2 Artificial Bee Colony-Based Hyper-heuristic (ABC-SPEA2) . . . 184
7.3 Experimental Results over Multi-objective examination timetabling problem with Fairness . . . 185
7.3.1 The Experimental Results with NSMA-II based Hyper-heuristic 188
7.3.2 Experimental Results with ABC-SPEA2 based Hyper-heuristic . 193
7.4 Experimental Results for Multi-Objective Examination Timetabling Problem with Multiple Stakeholder Perspectives . . . 195
7.4.1 Experimental Results with NSMA-II base Hyper-heuristic 197
7.4.2 Experimental Results with ABC-SPEA2 based Hyper-heuristic . 198
7.5 Conclusion . . . 203

8 Conclusions and Future Work 205
8.1 Research Summary . . . 205
8.2 Future Work . . . 211

Appendices 248

A Tables of Literature Survey 249

B Survey on Examination Timetable Preferences of University Students 255

C Detailed Experimental Result Data with 'Sum of Powers' Approach 263

D Complete Experimental Results with Classical Scalarisation Techniques-based Hyper-heuristics 271

List of Figures

2.1 The Basic Hyper-heuristics Framework 19
2.2 The Hyper-heuristic Classification 21

3.1 The objective function values during the iterations of the proposed hyper-heuristic over problem instance CAR-F-92 51
3.2 Objective function values during the search: comparison of the proposed SA-GD vs SA-HC-GD hyper-heuristic over problem instance CAR-F-92 54

4.1 Grouping the examination timetable's objective functions based on its Stakeholders. 91
4.2 Correlation between the number of iteration, standard objective function (OF) values, and anti Jain fairness index (AJFI) values. 96
4.3 The standard objective function values and unfairness objective function values during the iterations, with 'Sum of Powers' (SOP) objective function using different exponent numbers (q) 98

6.1 The scatter plot of the Pareto solution set from the experiments over the bi-objective problem with global unfairness. 142
6.2 The scatter plot: standard objective function and 'specified cohort unfairness'. 144
6.3 The scatter plot: standard objective function and 'exam-count-based cohort' unfairness. 146
6.4 Parallel coordinate of the Pareto solution set from the experiments over the three-objective problem. 149
6.5 Parallel coordinate of the selected solution in the Pareto solution set from the experiments over the three-objective problem of problem instance STA-F-83. 150
6.6 The parallel coordinate of problem instance EXAM9. 151

6.7 Penalty associated to each individual students of selected solution from the Pareto solution set with the three-objective functions over problem instance `STA-F-83`. 152
6.8 The penalty associated to each individual students over problem instance `EXAM9`. 153
6.9 Parallel coordinate representation of the Pareto solution set from the experiments over the three-objective problem with 'exam-count-based cohort unfairness' on four selected problem instances. 156
6.10 The parallel coordinate for problem instance `STA-F-83`. 157
6.11 Penalty associated to each individual students of selected solution from the Pareto solution set with the three-objective functions over problem instance `STA-F-83`. 158
6.12 The scatter plot of the Pareto solutions of problem instance `EXAM9` from the experiment of hyper-heuristics with different strategies tested over the bi-objective examination timetabling problem with multiple stakeholder perspective. 163
6.13 The parallel coordinate of the Pareto solutions from the experiment over the four-objective problem. 164
6.14 The scatter plot of the Pareto solutions from the experiment of SA-WSGD-HH with two different stopping conditions over the bi-objective problem. 165
6.15 The parallel coordinate of the Pareto solutions from the experiment of RL-WSGD-HH with two different stopping conditions over the four-objective problem. 165
6.16 The scatter plot of the Pareto solutions from the experiment of RL-WSGD-HH with two different stopping conditions over the four-objective problem. 166
6.17 The selected solutions of the Pareto solutions from the experiment of RL-WSGD-HH hyper-heuristic with two different stopping conditions over the four-objective problem. 167
6.18 The results of trade-off regions analysis over MOETP with fairness . . . 173
6.19 The results of multiobjective scatter-plot analysis over MOETP with fairness 174
6.20 The results of trade-off regions analysis over MOETP with multiple stakeholder perspectives 176
6.21 The results of multiobjective scatter-plot analysis over MOETP with multiple stakeholder perspectives 177

7.1 The scatter plot of the approximated Pareto optimal solution set from the experimentation of NSMA-II Algorithm with and without cross-over (XO) operator, over problem instance `EXAM9` over the bi-objective ITC examination timetabling problem. 182
7.2 The box plot visualising the objective function values of the Pareto optimal solution set from the experiment of NSMA-II algorithm over the bi-objective problem. 189
7.3 The box plot of the objective function values of the Pareto optimal solution set from the experiment of NSMA-II algorithm over the bi-objective problem. 191
7.4 The scatter plot of the objective function values of the Pareto optimal solution set from the experiment of the three different hyper-heuristics. 193
7.5 The box plot of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on ABC-SPEA2 (S1) and NSMA-II (S2) over the bi-objective problem. 196
7.6 The scatter plot of the Pareto optimal solution set from the experiment of two different hyper-heuristics: NSMA-II (S1) and SA-WSGD-HH (S2) over Problem instance `EXAM10` of the bi-objective problem. 199
7.7 The scatter plot of the Pareto optimal solution set from the experiment of two different hyper-heuristics: NSMA-II (S1) and SA-WSGD-HH (S2) over the other four problem instances of the bi-objective problem. 199

List of Tables

2.1 The characteristics of problem instances from Carter dataset 9
2.2 The characteristics of problem instances from ITC 2007 examination timetabling problem dataset 12
2.3 The associate weight ITC 2007 examination timetabling problem dataset 12
2.4 The characteristics of problem instances from Yeditepe benchmark datasets 13
2.5 The problem instance properties of the other examination timetabling problems benchmark datasets 13
2.6 The Application of Hyper-heuristics 21
2.7 The Objective Functions of Nine-Criteria Examination Timetabling Problem 31
2.8 Summary of Fairness Quality Indicator 36

3.1 Low-level heuristics (LLH) within the uncapacitated examination timetabling problem domain 43
3.2 List of variables used in Algorithm 2, Algorithm 3 and Algorithm 4 . . . 44
3.3 List of methods used in Algorithm 2, Algorithm 3 and Algorithm 4 . . . 45
3.4 Combinations of two low-level heuristic selection strategies. 50
3.5 Comparing differing values of parameter L in the late acceptance strategy 50
3.6 Experimental results over Carter Dataset (standard objective function) using combination different low-level heuristic selection and move acceptance strategies. 52
3.7 Comparing SA-GD and SA-LA approaches with results reported in the selected recently published scientific literature. 52
3.8 Comparing The performance of SA-GD versus SA-HC-GD, from 11 runs within 3000000 iterations each 53
3.9 List of Basic Graph-Colouring Based Ordering Heuristic 56
3.10 List of Hybrid Ordering Heuristic 57
3.11 List of variables used in Algorithm 5 57

3.12 List of methods used in in Algorithm 5 57
3.13 List of variables used in Algorithm 6 59
3.14 List of methods used in Algorithm 6 60
3.15 Perturbation Low-Level Heuristics (LLHs) for Examination Timetabling Problems 64
3.16 List of variables used in Algorithm 7- 9 65
3.17 List of methods used in Algorithm 7 - 9 66
3.18 The characteristics of Nottingham dataset in the ITC 2007 format . . . 69
3.19 Results obtained by the AHOHHGH and AOH-SWO initial solution construction methods over 10 runs for the Carter, ITC 2007, Yeditepe, and Nottingham benchmark instances. 70
3.20 Experimental results with three different hyper-heuristics strategies: RL-GD-HH, RL-XGD-HH, and RL-MXGD-HH. 72
3.21 Result obtained using combination of AOH-SWO and RL-XGD-HH compared with the best result reported in scientific literatures 74
3.22 The strategy within our Hyper-heuristics (HH) framework 75
3.23 The Experimental Results with three different strategies: SR-GD-HH, RL-GD-HH, and SA-GD-HH. 76
3.24 The Experimental Results of Comparing Three Different Strategies: SR-XGD-HH, RL-XGD-HH, and SA-XGD-HH. 77
3.25 The Experimental Results of Comparing Three Different Hyper-heuristic Strategies: SR-MXGD-HH, RL-MXGD-HH, and SA-MXGD-HH. 78
3.26 The Experimental Results of Comparing: Hyper-heuristic Strategies summarised in Table 3.22 in terms of the best solutions 79
3.27 The experimental results over ITC 2007 datasets with the standard single objective function from three variant of hyper-heuristics: SR-GD-HH, RL-GD-HH, and SA-GD-HH. 81

4.1 Illustration Example 1 88
4.2 Illustration Example 2 89
4.3 The number of cohort of the problem instances based on the specified cohort and exam-count-based Cohort 90
4.4 Variable Description for Equation 4.16- 4.18 93
4.5 An Illustrating Example of two solutions with different penalty distribution and their respective different objective functions: Linear summation (LS), Sum of Squares (SOS), and min-max (Max) 93

4.6 Correlation between different fairness quality indicators and the standard benchmark objective function. 94
4.7 Average computing time of different quality indicator of fairness index. . 95
4.8 Correlation between number of iteration, standard objective function values, and Anti Jain fairness Index (AJFI) values. 97
4.9 The trade-off between efficiency, i.e. standard objective function (std.OF) and unfairness (AJFI) of the solutions produced with standard objective function vs 'Sum of Square'. 99
4.10 The trade-off between efficiency, i.e. standard objective function (SOF) and unfairness (AJFI) of the solutions produced with SA-GD hyper-heuristics using different objective functions: standard objective function vs 'Sum of Square'. 100
4.11 The trade-off between efficiency and unfairness of the solutions produced with SA-GD hyper-heuristics using three different objective functions. . 100
4.12 The average objective function values: the trade-off between efficiency and unfairness of the solutions produced with SA-GD Hyper-heuristics using three different objective functions. 101
4.13 Comparison of computation time between standard objective function and 'Sum of Powers' with `q=2` and `q=16`. 102

5.1 The Objective Functions' Values, i.e. Standard Objective Function (STD.OF), Global Unfairness, Specified Cohort Unfairness, and Exam-Count-Based Cohort Unfairness, of The Initial Solutions. 107
5.2 Experimental Results From Enforcing Global Fairness With RL-GD-HH and Deficiency <=5%. 110
5.3 Experimental Results From Enforcing Global Fairness With RL-HC-HH and Deficiency <=0%. 111
5.4 Experimental Results From Enforcing Global Fairness With RL-HC-HH and Deficiency <=5%. 112
5.5 Experimental Results From Enforcing Global Fairness With RL-LEX-HH Deficiency <=0%. 113
5.6 Experimental Results From Enforcing Specified Cohort Fairness With RL-HC-HH and Deficiency <=0%. 115
5.7 Experimental Results From Enforcing Specified Cohort Fairness With RL-HC-HH and Deficiency <=5%. 116
5.8 Experimental Results From Enforcing Specified Cohort Fairness With RL-LEX-HH and Deficiency <=5%. 117

5.9 Experimental Results From Enforcing Exam-count-based Cohort Fairness With RL-HC-HH and Deficiency <=0%. 118
5.10 Experimental Results From Enforcing Exam-count-based Cohort Fairness With RL-HC-HH and Deficiency <=5%. 119
5.11 Experimental Results From Enforcing Exam-count-based Cohort Fairness With RL-LEX-HH and Deficiency <=0%. 120

6.1 List of methods added to `ProblemDomain` interface in HyFlex to deal with multi-objective problems with scalarisation technique 125
6.2 List of methods as part of Hyper-heuristic strategy given in Algorithm 11 127
6.3 The best parameter settings for the experiments over the bi-objective problem using Algorithm 11. 130
6.4 Experimental Scenarios of Pareto Scanning 130
6.5 The range of objective function values of the Pareto solution set from the experiments with different Pareto scanning scenarios. 131
6.6 The solution with the best standard objective function values in the Pareto solution set. 132
6.7 The best solution in terms of global fairness function values in the Pareto solution set. 133
6.8 The quality metric: coverage, hypervolume, and Spread of the Pareto solution set. 134
6.9 The parameter settings for the experiments to compare reinforcement learning and self adaptive learning, over bi-objective examination timetabling problem with fairness. 136
6.10 Experimental Scenarios of Low-level heuristic (LLH) selection strategy . 136
6.11 Coverage, hypervolume, and Spread of the Pareto solution set from the experiments of a hyper-heuristic with reinforcement learning (S1) and self-adaptive learning (S2) using 'Weighted Tchebycheff'. 137
6.12 The parameter settings for the experiments to compare 'weighted sum' and 'weighted Tchebycheff'. 138
6.13 Experimental Scenarios Comparing two Scalarisation Techniques 138
6.14 Coverage, hypervolume and Spread of the Pareto solution set from the experiments using 'Weighted Sum' (S1) and 'Weighted Tchebycheff' (S2). 139
6.15 The parameter settings for the experiments to compare reinforcement learning and simple random. 139
6.16 Experimental Scenarios Comparing Reinforcement Learning and Simple Random 140

6.17 Coverage, hypervolume, and Spread of the Pareto solution set from the experiments of a hyper-heuristic with reinforcement learning (S1) and simple random (S2) using 'Weighted Summation'. 140
6.18 The best parameter settings for experiments over bi-objective examination timetabling problem: (1) Standard objective function, (2) Global fairness using Algorithm 11 141
6.19 The objective function values range of the Pareto solution set from the experiments over the bi-objective problem with global unfairness. 142
6.20 The objective function values of the solution with the best standard objective function and the best global fairness in the Pareto solution set from the experiments over the bi-objective problem with global unfairness.142
6.21 The objective function values range of the Pareto solution set from the experiments over the bi-objective problem with 'specified cohort unfairness'.143
6.22 The objective function values of the solution with the best standard objective function and the solution with the best global fairness in the Pareto solution set from the experiments over the bi-objective problem with 'specified cohort unfairness'. 144
6.23 The objective function values range of the Pareto solution set from the experiments over the bi-objective problem with 'exam-count-based cohort' unfairness. 146
6.24 The objective function values of the solution with the best standard objective function and the best global fairness in the Pareto solution set from the experiments over the bi-objective problem with 'exam-count-based cohort' unfairness. 146
6.25 The objective function values range of the Pareto solution set from the experiments over three-objective examination timetabling problem: standard objective function, global unfairness and 'specified cohort unfairness'.147
6.26 The objective function values of the solution with the best standard objective function in the Pareto solution set from the experiments over the three-objective problem. 148
6.27 The objective function values of the solution with the best global fairness in the Pareto solution set from the experiments over the three-objective problem. 148
6.28 The objective function values of the solution with the best specified cohort fairness in the Pareto solution set from the experiments over the three-objective problem. 148

6.29 The objective function values of the selected solution over problem instance `STA-F-83` from the Pareto solution set given in Figure 6.5 151
6.30 The changes in objective function values (Δ) from solution `0` to the other solutions of selected solutions over problem instance `STA-F-83` given in Table 6.29 151
6.31 The Objective Function Values of selected solution over problem instance `EXAM9` from Pareto solution set given in Figure 6.6 151
6.32 The changes in objective function values (Δ) from solution `0` to the other solutions of selected solutions over problem instance `EXAM9` given in Table 6.31 151
6.33 The objective function values range of the Pareto solution set from the experiments over three-objective examination timetabling problem: standard objective function, global unfairness and 'exam-count-based cohort unfairness'. 155
6.34 The objective function values of the solution with the best standard objective function in the Pareto solution set from the experiments over the three-objective problem with 'exam-count-based cohort unfairness'. . . . 155
6.35 The objective function values of the solution with the best global fairness in the Pareto solution set from from the experiments over the three-objective problem with 'exam-count-based cohort unfairness'. 155
6.36 The objective function values of the solution with the best exam-count-based cohort fairness in the Pareto solution set from the experiments over the three-objective problem with 'exam-count-based cohort unfairness'. . 155
6.37 The objective function values of the selected solution over problem instance `STA-F-83` with the three-objective functions with 'exam-count-based cohort unfairness', from Pareto solution set given in Figure 6.10 . 157
6.38 The changes in objective function values (Δ) from solution `0` to the other solutions of selected solutions over problem instance `STA-F-83` with the three-objective functions with 'exam-count-based cohort unfairness', given in Table 6.37 157
6.39 Four variants of hyper-heuristics scenarios 160
6.40 The parameter settings for experiment with Hyper-heuristics presented in Algorithm 11 and Algorithm 12 over multi-objective examination timetabling problem with multiple stakeholders' perspective. 160
6.41 Coverage Definition given in Table 6.42 and Table 6.43 160

6.42 The quality indicators of the Pareto solutions from the experiment of hyper-heuristics with different strategies tested over the bi-objective examination timetabling problem with multiple stakeholder perspective. . 161
6.43 The quality indicators of the Pareto solutions from the experiment of hyper-heuristics with different strategies tested over the four-objective examination timetabling problem with multiple stakeholder perspective. 161
6.44 Comparing the best solutions from different approaches: single objective with RL-GD/XGD/MXGD-HH versus two Objectives with RRL-WSGD/WTGD-HH. 168
6.45 Comparing the best solutions from different approaches: single objective with RL-GD/XGD/MXGD-HH versus two Objectives with RL-WSGD/WTGD-HH. 169
6.46 Comparing the best standard objective function value from the experiment with `RL-WSGD-HH` over the single objective problem and the multiple objective problems. 170
6.47 Solutions from our multi-objective approach compared to the best-known solutions 171
6.48 The results for the global pairwise relationship analysis over MOETP with fairness 172
6.49 The results of objective range analysis over MOETP with fairness . . . 172
6.50 The results for the global pairwise relationship analysis over MOETP with multiple stakeholder perspectives 175
6.51 The results of objective range analysis over MOETP with multiple stakeholder perspectives 175

7.1 List of local search operators. 182
7.2 The experimental scenarios over NSMA-II based hyper-heuristic to evaluate the effect of initial population and the number of generation. . . . 188
7.3 The parameters setting of the experiment over NSMA-II based hyper-heuristic to evaluate the effect of initial population and the number of generation. 188
7.4 The objective function values of the Pareto optimal solution set from the experiment of NSMA-II over bi-objective examination timetabling problem: standard objective function and global fairness. 189
7.5 The coverage metric of the Pareto optimal solution set from the experiment of NSMA-II algorithm over the bi-objective problem. 189

7.6 The other multi-objective metric of the Pareto optimal solution set from the experiment of NSMA-II algorithm over the bi-objective problem. . . 190
7.7 The experimental scenarios over NSMA-II algorithm to evaluate the effectiveness of strategy for adding the new solutions produced after applying mutation and local search operator to the offspring population. 190
7.8 The parameters setting of the experiment over NSMA-II algorithm to evaluate the effectiveness of strategy for adding the new solutions produced after applying mutation and local search operator to the offspring population. 190
7.9 The objective function values of the Pareto optimal solution set from the experiment of NSMA-II algorithm over the bi-objective problem. 191
7.10 Coverage metric of the objective function values of the Pareto optimal solution set from the experiment of NSMA-II algorithm over the bi-objective problem. 191
7.11 Other multi-objective Metric of The objective function values of the Pareto optimal solution set from the experiment of NSMA-II algorithm over the bi-objective problem. 191
7.12 The objective function values of the Pareto optimal solution set from the experiment of three different hyper-heuristics based on: NSMA-II (S1), weighted sum scalarisation (S2), and weighted Tchebycheff scalarisation (S3) over the bi-objective problem. 192
7.13 The coverage metric of the Pareto optimal solution set from the experiment with the three different hyper-heuristics. 192
7.14 The other metrics of the Pareto optimal solution set from the experiment of the three different hyper-heuristics. 192
7.15 The experimental scenarios over ABC-SPEA2 based hyper-heuristic to evaluate the effect of initial population and the number of stopping condition. 194
7.16 The parameters setting of the experiment over ABC-SPEA2-based hyper-heuristic to evaluate the effect of initial population and the number of generation. 194
7.17 The objective function values of the Pareto optimal solution set from the experiment with ABC-SPEA2 over bi-objective examination timetabling problem: standard objective function and global fairness. 194
7.18 The coverage metric of the Pareto optimal solution set from the experiment with ABC-SPEA2 over the bi-objective problem. 195

7.19 The other metrics of the Pareto optimal solution set from the experiment with ABC-SPEA2 over the bi-objective problem. 195
7.20 The objective function values of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on ABC-SPEA2 (S1) and NSMA-II (S2) over the bi-objective problem. 195
7.21 The coverage metric of of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on ABC-SPEA2 (S1) and NSMA-II (S2) over the bi-objective problem. 196
7.22 The other metrics of of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on ABC-SPEA2 (S1) and NSMA-II (S2) over the bi-objective problem. 196
7.23 The parameters setting of the experiment over NSMA-II based hyper-heuristic over multi-objective examination timetabling problem with multiple stakeholder perspectives. 197
7.24 The quality metric of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on NSMA-II (S1) and SA-WSGD-HH (S2) over the bi-objective problem. 198
7.25 The quality metric of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on NSMA-II (S1) and SA-WSGD-HH (S2) over the four-objective problem. 198
7.26 The parameters setting of the experiment of ABC-SPEA2-based hyper-heuristic over multi-objective examination timetabling problem with multiple stakeholder perspectives. 200
7.27 The coverage metric of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on ABC-SPEA2 (S1) and NSMA-II(S2) over the bi-objective problem. 200
7.28 The coverage metric of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on ABC-SPEA2 (S1) and NSMA-II (S2) over the four-objective problem. 200
7.29 The other quality metrics of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on ABC-SPEA2 (S1) and NSMA-II (S2) over the bi-objective problem. 201
7.30 The other quality metrics of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on ABC-SPEA2 (S1) and NSMA-II(S2) over the four-objective problem. 201

7.31 The quality metric of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on ABC-SPEA2 (S1) and SA-WSGD-HH (S2) over the bi-objective problem. 201
7.32 The quality metric of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on ABC-SPEA2 (S1) and SA-WSGD-HH (S2) over the four-objective problem. 201
7.33 The coverage metric of the Pareto optimal solution set from the experiment of ABC-SPEA2 based hyper-heuristic with two scenarios of stopping generation over the bi-objective problem. 202
7.34 The coverage metric of the Pareto optimal solution set from the experiment of ABC-SPEA2 based hyper-heuristic with two scenarios of stopping generation over the four-objective problem. 202
7.35 The other metrics of the Pareto optimal solution set from the experiment of ABC-SPEA2 based hyper-heuristic with two scenarios of stopping generation over the bi-objective problem. 202
7.36 The other metrics of the Pareto optimal solution set from the experiment of ABC-SPEA2 based hyper-heuristic with two scenarios of stopping generation over the four-objective problem. 203

Chapter 1

Introduction

1.1 Research Background and Motivation

The examination timetabling problem is one of many difficult real-world combinatorial optimisation problems proven NP-hard. Examination timetabling itself has attracted researchers for over five decades, and has inspired many approaches to solve the problem. These approaches include a variety of approaches from traditional graph-colouring to hyper-heuristic techniques. However, we believe that the state-of-the-art, real-world scheduling problem formulations solved with current approaches in the research community are over simplified. This leads to a wide gap between research and real application in the real world. The solution considered best in the research is not guaranteed to be the best in real-world applications.

In addition, hyper-heuristics have attracted many researchers in different research areas, especially in operations research and AI, for decades. Many real-world combinatorial problems, including the timetabling and scheduling problems, have been successfully solved with hyper-heuristics. The strength of hyper-heuristics against advanced meta-heuristic approaches is its generality and independence from the nature of the problem domains in hand. Rather than producing an optimal solution for a specific problem, hyper-heuristics are used to produce a good-enough solutions over cross-problem domains. Hence, hyper-heuristics are a promising and rich research area.

This work aims to extend advanced examination timetabling problems with real-world structures, and then propose hyper-heuristic approaches to solve the new problem formulation. Having better representation of the real-world structures of examination timetabling problem formulation, it is hoped that it will be more applicable and prefer-

able in the real-world scenario rather than the existing simplified model currently studied in the scientific literature.

One of the important real-world issues that needs to be considered is the fairness issue. In real-world problems, since equality or fairness among stakeholders of the problems is a sensitive issue, it should be taken into account seriously. Otherwise, there will be dissatisfaction or even rejection against the solution from the stakeholders. In other words, considering fairness in solving a particular problem will produce not only the best but also the most sensible solution. Therefore, this study focuses on investigating the fairness issue in examination timetabling problems.

Considering fairness in examination timetabling problems is very important in order to maintain student satisfaction in the university. As examinations are the most critical part of evaluation that determines the success of students, providing a good and fair examination timetable is critical; bad or unfair examination timetables could affect student satisfaction within a university.

1.2 Aim and Objectives

1.2.1 Aim

The aim of the research project is to answer the following research questions:

1. What kind of new examination timetabling problem models are more oriented toward students preference and address the issue of fairness among students?
2. What kind of algorithms are effective to solve the new models?

1.2.2 Objectives

Overall, the objectives of the study reported in this thesis are:

1. To investigate the real-world examination timetabling structures, critical issues, and other aspects of the current examination timetabling problems, especially the fairness issue.
2. To extend the existing examination timetabling problems model with the real-world structures.
3. To propose hyper-heuristic approaches in order to deal with the new problem formulations.

1.3 Research Contribution

In general, the main contribution of this research is that it is the first study investigating fairness in examination timetabling from the perspective of operations research. Specifically, the contributions are listed as below:

- Results of a survey amongst some undergraduate and taught postgraduate students on their own preferences for the properties of exam timetables confirm the expectation that fairness is a concern for them, particularly within their cohort.
- Suggesting a new examination timetabling problem model that is more student oriented and more representative to the real-world examination timetabling problems, by taking into consideration fairness issue as well as the facts found from the survey, especially regarding the fact that the exams have different difficulty levels and students expect more gaps before harder exams.
- A sum of powers and multi-stage approach to enforcing fairer solutions in multi-objective examination timetabling problem with fairness.
- Proposal of multi-objective hyper-heuristic approaches based on two classical scalarisation techniques: weighted sum and weighted Tchebycheff for solving multi-objective optimisation problem.
- Proposal of multi-objective hyper-heuristic approaches based on population-based algorithms: non-dominated sorting memetic algorithmic - II (NSMA-II), which is the hybridisation of NSGA-II and memetic algorithm, and ABC-SPEA2, which is hybridisation of artificial bee colony and strength pareto evolutionary 2 (SPEA2) algorithms for solving multi-objective optimisation problem.
- Better understanding on the benchmark standard single objective examination timetabling problems and the proposed multi-objective examination timetabling problems: multi-objective examination timetabling problem with fairness and multi-objective examination timetabling problem with multiple stakeholders perspective. In particular, we find that it is easier to improve cohort fairness than global fairness.

1.4 List of Publications

Below are some publications published in some conferences and submitted into journals.

1. Muklason A., Parkes A.J., McCollum B. and Ozcan E. Cohort Fairness in Examination Timetabling Problems, PATAT 2016, 11th International Conference on The Practice and Theory of Automated Timetabling (PATAT 2016), 23-27 August 2016, Udine, Italy, 2016. (Chapter 6 and Chapter 7).
2. Muklason A., Parkes A.J., McCollum B. and Ozcan E. Hyper-heuristic Approaches for Multi-Objective Examination Timetabling Problems With Fairness, to be submitted to Information Sciences.(Chapter 4 - 7).
3. Muklason A., Parkes A.J., McCollum B. and Ozcan E. Hyper-heuristics for Solving Multi-objective ITC 2007 Examination Timetabling Problem with multiple stakeholders perspective, to be submitted to EJOR.(Chapter 6 and Chapter 7).
4. Muklason A., Parkes A.J., McCollum B. and Ozcan E. Fairness in Examination Timetabling Problems, accepted for publication to Journal of Applied Soft Computing, 2017. DOI:10.1016/j.asoc.2017.01.026 (Appendix B, Chapter 4, and Chapter 6).
5. Muklason A., Parkes A.J., McCollum B. and Ozcan E. Fairness in Examination Timetabling Problems: A Survey and the New Problem Formulation, PATAT 2014, 10th International Conference on The Practice and Theory of Automated Timetabling (PATAT 2014), 26-29 August 2014, York, UK. (Appendix B and Chapter 4).
6. Muklason, A., Parkes, A. J., McCollum, B., & Ozcan, E. (2013). Initial Results on Fairness in Examination Timetabling. Multidisciplinary International Scheduling Conference: Theory and Application 2013 (MISTA 2013), 27-29 August 2013, Ghent, Belgium. (Chapter 4).

1.5 Thesis Outline

The structure of this thesis is organised as follows.

Firtsly, Chapter 1 presents the motivation, aims, objectives, as well as the claimed contribution of research to give rationale why this research is important.

To explain the research background and highlight the gap between the existing research in the literature and the research elaborated in this thesis. Chapter 2 surveys background knowledge from prior works in the related area focused on the related timetabling problems, especially examination timetabling problem, the approaches to solve the problems,

multi-objective optimisation problems, as well as the central issue of this research: fairness in related combinatorial optimisation problems.

Prior to addressing fairness, Chapter 3 investigates hyper-heuristic approaches to solve the benchmark examination timetabling problems with their standard problem formulation.

Further, based on survey findings reported in Appendix B, Chapter 4 discusses the proposed extension of examination timetabling problem formulation, especially taking fairness into account. The first attempt to enforce fairness using 'sum of powers' is also discussed in this chapter.

As the second attempt to enforce fairness, Chapter 5 presents our study in investigating multi-stage approaches. It is then followed by Chapter 6 that reports our study in solving multi-objective examination timetabling problem with complex fairness using hyper-heuristics based on classical scalarisation techniques (i.e. weighted sum and weighted Tchebycheff). Instead of considering 'whole' global fairness as second objective in addition to standard objective function, complex fairness notions yielding many objective optimisation problems are discussed.

Lastly, Chapter 7 presents hyper-heuristics based on population based algorithms, i.e. non-dominated sorting memetic algorithm - II (`NSMA-II`) and hybridisation of artificial bee colony and strength pareto evolutionary algorithm 2 `ABC-SPEA2` for solving the multi-objective examination timetabling problems. Finally, Chapter 8 concludes the thesis by highlighting the most important findings of the research and some recommendations for future work.

Chapter 2

Literature Review

This chapter elaborates the fundamental concepts as well as the state-of-the-art approaches of the examination timetabling problems as well as the related problems. The fundamental theoretical background and the current research development to tackle the problems are presented afterwards. Furthermore, a survey of fairness issues, especially those in related with combinatorial optimisation problems is given to set up the background of research.

2.1 Examination Timetabling Problem

Timetabling is an interesting problem that has been studied intensively for 4 decades or so in intertwined research areas, especially in the field of operations research and artificial intelligence. Timetabling problems are instances of combinatorial optimisation problems class which is defined By Lawler [184] as:

> "Mathematical study of finding an optimal arrangement, grouping, ordering, or selection of discrete objects usually finite in numbers."

In the literature, term timetabling could make confusion with other terms such as scheduling, sequencing and rostering. In order to make the different clearer, Wren in [325] used a more restrictive definition for these terms as follows:

> "*Scheduling* is the allocation, subject to constraints, of resources to objects being placed in space-time, in such a way as to minimise the total cost of some set of the resources used. Common examples are transport scheduling or delivery vehicle routing which seek to minimise the numbers of vehicles or drivers and within that minimum to minimise the total cost, and job shop

scheduling which may seek to minimise the number of time periods used, or some physical resource.

Timetabling is the allocation, subject to constraints, of given resources to objects being placed in space-time, in such a way as to satisfy as nearly as possible a set of desirable objectives. Examples are class and examination timetabling and some forms of personnel allocation, for example manning of toll booths subject to a given number of personnel.

Sequencing is the construction, subject to constraints, of an order in which activities are to be carried out or objects are to be placed in some representation of a solution. Examples are flow-shop scheduling and the travelling salesman problem.

Rostering is the placing, subject to constraints, of resources into slots in a pattern. One may seek to minimise some objective, or simply to obtain a feasible allocation. Often the resources will rotate through a roster. ”

From the above definitions, we can observe that the difference lies on the purposes. However, as Wren is aware of, some problems could fit in more than one these definitions and in the real-world the definitions might be different as well.

More specifically, Burke et al. [64] defines timetabling problem as:

“A timetabling problem is a problem with four parameters: T, a finite set of times; R, a finite set of resources; M, a finite set of meetings; and C, a finite set of constraints. The problem is to assign times and resources to the meetings so as to satisfy the constraints as far as possible.”

Furthermore, Schaerf [281] classified timetabling problems into three subclasses, namely school timetabling problems, course timetabling problems, and examination timetabling problems. Similarly, as he noticed, there is no strict distinction among these three problems. A particular problem could fall between two classes of them.

To convey the nature of examination timetabling problem, we can view the problem from different perspectives. For example, examination timetabling problems could be considered as resource allocation problems, search and optimisation problems, as well as constraint satisfaction problems.

From constraint satisfaction problem (CSP) as well as search and optimisation problem perspective, Burke et al. in [63] represents examination timetabling as a 3-tuple $< E, I, K >$, where E is a set of examinations, I is resource domain e.g. time period, and K is a set of constraints as shown by equation 2.1 and 2.2.

$$E = \{e_1, e_2, ..., e_N\} \tag{2.1}$$

$$K = \{k_1, k_2, ..., k_M\} \tag{2.2}$$

If $T = \{t_1, t_2, ..., t_L\}$ is an ordered list of possible timeslot for an examination. The exams timetabling problem can be defined as a searching problem for the best assignment $\forall_x$, $\exists_y(e_x = t_y)$ which means exam e_x is scheduled at timeslot t_y, where $e_x \in E$ and $t_y \in T$, such that, the given constraints are satisfied.

The common objective function of examination timetabling problems is to satisfy all hard constraints and minimise penalties with regard to soft constraints that vary from institution to institution. These varying constraints have been been investigated in [56]. A solution that satisfies all hard constraints is referred as a feasible solution. If a solution also satisfy all soft constraints, i.e. the violation penalty equal to zero, it is referred as a perfect solution. However, an optimal solution is not necessarily a perfect solution. For instance in the uncapacitated examination timetabling problem context, in an optimal solution it is not necessary that each student got penalty i.e. the gap between two exams is at least 5 timeslots.

In the sense of computational complexity theory, examination timetabling problems are classified in the class of NP-complete problems [136, 102, 171]. For NP-complete problems there is no known polynomial time exact algorithm to find an optimal solution.

In order to provide standard examination timetabling problems formulation and dataset including the real-word problem ones in examination timetabling research, some previous works have shared benchmark datasets. Two widely used benchmark datasets that have been used by many researchers are the Carter dataset [89] and International Timetabling Competition 2007 (ITC 2007) dataset [205, 207]. The details of these datasets are discussed in the following subsections.

2.1.1 Carter Dataset

The Carter dataset also known as the University of Toronto data set was firstly introduced by Carter et. al. in 1996 [89]. It consists of a set of 13 real-world exam timetabling problems from 3 Canadian highs schools, 5 Canadian, 1 American, 1 UK and 1 mid-East universities. It then became a standard in uncapacitated examination timetabling

problem. These 13 real-world exam timetabling problems are concisely summarised by Table 2.1.

Table 2.1: The characteristics of problem instances from Carter dataset

Problem Instances	No.Of Exams	No.Of Students	Conflict Density	No.Of Timeslots
CAR91	682	16925	0.13	35
CAR92	543	18419	0.14	32
EAR83	190	1125	0.27	24
HEC92	81	2823	0.42	18
KFU93	461	5349	0.06	20
LSE91	381	2726	0.06	18
PUR93	2419	30029	0.03	42
RYE92	486	11483	0.07	23
STA83	139	611	0.14	13
TRE92	261	4360	0.18	23
UTA92	622	21266	0.13	35
UTE92	184	2749	0.08	10
YOR83	181	941	0.29	21

Each column in Table 2.1 respectively refers to the name of problem, number of examination, number of students, number of enrolments, conflict density, and number of timeslot in each problem. The conflict density calculates the ratio between the number of elements of value "1" to the total number of elements in the conflict matrix. In a conflict matrix C, each element $C_{ij} = 1$ if exam i conflict, i.e. have at least one common student, with exam j or $C_{ij} = 0$ otherwise. Suppose N is the number of examinations, the conflict density is formally defined by equation 2.3.

$$\texttt{Conflict Density} = \frac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}(\frac{C_{ij}}{N})}{N} \tag{2.3}$$

The best-known solution for each problem in this dataset is reported and shared in [254]. In the Carter dataset the examination timetabling problem is simplified. The only hard constraint taken into account is the exam conflict, in which two exams that share students must not occur in the same period. In other words, there is no student that sits more than one examination at the same time. In addition, a soft constraint namely 'period spread' which means 2 exams that share students should be a number of periods apart, is also considered. Therefore, in this dataset the objective of examination timetabling is to minimise the sum of proximity costs per student by trying to make spread of examinations for a given student as wide as possible. It implies if a student is sitting one exam followed immediately by another, the penalty is greater than if the student has an exam, a rest period, and then another examination. If the student has two rest periods, this is even better, and so on, until there are five rest periods. If the

gap is more than 5 timeslots, then the penalty is zero. Formally, the proximity cost, P, is defined in Equation 2.5 below.

$$P = \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} C_{ij} W_{|t_j - t_i|}}{S} \tag{2.4}$$

where

$$\begin{aligned} W_{|t_j - t_i|} &= 2^{5-|t_j - t_i|} \text{ when } 1 \leq |t_j - t_i| \leq 5 \\ &\quad 0 \text{ otherwise} \end{aligned} \tag{2.5}$$

and subject to the hard constraint that no student has two exams at the same time:

$$\forall_{i \neq j}. \; t_i \neq t_j \text{ when } C_{ij} > 0 \tag{2.6}$$

Where N is the number of examinations, C_{ij} is the number of students enrolled in both examination i and j while t_i is the assigned time-slot for examination $i, W_{|t_j - t_i|}$ is the weight whenever a student who sit for two examinations are scheduled $\mid t_j - t_i \mid$ apart and S is the number of students. The penalty weight, $W_{|t_j - t_i|}$ is calculated as $2^{5-|t_j - t_i|}$ where, $\mid t_j - t_i \mid \in \{1, 2, 3, 4, 5\}$.

From student point of view, Equation 2.4 can be modified by Equation 2.7 as shown below. Where X_{pi} and X_{pj} are binary variables with value 1 if student p has to sit for exam i/j.

$$P = \frac{\sum_{p=1}^{S} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} X_{pi} X_{pj} W_{|t_j - t_i|}}{S} \tag{2.7}$$

2.1.2 ITC 2007 Dataset

Different from the carter dataset where the examination timetabling problem is simplified, ITC 2007 Dataset takes into account the complexity of real-world examination timetabling problem. The complexity of this problem is represented by the following hard and soft constraints.

The hard constraints consist of:

1. Conflicts: two conflicting exams in the same period.

2. Room occupancy: more seating required in any individual period than that available.
3. Period utilisation: more time required in any individual period than that available.
4. Period related: ordering requirements are not obeyed.
5. Room related: room requirements are not obeyed.

Formally, subject to all hard constraints are satisfied, the objective function is to minimise total penalty as result of soft constraints violation defined by Equation 2.8.

$$P = \sum_{s \in S} (w^{2R} C_s^{2R} + w^{2D} C_s^{2D} + w^{PS} C_s^{PS}) + w^{NMD} C^{NMD} + w^{FL} C^{FL} + C^P + C^R \quad (2.8)$$

For each student s, the total penalty is summation of:

C_s^{2R} ('Two in Row') is penalty given whenever a student s has to sit two distinct examinations scheduled in two consecutive timeslots within the same day.

C_s^{2D} ('Two in Day') is penalty given whenever a student s has to sit two distinct examinations scheduled in two non-consecutive timeslots within the same day.

C_s^{PS} ('Period Spread') is penalty given whenever a student s has to sit more than one examinations within specified period.

C^{NMD} ('Non-Mixed Duration') is penalty given whenever any room in any timeslot is allocated to examinations with different duration.

C^{FL} ('Front Load') is penalty incurred by allocating considered large examination in the end of examination period.

C^P is penalty associated to period/timeslot whenever it is allocated to examination.

C^R is penalty associated to room whenever it is allocated to examination.

The weight for each above mentioned penalty, e.g. w^{2R} as well as the other specifications, e.g. penalty associated to each room/timeslot are defined in 'institutional model index' file. The detail as well as the mathematical programming formulation of this problem could be consulted in [205]. Whereas the characteristics of ITC 2007 dataset is presented in Table 2.2. In addition, the associate weight for each dataset is given in Table 2.3.

Table 2.2: The characteristics of problem instances from ITC 2007 examination timetabling problem dataset

Problem Instances	No.Of T.Slots	No.Of Exams	No.Of Students	No.Of Rooms	Conflict Density	No.Of P.HC	No.Of R.HC
EXAM1	54	607	7891	7	0.05	12	10
EXAM2	40	870	12743	49	0.01	12	2
EXAM3	36	934	16439	48	0.03	170	15
EXAM4	21	273	5045	1	0.15	40	0
EXAM5	42	1018	9253	3	0.01	27	0
EXAM6	16	242	7909	8	0.06	23	0
EXAM7	80	1096	14676	15	0.02	28	0
EXAM8	80	598	7718	8	0.05	20	1
EXAM9	25	169	655	3	0.08	10	0
EXAM10	32	214	1577	48	0.05	58	0
EXAM11	26	934	16439	40	0.03	170	15
EXAM12	12	78	1653	1	0.18	9	7

Note: P.HC=period-related hard constraints, R.HC=room-related hard constraints

Table 2.3: The associate weight ITC 2007 examination timetabling problem dataset

Problem Instance	w^{2D}	w^{2R}	w^{PS}	w^{NMD}	LE	NLP	w^{FL}
EXAM1	5	7	5	10	100	30	5
EXAM2	5	15	1	25	250	30	5
EXAM3	10	15	4	20	200	20	10
EXAM4	5	9	2	10	50	10	5
EXAM5	15	40	5	0	250	30	10
EXAM6	5	20	20	25	25	30	15
EXAM7	5	25	10	15	250	30	10
EXAM8	0	150	15	25	250	30	5
EXAM9	10	25	5	25	100	10	5
EXAM10	0	50	20	25	100	10	5
EXAM11	50	10	4	35	400	20	10
EXAM12	10	35	5	5	25	5	10

Note: LE=the number of largest exam to be considered.
NLP=the number of last periods to be considered.

2.1.3 Yeditepe and Other Datasets

The Yeditepe dataset are from Yeditipe university that are basically similar to the ITC 2007 datasets. The characteristics of Yeditepe dataset are given in Table 2.4. Generally, compared to ITC 2007 dataset, Yeditepe dataset are smaller and less constrained. In ITC 2007, each exam has to be assigned to one of a set of rooms with different maximum capacity, whereas in Yeditepe dataset, there is only total room capacity constraint as shown in Table 2.4. The total room capacity represents the maximum number of students that could be accommodated in the same timeslot. In other words, in Yeditepe dataset, there is actually no room assignment. Similarly note that in Carter dataset the room capacity is neglected, therefore it can be assumed that the total room capacity in each timeslot is equal to the total number of students.

Table 2.4: The characteristics of problem instances from Yeditepe benchmark datasets

Problem Instances	No. of Exams	No. of Students	No. of Enrolments	Conflict Density	Days	Tot. Room Capacity
YUE20011	126	559	3486	0.18	6	450
YUE20012	141	591	3708	0.18	6	450
YUE20013	26	234	447	0.26	2	150
YUE20021	162	826	5755	0.18	7	550
YUE20022	182	869	5687	0.17	7	550
YUE20023	38	420	790	0.21	2	150
YUE20031	174	1125	6714	0.15	6	550
YUE20032	210	1185	6833	0.14	6	550

Other examination timetabling problem instances in the literature include Nottingham benchmark dataset [74], University of Melbourne benchmark dataset [211], MARA University Malaysia dataset [173], Universiti Kebangsaan Malaysia (UKM) [28], modified Carter dataset [41], Yeditepe bechmark dataset [233, 41, 241], Universiti Malaysia Pahang dataset [167], and KAHO Sint-Lieven [122]. The properties of these datasets are summarised in Table 2.5.

Table 2.5: The problem instance properties of the other examination timetabling problems benchmark datasets

Datasets	Instance	No. of Exams	No. of Students	No. of Timeslots	Days	No. of Enrolments	Conflict Density	Capacity	No. of Rooms
Nottingham	nott	800	7896	23		33997	0.23	1550	
Melbourne	mel01s1	521	20656	28					
	mel01s2	562	19815	32					
Mara (UiTM)	UiTM03s2	472,201	84,675			470793			
UKM	UKM06-1	818	14047	42	15	75857			
UMP	UMP07s1	157	3,550	20	10	12,731	0.05		
KAHO	KAH06s1	336	135 groups	40					71

Regarding the room assignment, the examination Timetabling problems can be classified into uncapacitated and capacitated problems. In uncapacitated problems, room capacities are neglected whilst in capacitated problems the room capacities are considered

as a hard constraint. The uncapicatated problems include Carter and MARA dataset whilst ITC 2007, Nottingham, Melbourne, UKM, Yeditepe, Universiti Malaysia Pahang (UMP), and KAHO Sint-Lieven are capacitated problems. Further, we compared these datasets, in term of the complexity of hard constraints and soft constraints considered in their problem formulation as summarised in Table A.1 in Appendix A. Considering the number of both hard and soft constraints, from Table A.1 we can observe that ITC 2007 examination timetabling problem formulation is the most comprehensive, nearly reflecting the real-world examination timetabling problem.

In the scientific literature, the most studied examination timetabling problem datasets are Carter and ITC 2007 dataset. Therefore we summarised the experimental results including the best known solutions reported in the scientific literature summarised by Table A.2 and Table A.3 in Appendix A for Carter and ITC 2007 datasets respectively. We used these experimental results as a benchmark for our experiments.

From this literature survey, we can observe that most of the current studies over examination timetabling problems focused on benchmark problems, i.e. Carter and ITC 2007. However, lack of studies focusing on new model that are more representative to the real-world structure, especially the model that is more student's preferences oriented. Fairness amongst students regarding their examination is also overlooked by the existing studies. Therefore, rather than only study on the benchmark problem models, this thesis more focus on study new models that are more student oriented especially regarding with fairness issue.

2.2 Examination Timetabling Problem Approaches: state-of-the-art

Surveys on state-of-the-art examination timetabling problem formulations, techniques and algorithms have been reported in prior work such as in [89], [281], [78], [204], [188], and [260]. In [281], the methods to solve the examination timetabling problems are classified into: direct heuristics [182, 88], graph colouring approach [209], simulated annealing [132, 162], genetic algorithms [103, 105, 236, 58], and other methods, e.g. network and Lagrangian technique [35]. On the other hand, in [260], which is claimed as the most comprehensive examination timetabling problems techniques survey, the methods are classified into: graph based sequential technique, constraint based technique, meta-heuristic, multi-criteria technique, hyper-heuristic, and decomposition/clustering technique. The methods within this classification are summarised in Table A.4. Further,

based on classification in [260], we extend the literature survey with recently published literature summarised in Table A.5 in Appendix A.

In the graph-based sequential method, a solution is constructed from empty set solution. The exams are assigned to timeslots one by one. The sequential method includes direct heuristic, exact vertex colouring method, and graph colouring heuristics method. In direct heuristic, each exam is assigned to timeslot gradually without any ordering criteria [281]. In exact vertex colouring method, each exam is represented as a vertex and two conflicting exams are represented by an edge connecting them. Each vertex is assigned to a colour making sure that there is not connected vertices assigned to the same colour [318, 89]. On the other hand, in graph colouring heuristics method, the vertices are assigned to colours based on some criteria such as largest degree first, largest weighted degree first, largest saturation degree first, and colour degree [89]. These methods are also successfully incorporated with random element [76], adaptive ordering [50], fuzzy logic [23], neural networks [106], clustering [230], rough set approach [303], adaptive selection [82], and matrix semi-product approach [326].

In clustering/decomposition methods, conflict-free sets of exams are clustered into the same cluster, then each cluster is assigned to the same time slot [319, 78]. In order to make the problem easier to solve, the exam timetabling problems also can be decomposed into smaller sub-problems such as in [71, 257]. Another approaches within this category also include granular modelling [262], nested partition [328], hybridisation clustering with graph colouring heuristics [230]. Whilst, in constraint-based method, exam timetabling problem is modelled as assignment of values to variables given some defined constraints. It consists of constraint logic programming [141, 211] and constraint satisfaction technique [141, 117].

Metaheuristics are the most successfully and widely methods to solve examination timetabling problems. Typically, metaheuristic methods are employed to improve a feasible solution initially constructed with sequential technique. Generally, according to the number of initial solutions, these methods can be classified into local search and population base technique. Local search based techniques, which work upon a single solution, include simulated annealing [308, 307, 309], great deluge algorithm [50], extended great deluge algorithm [208], tabu search [153, 125, 320, 237], large neighbourhood search [6], variable neighbourhood search [12, 55], iterated local search [1, 52], Greedy randomised adaptive search procedures (GRASP) [90], memetic algorithm [74], hybrid metaheuristic approaches [76, 146, 311], and intelligent water drop algorithm [15]. On the other hand, population based techniques include evolutionary algorithm e.g. genetic algorithm [57, 105, 271, 267, 107, 312, 249], Memetic Algorithm [74, 67, 8, 235], ant al-

gorithm [31, 126], artificial immune algorithm [199], harmony search [14], memetic algorithm with harmony search algorithm [13], evolutionary ruin and stochastic rebuild [189], Hybrid genetic algorithm [231], MOEA with micro-genetic algorithm [127], hill climber local search [97], hybrid fish swarm [310], honey-bee mating [278], hybrid artificial bee colony [20], hybrid bee colony [19], tabu-based memetic approach [10], hybrid imperialist swarm-based optimisation [139], double evolutionary pool memetic algorithm [186], adaptive artificial bee colony and late acceptance hill climbing [18], hybrid self-adaptive bees algorithm [7], and particle swarm optimisation algorithms [183].

In multi-criteria technique, constraints are defined from different point of view. Instead of simply considered as aggregate summation of cost/penalties caused by different constraint, multi-criteria techniques are introduced. Studies of the examination timetabling problems with multi-criteria techniques can be observed in [51, 287, 185, 97, 224].

In [224], multi-objective memetic algorithm was implemented to solve examination timetabling problem with two objectives, i.e. timetable length and proximity cost. A solution is represented as permutations of examinations with a greedy algorithm to allocate timeslots to exams sequentially. The proposed multi-objective framework, which is based on [314, 223] consists four stages, i.e. initialisation, multi-objective optimisation with bi-objective, multi-objective optimisation focusing on proximity cost, and local search. Grouping and reordering heuristics [115] instead of traditional sequential graph colouring technique is utilised. In each iteration the grouping, reordering and mutation operators are applied over the selected individual. There is not crossover operator in the proposed algorithm and there are two mutation operators, i.e. insertion (position based mutation by choosing two examinations randomly and insert the second exam before the first exam) [118] and Kempe chain swap. Kempe chain swap is also used in local search stage.

In general, memetic algorithm is hybridisation of evolutionary algorithm and local search that was firstly introduced by [216] and formalised by [261]. Prior work on investigating memetic algorithm for examination timetabling problems is [74]. In [74] two mutations, namely, heavy and light mutations were introduced. These operators were combined with hill climbing local search. Light mutation operates by selecting a number of examinations and move to another feasible timeslot while heavy mutation disrupts one or more selected timeslots. Each examination in selected timeslots is then randomly removed and rescheduled in the first feasible timeslot. The hill climbing operates by rescheduling each exam in the order of the current solution, to another feasible timeslot causing the least penalty. The time consuming process of hill climbing operators is reduced by delta evaluation [268]. The proposed algorithm was tested over Carter and

Nottingham datasets.

Hyper-heuristic methods aim at tackling the limitation of metaheuristic approaches that required experienced based parameter tuning. Studies of hyper-heuristics over the examination timetabling problems can be observed in [173, 70, 258, 72, 41, 232, 247, 63, 122]. Furthermore, graph colouring constructive hyper-heuristic was also introduced in [248, 280, 80, 82, 290, 3] as well as Monte carlo hyper-heuristic in [63], Harmony search hyper-heuristics [21, 22], grammatical evolution hyper-heuristics [276], hierarchical hybridizations graph colouring based construction heuristics [280, 248], and dynamic multi-armed bandit-gene expression programming [279].

One of the key literatures with new approach reported successfully to solve the examination timetabling problems is artificial bee colony algorithm. In [20], artificial bee colony (ABC) [168, 170, 169] with simulated annealing and disruptive neighbourhood structures strategy was employed. In ABC, the honey bees are classified into three types: employed bee, onlooker bee, and scout bee. A solution is mimicked as a food resource. Employee bees perform exploration search, while onlooker bees perform exploitation search. Onlooker bee select a solution from the employed bee. Employed bee whose abandoned solution, in turn, become scout bee and undergo random exploration search. The number of employed bee corresponded to the number of solution in population. The initial solutions representing employed bee are constructed by largest degree heuristic sequential approach. In the basic ABC, onlooker bee select a solution from employed bees base on Roulette Wheel Selection mechanism, in which the probability of solution to be chosen is proportional with its objective function value. This selection mechanism result in premature convergence and lack of solution diversity. To tackle this problem disruptive selection strategy [181] is applied.

In disruptive selection strategy, a solution with extreme objective function value (the best or the worst one) has the highest probability to be chosen. It is because the fitness value of a solution is proportional to absolute different between its objective function value and average objective function value of solutions in the population. In each iteration, both employed bee and onlooker bee employ neighbourhood search with one of four neighbourhood structure (i.e. move and swap operations). The neighbourhood structures are selected base on self-adaptive method [238]. By this method, a neighbourhood structures is chosen from a fixed-length list consisting of mainly well-performed (during the search process) neighbourhood structures. Different from the original ABC, where only an improved solution is accepted. In order to skip from local optima, onlooker bees execute local search with simulated annealing mechanism rather than greedy neighbourhood search.

A similar work with [20] on employing artificial bee colony algorithms for solving exam timetabling problems was reported in [18]. In this work the only difference is that simulated annealing was replaced with late acceptance hill climbing(LAHC) [49]. In [48], LAHC list length is set to 500. The experimental results showed that the proposed approach outperforms the basic ABC quite significantly. Further, in [19], Bee Colony Optimisation (BCO) algorithm [192, 194, 297] was employed. Four selection strategies, i.e. tournament [42], ranking [291], disruptive [181], and roulette wheel are evaluated [34]. For the most recent approaches exclusively dedicated for timetabling problems can be tracked from two bi-annual conferences, namely, international conference on the practice and theory of automated timetabling (PATAT) and multidisciplinary international scheduling conference:theory and application (MISTA).

From the survey over the existing approaches for solving examination timetabling problems we can observe that the most successful approaches are mainly the combination of graph colouring based ordering heuristics and meta-heuristics. The graph colouring based ordering heuristics are employed to generate an initial solution. Furthermore, the quality of the initial solution is improved with meta-heuristic approach.

2.3 Hyper-heuristics

The basic idea of hyper-heuristic is a notion to develop a more generic algorithm as opposed to the standard metaheuristic approaches that are generally require intensive empirical based parameter tuning to solve different problem domains. Within standard metaheuristics, problem domain knowledge required for empirical based parameter tuning knowledge is essential [111]. To cope with this problem, a hyper-heuristic comes up with the idea of working upon higher level, i.e. low-level heuristics search space instead of directly to the solution search space. Lifting the search space is intended to make hyper-heuristics more generic and perform well over cross problem domains.

Since it was firstly introduced in [111], the broadly-accepted definition of hyper-heuristic is (meta)heuristics (later on called high level heuristics) to select (meta)heuristics (later on called low level heuristics) to solve problems in hand [65]. Hyper-heuristic is a heuristic approach which is incomplete search methods that have no guarantees of success to find the optimal solution. Sometime, the term hyper-heuristic is interchangeable with metaheuristics, an iterative process which guides a subordinates heuristics by combining intelligently different concepts for exploring and exploiting the search space using learning strategies to structure information in order to find efficiently near-optimal solutions [229].

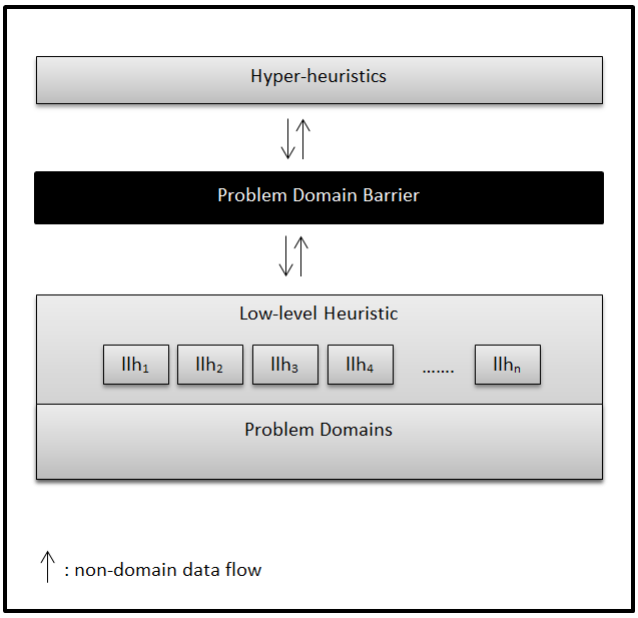


Figure 2.1: The Basic Hyper-heuristics Framework

The basic hyper-heuristic framework is shown in Figure 2.1 as proposed in [65, 109, 111, 112, 113, 114, 151]. As shown by Figure 2.1, which is adapted from [65], the main part of hyper-heuristic framework is a domain barrier. It makes hyper-heuristics have no information about the domain under which it is operating. The only thing hyper-heuristic know is n low level heuristics and the evaluation function result from the problem domains. In addition, there is an interface defined between hyper-heuristic and low level heuristics. Therefore, hyper-heuristic works upon heuristic space instead of directly upon solution space. In other words, hyper-heuristics works at higher level of abstraction than the standard metaheuristics approach. Another important point of hyper-heuristics against metaheuristic is that in hyper-heuristics framework a solution can be constructed by jumping to the neighbourhoods while in metaheuristics or heuristic it moving to the neighbourhoods.

Since the incorporation of genetic programming and other methods such as "squeaky wheel" [163] into hyper-heuristic, the definition of hyper-heuristics evolved. It is not only defined as heuristics to select heuristics but also heuristic to generate heuristics. In addition, the higher level strategy should not be restricted to be a heuristic, but could be other strategies such as case base reasoning. Clearly, in [60], a hyper-heuristic is redefined as follow:

> "Hyper-heuristics comprise a set of approaches with the goal of automating, often by the incorporation of machine learning techniques, the process of either (i) selecting and combining simpler heuristics, or (ii) generating

> new heuristics from components of existing heuristics; in order to solve hard computational search problems."

The higher level strategy of hyper-heuristic could be metaheuristics e.g. tabu search, simulated annealing, and genetic algorithm; cost function, case base reasoning and other machine learning techniques. On the other hand, the lower level heuristics could be constructive heuristics e.g. constraint satisfaction and graph heuristic or improvement/perturbation heuristics e.g. move strategy.

There are some prior works on classifying hyper-heuristics such as in [94, 61], and the most recent work is in [59]. In [61, 59], according to the nature of the heuristic search space, hyper-heuristics are categorised into heuristic selection and heuristic generation methodology in the first level that further classified the low level heuristics into construction and perturbation (i.e. local search) in the second level.

Furthermore, with respect to the feedback source during learning, hyper-heuristics are classified into online, offline, and no learning hyper-heuristic. In online hyper-heuristics there is a learning mechanism while solving problem in hand using approaches such as reinforcement learning and metaheuristics. Offline hyper-heuristics, on the other hand, infer generalised rules from training problem instances to solve unseen problem instances. An example of offline learning approaches in the hyper-heuristics is the incorporation of case base reasoning. This classification illustrating [61] is summarised by Figure 2.2. Classification A is based on the nature of the heuristic search space. Classification L is based on the feedback/learning mechanism.

In a nutshell, selection hyper-heuristics consist of two main components, i.e. heuristic selection method and move acceptance method. The heuristic selection methods which are the strategies on how higher level methodology chooses the lower level heuristic have been surveyed in [94]. In [94] the strategies are classified into random selection (including pure random, random descent, unbiased random process, monte carlo, and random with deterministic acceptance), greedy (include accepting only improving solution and accepting non-improving solution) and peckish that incorporate greedy with random selection, metaheuristics (including GA-based hyper-heuristics, simulated annealing, Tabu Search, and VNS), and learning base (include reinforcement learning, choice function, learning subset of low level heuristic, learning classifier system, and case base reasoning).

On the other hand, move acceptance methods include all moves are accepted (AM) and only improving moves accepted (MO). The comparative studies of heuristic selection and move acceptance on examination timetabling problems can be seen in [41].

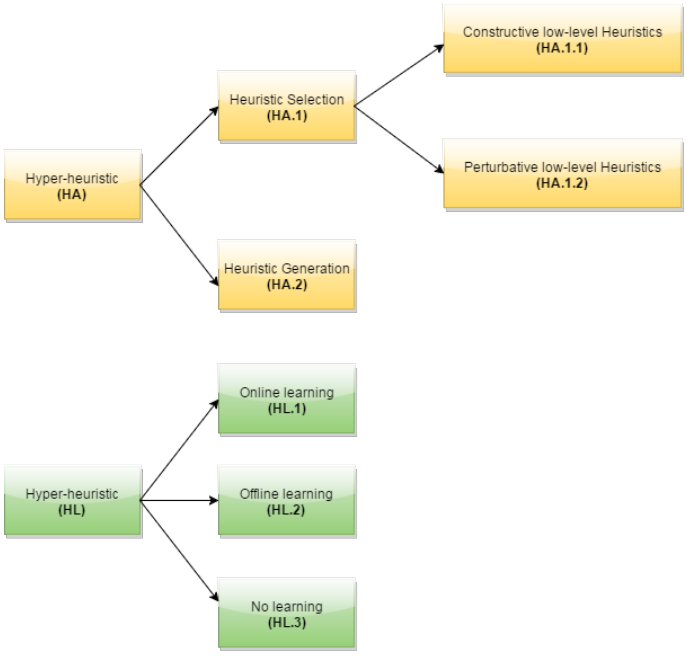


Figure 2.2: The Hyper-heuristic Classification

Hyper-heuristic approaches have been successfully implemented to solve hard search problems such as timetabling [70] including course timetabling [72], examination timetabling [41, 79], nurse rostering; bin packing [200], stock cutting [298], presentation scheduling [62], sales summit scheduling [111], channel assignment problem in mobile communications [175], shelf space allocation problem [33], vehicle routing [32], etc.

The current development of hyper-heuristics is comprehensively reported in [59]. In [59], the application of hyper-heuristics are classified into several problem domains as summarised in Table 2.6

Table 2.6: The Application of Hyper-heuristics

Code	Problem Domain
C.1	Personal Scheduling
C.2	Educational Timetabling
C.3	Space Allocation
C.4	Cutting and Packing
C.5	Vehicle Routing
C.6	Sport Scheduling
C.7	Production Scheduling
C.8	Satisfiability
C.9	Traveling Salesman Problem
C.10	Function Domain
C.11	Cross Domain
C.12	Others

Hyper-heuristics also have been successfully extended to solve multi-objective optimisation problems. In [316] two-phases hyper-heuristic applied for multi-objective travelling

salesman problem whilst in [203, 203], a markov-chain hyper-heuristic is applied for solving water distribution network design problem. In addition, in [149], a NSGAII-based hyper-heuristic is employed for solving cutting stock problems.

In [195, 197, 196] choice function based hyper-heuristics are investigated to solve multi-objective optimisation problem. Three well-known multi-objective algorithms, namely, NSGAII, SPEA2, and MOGA, are used as low-level heuristic controlled by great deluge and late acceptance move acceptance strategy which is incorporated with hyper-volume metric within hyper-heuristic framework. The proposed approach were tested over standard multi-objective optimisation problems from walking fish group test suite, and a real-world problem, i.e. vehicle crashworthiness design problem. Compared to other approaches, it was shown that the proposed approach is more effective.

2.4 Hyper-heuristics for Examination Timetabling Problems

The majority of existing hyper-heuristics approaches for solving examination timetabling problems are from type of heuristic selection hyper-heuristics. The low-level heuristics are constructive heuristics adopted from graph colouring heuristics. The prior works with graph constructive heuristics selection hyper-heuristics include [270, 300, 12, 273, 272, 23, 53, 79, 70, 250, 244, 258, 228, 190, 280]

In [270, 300], genetic algorithm with non-direct chromosome representation was used to control the configuration of constraint satisfaction problem (CSP) method. Whereas in [12], variable neighbourhood search (VNS) was utilised to select the combination of parameterised constructive heuristics.

Furthermore, in [273], simple messy GA [148] was utilised to learn the association between construction heuristics and problem state to decide which heuristic is adequate to be used for problem at hand. The heuristics consist 44 heuristics, i.e. 16 ways of choosing an event, and 28 ways of choosing a slot. In the follow-up paper [272], the proposed algorithm was improved and proven to be be able to solve more problems.

In [23], a fuzzy multiple heuristics ordering method was proposed. The method was based on three graph colouring heuristics: largest degree first (LD), least saturation degree first (SD) and largest enrolment first (LE). In addition, in [53], tabu search was employed to control the selection of two constructive graph colouring heuristics: SD, LD. Further, with data from prior tabu search hyper-heuristic experiment, a case base reasoning (CBR) method was developed. This CBR method was then improved in [79],

in which at each decision point, CBR was employed to intelligently select a heuristic from a set of constructive heuristics: LD, largest colour degree (LCD) with tournament selection, colour degree (CD), SD, and hill climbing (HC).

Later, in [70], tabu search was utilised to control search of optimal permutation of consturctive low-level heuristics, i.e. graph colouring heuristics: SD, largest colour degree first (CD), LD, LE, largest weighted degree first (LWD), and random ordering (RO). Each heuristic in the permutation is sequentially used to order the unassigned exams, to be assigned to timeslot/room. Moreover, in [250], genetic programming (GP) approach was proposed. In this approach, a program in the proposed GP is an analogy of a sequence of graph colouring heuristics to construct a complete examination timetable. Each individual in the population is represented as variable-length string in which each character represent a graph colouring heuristic. The experimental results showed that, tested over Carter dataset, the GP-based hyper-heuristic could produce feasible solutions for 9 problem instances and 5 of them outperformed results from similar hyper-heuristic approach.

Furthermore, in [258], which is extended work of [70], compared some metaheuristic methods: steepest descent method (SDM), iterated local search (ILS), tabu search (TS), and variable neighbourhood search (VNS) as a high level strategy i.e. working over heuristic space in hyper-heuristics. It was shown that iterative techniques (ILS, VNS) are more effective than non-iterative techniques. In this work, a hyper-heuristic was also hybridised with local search within solution space that showed better results. Other extended works of [70] are [228] and [190]. In [228], the notion of fitness landscape was proposed to analyse the search space of graph colouring heuristic. It was concluded that an optimal solution would be surrounded by many local optima. Whereas in [190], with the data from [70], artificial neural network (ANN) and binary logistic regression (BLR) techniques were utilised to find global patterns of heuristic sequence. The performance of the heuristics (whether the resulting solution will be rejected/approved) can be predicted with the generated classification rules, thus there is no need for computationally expensive processes, i.e. objective function calculation and acceptance determination. Experiment showed significant speed-ups of the search process. All above mentioned methods were tested over Carter dataset. Except for [12] that was tested over Nottingham dataset.

The most recently published studies on examination timetabling problems with hyper-heuristics are discussed in [41, 70, 260, 232, 247, 63, 80, 122, 290, 4, 3, 280, 279]. Bilgin et al. in [41] carried out a comprehensive in-depth comparative analysis on the performance of hyper-heuristics with differing combination of low-level heuristics selection mechanism

and move acceptance strategies over examination timetabling problem benchmark instances. The selection mechanisms consist of seven mechanism, namely, simple random, random descent, random permutation, random permutation descent, choice-function, tabu search and greedy. Whereas the move acceptance strategies consist of five strategies, namely, all moves accepted (AM), only improving moves accepted (OI), improving and equal moves accepted (IE), great deluge and monte carlo strategy. This combination result in 35 different hyper-heuristics.

To evaluate the performance of hyper-heuristics, the study was carried out over 14 well-known benchmark mathematical or bit string functions as well as 21 examination timetabling problem instances from Carter benchmark datasets [89] and Yeditepe bechmark datasets [241]. The experimental results showed that the combination of choice function heuristic selection method with monte carlo acceptance criterion [30] are the superior one over the other combinations.

Graph-based hyper-heuristic incorporated with tabu search (TS) evaluated over Carter datasets was reported successful with good results in [70]. Further, in [254] the graph-based hyper-heuristics incorporated with steepest descent method (SDM), iterated local search (ILS), and variable neighbourhood search (VNS) were studied over Carter datasets. The computational result showed that iterative technique e.g. VNS and ILS are more effective than TS and SDM.

In addition, hyper-heuristics with late acceptance strategy was studied in [232]. This strategy compares candidate solution with solution from earlier iterations instead of the current solution. The proposed approach was tested over Carter dataset. The experimental study showed that late acceptance strategy is best suited with simple random low-level heuristic selection. This combination outperforms the combination of late acceptance strategy with reinforcement learning or statistical based heuristic selection.

Evolutionary algorithm based hyper-heuristic for examination timetabling problem with Carter datasets was studied in [247]. The study which is an extension work of [251] examined three different proposed representation of low-level heuristics combination: fixed length heuristic combination(FHC), variable length heuristic combination(VHC), and N-times heuristic combination (NHC). The experimental results show that NHC and VHC performs much better than FHC. The results also show that the combination of three representations yields better performance than FHC, VHC, and NHC.

Burke et al. [63] compared the performance of different Monte Carlo based hyper-heuristics over Carter datasets. Four low-level heuristic selection methods, i.e. simple

random, greedy, choice function and learning scheme; and three Monte Carlo based move acceptance methods, i.e. standard simulated annealing, simulated annealing with reheating and exponential Monte Carlo were evaluated. The results indicate the success of a hyper-heuristic combining a reinforcement learning based method, namely choice function and simulated annealing with reheating.

Tournament-based hyper-heuristics for examination timetabling problems were investigated in [122]. The study evaluated tournament based random selection of low-level heuristics coupled with four move acceptance criteria: 'improving or equal', simulated annealing, great deluge, and an adapted version of the late acceptance strategy. The proposed hyper-heuristics were tested over three benchmark datasets, namely Carter, ITC 2007, and KAHO datasets, a new examination timetabling problems benchmark. Different from prior problem instances, in KAHO problem instances there are two types of examinations i.e. written and oral examinations. The experimental result showed that the proposed approach could improve best results in the literature, seven out of thirteen Carter datasets. For ITC dataset, though it is failed to improve the best known result in the literature, but it still could produce competitive results.

In [290], in order to assign exams to time slots and rooms, bin packing heuristics are hybridised using random iterative hyper-heuristic. The study over ITC 2007 datasets, shows that the hybridisation of mainly heuristics that produced the best solution when used on its own is the one that could produce the best solution. Compared to the best known solutions in the literatures, the results from the proposed approach are reported very competitive.

Abdul-Rahman et al. in [4] used adaptive decomposition and heuristics ordering. In the process of assignment, the examinations are divided into two subsets, namely difficult examinations and easy examination. Further, to determine which examination should be assigned to timeslot first, within each subset the examinations are ordered based on differing strategy of graph colouring heuristics. Initially, all examinations are members of easy examination set. Then, during the process of assignment if an examination could not be assigned to any feasible timeslot, it is moved to hard examination subset. The process is iterated until all examinations are assigned to feasible timeslot. The experimental study over Carter datasets shows that the proposed approach is competitive with prior works in the literature.

In [3] constructive approach so-called adaptive linear combination of heuristics was proposed. This approach is based on squeaky wheel optimisation [163]. During the process of assignment, each examination is represented by difficulty score which is based on graph colouring heuristics and heuristics modifier that will change dynamically. The

examination that will be firstly assigned to resource (i.e. timeslot and room) is the exam with the highest difficulty score. Starting with difficulty score equal to chosen graph-colouring heuristic order, the difficulty score of an examination will be increased based on heuristics modifier function whenever it could not be assigned to any feasible resource. As consequence, the difficult score will change the previous order dynamically. The process stops whenever all examinations have been assigned to feasible resource. In order to get high quality solution, in each assignment process, the resource to be allocated is chosen from those incurring penalty as least as possible. Testing over Carter and ITC 2007 dataset show that, in addition to its simplicity and practicality, the proposed approach is competitive with previous approaches in the literature.

In [280], constructive heuristics selection hyper-heuristics approach was proposed. In this work, exams are ordered based on difficulty index (DI). DI is calculated base on hierarchical hybridisation of graph colouring heuristics: LD, SD, LCD, and LE. Roulette wheel selected was utilised to assign exam to timeslot. Experiment results over Carter and ITC 2007 datasets showed competitive results with reporting result in the literature.

Later, in [279] proposed hyper-heuristic with low-level heuristics selection mechanism based on dynamic multi-armed bandit-extreme value-based reward. Moreover, the move acceptance criteria are generated automatically using the proposed gene expression programming framework. The proposed approach were tested over two different problem domains, namely ITC 2007 examination timetabling problem and dynamic vehicle routing. The experimental study show that compared to ITC 2007 winner as well as post-ITC 2007 methods, the proposed approach outperforms for 4 out 8 problem instances.

The previous studies using hyper-heuristics for solving examination timetabling problems show that though the algorithms were developed to solve multiple cross problem domains instead of a specific problem domain, the performance were very competitive with other algorithms specially developed for solving exam timetabling problem. In addition, employing hyper-heuristics for solving examination timetabling problem give an advantage of easiness to employ the algorithms to solve other problems domains without parameters changing on the algorithms. Therefore, in the study discussed in this thesis, hyper-heuristic is chosen.

2.5 Multi-Objective Optimisation Algorithm

Generally, a multi-objective optimisation problem (MOP) also known as multi-criteria decision making (MCDM) includes a set of n decision variables, a set of k objective function, and a set of m constraints. The goal of MOP is to:

$$\begin{aligned} minimise/maximise\ \ y = f(x) &= (f_1(x), f_2(x), ..., f_k(x)) \\ subject\ to\ \ e(x) &= (e_1(x), e_2(x), ..., e_m(x)) \leqslant 0 \\ where\ \ x &= (x_1, x_2, ..., x_n) \in X \\ y &= (y_1, y_2, ..., y_k) \in Y. \end{aligned} \tag{2.9}$$

In which x and y are the decision and objective vectors (respectively) while X and Y are the decision and objective spaces (respectively). In MOP, the set of feasible solutions is defined as:

$$X_f = \{x \in X | e(x) \leqslant 0\}. \tag{2.10}$$

The image of X_f is a feasible region in objective space, defined as: $Y_f = f(X_f) = \bigcup_{x \in X_f} \{f(x)\}$. For any two objective vectors u and v, the relations between them are defined as:

$$\begin{aligned} u = v\ \ &iff\ \ \forall i \in \{1, 2, ..., k\} : u_i = v_i \\ u \leqslant v\ \ &iff\ \ \forall i \in \{1, 2, ..., k\} : u_i \leqslant v_i \\ u < v\ \ &iff\ \ u \leqslant v \wedge u \neq v. \end{aligned} \tag{2.11}$$

The relation $\geqslant$ and $>$ could be defined similarly. By this relation definition, the Pareto dominance of any two decision vectors a and b (in minimisation problem) is defined as:

$$\begin{aligned} a \prec b\ \ (a\ \text{dominates}\ b)\ \ &iff\ \ f(a) < f(b) \\ a \preceq b\ \ (a\ \text{weakly dominates}\ b)\ \ &iff\ \ f(a) \leqslant f(b) \\ a \sim b\ \ (a\ \text{is indifferent to}\ b)\ \ &iff\ \ f(a) \nleqslant f(b) \wedge f(b) \nleqslant f(a). \end{aligned} \tag{2.12}$$

For the maximisation problem, the relation $\succ$, $\succeq$, and $\sim$ could be defined analogically. Based on the Pareto dominance definition, Pareto optimality could be defined. A solution $\mathbf{s}$ is a Pareto optimal or non-dominated/ non-inferior solution, if it cannot be

improved its any corresponding objective functions without degrading at least one other objective functions. Formally, a solution consists of a decision vector $x \in X_f$ is said non-dominated regarding a set $A \subseteq X_f$ *iff*:

$$\nexists a \in A : a \prec x \tag{2.13}$$

The A in the definition could be left out, if the context which set A is meant. Furthermore, we can say that x is Pareto optimal *iff* x is nondominated regarding X_f. The entire Pareto solutions and their corresponding objective vectors are called *Pareto optimal set* and *Pareto optimal front or surface*, respectively. Pareto optimal set ideally consists of globally optimal solution but may also be local optimal. Suppose a set of decision vector $A \subseteq X_f$, as introduced in [120]; the set A could be defined as local Pareto optimal set and global Pareto optimal set by Equation 2.14 and Equation 2.15, respectively.

$$\begin{gathered} \forall a \in A : \nexists x \in X_f : x \succ a \wedge \|x - a\| < \epsilon \wedge \|f(x) - f(a)\| < \delta \\ \textit{where } \|.\| \textit{ is a corresponding distance metric and } \epsilon > 0, \delta > 0 \end{gathered} \tag{2.14}$$

$$\forall a \in A : \nexists x \in X_f : x \prec a \tag{2.15}$$

Since the global Pareto optimal solution is usually not known, candidate non-dominated solutions that approximate the real global Pareto optimal set are searched instead. Similar definition of the basic concept of multi-objective problem and Pareto optimality given above can be found in [332, 101, 99].

The majority of successful metaheuristic approaches for multi-objective optimisation problems are dominated by evolutionary algorithms. The most successful algorithms include NSGA-II [121], PAES (Pareto Archived Evolution Strategy) [177], MPAES (Memetic-Pareto Archived Evolution Strategy) [178], SPEA (Strength Pareto Evolutionary Algorithm) [334], and SPEA2 [334, 333]. These approaches have been successfully implemented to solve real-world and benchmark multi-objective continue problems. In [226] convergence speed in multi-objective optimisation problem was compared to indicate that NSGA-II [121] achieved the best solution and placed fourth in convergence speed after MOCell [225], AMOPSO [286], and AbYSS [227]. Since no previous studies found that NSGA-II algorithm performs well on combinatorial optimisation problem (COP), modification on the original version of NSGA-II in order to cope with COP is necessary.

However, few previous works investigated non-evolutionary algorithm for multi-objective optimisation problems, such as ones based on simulated annealing algorithm, e.g. Multi Objective Simulated Annealing (MOSA) [283] and MOSA-like algorithms [313, 116], Ishibuchi Murata Multi Objective Genetic Local Search I(MMOGLS) [157], Multi Objective Genetic Local Search(MOGLS) [160], and many others.

As noted by [287], the prior works reported in the literature on utilising multi-objective metaheuristics for timetabling are relatively scarce. One of the few studies that considered this was [86], which implemented multi-objective genetic algorithm for bi-objective class/teacher timetabling problem. The work was based on NSGA algorithm [293]. Rather than a single objective, the objective was divided into two objectives: soft constraint violation penalty associated with class and ones associated with teacher, respectively.

In addition to multi-objective algorithm, some framework tools to solve multi-objective problems have been proposed in the literature, one of which is jMetal [131, 130]. It is an open source Java-based framework for multi-objective optimisation problems with some classical and modern metaheuristic approaches widely used in the literature. The latest version includes a popular multi-objective genetic algorithm, NSGA-II [121]. jMetal is also claimed to be easy to be extended with new algorithms and problems. Other frameworks are ParadisEO [83], Opt4j [193], and MOEA [215], which are designed to facilitate experiments to compare the performance of multi-objective evolutionary algorithms.

In order to be able develop effective algorithms to solve a particular problem, knowing the fundamental nature of the problem is important. To analyse the nature of multi-objective problems, several methodologies have been proposed in the literature [253, 92, 156, 252]. In [253], three relationships (i.e. conflict, harmony and independence) between two objectives within large number criteria evolutionary optimisation problem were identified. To identify the relationship, qualitative methods such as parallel coordinate and scatter matrix could be used. Alternatively, quantitative method, such as Kendall sample correlation metric [176] and Schroder metric [282], can also be used.

Furthermore, based on the platform proposed by [253], in [92] the nature of various objectives was considered to investigate whether there is pairwise conflict in the vehicle routing problem with Time Windows (VRPTW). The study identified conflicting relationships between five objectives, whereas in [156], highly correlated objectives were identified on many-objective 0/1 knapsack problems. Similarly, in [252], tested over multiobjective multidimensional knapsack problem, a technique to identify different kinds of relationships (i.e. global, local and composite) between objectives was proposed. The

technique also could identify interesting trade-offs in the fitness landscape. Basically, the proposed technique consists of four steps: global pairwise relationship analysis, objective range analysis, trade-off regions analysis and multi-objective scatter-plot analysis.

Solving a single objective as a multi-objective optimisation problem could also help in dealing with local optima by decomposing a single objective function into some (conflicting) objective function, as implemented independently by Knowles et al. [179] and Jensen et al. [161].

2.6 Multi-Objective Examination Timetabling Problems

Examination timetabling problems are inherently multi-objective optimisation problems [51, 96, 69]. Rather than being solved as a single objective optimisation problem, in [51], a multi-criteria approach was used. The number of criteria is defined with respect to a number of exam timetabling constraints, i.e. nine criteria representing room capacities, proximity of exams, and time and order or exams, over the Nottingham dataset. In [96] two conflicting objective functions were investigated, number of clashes and timetable length or the number of timeslots. In [69] the multi-objective aspect of ITC 2007 examination timetabling problem formulation was studied by grouping the standard single (aggregated) objective function based on four stakeholders: students, invigilators, students and estates.

Most prior works investigating multi-objective approach for examination timetabling problems were conducted over Carter Dataset, an uncapacitated examination timetabling problem. Moreover, in addition to treating each constraint as separated objective function, such as in [239, 243], the objectives are minimising the standard objective function, e.g. proximity cost (see Section 2.1.1) and minimising the total number of timeslots. Such works are reported in [323, 108, 96, 97, 224].

In [239] a bi-objective examination timetabling problem was studied. The objectives are based on the soft constraints defined in the problem formulations, i.e. **within-group** and **between-group**. In **within-group** constraint examinations in the same group must occur on different days whereas in **between-group** constraint, examinations in a group must not occur simultaneously with any examination in certain other groups. In this study, the problem was solved using multi-objective evolutionary algorithms.

Furthermore, in [243], two multi-objective examination timetabling problems, bi-criteria and nine-criteria problem, were also studied. The bi-criteria problem is based on [71],

in which the objective function consists of the number of conflicts where students have to sit two exams in adjacent periods, and the number of conflicts where students have exams in overnight adjacent periods. The nine-criteria problem in which the objective functions are shown by Table 2.7 is based on [51]. In this study, a great deluge local search algorithm guided with a trajectory to the reference point was employed. During the search, the weight of each criteria is changed dynamically. Within this approach a reference point should be specified in advance, either manually or by choosing among the set of solutions, generated by any automated method.

Table 2.7: The Objective Functions of Nine-Criteria Examination Timetabling Problem

Criterion	Description
C_1	Number of times that room capacities are exceeded.
C_2	Number of conflicts where students have exams in adjacent periods on the same day.
C_3	Number of conflicts where students have two or more exams in the same day.
C_4	Number of conflicts where students have exams in adjacent days.
C_5	Number of conflicts where students have exams in overnight adjacent periods.
C_6	Number of times that students have exams that are not scheduled in period of proper duration.
C_7	Number of times that students have exams that are not scheduled in the required time period.
C_8	Number of times that students have exams that are not scheduled before/after another specified exams.
C_9	Number of times that students have exams that are not scheduled immediately before/after another specified exams.

In [323], hybrid multi-objective evolutionary algorithm (MOEA) was implemented with two objectives namely minimising the number of students having to take consecutive exams and the number of timeslots or timetable length, with direct representation, in which a solution is represented as a vector of exams and the value of each element is the timeslot. Different from the original MOEA, the crossover operators are replaced by two local search operators based on Tabu search algorithm. The first local search is to repair infeasible solution and the second local search is to minimise proximity cost. The algorithm starts with a random solution disregarding its feasibility. Simple 1-move neighbourhood structure, i.e. move a conflicting exam to another timeslot, is utilised in the first local search. The second local search is a simplified variable neighbourhood search [213] with two neighbourhood structures, i.e. Kempe chain and 1-move. Furthermore, the solutions are ranked base on Pareto Strength concept in SPEA-II multi-objective evolutionary algorithm [177]. Mutation operator is implemented by assigning a random timeslot to selected exams. Each exam has a pre-defined mutation rate. The proposed algorithm was tested over 5 Carter dataset and 1 Nottingham dataset. The

same algorithm with two objectives (Carter proximity cost and timetable length) was implemented in [108] and tested over 15 uncapacitated examination timetabling problem datasets from the Carter, Nottingham, and Melbourne datasets.

In [97] which extended the work of [96], multi-objective evolutionary algorithm was also implemented. Similar to [323], the objectives are minimising the number of clashes and minimising timetable length. A variable-length chromosome representation [295] was used. A chromosome directly represents a complete and feasible timetable. The standard crossover operator is substituted by a day-exchange operator with period control operator. The mutation operator is a light mutation operator, such as in [74], in which a number of exams are removed from selected chromosomes and reinserted into randomly selected periods while maintaining its feasibility. Extended from [74], the reinsertion process utilises graph colouring heuristics to order the removed exams. The solutions are ranked based on a goal-based Pareto fitness ranking scheme that is an extension of Pareto fitness ranking scheme [140] by increasing the rank if the timetable length is not within the desired timetable length range. It is essentially similar to the goal-sequence domination scheme [296]. For exploiting optimal solution, two local search operators are incorporated: micro-genetic algorithm (GA with small population and short evolution) [127] and hill-climber. The local search operators over single objective function. The proposed algorithm was tested over 5 Carter dataset and 1 Nottingham datasets. The desired timetable length is set to three above and three below the specified timeslots.

In [97] it was claimed that the proposed approach performed well in comparison with seven other recent and established optimization techniques and produced the best results for four out of the seven publicly available datasets tested. However, it should be noted that the objective was modified from the original version over which the compared techniques tested. It is also questionable how the results of [97] could be compared with prior results due to their different objectives, i.e. number of clashes (total students having two consecutive exams versus proximity cost).

A similar work was carried out in [224], in which two objective functions optimising the standard objective function (i.e. the spread of examinations for individual students) and minimising timetable length were optimised simultaneously. In this study, a Memetic algorithm together with greedy algorithm were employed to solve the problem.

While there have been initial works for solving examination timetabling approach, no previous literature reported multi-objective approach for ITC 2007, though the multi-objective nature of this problem was identified in [69].

2.7 The Performance Assessment of Algorithms over Multi-Objective Problems

One of important study of algorithms for solving multi-objective problem (MOP) is how to measure the performance of the algorithm or in other words what quality indicators could be used over the solutions of multi-objective problems. As stated in [334], the goals of optimisation within a MOP itself consist of: minimal distance to the Pareto-optimal front, good distribution, and maximum spread. To measure the performance of algorithms over multi-objective optimisation problems, some quantitative metrics have been proposed in the scientific literatures. The commonly used quality indicators are two complementary measures: coverage metric ($\mathcal{C}$) [334], and hypervolume [335].

Given $A, B \subseteq X$ are two sets of Pareto solutions represented by decision vectors, the $\mathcal{C}(A, B)$ is defined as:

$$\mathcal{C}(A, B) = \frac{|\{b \in B \mid \exists a \in A : a \preceq b\}|}{|B|} \tag{2.16}$$

As a reminder, the symbol $\preceq$ means 'dominated'. The range of value $\mathcal{C}(A, B)$ is $[0, 1]$. The value of coverage indicates the proportion of Pareto solution `B` dominated by at least one solution from Pareto solution `A`. The value `0` means there is not a solution in `B` dominated by any solution from `A`. Conversely, the value `1` means that all solutions in `B` are dominated by at least a solution from `A`. By this definition, with respect to Pareto solution `A`, the greater of the value the better.

Hypervolume, on the other hand, is a measure of the size of dominated space, therefore the bigger the better. Suppose $A = x_1, x_2, x_3, ..., x_k \subseteq X$ is a set of k decision vectors. Hypervolume of A, $\mathcal{S}(A)$ in [335] is defined as: "the volume enclosed by the union of the polytopes $p_1, p2, p3, ...p_k$ where each p_i is formed by the intersections of the following hyperplanes arising out of x_i , along with the axes: for each axis in the objective space, there exists a hyperplane perpendicular to the axis and passing through the point $(f_1(x_i), f_2(x_i), ..., f_k(x_i))$. In the two-dimensional case, each p_i represents a rectangle defined by the points $(0, 0)$ and $(f_1(x_i), f_2(x_i))$."

The other quality indicators include spread [121], generalised spread [330], additive epsilon [331], generational distance [315], and inverted generational distance [315]. All these quality metrics are to be minimised.

Spread (Δ) measures the extent of spread achieved among the obtained solutions and it is formally defined as:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} \left| d_i - \bar{d} \right|}{d_f + d_l + (N-1) + \bar{d}}. \tag{2.17}$$

In Equation 2.17, the parameters d_f and d_l are the Euclidean distances between the extreme solutions and the boundary solutions of the obtained non-dominated set respectively. Whereas parameter d_i is Euclidian distance between consecutive solutions in the obtained non-dominated set of solutions, parameter $\bar{d}$ is the average Euclidian distance, and parameter N is the number solutions. Since the original spread definition only works over two-objective problems, in [330], it is extended to cope with multi-objective problems with number of objectives greater than two. It is then called generalised spread.

The idea of Epsilon and also additive epsilon [331] are similar to Coverage, but instead of standard domination concept, the notion $\epsilon-$domination is used instead. Generational distance and also inverted generational distance [315] basically measure the distance between the known Pareto solutions and the true Pareto solutions. Therefore, these metrics only suitable for the problems with known true Pareto solutions.

In this study, we use coverage and hypervolume as quality indicators. These metrics are chosen because within this metric the true Pareto-optimal solution is not required and in the examination timetabling problems studied in this thesis the true Pareto-optimal solution is not known. On the other hand, compared to other metrics, e.g epsilon and additive epsilon, the calculation of these metrics are simpler. In addition, to measure different aspect of MOP solutions, Spread is also used to measure the distribution or spread of the solutions in the objective space.

2.8 Fairness in General

The concept of fairness (also 'balance' or 'evenness') have been thoroughly investigated in the field of political science and political economics. Some common sense definitions of fairness in these fields are discussed in [137] and [242]. In [137], fairness is defined as an allocation where no person in the economy prefers anyone else's consumption bundle over his own whilst [242] defines fairness as a fair allocation that is free of envy. The fairness issues have well studied in the field of computer networks in terms of, such as, fair resource distribution among entities [123, 16, 187, 266, 329] and fair congestion control [158, 214]. In the field of operations research, fairness issues has been investigated for particular problem such as in flight landing scheduling (see [292] and [317]).

In general resource allocation, there are two well-accepted notions of fairness criteria i.e.

max-min fairness (or min-max in minimisation problem) and proportional fairness. The max-min fairness is stemmed from political sciences [263] that in essence is to maximise the utility of the worst-off individual. Max-min fairness is a qualitative measure to evaluate whether an allocation is fairer than another allocation but does not measure how much fairer.

Proportional fairness, on the other hand, was firstly introduced by [172] for network resource allocation problems. Proportional fairness is the generalisation of two-player problem based Nash solution notion. In Nash solution, resources allocation between two players is considered as favourably fair if the percentage increase in the utility of one player is larger than the percentage decrease in utility of the other players (see [40] for detailed elaboration). Proportional fairness is Pareto efficient, while max-min fairness is not necessarily Pareto efficient. However, the refinement of max-min called leximin is always Pareto optimal [202].

There have been some quantitative measurement of inequality in resource allocation such as Gini Index [144] and Jain Fairness Index [159]. The Gini index can be written as an expression shown by Equation 2.18 and Equation 2.19 as elaborated in [37] and [38] respectively. In the context of examination timetabling problem, proportional fairness could be improved by making the distribution of penalty associated to each individual student as even as possible. Whereas the max-min fairness could be improved by decreasing penalties associated to the student with the highest penalty.

$$I_G = \sum_{j=1}^{n} s_j \left[(n-j)/n - (j-1)/n \right] \tag{2.18}$$

$$I_G = (e' G s) \tag{2.19}$$

Where s_j is the proportion of resource allocated to individual whose resource has j^{th} rank in resource allocation for n individuals. It is assumed that $s_1 \geqslant s_2 \geqslant ... \geqslant s_j \geqslant ... \geqslant s_n$. Further, e' is row vector of n elements equal to $1/n$, G is an n by n matrix with element:

$$g_{ij} \begin{cases} -1 & \text{if } j > i \\ +1 & \text{if } i > j \\ 0 & \text{if } i = j \end{cases} \tag{2.20}$$

Jain fairness index and min-max fairness , can be defined as given by Equation 2.21 and Equation 2.22. s_i is total of resource allocated to individual i.

$$I_J = \frac{|\sum_{i=1}^{n} s_i|^2}{n \sum_{i=1}^{n} s_i^2} \tag{2.21}$$

$$I_M = \frac{min\{s_1, s_2, ..., s_i, ..., s_n\}}{max\{s_1, s_2, ..., s_i, ..., s_n\}} \tag{2.22}$$

The value of I_G and I_J range from 0 to 1. In I_G, the value 0 shows complete equality and 1 shows complete inequality. Contrarily, in I_J the value 0 shows complete inequality and 1 shows complete equality.

Other options, given the average resource allocated to individual is μ, the classical statistical metric such as variance (I_v) and relative standard deviation (I_R) or coefficient of variance could be used to measure fairness (see Equation 2.23-Equation 2.25). However, different from I_G and I_J, the value range of Variance is not bounded between 0-1. It makes difficult to interpret its value. These fairness quality indicators are summarised in Table 2.8. How these fairness metrics are used within the context of examination timetabling problems will be discussed in the problem formulation in Chapter 4.

$$I_v = \frac{1}{n} \sum_{i=1}^{n} s_i - \mu \tag{2.23}$$

$$\mu = \frac{1}{n} \sum_{i=1}^{n} s_i \tag{2.24}$$

$$\frac{\sqrt{I_v}}{\mu} \tag{2.25}$$

Table 2.8: Summary of Fairness Quality Indicator

Fairness Quality Indicator	Symbol	Bounded Value Range	To be minimised /maximised
Jain Fairness Index (JFI)	I_J	(0-1]	To be maximised
GINI Index	I_G	(0-1]	To be minimised
Min Max Index	I_M	[0-1]	To be maximised
Variance	I_V	Unbounded	To be minimised
Relative Standard Deviation(RSD)	I_R	Unbounded	To be minimised

2.9 Fairness in Timetabling Problems

Standard examination timetabling problems formulations [89, 233, 207] have penalties for violations of various soft constraints, including those which influence spread of examinations across the overall time period, and so give students more time for prepara-

tion. There is a limited number of prior work explicitly dealing with fairness issues in timetabling.

Castro and Manzano [91] proposed a formulation of the balanced academic curriculum problem, which requires assignment of courses (modules) to periods for teaching, while respecting the prerequisite structure among the courses and balancing the students load, which can be considered as a fairness issue. In this study, the notion of max-min fairness is used. The objective function is to minimise the maximum academic load for all periods.. Let c_i be the academic load of period i and $\forall i = 1, ..., n$. The objective is formally written as:

$$Min \;\; c = Max \;\; \{c1, ..., cn\} . \tag{2.26}$$

Constraint programming techniques were used to solve the problem. This formulation was later extended by Gaspero and Schaerf [143] and Chiarandini et al. [98]. In [143] and [98], generalised balanced academic curriculum problem was proposed. Instead of considering 'balance' as the maximum academic load for all periods, in this work 'balance' is counted by summing up all the deviations from the average number of credits per period for each curriculum. In other words, proportional fairness notion was used instead of max-min fairness. A new objective, professors preference, was added to 'balance' objective function. However, this problem was still treated as single objective function using weighted summation. Hybridisation of simulated annealing, dynamic tabu search and large-neighbourhood search was employed to solve the problem.

The most relevant work was the study of fairness in curriculum based course timetabling problem by [217], extended in [218]. This study aims to distribute the penalty (i.e. soft constraints violation) among stakeholders in a fair way. Two notions of fairness, max-min fairness [39] and Jain fairness index (JFI)-based fairness [159], were studied over course timetabling problem dataset from course timetabling problem track of international timetabling competition 2007 (ITC 2007) [43]. In the first variant, the optimisation problem aims at optimising the penalties associated to the worst-off stakeholder (curricula in this case), whereas in the second variant the optimisation problem aims at seeking optimal trade-off (i.e. Pareto optimal solutions) between two objectives: total penalties from the benchmark problem formulation (see [43]) and fair distribution of penalties among curricula measured with JFI. Over the first problem variant, firstly the initial feasible solution is generated using sequential heuristics [70], then the max-min fairness is optimised using simulated annealing algorithm. It is basically a single objective optimisation problem. Over the second problem variant, using the best known solutions as initial solutions, the two objectives are optimised using AMOSA [36], a multi-objective

optimisation algorithm based on simulated annealing. The experimental results showed that over the first problem variant, compared to the best known solutions (in terms of benchmark objective function) the proposed algorithm could produce fairer solutions (in terms of max-min fairness notion) for 20 out of 32 instances, whereas over the second problem variant the proposed algorithm could make the best-known solutions fairer in exchange for a small increase in the total number of soft constraint violations.

In addition, in [93], a notion of 'balanced' examination timetable was proposed. Firstly, rather than assuming that all exams have the same 'critical' level, a fuzzy-criticality level based on credits, success ratios, and type of the classes for each exam was proposed. Secondly, balanced workloads of the students in terms of the criticality levels of the classes and even distribution of exams for each professor are set as objective functions in the problem formulation. Goal programming model, which assumes that each goal has an equal importance used to solve the problem, was used to solve the problem. The limitation of this work is that the proposed model was implemented over a small and unrealistic problem. In addition, within the goal programming approach, it is assumed that the ideal goal is known, which is not always the case in the real-world problems.

Apart from academic timetabling problems, fairness also has been studied in other timetabling/scheduling areas such as nurse rostering problems. Aiman et al. [155] discussed the results from a survey conducted among nurses in Malaysian public hospitals, emphasizing the importance of fairness among nurses in rosters to the nurses in terms of workload balance and respecting their preferences. No model was proposed in this work. Smet et al. [289] proposed the use of fairness model within objective functions to produce fair nurse rosters. Max-min fairness is used as the objective function, in which the quality of the worst individual roster determines the fairness measure. To solve the proposed problem model, similar to [288], a hyper-heuristic was used. The experiment over the real-world dataset shows that the max-min fairness objective function could produce fairer solutions (measured by relative standard deviation and difference between the worst and the best roster) than solutions produced with the standard benchmark objective function for all eight problem instances. However, for half of the problem instances, the max-min fairness objective function causes the standard objective function to worsen.

Martin et al. [201] tested a range of fairness models embedded into objective functions under an agent-based cooperative meta-heuristic search for fair nurse rostering using the Belgian hospital benchmark [288]. In this study, four fairness-based objective functions were evaluated. Let $q_{ros}(n)$ be the cost of assigning nurse n to a given shift, p_{viol}

is the coverage constraints, and the average individual roster quality of the nurses is defined as $\mu = \frac{1}{|N|}\sum_{n\in N} q_{ros}(n)$. The four fairness objective functions are given in Equations 2.27- 2.30:

$$MinMax = \min\left(|N|\left(\underset{n\in N}{\text{Max}}\ q_{ros}(n)\right) + p_{viol}\right) \tag{2.27}$$

$$MinDev = \min\left(\left(\sum_{n\in N}(|\mu - q_{ros}(n)|) + |N|\right) + p_{viol}\right) \tag{2.28}$$

$$MinError = \min\left(|N|\left(\left(\underset{n\in N}{\text{Max}}\ q_{ros}(n) - \underset{n\in N}{\text{Min}}\ q_{ros}(n)\right) + \mu\right) + p_{viol}\right) \tag{2.29}$$

$$MinSS = \min\left(w_1 \sum_{n\in N} q_{ros}(n)^2 + w_2\ p_{viol}^2\right) \tag{2.30}$$

The agent-based cooperative search incorporates three meta-heuristics: tabu search, simulated annealing, and variable neighbourhood search. The results show that all fairness objective function could produce fairer rosters. Moreover, if all agents using the same fairness objective function, $MinDev$ resulted the fairest rosters whereas the generic weighted sum objective function produced the least fair roster. However, overall, the combined objective functions, in which each cooperating meta-heuristic using a different fairness objective function yielded the fairest rosters.

In [150], a new fairness objective based on a lexicographic rule that offers a beneficial trade-off between roster quality and fairness was proposed for personal rostering problem. P^{lexi} is defined by Equation 2.31, which is a permutation of all individual penalties, sorted in a non-increasing order. Under this definition, roster a is said to be fairer than roster b, if the first non-zero component of the vector $P^{lexi}(a) - P^{lexi}(b)$ is negative.

$$P^{lexi} = (P'_{H,1}, P'_{H,2}, ..., P'_{H,n})\ \ s.t.\ \ P'_{H,1} \geqslant P'_{H,2} \geqslant ... \geqslant P'_{H,n} \tag{2.31}$$

From the above survey on previous works tackling fairness in timetabling problems, we know that some fairness objective functions have been proposed. However, none of the previous works, treated fairness objective function within truly multi-objective optimisation problem; a single objective optimisation approach was used instead. In addition, to the best of our knowledge, there is no prior study explicitly investigating

fairness among students in examination timetabling problems.

Chapter 3

Solving Benchmark Examination Timetabling Problems With a Hyper-heuristic

3.1 Introduction

Before solving the examination timetabling problem with fairness, this chapter discusses our proposed hyper-heuristic approaches to solving the benchmark examination timetabling problem instances widely studied in the scientific literatures with their standard single objective function. The aim is to show that our proposed hyper-heuristics perform good enough and are competitive to currently reported approaches in the scientific literature. Hence, the results over examination timetabling problems with fairness will be reliable. We tested our proposed approach over both uncapacitated and capacitated examination timetabling problem instances. Furthermore, in the following chapters we will extend the approaches discussed in this chapter to deal with our new proposed multi-objective examination timetabling problems with fairness as a new objective function.

3.2 Solving The Uncapacitated Examination Timetabling Problems

As discussed in the literature review in Chapter 2, within the uncapacitated examination timetabling problem only timeslot assignment is considered; the room assignment is neglected. The Carter, also known as Toronto examination timetabling problem dataset is the most intensively studied problem instances of this kind of problem. Therefore, we test our proposed algorithm over the Carter dataset.

3.2.1 Method

Basically, in this study, we employ hyper-heuristic approaches within HyFlex[1] framework. As discussed in Section 2.3, the distinctive feature of a hyper-heuristic approach is that there is a barrier between problem domain and hyper-heuristic strategy. This separation is aimed at maintaining the generality of the hyper-heuristic strategy.

3.2.1.1 Constructive Algorithms

Since in the existing version of Hyflex examination timetabling problem is not part of its current available problem domains, we added examination timetabling problem as a new problem domain in the current version of Hyflex. The main methods that have to be implemented by a problem domain within Hyflex framework are method to construct an initial feasible solution and to apply set of pre-defined low-level heuristics. These low-level heuristics are chosen and applied during the iteration of optimal solution search.

To construct an initial feasible solution, we basically adopted saturation degree heuristic method [45] commonly used in a graph colouring problem as shown by Algorithm 1. First, the examination timetabling problem is represented as a graph. In the representative graph, each vertex represents an exam and each edge between two vertices indicates that there is at least one student enrolled in two exams represented by the connected vertices. Each connected vertices must not be assigned to the same colour. It reflects two exams with common students must not scheduled in the same timeslot.

To estimate the minimum number of different colours required to colour all vertices in a graph, in a graph theory, it is known a maximal clique size. A clique is a complete subgraph in which each pair of vertices is connected by an edge(see [87]). In examination

[1] http://www.hyflex.org/hyflex/

timetabling problem context, a clique represents a set of exams, in which any pair of the exam are conflicting. Therefore, the number of vertices in the maximum clique indicates the minimal number of timeslots required to construct a clash-free timetable.

In our proposed algorithm, we employed randomized heuristic adopted from [284] to find a maximal clique in a graph. Then, each vertex in the maximal clique is assigned to a different colour. Having the vertices in the maximal clique are coloured, the remaining uncoloured vertices are ordered by its saturation degree and coloured sequentially. Saturation degree of a vertex, `v` is the number of remaining feasible colour for `v` during the process of assignment. The vertex with smaller saturation degree number have to be coloured earlier than the vertex with greater saturation degree. Ties are solved by choosing the vertices with the same saturation degree randomly. The algorithm stops when all vertices in the graph are assigned to a feasible colour.

Algorithm 1 Sequential Construction Method with Saturation Degree-based Heuristic Algorithm

```
1: procedure initiateSolution(id)
2:     Generate a Graph G from the problem instance.
3:     Find maximal clique MC from G.
4:     //Colour starts from 1
5:     Assign a differing colour number to each vertex, v ∈ MC.
6:     L ← list of remaining uncoloured vertices.
7:     repeat
8:         Order vertices in L by non-increasing saturation degree.
9:         Assign the first vertex, f, from L with a feasible lowest number colour.
10:        Remove f from L.
11:        Update the saturation degree of vertices connected to f.
12:    until All vertices are assigned to colour
13:    store the coloured graph G in solution memory index i.
14: end procedure
```

Another main component of a problem domain within HyFlex as a hyper-heuristic framework is a set of pre-defined low-level heuristic. In this study, four different low-level heuristics i.e. random move, random move two, swap, and swap Kempe chain are employed. These low-level heuristics are summarised in Table 3.1

Table 3.1: Low-level heuristics (LLH) within the uncapacitated examination timetabling problem domain

LLH	Name	Description
$\texttt{LLH}_1$	Random Move	A random exam is chosen and its allocated timeslot is moved to another random timeslot.
$\texttt{LLH}_2$	Random Move Two	Two random exams are chosen and their allocated timeslots are moved to another random timeslot.
$\texttt{LLH}_3$	Swap	Two random exams are chosen and their allocated timeslots are swapped
$\texttt{LLH}_4$	Swap Kempe Chain	Two sets of conflict free exams are chosen and their timeslots are swapped (See [309, 55]).

3.2.1.2 Improvement Method with Hyper-heuristic Strategy

In selection hyper-heuristics, during the searching iteration, as pointed out in [41], there are two strategies that play as key success factors, i.e. low-level heuristic selection and move acceptance strategy. For the low-level heuristic selection strategy, we study two strategies, i.e. simple random selection and self-adaptive selection [238]. Whereas, for move acceptance strategy, we study three strategies, i.e. hill climbing, great deluge algorithm [128], and late acceptance [49] algorithm. To best of our knowledge, the hybridisation we give here of self-adaptive learning as low-level heuristic selection strategy with hill climbing, great deluge, and hill climbing as move acceptance strategies within hyper-heuristic framework especially for solving examination timetabling problems is firstly investigated in this study. Self-adaptive strategy is originally proposed in [238] to choose neighbourhood structures within artificial bee colony algorithm for solving lot-streaming flowshop scheduling problem. These new design aims at improving the previous designs in the literature in order to get new solutions improving the best-known solutions.

Our proposed hyper-heuristics based on self-adaptive combined with hill climbing, self-adaptive combined with great deluge, and self-adaptive combined with late acceptance are given in Algorithm 2, Algorithm 3 and Algorithm 4 respectively. Note that, the variables and methods used in these algorithms are briefly explained in Table 3.2 and Table 3.3.

Table 3.2: List of variables used in Algorithm 2, Algorithm 3 and Algorithm 4

Varible	Description
`P`	Problem domain, i.e. Examination timetabling problem.
`St`	Stopping condition.
`M`	Number of predefined low-level heuristics in `P`.
`LH`	An `ArrayBlockingQueue` of `Integer` to store the temporary indices of low-level heuristics.
`S`	Size of `LH`.
`WH`	A `LinkedList` of `Integer` to store the indices of well-performing low-level heuristics.
S_{best}	The best solution's objective function value so far.
`l`	A chosen low-level heuristic.
`newFunctionVal`	The objective function value of new solution produced by applying a low-level heuristic `l`.
S_{cur}	The current solution's objective function during iteration.
`B`	Boundary level.
α	Decay rate.
`N`	number of runs/iterations.
`R`	An `ArrayBlockingQueue` of `Double` to store the values of the accepted solutions' objective function.
`L`	Size of `R`.
`c`	selected element index in `R`.

Table 3.3: List of methods used in Algorithm 2, Algorithm 3 and Algorithm 4

Method	Description
`P.getNumberOfHeuristics()`	Get the number of low-level heuristics in problem domain `P`.
`getRand(M)`	Generate a positive random real number less than `M`.
`P.initiateSolution(i)`	Generate an initial feasible solution using Algorithm 1 and store it in solution memory index `i`.
`P.getFunctionValue(i)`	Get objective function value of solution stored in solution memory index `i`.
`LH.poll()`	Get the element of `LH` with index `0` and delete it from `LH`.
`P.applyLLH(l,i,j)`	Apply a predefined low-level heuristics in problem domain `P` with index `l` over solution stored in solution memory `i` and stored the new solution in solution memory index `j`. Return the objective function value of the new solution.
`P.copySolution(i,j)`	Copy solution in solution memory index `i` to solution memory index `j`.
`WH.add(l)`	Add low-level heuristic index `l` to well-performed low-level heuristic indices' list, `WH`.
`R.add(x)`	Add a value `x` to `ArrayBlockingQueue R` (see Algorithm 4).
`R.get(x)`	Get value from `ArrayBlockingQueue R` in index `x` (see Algorithm 4).
`R.poll()`	Delete permanently the element in `ArrayBlockingQueue R` with index `0` (see Algorithm 4).
`R.put(a)`	Insert a into `ArrayBlockingQueue R` (see Algorithm 4).

Within the self-adaptive strategy, there is a mechanism in order to make better performing low-level heuristics to be chosen more frequently than the worse performing ones. As shown by Algorithm 2-4, a fixed sized list of low-level heuristics indices predefined in problem domain, `LH`, is maintained. Initially, the list is filled with randomly chosen indices of problem domain's low-level heuristics (see line 7-9 in Algorithm 2-4). The low-level heuristics whose indices stored in the list is applied and then deleted from the list sequentially during the iterations. If a low-level heuristic produced an accepted solution, then its index is stored in a dynamic sized list of well-performed low-level heuristics indices, `WH` (see line 27, 30, 35 in Algorithm 2, 3, and 4 respectively). Whenever the list of low-level heuristic indices is empty, 75% of the capacity are filled with element of well-performed low-level heuristic indices list, whereas the rest are filled with randomly chosen indices of predefined low-level heuristic in problem domain (see line 17-19 in Algorithm 2).

Within the hill-climbing move acceptance strategy, only an improving candidate solution is accepted to replace the current solution. Whereas in great deluge and late acceptance strategies, there is a mechanism to skip from local optima trap by allowing to accept worse solution. As shown by Algorithm 3, to escape from local optima trap, within great deluge algorithm, in each iteration, a better solution is always accepted. In addition, the worse solution is also accepted as long as it does not exceed the boundary level. During the iteration, the boundary level is decreasing gradually.

In late acceptance method as shown by Algorithm 4, the new candidate solution is compared to previous solutions from prior iterations. A list `R` of size `L` is maintained to record the objective function values of accepted solution from each iteration. In each iteration, a candidate solution is compared to solution from the list at position `c`. Where `c` is defined by Equation 3.1. The variable `i` in the equation is the iteration number.

$$c = i \mod L \tag{3.1}$$

3.2.2 Experimental Environment

In this study, the algorithm was implemented in Java operating under Windows 7. The hardware specification where this experiment was run was: Processor: Intel(R) Core(TM)i7-3820 CPU @ 3.60GHz 3.60 GHz, Installed memory (RAM): 16.0 GB.

Algorithm 2 Self-adaptive - Hill Climbing (SA-HC) Based Hyper-heuristic. The methods and variables are briefly explained in Table 3.2 and Table 3.3 respectively

```
1: procedure SOLVE(ProblemDomain P)
2:     Set stopping condition St
3:     Set M ← P.getNumberOfHeuristics()
4:     Set an ArrayBlockingQueue of Integer with size S to store the indices of low-level heuristics, LH
5:     Set a LinkedList of Integer to store the indices of well-performing low-level heuristics, WH
6:     // Fill LH with the index of low-level heuristics predefined in P chosen randomly
7:     for i=0 to S-1 do
8:         LH.add(getRand(M))
9:     end for
10:    P.initiateSolution(0)
11:    S_best ← P.getFunctionValue(0)
12:    while St is not met do
13:        if LH is not empty then
14:            // get the earliest element of LH and delete it from LH
15:            l ← LH.poll()
16:        else
17:            fill 75% elements of LH with elements from WH
18:            fill 25% elements of LH with the index of low-level heuristics predefined in P chosen randomly.
19:            Clear the element of WH
20:            l ← LH.poll()
21:        end if
22:        newFunctionVal ← P.applyLLH(l,0,1)
23:        if newFunctionVal ≤ S_best then
24:            //the new solution is accepted
25:            P.copySolution(1,0)
26:            S_best ← newFunctionVal
27:            WH.add(l)
28:        end if
29:    end while
30:    return S_best
31: end procedure
```

Algorithm 3 Self-adaptive - Great Deluge (SA-GD) Based Hyper-heuristics. The methods and variables are briefly explained in Table 3.2 and Table 3.3 respectively

```
1: procedure SOLVE(ProblemDomain P)
2:     Set stopping condition St
3:     Set M ← P.getNumberOfHeuristics()
4:     Set a list of low-level heuristic indices, LH of size S
5:     Set a list of Well-performed low-level heuristic indices, WH
6:     // Fill LH with the index of low-level heuristics predefined in P chosen randomly
7:     for i=0 to S-1 do
8:         LH.add(getRand(M))
9:     end for
10:    P.initiateSolution(0)
11:    S_best ← P.getFunctionValue(0)
12:    S_curr ← P.getFunctionValue(0)
13:    B ← P.getFunctionValue(0)
14:    Set decay rate α
15:    while St is not met do
16:        if LH is not empty then
17:            // get the earliest element of LH and delete it from LH
18:            l ← LH.poll()
19:        else
20:            fill 75% elements of LH with elements from WH
21:            fill 25% elements of LH with the index of low-level heuristics predefined in P chosen randomly.
22:            Clear the element of WH
23:            l ← LH.poll()
24:        end if
25:        newFunctionVal ← P.applyLLH(l,0,1)
26:        if newFunctionVal≤ S_curr OR newFunctionVal ≤ B then
27:            //the new solution is accepted
28:            P.copySolution(1,0)
29:            S_curr ← newFunctionVal
30:            WH.add(l)
31:            if newFunctionVal<S_best then
32:                S_best ← newFunctionVal
33:            end if
34:        end if
35:        B ← B-α
36:    end while
37:    return S_best
38: end procedure
```

Algorithm 4 Self-adaptive - Late Acceptance (SA-LA) Based Hyper-heuristic. The methods and variables are briefly explained in Table 3.2 and Table 3.3 respectively

```
1: procedure SOLVE(ProblemDomain P)
2:     Set stopping condition St
3:     Set M ← P.getNumberOfHeuristics()
4:     Set a list of low-level heuristic indices, LH of size S
5:     Set a list of Well-performed low-level heuristic indices, WH
6:     // Fill LH with the index of low-level heuristics predefined in P chosen randomly
7:     for i=0 to S-1 do
8:        LH.add(getRand(M))
9:     end for
10:    P.initiateSolution(0)
11:    S_best ← P.getFunctionValue(0)
12:    //set R, list (queue) of size L to record the accepted solutions' objective function
13:    Set R ← new Queue(L)
14:    for i=0 to L-1 do
15:       R.add(P.getFunctionValue(0))
16:    end for
17:     i ← 0
18:    while St is not met do
19:       if LH is not empty then
20:          // get the earliest element of LH and delete it from LH
21:          l ← LH.poll()
22:       else
23:          fill 75% elements of LH with elements from WH
24:          fill 25% elements of LH with the index of low-level heuristics predefined in P chosen randomly.
25:          Clear the element of WH
26:          l ← LH.poll()
27:       end if
28:       newFunctionVal ← P.applyLLH(l,0,1)
29:       c ← i mod L
30:       if newFunctionVal≤ R.get(c) then
31:          //the new solution is accepted
32:          P.copySolution(1,0)
33:          R.poll()
34:          R.put(newFunctionVal)
35:          WH.add(l)
36:          if newFunctionVal<S_best then
37:             Set S_best ← newFunctionVal
38:          end if
39:       end if
40:       i++
41:    end while
42:    return S_best
43: end procedure
```

3.2.3 Data Structure and Algorithm Implementation

In Algorithm 1 the examination timetabling problem is modelled as graph-colouring problem. In this graph colouring model, each examination is represented as a vertex or node object. The conflict between two examinations, meaning that at least there is a student enrols both examinations, is represented as an edge connecting between those two vertices.

Within the examination timetabling problem defined, a two-dimensional array is devised to store the conflict matrix. The solution is stored in 1-dimensional plain array, where the index represents the examination index and the value represents the timeslot. To speed up the computation of objective function value or the fitness, delta evaluation [268], which is computing the different objective function value between moves instead of typical cost evaluation for each single move, is adopted.

3.2.4 Parameter Setting

It is broadly known that parameter tuning plays important role in meta-heuristic approach. However in our approach, we attempt to make the parameter tuning as less as possible and as general as possible. In self-adaptive-great deluge based hyper-heuristic shown by Algorithm 3, the decay rate is set by Equation 3.2. Where B_0 is the objective function value of initial solution, Sol_{ref} is a reference or target solution, and N_{iter} is number of iterations. The target solutions are best-known solutions reported in scientific literature (see Table A.2 in Appendix A). It is very common way to set up the decay rate in great deluge algorithm. For the other parameters such as the size of low-level heuristic list in self-adaptive learning mechanism, and the size of a list of accepted solutions' objective function value, we will conduct preliminary experiment discussed in the following section.

$$\delta R = \frac{B_0 - f(Sol_{ref})}{N_{iter}} \tag{3.2}$$

3.2.5 Experimental Scenario

We tested our proposed hyper-heuristics given in Algorithms 2-4 over Carter dataset. Moreover, as a comparators, we also combined the three move acceptance strategies with simple random strategy to select the low-level heuristic instead of self-adaptive strategy. Therefore, we have six hyper-heuristics summarised in Table 3.4. These hyper-heuristics include simple random(SR) and self-adaptive(SA) learning with three move acceptance

strategies: late acceptance(LA), hill climbing(HC), and great deluge(GD) algorithm. The combination with self-adaptive is the innovative design in this study.

Table 3.4: Combinations of two low-level heuristic selection strategies.

No.	Combinations
1	SR-GD
2	SR-HC
3	SR-LA
4	SA-GD
5	SA-HC
6	SA-LA

3.2.6 Preliminary Experiment

We conducted preliminary experiments to find optimal parameter setting in the proposed algorithms. Firstly, the preliminary experiment was aimed at finding optimal size of low-level heuristic list of self-adaptive learning mechanism to select a low-level heuristic as shown by the Algorithm given in Algorithm 2-4. We tested six different size of low-level heuristic list (`S`), i.e `5, 10, 50, 100, 150, and 200` over five randomly chosen problem instances of Carter dataset. From the preliminary experiment we found that the best size of low-level heuristics, `S`, is `200`, though, the differences were small.

Secondly, the preliminary experiment was aimed at finding the optimal size of list in the late acceptance strategy (`L`). Our preliminary experiment by comparing the size of list: `100, 200, 500, 1000, and 2000` suggests that the size of `500` is in the best rank as shown by Table 3.5.

Table 3.5: Comparing differing values of parameter L in the late acceptance strategy

Instance	l=100		l=200		l=500		l=1000		l=2000	
	OF	Rank	OF	Rank	OF	Rank	OF	Rank	OF	Rank
CAR-F-92-I	4.26	3	4.13	1	4.25	2	4.47	4	4.73	5
CAR-S-91-I	4.96	1	5.01	2	5.13	3	5.44	4	5.96	5
EAR-F-83-I	157.12	3	157.13	4	157.03	1	157.18	5	157.06	2
HEC-S-92-I	8.09	1	8.24	3	8.16	2	8.32	4	8.51	5
KFU-S-93	3.42	2	3.41	1	3.45	3	3.64	4	3.98	5
LSE-F-91	25.83	5	25.17	3	25.39	4	24.95	1	24.96	2
PUR-S-93-I	37.35	4	36.34	2	36.78	3	36.12	1	38.41	5
RYE-F-92	34.99	5	33.62	2	33.51	1	33.73	3	34.58	4
STA-F-83-I	10.74	5	10.60	4	10.47	2	10.50	3	10.43	1
TRE-S-92	13.76	5	13.53	3	13.06	1	13.76	4	13.46	2
UTA-S-92-I	11.07	5	10.69	3	10.18	1	10.43	2	10.96	4
UTE-S-92	4.88	1	5.22	2	5.80	3	6.31	4	6.90	5
YOR-F-83-I	8.60	2	8.27	1	8.68	3	9.16	4	10.08	5
Avg. Rank		3.23		2.38		**2.23**		3.31		3.85

3.2.7 Experimental Result and Discussion

Before testing over all 13 problem instances of Carter dataset, we are interested to know the different correlation between number of iterations and the objective function value, of the proposed approaches. Therefore, we tested the proposed approaches over `CAR-F-92` and plot the objective function in each iteration. We set `3,000,000` iterations as stopping condition (`St`). The objective function values during the iteration are visualised in Figure 3.1.

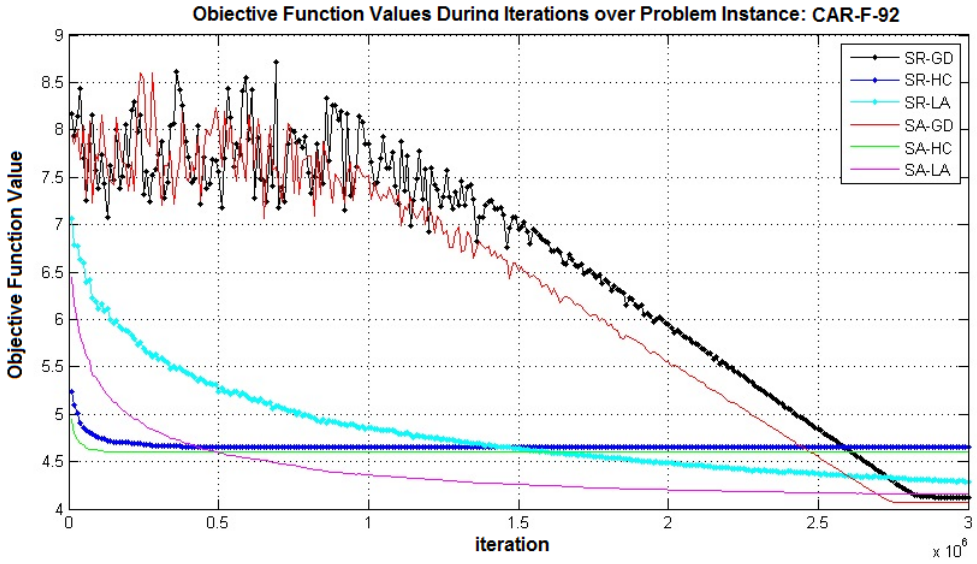


Figure 3.1: The objective function values during the iterations of the proposed hyper-heuristic over problem instance CAR-F-92

As shown by Figure 3.1, hill climbing search is quick to converge, but it is also easily too early trapped in local optimum, since only improving solution will be accepted. Contrarily, the great deluge search is scattered at the beginning but eventually converge better at the end. Because in great deluge search, a worsening solution, as long as it is not worse than boundary level, is accepted, it prevents the search to be trapped into local optimum too early. On the other hand, late acceptance performs in between. Overall, great deluge algorithm outperforms hill climbing and late acceptance mechanism. It is also clear that self-adaptive low-level heuristic selection method could improve the performance of the algorithm, whichever move acceptance strategy is used. The figure also shows that the combination of self-adaptive as low-level heuristic selection strategy and great deluge algorithm as move acceptance strategy (SA-GD) outperforms the other combinations.

In order to know whether the experimental results over `CAR-F-92-I` are consistent on the rest of problem instances, we run the proposed approaches over `13` Carter dataset.

Table 3.6: Experimental results over Carter Dataset (standard objective function) using combination different low-level heuristic selection and move acceptance strategies.

Instance	BEST						AVERAGE					
	SA-GD	SA-HC	SA-LA	SR-GD	SR-HC	SR-LA	SA-GD	SA-HC	SA-LA	SR-GD	SR-HC	SR-LA
CAR-F-92-I	**3.93**	4.46	4.00	4.07	4.47	4.21	**4.05**	4.60	4.14	4.15	4.69	4.30
CAR-S-91-I	**4.78**	5.29	4.86	4.93	5.29	5.22	**4.87**	5.47	4.97	4.98	5.49	5.33
EAR-F-83-I	33.03	36.77	**32.97**	33.15	37.63	**32.97**	33.56	37.95	33.97	33.75	39.11	**33.45**
HEC-S-92-I	10.39	11.24	**10.31**	10.41	11.24	10.32	10.64	11.49	10.66	10.61	11.53	**10.50**
KFU-S-93	13.24	13.90	13.37	**13.23**	13.79	13.27	**13.50**	14.60	13.71	13.53	14.72	13.67
LSE-F-91	**10.07**	11.50	10.33	10.33	11.66	10.36	**10.44**	11.95	10.74	10.57	12.10	10.68
PUR-S-93-I	**4.71**	5.22	5.21	4.82	5.22	6.06	**4.76**	5.33	5.34	5.00	5.38	6.15
RYE-F-92	**8.27**	9.18	**8.27**	8.31	8.98	8.67	8.37	9.42	**8.35**	8.49	9.32	8.94
STA-F-83-I	157.08	157.37	157.13	157.28	157.36	**157.05**	157.08	157.37	157.13	157.28	157.36	**157.05**
TRE-S-92	8.08	8.75	8.05	8.20	8.77	**8.02**	**8.21**	9.06	**8.21**	8.30	9.06	8.23
UTA-S-92-I	**3.28**	3.62	3.33	3.32	3.62	3.56	**3.35**	3.73	3.39	3.39	3.72	3.61
UTE-S-92	**24.81**	25.36	24.85	24.95	25.84	24.91	**25.01**	25.83	25.00	25.09	26.54	24.93
YOR-F-83-I	**35.35**	38.81	35.79	35.94	38.68	35.90	**36.25**	39.83	36.68	36.66	39.76	36.42

For each problem instance, we run the algorithm 11 times with `3,000,000` iterations as stopping condition in each run.

Table 3.6 presents the experimental results over all `13` problem instances and Table 3.7 compares our approaches with other selected approaches recently published in scientific literature. From Table 3.6 it is clear that the combination of self-adaptive and great deluge algorithm (SA-GD) outperforms the other combinations over almost all problem instances. It can also be observed from Table 3.7 that our proposed approaches outperforms the selected benchmark recently published in the literature over almost all problem instances.

Table 3.7: Comparing SA-GD and SA-LA approaches with results reported in the selected recently published scientific literature.

Instance	Our Approach		BENCHMARK (BEST)				
	SA-GD	SA-LA	(Burke,2012) [63]	(Sabar,2012) [280]	(Rahman,2014) [3]	(Burke,2014) [82]	(Abdullah,2013) [7]
CAR-F-92-I	**3.93**	4.00	4.22	4.7	4.41	4.31	3.94
CAR-S-91-I	4.78	4.86	5.03	5.14	5.12	5.19	**4.76**
EAR-F-83-I	33.03	**32.97**	36.06	37.86	36.91	35.79	33.61
HEC-S-92-I	10.39	**10.31**	11.71	11.9	11.31	11.19	10.56
KFU-S-93	**13.24**	13.37	16.02	15.3	14.75	14.51	13.44
LSE-F-91	**10.07**	10.33	11.15	12.33	11.41	10.92	10.87
PUR-S-93-I	**4.71**	5.21	NA	5.37	5.87	NA	NA
RYE-F-92	**8.27**	8.27	9.42	10.71	9.61	NA	8.81
STA-F-83-I	**157.08**	157.13	158.86	160.12	157.52	157.18	157.09
TRE-S-92	8.08	8.05	8.37	8.32	8.76	8.49	**7.94**
UTA-S-92-I	3.28	3.33	3.37	3.88	3.54	3.44	**3.27**
UTE-S-92	**24.81**	24.85	27.99	32.67	26.25	26.7	25.36
YOR-F-83-I	**35.35**	35.79	39.53	40.53	39.67	39.47	35.74

If we observe Figure 3.1, we can notice that in the first `1,000,000` iteration of SA-GD trajectory search, the optimal solution progresses very slow. It is because the new solution is always accepted as the new solution is always lower than the boundary (see Figure 3.2). Therefore, we aim to speed up the progress by combining hill climbing at the beginning of iteration. We have evaluated the optimal number of iterations to employ

the hill climbing before the SA-GD between 0.0001-10% of the total number of iterations. The preliminary experiment showed that 0.001% of the total number of iterations is the best number of iteration to employ hill climbing before the SA-GD. The comparison of the performance between SA-GD and SA-GD with hill climbing, henceforth, it is referred to as SA-HC-GD, is shown by Figure 3.2. Figure 3.2 visualises the objective function during the iteration of SA-GD hyper-heuristic compared to SA-HC-GD hyper-heuristic over `CAR-F-92-I`. It is clear from the Figure that SA-HC-GD outperforms SA-GD. However, when we tested over all problem instances with multiple runs, i.e. 11 times, SA-GD and SA-HC-GD are very competitive as shown by Table 3.8.

Table 3.8: Comparing The performance of SA-GD versus SA-HC-GD, from 11 runs within 3000000 iterations each

Instance	BEST*		AVERAGE		BEST**	
	SA-GD	SA-HC-GD	SA-GD	SA-HC-GD	SA-GD	SA-HC-GD
CAR-F-92-I	**3.93**	3.95	**4.05**	4.06	4.19	**4.06**
CAR-S-91-I	**4.78**	4.83	**4.87**	4.90	4.88	**4.84**
EAR-F-83-I	**33.03**	33.32	**33.56**	33.60	33.88	**33.32**
HEC-S-92-I	10.39	**10.31**	**10.64**	10.69	10.89	**10.27**
KFU-S-93	13.24	13.24	13.50	**13.41**	13.48	**13.45**
LSE-F-91	10.07	**10.02**	10.44	**10.34**	**10.30**	10.33
PUR-S-93-I	4.71	**4.62**	4.76	4.76	4.70	**4.62**
RYE-F-92	**8.27**	8.29	**8.37**	8.39	8.25	8.25
STA-F-83-I	157.08	**157.06**	157.08	**157.06**	157.08	**157.06**
TRE-S-92	8.08	**8.05**	8.21	**8.18**	8.18	8.18
UTA-S-92-I	**3.28**	3.32	**3.35**	3.37	**3.35**	3.36
UTE-S-92	**24.81**	24.86	**25.01**	25.05	**25.08**	25.22
YOR-F-83-I	**35.35**	35.80	**36.25**	36.28	**36.92**	37.15

*) in each run using different initial solution **) in each run always using the same initial solution

3.3 Solving The Capacitated Examination Timetabling Problem

Different from uncapacitated examination problem, in which only timeslots assignment is considered, in capacitated examination timetabling, the assignment of each exam into one of a set of rooms with different maximum capacity is also considered. The capacitated examination timetabling problem widely studied in the scientific literature is dataset from ITC 2007. As discussed in Section 2.1.2 of Chapter 2, ITC 2007 is much more complex than Carter examination timetabling problem. In addition to timeslot and room assignment, ITC 2007 examination timetabling problem incorporates more soft and hard constraints. Therefore, since in Carter problem only timeslot assignment is considered, in which it could be viewed as graph colouring problem, the graph colouring problem based method used in the problem domain of Carter examination timetabling problem discussed in the previous section, practically, can not be employed for this problem.

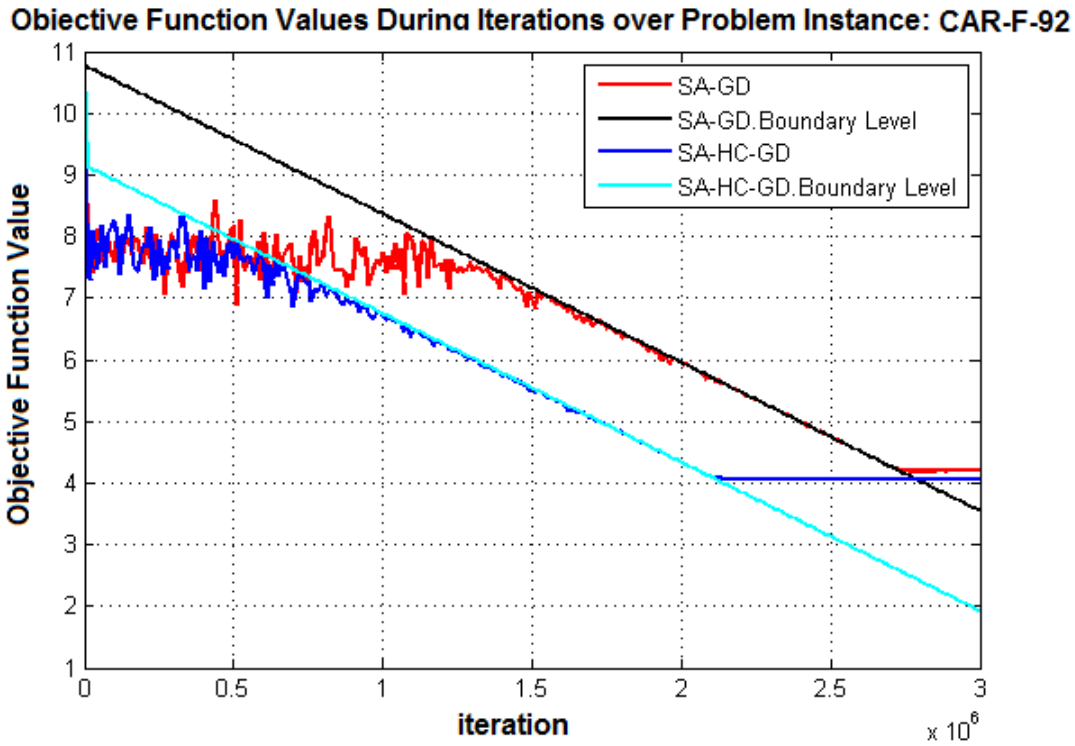


Figure 3.2: Objective function values during the search: comparison of the proposed SA-GD vs SA-HC-GD hyper-heuristic over problem instance CAR-F-92

3.3.1 Method

Generally, the method used for solving the capacitated examination timetabling problem is the same as the method used for uncapacitated examination timetabling problem, i.e. hyper-heuristic. Similarly, it is also implemented within HyFlex framework. The only difference is that we have to extend the problem domain in order to deal with the complexity of capacitated examination timetabling problem. Though, the problem domain is aimed for capacitated examination timetabling problem, we intend to generalise it to cope with both uncapacitated and capacitated examination timetabling problems. The uncapacitated examination timetabling problem could be considered as capacitated examination timetabling problem by assuming that there is only one room with the capacity equal to total number of students defined in the problem. The problem domain for capacitated examination timetabling problem and the hyper-heuristic strategy are detailed as follows:

3.3.1.1 Constructive Algorithm

As discussed in previous section, the main component of problem domain are a method to get the objective function value, a method to construct initial feasible solution, and

a set of low-level heuristic. The objective function for capacitated problem, which is represented by ITC 2007 examination timetabling problem is given in Section 2.1.2 of Chapter 2.

Basically, to construct an initial feasible solution for the capacitated examination timetabling problem, we investigated adaptive ordering heuristic (AOH) based methods that are more sophisticated than simply order the exams by their saturation degree as the method for uncapacitated examination timetabling problem given by Algorithm 1. The central idea of AOH is that an exam deemed as more 'difficult' (in term of the availability of feasible time-slots and room) is assigned to a time-slot and a room earlier than the easier ones. In other words, the exams are ordered by their difficulty level indicator first, before being assigned to timeslot and room sequentially. There are various criteria or basic ordering heuristics that could be utilised to predict the level of difficulty of each examination during the process of assignment. The basic ordering heuristics include ordering heuristics adopted from graph-colouring problems e.g. largest degree, colour degree, and saturation degree (see [89]).

In this chapter, we intended to study two different approaches of adaptive ordering heuristics to examination timetabling problem recently proposed by [280] and [3] with some proposed modifications. The former approach predicts the difficulty level of each examination by hierarchical hybridisations of graph colouring ordering method. Whereas, the latter approach predicts the difficulty level by adaptive linear combinations of graph colouring heuristics combined with a heuristic modifier. The following sections explain both approaches in detail.

Adaptive Ordering Heuristic With Hierarchical Hybridisation of Basic Graph Colouring Heuristics

The common criteria widely used in the scientific literature to estimate how difficult an exam to be assigned to resource, i.e. timeslot or period and room, during assignment process are the notions from graph colouring problem such as node degree and saturation degree. The other criteria include the number of students enrolled in the exam. Rather than ordering the exams by a basic single criterion only that have been proven does not work for hard problems, in this approach as elaborated in [280], the exams are ordered by a hierarchical hybridisation of some basic difficulty criteria. The basic ordering heuristics and its hybridisation are given in Table 3.9 and Table 3.10 respectively.

Different from [280] in our proposed constructive algorithm we used largest weighted degree (LWD) instead of largest degree as shown by Table 3.9. The expected benefit

Table 3.9: List of Basic Graph-Colouring Based Ordering Heuristic

Heuristic	Explanation
Saturation Degree (SD)	Exams are ordered in non-decreasing order by the number of remaining feasible timeslots
Largest Degree (LD)*	Exams are ordered in non-increasing order by the number of exams with common students
Largest Weighted Degree (LWD)**	Exams are ordered in non-increasing order by the number of exams with common students weighted by its number of the enrolled students
Largest Enrolment (LE)	Exams are ordered in non-increasing order by the number of the enrolled students
Largest Colour Degree (LCD)	Exams are ordered in non-increasing order by the number of exams with common students and already assigned to time-slot

*) It is used in [280] but not used in our algorithm.

**)It is used in our algorithm but is not used in [280].

of using LWD is that it could work better than largest degree (LD) in predicting the difficulty level of each exam in term of finding feasible timeslot and room during assignment process. In which based on this difficulty level the exams will be sorted in advance before being assigned to timeslot and room one by one. In addition, by using LWD, less number of heuristics could be hierachically hybridised, thus make simpler method. Furthermore, in the original work in [280] the hybridisations used to solve uncapacitated examination timetabling problem, i.e. Carter dataset are different from the hybridisation used to solve capacitated examination timetabling problem, i.e. ITC 2007 dataset. For Carter dataset, in [280], the hybridisation consist of four basic heuristics, whereas for ITC 2007 dataset, due to the size of problem, the hybridisation only consist of two basic heuristics. However, in our proposed algorithm we used exactly the same hybridisation of two basic heuristic for both the capacitated and uncapacitated datasets (see Table 3.10).

The hierarchical hybridisation in Table 3.10 can be explained as follows. The $\mathtt{H}_1$ that is a hybridisation of `LD+LE+SD+LCD` implies that firstly the exams are ordered by the number of other exams shared common students with the exams (non-increasing order). The tie is broken by ordering the exams by the number of students enrolled in the exams (non-increasing degree), then by the number of remaining feasible timeslot (non-decreasing order), and finally by the number of exams with common students and already assigned to time-slot and room.

As an illustration, given four ordering list $\mathtt{L}_1$, $\mathtt{L}_2$, $\mathtt{L}_3$, $\mathtt{L}_4$ of exams ordered by $\mathtt{H}_1$, $\mathtt{H}_2$, $\mathtt{H}_3$, $\mathtt{H}_4$ respectively. The difficulty index (DI) of each exam i i.e. e_i is defined by Equation 3.3.

Table 3.10: List of Hybrid Ordering Heuristic

	Original Approach [280]		Our Algorithm
	Carter	ITC 2007	For Both Carter and ITC 2007
H_1	LD+LE+SD+LCD	LD+LE	LWD+LE
H_2	SD+LCD+LD+LE	SD+LCD	SD+LCD
H_3	LCD+SD+LD+LE	LCD+SD	LCD+SD
H_4	LE+LD+SD+LCD	LE+LD	LE+LWD

$$DI(e_i) = \sum_{l=1}^{4} I_l^i \tag{3.3}$$

Where I_l^i is the order (position) of exam e_i in ordering list L_l. Hence, suppose exam e_1 is placed first in L_1 and L_2; and placed second and third in L_3 and L_4 respectively. Then, the difficulty index of e_1 is $1 + 1 + 2 + 3 = 7$.

Table 3.11: List of variables used in Algorithm 5

Variables	Description
`E`	List of all exams to be assigned to time-slot and room.
`X`	List of exams requiring exclusive rooms, i.e. could not be merged in the same room with other exams.
`C`	List of exams with coincidental exams, i.e. must be scheduled in the same timeslot.
`D`	List of exams coincidence with exam `c` *in* `C`.
`U`	List of all unassigned exams.

Table 3.12: List of methods used in in Algorithm 5

Methods	Description
`assignExam(e,t,r)`	Assign an exam `e` to random feasible timeslot `t` and random feasible room `r`.
`orderExam(U,c1,c2)`	Order a list of exams `U` by criteria `c1`, then ties are solved by criteria `c2`.
`getExamWithMinDiffIndex(L1,L2,L3,L4)`	Count the difficulty index of each unassigned exam according to their position in the list `L1,L2,L3,L4`. Return the exam with the minimum difficulty index.
`UpdateSatDegree(e)`	Update the saturation degree of all exams related, meaning have students in common with exam `e`.
`Reorder(L)`	Reorder the position of exams in the list `L`.

Algorithm 5 presents the pseudo-code of our proposed algorithm. Note that the variables and methods used in Algorithm 5 are briefly explained in Table 3.11 and Table 3.12 respectively. Compared to the original algorithm [280], we made some modifications in our proposed algorithm. Firstly, in order to prevent repair or backtracking mechanisms, i.e. timeslot repair mechanism and room repair mechanism, as in the original version, we make pre-assignment, i.e. assign examination to timeslot and room, for examinations

requiring exclusive room and examinations with coincidental exams (i.e. exams that have to be scheduled in the same time) (see line 3-14 in Algorithm 5) before following the original algorithm. Secondly, in our proposed algorithm, due to the initial feasible solution will be improved further later, during the assignment process, an exam is assigned to any feasible timeslot and room chosen randomly while in the original algorithm an exam is assigned to timeslot and room that caused minimum penalty. In [280], for Carter dataset, timeslot is chosen based on roulette wheel mechanism based on the sum of penalties; while for ITC 2007 dataset, each exam is randomly assigned to the best feasible timeslot and a best fit room with repair mechanism.

Algorithm 5 Pseudo-code of algorithm to initiate a feasible solution with adaptive ordering heuristic based on hierarchical hybridisations of graph colouring heuristics(adapted from [280]): AOH-HHGH. The variables and methods used are briefly explained in Table 3.11 and Table 3.12 respectively

```
1: procedure initiateSolution(i)
2:     E ← list of all exams
3:     X ← list of exams ⊂ E requiring exclusive room
4:     for all  e ∈ X do
5:         assignExam(e,t,r)
6:     end for
7:     C ← list of exams with coincidental exams
8:     for all c ∈ C do
9:         assignExam(c,t,r)
10:        D ← list of exams coincidence with exam c
11:        for all d ∈ D do
12:            assignExam(d,t,r)
13:        end for
14:    end for
15:    U ← list of unassigned exams
16:    L1 ← orderExam(U,LWD,LE)
17:    L2 ← orderExam(U,SD,LCD)
18:    L3 ← orderExam(U,LCD,SD)
19:    L4 ← orderExam(U,LE,LWD)
20:    while L1 ≠ ∅ do
21:        e ← getExamWithMinDiffIndex(L1,L2,L3,L4)
22:        assignExam(e,t,r)
23:        //remove exam e from L1, L2, L3, L4
24:        L1 ← L1-e
25:        L2 ← L2-e
26:        L3 ← L3-e
27:        L4 ← L4-e
28:        UpdateSatDegree(e)
29:        Reorder(L1)
30:        Reorder(L2)
31:        Reorder(L3)
32:        Reorder(L4)
33:    end while
34:    if L1 is empty then
35:        return a feasible solution
36:    else
37:        return an infeasible solution
38:    end if
39: end procedure
```

After pre-assignment process, firstly four lists of unassigned exams containing unassigned exams ordered by different criteria are created. Then, difficulty index of each unassigned exam is counted according to its position in these four list using Equation 3.3. Furthermore, the unassigned exams are assigned to resource, i.e. timeslot and room, iteratively. In each iteration the exam with the lowest difficulty index is assigned to resource and remove from the four lists. During the process of assignment, whenever an exam is

assigned to a timeslot, the saturation degree of each exam related to the exam, i.e. has common students, is changed. Therefore, in each iteration of the assignment process, saturation degree of all exams related to the assigned exam are updated. The process stops when all the exams are successfully assigned to feasible resource and a feasible solution is returned. If, there is an exam could not be assigned to any feasible resource, an infeasible solution is returned instead.

Adaptive Linear Combination Ordering Heuristic With a Heuristic Modifier

The second approach we evaluated to construct initial feasible solution is adaptive linear combination of graph colouring problem based ordering heuristics with heuristic modifier. This approach as elaborated in [2, 3] is an extended work of adaptive ordering heuristic proposed by [50]. In [50], in order to improve the performance of basic ordering heuristic, an experienced based difficulty metric with so-called a heuristic modifier was proposed. Basically, the mechanism is based on squeaky wheel optimisation [163].

Squeaky wheel optimisation approach consists of three main components i.e. constructor, analyser, and prioritiser. Within the case of examination timetabling problem, after the exams are initially ordered by the given initial ordering criteria, the constructor attempts to assign each unscheduled exam into best timeslot and best room (cause minimum soft constraint violation penalty). If an exam could not be assigned to any feasible timeslot and room, it remains unscheduled. In the following step, the analyser evaluates the process of assignment in previous step. The exam with more problematic assignment, e.g. it could not be assigned to any feasible timeslot and room, will be given higher priority in the next iteration. The priority is represented by a difficulty score associated to each exam. In other words, the exam that is more difficult to find feasible timeslot and room in a iteration, will be given higher difficulty score in the following iteration. Finally, the prioritiser reorder the exams based on the updated difficulty score in non-increasing order. The above processes are repeated until the stopping criteria is met. The detail of our proposed algorithm is shown by pseudo-code in Algorithm 6. Note that the variables and methods used in the algorithm are briefly explained in Table 3.13 and Table 3.14 respectively.

Table 3.13: List of variables used in Algorithm 6

Variables	Description
`E`	List of all exams to be assigned to time-slot and room.
`NOI`	Number of iterations.
`R`	a set of related exams.

Table 3.14: List of methods used in Algorithm 6

Methods	Description
`getDegree(e)`	return the degree, i.e. number of exams shared at least one the same student with exam **e**.
`orderExamByDS(E)`	order exams in the list of exams **E** by its difficulty score, in non-decreasing order.
`SHUFFLETOPWINDOW(E,s)`	Some exams of size `s` on the top of the ordered list `E` are randomly shuffled.
`isAssigned(x)`	if exam `x` already assigned to resource, a timeslot and a room return **True**, otherwise return **False**.
`assignTimeslot2Exam(x,bt)`	assign exam `e` to the best timeslot `bt`. If there is no feasible timeslot return **False**, otherwise return **True**.
`assignRoom2Exam(x,br)`	assign `x` to the best room, `br`. If there is no feasible room return **False**, otherwise return **True**.
`getRelatedExam(x)`	return exams related with `x` defined in hard constraints, i.e. coincidence exams, exams that should be scheduled before/after exam **x**.
`assignExam(x,ft,fr)`	assign x_r to any feasible timeslot,`ft` and feasible room, `fr`. If there is no feasible timeslot or there is no feasible room return **False**, otherwise return **True**.
`REASSIGNMENT(x)`	reassign exam `x` to another timeslot and room. If `x` could not be reassigned to any another timeslot and room return **False**, otherwise return **True**.
`isComplete()`	if all exams have been assigned to timeslot and room return **False**, otherwise return **True**.

Algorithm 6 Pseudo-code of an algorithm to initiate a feasible solution with Adaptive Linear Combination of Ordering Heuristic With Heuristic Modifier based on squeaky wheel optimisation adapted from [3]:AOH-SWO.The variables and methods used in this algorithm are briefly explained in Table 3.13 and Table 3.14 respectively.

```
1: procedure initiateSolution(0)
2:     E ← list of exams from the problem instance P
3:     //Initiate the difficulty score, i.e.  ds of exam e by its degree
4:     for all e ∈ E do
5:        e.ds ← getDegree(e)
6:     end for
7:     //NOI: number of iterations
8:     for i ← 1, i ← NOI do
9:        //Order the exams in E by their difficulty score in non-increasing order
10:       E ← orderExamByDS(E)
11:       for e=0, NumberOfExams-1  do
12:          SHUFFLETOPWINDOW(E,size)
13:          //assign x to element of E at index e
14:          x ← E[e]
15:          if (!isAssigned(x)) then
16:             a ← assignTimeslot2Exam(x,bt)
17:             if a = False then
18:                //Increase difficulty index of exam x by 2
19:                x.ds ← x.ds+2
20:                continue
21:             end if
22:             a ← assignRoom2Exam(x,br)
23:             if a=False then
24:                x.ds ← x.ds+2
25:                continue
26:             end if
27:              R ← getRelatedExam(x)
28:             for all x_r ∈R  do
29:                a ← assignExam(x_r,ft,fr)
30:                if a=False then
31:                   ra ← REASSIGNMENT(x)
32:                   if ra=False then
33:                      x.ds ← x.ds+2
34:                   end if
35:                end if
36:             end for
37:          end if
38:       end for
39:       if  isComplete() then
40:          return a feasible solution
41:       else
42:          continue
43:       end if
44:    end for
45:    return an infeasible solution
46: end procedure
```

In our proposed algorithm, in order to decide which examination should be assigned to which resource (i.e. timeslots and room) first, second, and so on, initially the examinations are ordered by its degree, i.e. the number of other examinations with common students. In other words, the initial difficulty score value of each examination is set to equal with its degree (see line 4-6, in Algorithm 6).

Then, to incorporate randomness as in [3] top window shuffle mechanism is incorporated, in which some exams in the top of the ordered list of exams are randomly shuffled (see line 12 Algorithm 6). We use top window shuffle of size 3 as suggested in [3]. By this mechanism, the exam that initially at the first place in the initial ordered list could move to second or third place as well as remain at the first place. Similarly, exam at position two and three also could move to two other positions or remain in their initial position.

In each assignment, an exam is assigned to the best timeslot and best room. The 'best' here means that the chosen timeslot and room incur the least penalty. If there is a case in which no feasible timeslot or room found for an exam, the heuristic modifier is applied for the exam. In our proposed algorithm, we used additive heuristic modifier (see line: 19, 24 and 33 of Algorithm 6), which is by adding the difficulty score with a constant number, e.g. `2` in this case.

Then, if the assignment of an exam is successful, then all related exams must be assigned to timeslot and room as well (see line 27-36 in Algorithm 6). Related exams of an exam `x` are all exams that are related to exam `x` as specified in time-related hard constraints. These include exams that must be scheduled in the same timeslot with exam `x` and exams that must be scheduled after or before exam `x`.

If prior assignment of exam `x` to the best timeslot and the best room causes at least one of the related exams to become unscheduled, the reassignment mechanism is called (see line 31 in Algorithm 6). In this mechanism, if possible the exam `x` is firstly reassigned to another timeslot/room result in the same cost with the prior timeslot/room. Otherwise, the exam is reassigned to any feasible timeslot/room such that all related exams could be assigned to feasible timeslot/room. Again, if the reassignment process is failed, the additive heuristic modifier is applied to the exam.

The above processes are repeated until completed, i.e. all examinations are assigned to feasible timeslots and rooms or the number of iteration limit is reached. We set 1000 as the number of iteration limit. Consult [2, 3] for detailed study of how the initial order, shuffle mechanism, as well as heuristic modifier affect the performance of algorithm.

Compared to the original algorithm, we have made some modification. First, we use exactly the same algorithm with the same configuration for Both Carter and ITC 2007 datasets whereas in [3] different configuration was used. While in [2], the study is tested over Carter datasets only. Second, in [3] the algorithm was tested over ITC 2007 datasets are with multiple heuristics, i.e. largest degree and saturation degree, and a heuristic modifier, while in our proposed algorithm we use only a single heuristic i.e. largest degree with a heuristic modifier. It aims at proposing simpler method with the same/better performance that corresponding to the cheaper computational time. Third, different from [3], in which in each iteration only the exam with highest difficulty score will be assigned to resource, in our algorithm, in order to minimise reassignment/backtracking mechanism, all related exams are also assigned to resource subsequently (see line 27-36 of Algorithm 6) .

Lastly, another component of a problem domain within a hyper-heuristic framework is a set of predefined low-level heuristics. The low-level heuristics employed in our capacitated examination timetabling problem domain are given in Table 3.15. As shown by Table 3.15, we employ 14 low-level heuristics instead of 4 low-level heuristics employed for uncapacitated examination timetabling problem given in Table 3.1. These low-level heuristics are collated from commonly used low-level heuristic for examination timetabling problem in prior works reported in the scientific literature mainly from [55]. The low-level heuristics in Table 3.15 are natural and often used.

3.3.1.2 Improvement Method with Hyper-heuristic Strategy

For hyper-heuristic strategy, in addition to self-adaptive great deluge algorithm that has been proven performs well for uncapacitated examination timetabling problem discussed in Section 3.2, we employed a selection hyper-heuristic approach with reinforcement learning as low-level heuristics selection strategy and great deluge as move acceptance. This approach is adapted from [234].

The original approach in [234] was tested over Carter and Yeditepe dataset only. Whereas, in our proposed approach, in addition to work for both datasets, we adapted to work for ITC 2007 datasets too. Compared to Carter datasets, ITC 2007 dataset are much more complex in terms of the number of hard and soft constraints as discussed in Chapter 2.

In general, the algorithm is shown by Algorithm 7. In order to improve the quality of initial solution (i.e. minimising the objective function), the initial solution is perturbed iteratively by a set of problem-specific low-level heuristics (for capacitated examination

Table 3.15: Perturbation Low-Level Heuristics (LLHs) for Examination Timetabling Problems

LLH	**Description**
$\texttt{LLH}_1$	Select one examination at random and move to a new random feasible timeslot and room.
$\texttt{LLH}_2$	Select two examinations at random and move each examination to a new random feasible timeslot.
$\texttt{LLH}_3$	Select three examinations at random and move each examination to a new random feasible timeslot.
$\texttt{LLH}_4$	Select four examinations at random and move each examination to a new random feasible timeslot.
$\texttt{LLH}_5$	Select two examinations at random and swap the timeslots between these two examinations while maintaining the feasibility of the two examinations.
$\texttt{LLH}_6$	Select one exam at random and select another timeslot then apply the Kempe-chain move.
$\texttt{LLH}_7$	Select one highest penalty examination selected from a random 10% selection of the examinations and select another timeslot then apply the Kempe-chain move [309].
$\texttt{LLH}_8$	Select one highest penalty examination selected from a random 20% selection of the examinations and select another timeslot then apply the Kempe-chain move.
$\texttt{LLH}_9$	Select two timeslots at random and swap the examinations between them.
$\texttt{LLH}_{10}$	Select one timeslot at random and move the examinations assigned to that timeslot to a new feasible time-slot.
$\texttt{LLH}_{11}$	Shuffle all time-slots at random.
$\texttt{LLH}_{12}$	Select one exam at random and move it to a randomly selected feasible room.
$\texttt{LLH}_{13}$	Select two exams at random and swap their rooms (if feasible).
$\texttt{LLH}_{14}$	Select one big examination at random and move to a new random earlier feasible timeslot [122].

timetabling problem domain see Table 3.15) until the pre-defined time or number of iterations limit is reached. This iterative process is shown by lines of code between number 12 and 27 in Algorithm 7).

One of more intelligent ways of choosing a low-level heuristic in each iteration is by employing reinforcement-learning mechanism [164, 294, 234]. Whereas for move acceptance strategy, great deluge [128, 206, 166] as discussed in Section 3.2 was proven performs well for solving examination timetabling problem.

The idea of reinforcement learning base mechanism for low-level heuristics selection is simply by giving each low level heuristic reward and punishment. Initially, each low-level heuristic is associated with the same initial score, i.e. 10, in our proposed algorithm (see line 6 in Algorithm 7. If a heuristic is selected and result in an accepted solution, then its score is increased by 1 (see line 19 in Algorithm 7)until upper bound score is reached (20, in our implementation). Contrarily, if a heuristic is selected but result in rejected solution, then its score is decreased by 1 (see line 24 in Algorithm 7)until lower bound score (i.e. 0, in our setting) is reached. In each iteration, a low-level heuristic with the highest score is chosen(see line 13 in Algorithm 7). If there are more than one low-level heuristics with the highest score, one of them is chosen randomly. Consult [234], for study on how parameter setting(e.g. reward and mechanism procedure, lower and upper bond) affects the performance of algorithm.

The great deluge algorithm as move acceptance is basically similar with great deluge algorithm discussed in Section 3.2.1.2. In addition to standard great deluge algorithm, we also evaluated an extended great deluge algorithm [206] and modified great deluge algorithm [154] given in Algorithm 8 and Algorithm 9 respectively.

Table 3.16: List of variables used in Algorithm 7- 9

Variables	Description
`P`	Exam timetabling problem instance.
$\mathtt{S}_t$	The stopping condition.
`M`	Number of low-level heuristics.
`R`	Array of reward given to each low-level heuristics.
$\mathtt{S}_{best}$	Objective function value of the best solution so far.
$\mathtt{S}_{curr}$	Objective function value of the current solution.
`B`	Boundary level.
α	Decay rate.
`l`	a chosen low-level heuristics.
`newFunctionVal`	The objective function value of the new solution.
`Ic`	Idle counter.
`Rp`	Reheating point.

Table 3.17: List of methods used in Algorithm 7 - 9

Methods	Description
`P.initiateSolution(i)`	Generate an initial feasible solution and stored it in solution memory in index `i`.
`initiateReward(R)`	method to give the same initial reward for each low-level heuristics. We set to `10` .
`P.getFunctionValue(i)`	Get objective function value of solution stored in solution memory index `i`.
`selectLLH(R)`	return the index of the low-level heuristic with the highest reward.
`P.applyLLH(l,i,j)`	Apply llh `l` over solution stored in solution memory `i` and stored the new solution in solution memory index `j`. Return the objective function value of the new solution.
`P.copySolution(1,0)`	copy solution in solution memory index `i` to solution memory index `j`.
`upgradeReward(R[l])`	update the reward for low-level heuristic `R[l]` by adding the value with `reward` (we set to `1`) if the value of `R[l]` does not reach the `upperbound` (we set to `20`).
`downgradeReward(R[l])`	update the reward for low-level heuristic `R[l]` by decreasing the value with `reward` (we set to `1`) if the value of `R[l]` does not reach the `lowerbound` (we set to `-20`).
`isStuck()`	If the objective function is not changed for a given number of iteration (we set to 40) return **True**, otherwise return **False**.

Algorithm 7 Reinforcement Learning Great Deluge-based Hyper-heuristics (RL-GD-HH). The variables and methods used are briefly explained in Table 3.16 and Table 3.16 respectively.

```
1: procedure SOLVE(ProblemDomain P)
2:     Set stopping condition St
3:     Set M ← P.getNumberofHeuristics()
4:     // R is an array to store reward given to each low-level heuristic.
5:     Set R ← new int[M]
6:     initiateReward(R)
7:     P.initiateSolution(0)
8:     S_best ← P.getFunctionValue(0)
9:     S_curr ← P.getFunctionValue(0)
10:    B ← P.getFunctionValue(0)
11:    Set decay rate α
12:    while S_t is not met do
13:        l← selectLLH(R)
14:        newFunctionVal ← P.applyLLH(l,0,1)
15:        if (newFunctionVal ≤ S_curr OR newFunctionVal ≤ B) then
16:            //the new solution is accepted
17:            P.copySolution(1,0)
18:            S_curr ← newFunctionVal
19:            upgradeReward(R[l])
20:            if newFunctionVal<S_best then
21:                Set S_best ← newFunctionVal
22:            end if
23:        else
24:            downgradeReward(R[l])
25:        end if
26:        B ← B.α
27:    end while
28:    return S_best
29: end procedure
```

Algorithm 8 Reinforcement Learning Extended Great Deluge based Hyper-heuristics (RL-XGD-HH). The variables and methods used are briefly explained in Table 3.16 and Table 3.16 respectively.

```
1: procedure SOLVE(ProblemDomain P)
2:     Set stopping condition St
3:     Set M ← P.getNumberofHeuristics()
4:     // R is an array to store reward given to each low-level heuristic.
5:     Set R ← new int[M]
6:     initiateReward(R)
7:     //Set idle counter and reheating point
8:     Initiate Idle counter Ic ← 0
9:     Set Reheating Point Rp
10:    P.initiateSolution(0)
11:    S_best ← P.getFunctionValue(0)
12:    S_curr ← P.getFunctionValue(0)
13:    B ← P.getFunctionValue(0)
14:    Set decay rate α
15:    while S_t is not met do
16:        l ← selectLLH(R)
17:        newFunctionVal ← P.applyLLH(l,0,1)
18:        if (newFunctionVal ≤ S_curr OR newFunctionVal ≤ B) then
19:            //the new solution is accepted
20:            P.copySolution(1,0)
21:            S_curr ← newFunctionVal
22:            upgradeReward(R[l])
23:            if newFunctionVal<S_best then
24:                Set S_best ← newFunctionVal
25:            end if
26:        else
27:            downgradeReward(R[l])
28:            Ic++
29:        end if
30:        B ← B.α
31:        if Ic > Rp then
32:            B ← S_curr
33:            Ic ← 0
34:        end if
35:    end while
36:    return S_best
37: end procedure
```

Algorithm 9 Reinforcement Learning Modified Extended Great Deluge based Hyper-heuristics (RL-MXGD-HH). The variables and methods used are briefly explained in Table 3.16 and Table 3.16 respectively.

```
1: procedure SOLVE(ProblemDomain P)
2:     Set stopping condition St
3:     Set M ← P.getNumberofHeuristics()
4:     // R is an array to store reward given to each low-level heuristic.
5:     Set R ← new int[M]
6:     initiateReward(R)
7:     Initiate Idle counter Ic ← 0
8:     P.initiateSolution(0)
9:     S_best ← P.getFunctionValue(0)
10:    S_curr ← P.getFunctionValue(0)
11:    B ← P.getFunctionValue(0)
12:    B_upper ← 1.08
13:    B_lower ← 0.92
14:    Set decay rate α
15:    while S_t is not met do
16:        l ← selectLLH(R)
17:        newFunctionVal ← P.applyLLH(l,0,1)
18:        if (newFunctionVal ≤ S_curr OR newFunctionVal ≤ B) then
19:            //the new solution is accepted
20:            P.copySolution(1,0)
21:            S_curr ← newFunctionVal
22:            upgradeReward(R[l])
23:            if newFunctionVal<S_best then
24:                Set S_best ← newFunctionVal
25:            end if
26:        else
27:            downgradeReward(R[l])
28:        end if
29:        B ← B.α
30:        if isStuck() then
31:            Ic ← Ic+0.5
32:            B ← (B_upper)^Ic.S_currs
33:        else
34:            B_level ←(B_lower)^(Ic+1).S_best
35:            if B ≤ B_level then
36:                Ic ← Ic+1
37:                B ← (B_upper)^Ic.S_best
38:            end if
39:        end if
40:    end while
41:    return S_best
42: end procedure
```

3.3.2 Experimental Results and Discussion

Overall, the aims of experiments reported in this chapter consist of two folds. Firstly, to evaluate the initial feasible solution construction within problem domain using two proposed algorithms in Section 3.3.1.1. Secondly, to examine the proposed hyper-heuristics: reinforcement learning - great deluge based heuristic (see Algorithm 7-9) and self-adaptive learning - great deluge based heuristic (see Algorithm 3) in optimising the benchmark standard objective function.

The experiments were conducted over four datasets, namely, Carter, ITC 2007, Yeditepe, and Nottingham. The characteristic of each these datasets has been discussed in Section 2.1 of Chapter 2. Though, basically, Carter dataset is uncapacitated examination timetabling problem, it could be easily converted to a capacitated examination timetabling problem in ITC 2007 format by assuming there is only one room with capacity equal to the total number of students in the problem domain[2]. The original objective function and hard constraints are retained. The Yeditepe datasets are in the same format with ITC 2007 format.

However, for Nottingham dataset, since there is lack of clarity, consistency, and standard format of the dataset in the literatures (probably, it is the reason why there are few prior works using these dataset), we have made some modifications on the period related constraints. We also convert the Nottingham dataset into ITC 2007 dataset format[3]. The characteristic of our modified Nottingham dataset is summarised in Table 3.18.

Table 3.18: The characteristics of Nottingham dataset in the ITC 2007 format

Problem	No.Of T.Slots	No.Of Exams	No.Of Students	No.Of Enrolment	No.Of Rooms	C.Density	No.Of P.HC	No.Of R.HC
NOTT-94-I	32	800	7896	33997	14(1630)	0.23	58	0
NOTT-94-II	32	800	7896	33997	1(1550)	0.23	0	0

	w^{2D}	w^{2R}	w^{PS}	w^{NMD}	LE	NLP	w^{FL}
NOTT-94-I	10	20	5	20	100	10	5
NOTT-94-II	0	1	0	0	0	0	0

Note:P.HC:period hard constraints,R.HC:room hard constraints
LE: the number of largest exam to be considered.
NLP: the number of last periods to be considered.

3.3.2.1 Initial Feasible Solutions Construction

As discussed in Section 3.3.1.1, in order to generate initial feasible solutions two adaptive ordering heuristic (AOH) algorithms are studied. Firstly, AOH by difficulty index based

[2]http://www.cs.nott.ac.uk/∼abm/research/exam-fairness-data/examproblemwithfairness/index.html
[3]http://www.cs.nott.ac.uk/∼abm/research/exam-fairness-data/examproblemwithfairness/index.html

on hierarchical hybridisations of graph colouring heuristics (henceforth, it will be referred to as AOH-HHGH). Secondly, AOH with difficulty score calculated from adaptive linear combination with heuristic modifier based on squeaky wheel (henceforth, it will be referred to as AOH-SWO). To evaluate their performance in generating feasible solutions, each algorithm was run 10 times over all problem instances from the four benchmark datasets. The experimental results are given in Table 3.19.

Table 3.19: Results obtained by the AHOHHGH and AOH-SWO initial solution construction methods over 10 runs for the Carter, ITC 2007, Yeditepe, and Nottingham benchmark instances.

Instance	Mean		Best		Avg.Time (sec)	
	AOH-HHGH	AOH-SWO	AOH-HHGH	AOH-SWO	AOH-HHGH	AOH-SWO
CAR91	8.93	**5.72**	8.54	**5.52**	**3429.9**	17385.5
CAR92	7.75	**4.98**	7.28	**4.83**	27330.8	**13757.4**
EAR83	57.1	**42.31**	53.5	**40.39**	4496.6	**359.1**
HEC92	20.3	**15.1**	16.99	**14.32**	328.1	**65.3**
KFU93	27.26	**18.01**	24.2	**16.64**	**138.4**	1379.9
LSE91	21.12	**13.11**	19.52	**11.86**	**95.7**	644.5
PUR93	10.21	**5.21**	9.91	**5.07**	**4801.4**	18910.2
RYE92	20.35	**13.68**	18.36	**11.84**	**614.3**	5143.5
STA83	**178.05**	184.153	169.2	**169.05**	1165.5	**64.3**
TRE92	13.44	**10.23**	12.38	**9.84**	**166.3**	547.6
UTA92	6.23	**4.01**	3.8	**5.46**	**862.4**	6054
UTE92	42.38	**34.34**	38.45	**31.82**	640.3	**84**
YOR83	53.23	**45**	52.2	**42.27**	**295.6**	4872.5
EXAM1	30689.2	**17829.9**	28972	**14628**	**720.8**	62527.5
EXAM2	36586.7	**14177.7**	30208	**3248**	**131.5**	13652.4
EXAM3	107017.8	**13675.4**	95957	**12423**	**261.8**	112013.5
EXAM4	NA	**35331.3**	NA	**28272**	NA	16302.5
EXAM5	134850.3	**5025**	117096	**4572**	**109.8**	8018.2
EXAM6	56490.5	**30945**	53955	**29080**	**61.9**	406.2
EXAM7	66631.2	**6322.5**	57794	**5867**	**173.5**	39139
EXAM8	134071.5	**66223.3**	74474	**26061**	**105.1**	58107.7
EXAM9	8741.2	**4744.5**	7738	**3100**	**42.1**	362.9
EXAM10	70916.7	**31297**	57481	**23898**	**53**	358.9
EXAM11	214147.7	**44769.8**	182745	**37743**	**4210.9**	110083.4
EXAM12	12712.7	**9263.9**	10612	**7033**	1959	**1477**
YUE20011	62366.6	**362.6**	49681	**233**	**41.9**	56.5
YUE20012	70717.4	**431.6**	61831	**292**	2454	**184.5**
YUE20013	68.1	**61.1**	53	**50**	0.8	**0.7**
YUE20021	80739.1	**768.7**	74165	**286**	**14**	19.1
YUE20022	92060.6	**511.6**	87050	**403**	1293.8	**124.4**
YUE20023	108.3	**87.2**	78	**60**	**6**	14.7
YUE20031	87430.8	**7990.3**	75291	**2379**	**12.8**	30.2
YUE20032	106483.5	**4814.8**	101342	**2647**	**91.1**	164.7
NOTT-94-I	77932.6	**30310.6**	71430	**23327**	**170.5**	3932.8
NOTT-94-II	2508.1	**408.2**	2063	**39**	**91**	2204

The tables show the average and the best quality of the generated initial solutions, as well as average times required to produce a feasible solution (in milliseconds) from 10 trials. A solution is feasible if all defined hard constraints are satisfied. Further, the quality of solution is measured by benchmark standard objective function value.

The experimental result showed that in each trial, AOH-SWO was always successful in

generating initial feasible for all instances from all datasets. Contrarily, though AOH-HHGH was always successful in generating initial feasible solution for all the other instances, it failed to generate a feasible solution for instance EXAM4 from ITC 2007 datasets. It is noteworthy that compared to the the original algorithm [280], our proposed AOH-HHGH, which could generate feasible solution for all problem instance, is simpler, i.e. using two heuristics instead of four heuristics. In addition, with pre-assignment mechanism, repair/backtracking mechanism as in the original version, could be omitted. Similarly, our proposed AOH-SWO algorithm is also simpler than the original version, initial difficulty score is based on single graph heuristic only instead of two graph colouring heuristics. It works because we employed subsequent assignment for related exams of an exam with highest score in each iteration.

In terms of quality of the solutions, it is unsurprisingly shown that AOH-SWO outperforms AOH-HHGH. The reason is that during the assignment process within AOH-SWO approach, each exam is assigned to the best timeslot/room (causing minimum penalties), while in AOH-HHGH each exam is just simply assigned to any feasible timeslot/room randomly.

Lastly, regarding the times required to construct initial feasible solutions, AOH-HHGH could produce initial feasible solution faster for 8 out 13 instances from Carter dataset, 10 out 12 instances from ITC dataset, 5 out 8 instances from Yeditepe dataset, and all instances from Nottingham dataset or 25 out 35 instances overall. In overall average, in order to constructing a feasible solution AOH-SWO needs running-time 1.5, 52, 1.2, and 23 longer than AOH-HHGH for Carter, ITC 2007, Yeditepe, and Nottingham datasets respectively.

Overall, from the tables we can conclude that in generating initial feasible solutions, AOH-SWO performs better than AOH-HHGH. Though required relatively slight greater amount of running times, it is not always the case, as result of finding the best feasible assignment instead of simply any feasible assignment during the process of assignment of exams to timeslot and room. The quality of this initial solution determine the quality of final solution after the improvement phase of the algorithm.

3.3.2.2 Results on optimising the standard objective function

Firstly, we conduct experiments with three proposed hyper-heuristic strategies i.e. Reinforce Learning - Great Deluge based Hyper-heuristic (RL-GD-HH) given in Algorithm 7, Reinforce Learning - Extended Great Deluge based Hyper-heuristic (RL-XGD-HH) given in Algorithm 8, and Reinforce Learning - Modified Extended Great Deluge based Hyper-

heuristic (RL-MXGD-HH) given in Algorithm 9. Within all these hyper-heuristic strategies, the initial feasible solution is generated using AOH-SWO, which has been proven outperforms AOH-HHGH in the previous experiment. The proposed hyper-heuristic was tested over four different datasets, i.e. Carter, ITC 2007, Yeditepe, and Nottingham. In addition, we also extended a problem instance from Nottingham dataset, i.e. `NOTT-94-II`, given in Table 3.18 by varying the number of timeslots. From the original version with 32 timeslots, we varies into four other versions with 31, 30, 26, and 23 timeslots. For each problem instance, the algorithm was run 21 times with running time limited to 360 seconds per run.

Table 3.20: Experimental results with three different hyper-heuristics strategies: RL-GD-HH, RL-XGD-HH, and RL-MXGD-HH.

Dataset	Instance	RL-GD-HH			RL-XGD-HH			RL-MXGD-HH		
		Best	Avg	Med	Best	Avg	Med	Best	Avg	Med
CARTER	CAR91	**5.24**	5.44	5.44	5.30	5.44	5.41	5.30	5.53	5.55
	CAR92	**4.42**	4.65	4.66	4.51	4.66	4.62	4.49	4.65	4.64
	EAR83	36.90	38.53	38.11	**36.73**	38.27	38.23	36.74	38.69	38.34
	HEC92	**10.87**	11.56	11.43	10.91	11.43	11.35	10.92	11.42	11.46
	KFU93	14.27	15.07	15.07	14.36	15.08	15.12	**14.14**	15.02	14.96
	LSE91	11.29	12.32	12.39	**11.02**	12.00	12.09	11.29	12.20	12.27
	PUR93	5.02	5.20	5.21	5.03	5.20	5.22	**5.00**	5.16	5.13
	RYE92	9.27	9.77	9.79	**9.01**	9.53	9.58	9.08	9.73	9.73
	STA83	157.09	157.35	157.33	157.12	157.39	157.32	**157.06**	157.32	157.37
	TRE92	8.97	9.28	9.27	8.75	9.19	9.13	**8.74**	9.10	9.06
	UTA92	**3.58**	3.73	3.72	3.60	3.72	3.72	3.63	3.78	3.79
	UTE92	25.56	26.62	26.5	**25.20**	26.39	26.4	25.42	26.30	26.28
	YOR83	38.64	40.36	40.28	**38.03**	39.72	39.56	38.35	39.73	39.77
ITC 2007	EXAM1	**6848**	7666	7631	6856	7146	7176	7363	7937	7954
	EXAM2	663	769	759	**632**	720	724	646	796	780
	EXAM3	11822	12463	12312	**11659**	12574	12429	11806	12424	12344
	EXAM4	17785	19179	19271	**16325**	18874	18991	16464	18726	18682
	EXAM5	3896	4272	4288	**3837**	4157	4050	3786	4280	4181
	EXAM6	27575	28656	28930	**27370**	28440	28250	27655	28875	29020
	EXAM7	5561	5860	5803	**5528**	6006	5848	5662	6207	5956
	EXAM8	10078	10553	10560	**9798**	10272	10178	9998	10380	10349
	EXAM9	1363	1504	1492	**1246**	1320	1320	1322	1438	1421
	EXAM10	15124	15932	15855	**14556**	15218	15239	14745	15414	15369
	EXAM11	36878	40457	40095	36810	40752	40109	**34412**	39714	39500
	EXAM12	5620	5966	5949	**5300**	5577	5581	5436	5968	5953
YEDITEPE	YUE20011	74	111	112	**56**	71	68	64	84	82
	YUE20012	135	187	185	**122**	159	161	127	158	156
	YUE20013	29	33	33	**29**	31	29	29	32	30
	YUE20021	115	148	142	**76**	113	111	96	139	128
	YUE20022	203	261	268	**162**	209	212	189	257	260
	YUE20023	56	64	62	**56**	62	61	56	64	61
	YUE20031	204	273	270	**143**	254	206	183	293	249
	YUE20032	461	519	518	**434**	673	479	433	553	491
NOTT	NOTT-I	12122	12941	12970	**11498**	12256	12329	12402	13145	13010
	NOTT-II-32	**0**	4	4	**0**	5	5	**0**	4	4
	NOTT-II-31	**0**	7	8	3	12	10	2	14	14
	NOTT-II-30	5	17	15	3	17	18	**0**	19	20
	NOTT-II-26	49	76	71	43	63	65	**37**	70	72
	NOTT-II-23	148	239	242	148	193	180	**124**	170	168

Table 3.20 shows the comparison of experimental results from three different hyper-heuristic strategies. It can be observed from Table 3.20 that having reinforcement learning as a low-level heuristic selection strategy, generally the hyper-heuristic base on great deluge algorithm with reheating (RL-XGD-HH) outperforms the other hyper-heuristic strategies. Especially, over problem instances from ITC 2007 and Yeditepe datasets, in which RL-XGD-HH outperforms the other hyper-heuristics over almost all the problem instances. However, over problem instances from Yeditepe dataset, RL-MXGD-HH looks perform better than RL-XGD-HH.

Further, Table 3.21 compares the results from RL-XGD-HH with results reported from the scientific literature. For Carter and ITC 2007 dataset, we compare our results with (Müller,2009) [222], (Burke, 2012) [63], (Sabar, 2012) [280], (Rahman, 2014) [3], and (Burke, 2014) [82]. Whereas for Yeditepe datasets, since lack of prior experimental results over this dataset are reported, we compare our results with results generated from solver submitted by (Müller,2009) [222] into ITC 2007. Müller solver is the winner of international timetabling competition 2007.

From the best solution shown in Table 3.21, we can observe that our proposed algorithm that is the hybridisation of adaptive heuristic ordering based on squeaky wheel optimisation (AOH-SWO) for initial feasible solution construction and reinforcement learning, extended great deluge algorithm hyper-heuristic (RL-XGD-HH) is well performing with promising results. The result beat the benchmarks for 8 out 13 from Carter datasets, 3 out 12 from ITC 2007 datasets, and 6 out 8 (1 draw) from Yeditepe datasets. It is also noteworthy that the proposed algorithm is not tuned for specific datasets, rather using exactly the same algorithm with the same parameter setting for all problem instances from all datasets. It could be the strength of the hyper-heuristic approach. However, in hyper-heuristics, though problem specific parameter tuning is not required, many configuration of algorithms are tried. It could be the trade-off hyper-heuristic and the standard meta-heuristic approaches.

Furthermore, we evaluated three different low-level selection strategies, i.e. simple random (SR), reinforcement learning (RL), and self-adaptive (SA) learning combined with three different move acceptance strategies, i.e. great deluge (GD), extended great deluge (XGD), and modified extended great deluge (MXGD). These hyper-heuristic strategies are summarised in Table 3.22. The SR-GD-HH and SA-GD-HH algorithms have been discussed in Section 3.2.1.2 (See Algorithm 3) whereas RL-GD-HH is given in Algorithm 7. However, different from Algorithm 7, in which the decay rate is set to be fixed, in this experiment the decay rate set by the Equation 3.2. Since the algorithm is run within given fixed time and the Equation 3.2 required number of iteration and target

Table 3.21: Result obtained using combination of AOH-SWO and RL-XGD-HH compared with the best result reported in scientific literatures

Instance	Our Result		Benchmark (Best)			
	MEDIAN	BEST	(Burke,2012) [63]	(Sabar,2012) [280]	(Rahman,2014) [3]	(Burke,2014) [82]
CAR91	5.41	5.30	**5.03**	5.14	5.12	5.19
CAR92	4.62	4.51	**4.22**	4.7	4.41	4.31
EAR83	38.23	36.73	**36.06**	37.86	36.91	35.79
HEC92	11.35	**10.91**	11.71	11.9	11.31	11.19
KFU93	15.12	**14.36**	16.02	15.3	14.75	14.51
LSE91	12.09	**11.02**	11.15	12.33	11.41	10.92
PUR93	5.22	**5.03**	NA	5.37	5.87	NA
RYE92	9.58	**9.01**	9.42	10.71	9.61	NA
STA83	157.32	**157.12**	158.86	160.12	157.52	157.18
TRE92	9.13	8.75	8.37	**8.32**	8.76	8.49
UTA92	3.72	3.60	3.37	3.88	3.54	**3.44**
UTE92	26.4	**25.20**	27.99	32.67	26.25	26.7
YOR83	39.56	**38.03**	39.53	40.53	39.67	39.47
			(Müller,2007) [222]	(Sabar,2012) [280]	(Rahman,2014) [3]	(Burke,2014) [82]
EXAM1	7176	6856	4370	6234	5231	6235
EXAM2	724	632	400	**395**	433	2974
EXAM3	12429	11659	10049	13002	**9265**	15832
EXAM4	18991	**16325**	18141	17940	17,787	35106
EXAM5	4050	3837	**2988**	3900	3083	4873
EXAM6	28250	27370	26585	27000	**26,060**	31756
EXAM7	5848	5528	**4213**	6214	10,712	11562
EXAM8	10178	9798	**7742**	8552	12,713	20994
EXAM9	1320	1246	**1030**	NA	1111	NA
EXAM10	15239	**14556**	16682	NA	14,825	NA
EXAM11	40109	36810	**34129**	NA	28,891	NA
EXAM12	5581	**5300**	5535	NA	6181	NA
			(Müller,2007) [222]	(Sabar,2012) [280]	(Rahman,2014) [3]	(Burke,2014) [82]
YUE20011	68	**56**	62	NA	NA	NA
YUE20012	161	**122**	125	NA	NA	NA
YUE20013	29	**29**	29	NA	NA	NA
YUE20021	111	76	**70**	NA	NA	NA
YUE20022	212	**162**	170	NA	NA	NA
YUE20023	61	**56**	70	NA	NA	NA
YUE20031	206	**143**	223	NA	NA	NA
YUE20032	479	**434**	440	NA	NA	NA
			(Burke,1996) [74]	(Caramia,2008) [85]	(Rahim,2009) [262]	(Cote,2005) [108]
NOTT-I	12329	11498	N.A	N.A	N.A	N.A
NOTT-II-32	5	0	0	N.A	N.A	N.A
NOTT-II-31	10	3	0	N.A	N.A	N.A
NOTT-II-30	18	3	0	N.A	N.A	N.A
NOTT-II-26	65	43	49	44	N.A	5
NOTT-II-23	180	148	348	98	19.61	6.9

objective function values, we conducted a preliminary experiment to approximate the number of iteration within the given time and the target objective function values are set to the best known solution given in Section 2.1 of Chapter 2.

Table 3.22: The strategy within our Hyper-heuristics (HH) framework

No	HH	Method for Heuristics Selection	Method for Move Acceptance
1	SR-GD-HH	Simple Random	Great Deluge
2	RL-GD-HH	Self-adaptive	Great Deluge
3	SA-GD-HH	Reinforcement Learning	Great Deluge
4	SR-XGD-HH	Simple Random	Extended Great Deluge
5	RL-XGD-HH	Self-adaptive	Extended Great Deluge
6	SA-XGD-HH	Reinforcement Learning	Extended Great Deluge
7	SR-MXGD-HH	Simple Random	Modified Extended Great Deluge
8	RL-MXGD-HH	Self-adaptive	Modified Extended Great Deluge
9	SA-MXGD-HH	Reinforcement Learning	Modified Extended Great Deluge

Similar to previous experiment, we tested our algorithm over all problem instances of four datasets. For each instance we run our algorithm 21 times. Time allocated for each run was 360 seconds. It is an approximation to the time allowed in the ITC 2007 competition. Table 3.23-Table 3.25 show the experimental results with Algorithms summarised in Table 3.22.

From Table 3.23, we can observe that as expected over ITC 2007 and Yeditepe datasets, SA-GD-HH algorithm outperforms SR-GD-HH and RL-GD-HH. However, over Carter dataset, SR-GD-HH outperforms SA-GD-HH and RL-GD-HH, whereas over Nottingham dataset, RL-GD-HH outperforms SR-GD-HH and SA-GD-HH. The fact that, SR-GD-HH outperforms SA-GD-HH is not consistent with the experimental results over uncapacitated examination timetabling problem given in Table 3.6 in which SA-GD-HH outperforms SR-GD-HH. Therefore, we can conclude that in addition to low-level heuristic selection and move acceptance strategies, the low-level heuristics defined in the problem domain also plays a very vital role on the performance of selection hyper-heuristics. In this case, more specific low-level heuristics to the problem domain is more effective than more general low-level heuristic.

When the different low-level heuristic selection strategies combined with extended great deluge algorithm instead of standard great deluge as shown by Table 3.24, generally, SR-XGD-HH outperforms RL-XGD-HH and SA-GD-HH over Carter dataset. Whereas over ITC 2007 dataset SR-XGD-HH is just comparable with RL-XGD-HH and over Yeditepe dataset, SR-XGD-HH is just comparable with SA-XGD-HH. And lastly, over Nottingham dataset, RL-XGD-HH outperforms SR-XGD-HH and SA-XGD-HH.

Furthermore, when the different low-level heuristic selection strategies combined with

Table 3.23: The Experimental Results with three different strategies: SR-GD-HH, RL-GD-HH, and SA-GD-HH.

Dataset	Instance	SR-GD-HH			RL-GD-HH			SA-GD-HH		
		Best	Avg	Med	Best	Avg	Med	Best	Avg	Med
CARTER	CAR91	**5.23**	5.51	5.51	5.53	5.71	5.72	5.29	5.47	5.45
	CAR92	**4.23**	4.43	4.41	4.79	4.96	4.91	4.27	4.60	4.7
	EAR83	**32.98**	33.81	33.5	38.67	42.00	41.37	32.99	34.71	34.76
	HEC92	**10.11**	10.39	10.39	12.13	14.43	14.71	10.18	10.55	10.56
	KFU93	**13.21**	13.66	13.56	16.70	17.87	17.72	13.57	14.04	13.92
	LSE91	**10.57**	11.55	11.57	11.68	13.32	13.22	10.94	11.84	11.78
	PUR93	5.05	5.18	5.17	5.00	5.18	5.18	**5.00**	5.18	5.18
	RYE92	**8.49**	8.76	8.72	10.82	12.65	12.77	8.78	9.18	9.1
	STA83	**157.03**	157.11	157.11	161.42	182.28	185.72	157.03	157.15	157.15
	TRE92	**8.12**	8.37	8.34	9.54	10.25	10.32	8.22	8.66	8.63
	UTA92	**3.43**	3.72	3.74	3.72	3.91	3.89	3.58	3.73	3.74
	UTE92	**24.91**	24.97	24.96	27.71	32.47	32.35	**24.91**	25.13	25.1
	YOR83	**35.27**	35.98	35.93	39.93	43.53	43.62	35.63	36.40	36.42
ITC 2007	EXAM1	8099	8778	8690	7146	10771	9935	**6942**	7738	7718
	EXAM2	1259	2855	2660	**588**	1825	871	788	949	940
	EXAM3	12029	13717	13265	**11749**	13001	12598	11928	13159	12988
	EXAM4	14538	16131	15749	27498	32651	32656	**13631**	15499	15160
	EXAM5	4376	4792	4637	**4499**	5697	5336	4486	4830	4766
	EXAM6	26605	27340	27260	28130	30447	29655	**26645**	27893	27805
	EXAM7	5715	6084	5978	5289	5562	5567	**5539**	5911	5859
	EXAM8	12330	19849	17330	11143	26444	12128	**10537**	11367	11400
	EXAM9	1512	2047	2014	1464	2985	2225	**1114**	1213	1215
	EXAM10	14458	15875	15702	15691	22876	20678	**14151**	14728	14715
	EXAM11	37910	42409	41481	35757	42336	41460	**35589**	40107	39772
	EXAM12	5551	6117	5982	5986	7719	7605	**5217**	5343	5348
YEDITEPE	YUE20011	88	113	107	69	257	258	**49**	54	54
	YUE20012	126	150	146	152	355	377	**98**	111	111
	YUE20013	**29**	29	29	45	53	55	**29**	29	29
	YUE20021	97	157	120	143	507	347	**46**	62	62
	YUE20022	150	181	185	281	462	480	**133**	147	146
	YUE20023	56	56	56	59	92	87	**56**	57	56
	YUE20031	186	915	865	274	6022	6440	**121**	167	167
	YUE20032	**362**	699	584	650	3904	3721	370	405	405
NOTTINGHAM	NOTT-I	12103	14318.48	14212	**11805**	20040	15430	12398	13193	13127
	NOTT-II-32	**0**	0.33	0	**0**	351	372	**0**	0	0
	NOTT-II-31	**0**	1.86	1	**0**	382	356	**0**	0	0
	NOTT-II-30	**0**	4.29	3	**0**	386	465	**0**	2	1
	NOTT-II-26	21	38.29	40	**0**	1081	1477	**9**	29	31
	NOTT-II-23	96	134	130	**84**	1678	2217	100	124	120

Table 3.24: The Experimental Results of Comparing Three Different Strategies: SR-XGD-HH, RL-XGD-HH, and SA-XGD-HH.

Dataset	Instance	SR-XGD-HH			RL-XGD-HH			SA-XGD-HH		
		Best	Avg	Med	Best	Avg	Med	Best	Avg	Med
CARTER	CAR91	5.35	5.51	5.5	**5.25**	5.61	5.61	5.34	5.48	5.45
	CAR92	**4.44**	4.65	4.66	4.79	5.04	5.06	4.46	4.65	4.65
	EAR83	**34.44**	37.61	37.35	38.35	41.66	41.48	36.94	37.92	37.76
	HEC92	10.64	11.25	11.3	11.64	15.19	15.16	**10.63**	11.30	11.27
	KFU93	13.92	14.66	14.74	16.21	17.87	17.69	**13.85**	14.61	14.56
	LSE91	**11.09**	11.96	12.05	11.99	13.52	13.46	11.42	12.09	12.07
	PUR93	**4.97**	5.18	5.18	5.03	5.17	5.2	5.06	5.20	5.18
	RYE92	8.97	9.33	9.34	11.46	13.42	13.5	8.96	9.50	9.57
	STA83	157.06	157.24	157.18	160.85	176.40	178.6	**157.03**	157.29	157.24
	TRE92	**8.74**	9.08	9.08	9.33	10.14	10.14	8.75	9.18	9.2
	UTA92	3.65	3.71	3.72	3.73	3.97	3.96	**3.60**	3.76	3.75
	UTE92	**25.06**	25.54	25.44	26.65	32.60	32.77	25.11	25.56	25.42
	YOR83	**37.17**	38.80	38.44	39.60	44.94	45.47	37.50	38.86	38.97
ITC 2007	EXAM1	8321	9208	9173	**7663**	14117	15633	8059	8622	8577
	EXAM2	1013	1539	1572	**603**	1390	846	806	987	1001
	EXAM3	12066	13375	13219	**11717**	14352	13242	11815	13613	12735
	EXAM4	**14865**	17150	17200	23505	32234	32258	15543	17479	17435
	EXAM5	**4081**	5110	4898	4657	6480	5513	4460	4792	4629
	EXAM6	**27000**	27861	27945	28050	30456	30170	27055	27683	27775
	EXAM7	5649	5981	5951	**5166**	5737	5698	5662	5949	5927
	EXAM8	12050	12670	12628	**12027**	35397	12937	11251	12084	11837
	EXAM9	**1169**	1251	1251	1456	3273	2688	1177	1276	1265
	EXAM10	**14598**	15084	15049	15709	23890	20955	14794	15225	15196
	EXAM11	37107	42473	41882	**35761**	45243	43618	36005	43366	42530
	EXAM12	5275	5522	5511	6363	8047	7699	**5247**	5492	5465
YEDITEPE	YUE20011	**50**	67	64	63	319	242	55	69	68
	YUE20012	**109**	142	139	161	384	389	113	141	140
	YUE20013	**29**	30	29	33	55	55	**29**	31	30
	YUE20021	64	97	80	101	430	392	**51**	91	89
	YUE20022	170	212	211	233	558	447	**163**	199	205
	YUE20023	**56**	63	64	61	79	76	**56**	63	61
	YUE20031	**124**	170	165	279	5573	4676	146	191	190
	YUE20032	405	452	452	648	3679	2787	**377**	436	436
NOTTINGHAM	NOTT-I	13574	15069.29	15093	**11028**	16539	13963	12572	13926	13489
	NOTT-II-32	**0**	0	0	**0**	285	359	**0**	0	0
	NOTT-II-31	**0**	0.14	0	**0**	349	449	**0**	1	0
	NOTT-II-30	**0**	0.81	0	**0**	562	631	**0**	2	1
	NOTT-II-26	15	27.9	27	**9**	886	1244	12	31	30
	NOTT-II-23	105	141	143	**65**	1921	2184	118	153	151

Table 3.25: The Experimental Results of Comparing Three Different Hyper-heuristic Strategies: SR-MXGD-HH, RL-MXGD-HH, and SA-MXGD-HH.

Dataset	Instance	SR-MXGD-HH			RL-MXGD-HH			SA-MXGD-HH		
		Best	Avg	Med	Best	Avg	Med	Best	Avg	Med
CARTER	CAR91	5.34	5.47	5.46	5.32	5.67	5.72	**5.26**	5.43	5.44
	CAR92	**4.24**	4.52	4.52	4.71	4.95	4.95	4.40	4.62	4.61
	EAR83	**32.92**	33.97	33.65	39.64	42.44	42.42	35.03	37.87	38.1
	HEC92	**10.50**	10.75	10.74	13.11	14.40	14.35	10.84	11.38	11.41
	KFU93	**13.23**	13.72	13.75	16.27	17.51	17.49	13.61	14.32	14.24
	LSE91	**10.89**	11.61	11.61	12.39	13.38	13.3	11.26	12.13	12.16
	PUR93	4.98	5.14	5.11	**4.90**	5.19	5.16	5.07	5.20	5.19
	RYE92	**8.55**	8.82	8.79	10.91	12.98	12.75	9.00	9.46	9.41
	STA83	**157.05**	157.18	157.18	159.50	176.28	175.73	157.06	157.27	157.28
	TRE92	**8.03**	8.29	8.28	9.55	10.05	10.02	8.76	9.15	9.17
	UTA92	**3.47**	3.69	3.69	3.75	3.93	3.90	3.64	3.74	3.73
	UTE92	**24.86**	25.01	24.99	30.26	34.22	34.03	25.04	25.86	25.68
	YOR83	**35.69**	36.60	36.67	41.69	44.69	44.4	38.01	39.60	39.48
ITC 2007	EXAM1	8905	9970	9671	7461	13145	11374	**7675**	8084	8053
	EXAM2	1179	2759	2488	**630**	3194	926	782	966	977
	EXAM3	**11590**	13487	12928	11717	14403	12833	12195	13303	12775
	EXAM4	14599	15536	15361	21997	32346	31338	**14549**	16593	16478
	EXAM5	**4407**	5060	4869	4335	6761	5774	4479	4906	4789
	EXAM6	**26610**	27464	27415	28165	31372	30920	27055	27923	27970
	EXAM7	5713	6193	6087	**5319**	5737	5638	5699	6206	5895
	EXAM8	13208	17413	15981	11655	22479	13021	**10971**	11510	11444
	EXAM9	1147	1257	1269	1367	2978	1905	**1128**	1278	1288
	EXAM10	**14223**	14880	14893	15355	25440	23851	14591	15023	15002
	EXAM11	36637	42785	42293	36233	43760	42676	**35355**	42155	41651
	EXAM12	**5227**	5394	5353	6578	8481	8484	5285	5574	5563
YEDITEPE	YUE20011	**48**	59	57	110	288	289	50	66	64
	YUE20012	**107**	120	120	213	443	400	117	141	139
	YUE20013	**29**	29	29	30	53	52	**29**	30	29
	YUE20021	**49**	64	65	126	420	322	55	89	88
	YUE20022	**125**	155	151	311	456	460	156	212	209
	YUE20023	**56**	59	58	65	89	84	**56**	60	58
	YUE20031	**148**	621	693	345	4874	5333	161	200	197
	YUE20032	**362**	512	433	685	3206	2904	400	464	456
NOTTINGHAM	NOTT-I	13279	15135.52	15154	**12059**	20981	18562	12364	13834	13781
	NOTT-II-32	**0**	0.19	0	**0**	254	233	**0**	0	0
	NOTT-II-31	**0**	1.81	0	**0**	302	362	**0**	0	0
	NOTT-II-30	**0**	4.67	5	**0**	563	589	**0**	2	0
	NOTT-II-26	13	32.57	29	**4**	716	65	16	29	29
	NOTT-II-23	113	145	146	**55**	1386	2002	106	146	139

Table 3.26: The Experimental Results of Comparing: Hyper-heuristic Strategies summarised in Table 3.22 in terms of the best solutions

Dataset	Instance	SR-GD	RL-GD	SA-GD	SR-XGD	RL-XGD	SA-XGD	SR-MXGD	RL-MXGD	SA-MXGD
CARTER	CAR91	**5.23**	5.53	5.29	5.35	5.25	5.34	5.34	5.32	5.26
	CAR92	**4.23**	4.79	4.27	4.44	4.79	4.46	4.24	4.71	4.4
	EAR83	32.98	38.67	32.99	34.44	38.35	36.94	**32.92**	39.64	35.03
	HEC92	**10.11**	12.13	10.18	10.64	11.64	10.63	10.50	13.11	10.84
	KFU93	**13.21**	16.70	13.57	13.92	16.21	13.85	13.23	16.27	13.61
	LSE91	**10.57**	11.68	10.94	11.09	11.99	11.42	10.89	12.39	11.26
	PUR93	5.05	5.00	5	4.97	5.03	5.06	4.98	**4.90**	5.07
	RYE92	**8.49**	10.82	8.78	8.97	11.46	8.96	8.55	10.91	9.00
	STA83	**157.03**	161.42	157.03	157.06	160.85	**157.03**	157.05	159.50	157.06
	TRE92	8.12	9.54	8.22	8.74	9.33	8.75	**8.03**	9.55	8.76
	UTA92	**3.43**	3.72	3.58	3.65	3.73	3.6	3.47	3.75	3.64
	UTE92	24.91	27.71	24.91	25.06	26.65	25.11	**24.86**	30.26	25.04
	YOR83	**35.27**	39.93	35.63	37.17	39.60	37.50	35.69	41.69	38.01
ITC 2007	EXAM1	8099	7146	**6942**	8321	7663	8059	8905	7461	7675
	EXAM2	1259	**588**	788	1013	603	806	1179	630	782
	EXAM3	12029	11749	11928	12066	11717	11815	**11590**	11717	12195
	EXAM4	14538	27498	**13631**	14865	23505	15543	14599	21997	14549
	EXAM5	4376	4499	4486	**4081**	4657	4460	4407	4335	4479
	EXAM6	**26605**	28130	26645	27000	28050	27055	26610	28165	27055
	EXAM7	5715	5289	5539	5649	**5166**	5662	5713	5319	5699
	EXAM8	12330	11143	**10537**	12050	12027	11251	13208	11655	10971
	EXAM9	1512	1464	**1114**	1169	1456	1177	1147	1367	1128
	EXAM10	14458	15691	**14151**	14598	15709	14794	14223	15355	14591
	EXAM11	37910	35757	35589	37107	35761	36005	36637	36233	**35355**
	EXAM12	5551	5986	**5217**	5275	6363	5247	5227	6578	5285
YEDITEPE	YUE20011	88	69	49	50	63	55	**48**	110	50
	YUE20012	126	152	**98**	109	161	113	107	213	117
	YUE20013	**29**	45	**29**	**29**	33	**29**	**29**	30	**29**
	YUE20021	97	143	**46**	64	101	51	49	126	55
	YUE20022	150	281	133	170	233	163	**125**	311	156
	YUE20023	**56**	59	**56**	**56**	61	**56**	**56**	65	**56**
	YUE20031	186	274	**121**	124	279	146	148	345	161
	YUE20032	**362**	650	370	405	648	377	362	685	400
NOTTINGHAM	NOTT-I	12103	11805	12398	13574	**11028**	12572	13279	12059	12364
	NOTT-II-32	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	NOTT-II-31	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	NOTT-II-30	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	NOTT-II-26	21	**0**	9	15	9	12	13	4	16
	NOTT-II-23	96	84	100	105	65	118	113	**55**	106

modified extended great deluge algorithm instead of standard great deluge as shown by Table 3.25, we can observe that SR-MXGD-HH outperforms RL-MXGD-HH and SA-MXGD-HH over Carter and Yeditepe datasets. Whereas over ITC 2007 dataset, SR-MXGD is just comparable with SA-MXGD-HH and over Nottingham dataset, RL-MXGD-HH outperforms SR-MXGD-HH and SA-MXGD-HH. From Table 3.23-Table 3.25, we can see that combined with whichever move acceptance strategy, over Nottingham dataset, reinforcement learning based low-level heuristic selection strategy outperforms the other simple random and self-adaptive strategies.

Table 3.26 compares the best solutions over 21 runs from all the hyper-heuristic strategies summarised in Table 3.22. From Table 3.26, we can observe that over Carter dataset, SR-GD-HH outperforms the other algorithms whereas over ITC 2007 and Yeditepe SA-GD-HH outperforms the other algorithms. Lastly, over Nottingham dataset, RL-XGD-HH outperforms the other algorithms.

Since compared to our proposed uncapacitated examination timetabling problem domain which has four low-level heuristics, our proposed capacitated examination timetabling problem domain has more number of low-level heuristics, i.e. fourteen low-level heuristic, we tried to tune the parameter of reinforcement learning and self-adaptive learning. In the reinforcement learning, we set the maximum reward from 20 in the previous setting to 70 and in self-adaptive learning we set the length of low-level heuristic list from 200 in the previous setting to 700. However, the experimental results showed that the the difference is not significant.

Finally, Table 3.27 compares the experimental results from our proposed algorithms/solver with results reported from scientific literature. From Table 3.27 we can observe that generally, our proposed solver outperforms the benchmarks over Carter, Yeditepe and Nottingham datasets. Whereas over ITC 2007 though our solvers do not outperform, they are still comparable with other approaches. It is worth noting that Müller solver [221] is the winner of the ITC 2007 competition. It is also worth noting that the instances: `EXAM9-EXAM12`, are hidden instances in the competition. Therefore, we can observe that Müller solver was not well-performed for the hidden instances.

3.4 Conclusion

Over both the uncapacitated examination timetabling problem, i.e. Carter dataset, and the capacitated examination timetbaling problems, i.e. ITC 2007, Yeditepe, and Nottingham dataset, three different initial feasible construction methods, i.e. sequen-

Table 3.27: The experimental results over ITC 2007 datasets with the standard single objective function from three variant of hyper-heuristics: SR-GD-HH, RL-GD-HH, and SA-GD-HH.

Dataset	Instance	Our Solvers			BENCHMARK				
		SR-GD-HH	RL-GD-HH	SA-GD-HH	[63]	[280]	[3]	[82]	[7]
CARTER	CAR91	5.24	5.27	5.24	**5.03**	5.14	5.12	5.19	4.76
	CAR92	4.23	4.57	4.25	4.22	4.70	4.41	4.31	**3.94**
	EAR83	**33.14**	38.86	33.36	36.06	37.86	36.91	35.79	33.61
	HEC92	**10.08**	11.62	10.1	11.71	11.9	11.31	11.19	10.56
	KFU93	**13.21**	16.48	13.33	16.02	15.3	14.75	14.51	13.44
	LSE91	10.56	12.07	**10.54**	11.15	12.33	11.41	10.92	10.87
	PUR93	4.96	**4.90**	4.95	NA	5.37	5.87	NA	NA
	RYE92	**8.51**	11.50	8.76	9.42	10.71	9.61	NA	8.81
	STA83	**157.03**	161.53	157.06	158.86	160.12	157.52	157.18	157.09
	TRE92	**7.85**	9.27	8.3	8.37	8.32	8.76	8.49	7.94
	UTA92	3.57	3.82	3.56	3.37	3.88	3.54	3.44	**3.27**
	UTE92	**24.86**	27.81	**24.86**	27.99	32.67	26.25	26.7	25.36
	YOR83	**35.03**	40.51	35.33	39.53	40.53	39.67	39.47	35.74
		SR-GD-HH	RL-GD-HH	SA-GD-HH	[221]	[122]	[3]	[18]	[82]
ITC 2007	EXAM1	6579	7019	5809	**4370**	6060	5231	5328	6235
	EXAM2	584	535	490	**400**	515	433	512	2974
	EXAM3	11153	11592	10819	**10049**	23580	9265	10178	15832
	EXAM4	**13233**	21992	14100	18141	NA	17787	16465	35106
	EXAM5	3658	4610	3596	**2988**	4855	3083	3624	4873
	EXAM6	26515	28130	**26075**	26950	27605	26060	26240	31756
	EXAM7	5145	5151	5185	**4213**	6065	10712	4562	11562
	EXAM8	9348	11405	9180	**7861**	9038	12713	8043	20994
	EXAM9	1074	1382	**1032**	1047	1184	1111	NA	NA
	EXAM10	**14201**	15573	14438	16682	15561	14825	NA	NA
	EXAM11	35445	34462	33607	34129	NA	**28891**	NA	NA
	EXAM12	**5163**	7013	5202	5535	5483	6181	NA	NA
		SR-GD-HH	RL-GD-HH	SA-GD-HH	[221]				
YEDITEPE	YUE20011	**48**	79	**48**	62				
	YUE20012	**98**	137	103	125				
	YUE20013	**29**	31	**29**	**29**				
	YUE20021	**46**	93	48	70				
	YUE20022	**123**	246	123	170				
	YUE20023	**56**	70	56	70				
	YUE20031	**115**	262	123	223				
	YUE20032	**340**	537	362	440				
		SR-GD-HH	RL-GD-HH	SA-GD-HH	[74]	[85]			
NOTT	NOTT-I	**11045**	11978	10837	N.A	NA			
	NOTT-II-32	**0**	0	0	0	NA			
	NOTT-II-31	**0**	0	0	0	NA			
	NOTT-II-30	**0**	0	0	0	NA			
	NOTT-II-26	**0**	5	0	49	44			
	NOTT-II-23	**32**	56	37	348	98			

tial construction with maximal clique and saturation degree heuristic, adaptive ordering heuristic with hierarchical hybridisation of basic graph colouring heuristics, and adaptive linear combination ordering heuristic with heuristic modifier based on squeaky wheel optimisation (SWO) were proposed and studied in this chapter. Moreover, to improve the initial solution, several hyper-heuristics with different strategies combining different low-level heuristic selection strategies, i.e. simple random, reinforcement learning, and self-adaptive learning with different move acceptance strategies, i.e. great deluge algorithm, hill climbing, and late acceptance were studied.

The experimental results showed that in constructing initial feasible solutions, adaptive linear combination ordering heuristic with heuristic modifier based on SWO is the most effective method capable of producing feasible solution for all problem instances from the four different datasets. Whereas, in improving the initial solutions, the experimental results suggested that over uncapacitated examination timetabling problem, the combination of self-adaptive learning and great deluge algorithm as low-level heuristic selection and move acceptance strategies respectively, i.e. `SA-GD-HH`, outperforms the other combinations. It is also shown that, compared to results from other methods recently reported in the literature, `SA-GD-HH` could produce very competitive results. It outperforms 8 out of 13 Problem instances. On other hand, over capacitated examination timetabling problems, SA-GD-HH outperforms the other approaches, though over Carter dataset which have been modified as capacitated examination timetabling problem, SR-GD-HH outperforms the others combination. Why the proposed algorithm well-performs in one problem instances but under-performs in other problem instances is interesting to be investigated in the future works.

The main contributions of this chapter are hyper-heuristic approach with self-adaptive learning as low-level heuristic selection strategy and great deluge algorithm as move acceptance strategy to solve examination timetabling problem and an extension of the current version of HyFlex, a hyper-heuristic framework, with the examination timetabling problem as a new problem domain. This chapter also showed that the basic hyper-heuristic approach proposed in this chapter, over which the algorithm will be extended to solved examination timetabling with fairness in the following chapters, is very competitive with other approaches reported in the scientific literatures.

Chapter 4

The Extended Examination Timetabling Problem Formulation and 'Sum of Power' Approach

4.1 Introduction

As shown in Appendix B, our finding from the survey of examination timetable preferences of university students shows that there are some issues that are not taken into account in the current state-of-the-art examination timetabling problem formulations. The most significant issues are fairness among students and the 'hardness' of the examinations. As discussed in Chapter 2, the existing examination timetabling formulations in the literature have not taken these issues into consideration. Therefore, in this chapter we intend to extend problem formulation defined in ITC 2007 in order to take the survey finding into account. Moreover, in this chapter, a first attempt to enforce fairness in examination timetabling using an approach so-called 'sum of powers' is also discussed.

4.2 Extended Problem Formulation

Mainly, there are three extensions of the current examination timetabling proposed in this thesis. First, an extension to consider hardness of examination. Second, an extension to take fairness issues into take account. Third, an extension to view the objective function of examination timetabling problem from multiple stakeholders. These extensions are explained as follows.

4.2.1 An Extension to Consider The Hardness of Examinations

Observing the problem formulation on ITC 2007 data set (see Chapter 2), we know that representing the real-world examination timetabling problems is complex and very challenging. In terms of soft constraint penalty, it has been quite representative for students, lecturer, and estates preferences over examination timetable. However, it can be observed that the formulation in ITC 2007 assumes that all exams have the same level of difficulty. Contrarily, findings from our survey as discussed in Appendix B show that the majority of students believed that some exams are more difficult than the others.

Adapted from [205], suppose E is a set of examinations, S is a set of students, and P is number of periods. Additionally, three binary variables y_{pq}, t_{is} and X_{ip}^{P} are defined as below:

$$y_{pq} = \begin{cases} 1 & \text{iff periods } p \text{ and } q \text{ are on the same day} \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

$$t_{is} = \begin{cases} 1 & \text{iff student } s \text{ is enrolled in exam } i \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

$$X_{ip}^{P} = \begin{cases} 1 & \text{iff exam } i \text{ is scheduled in period } p \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

In order to consider the differing difficulty level of examinations, we propose to extend soft constraints penalty formulation in ITC 2007 by introducing difficulty index for each exam. Suppose i is an exam in a set of E exams, for each i, d_i is defined as difficulty index of exam i, representing the weight of difficulty of exam i. The values of d_i range between 1 and 3, expressing exam difficulty (e.g. 1 = easy, 2 = medium, 3 = hard).

Given this extra data, the modified 'two exams in a row', 'two exams in a day', and 'period spread' are defined respectively as follows:

Two Exams in a Row Penalty

Provided that student s enrolled in both exam i and j (two distinct exams), and exam j is scheduled in the same day and immediately after exam i , two exams in a row (C_s^{TR}) penalty is defined as follow:

$$C_s^{TR} = \sum_{\substack{i,j \in E \\ i \neq j}} \sum_{\substack{p,q \in P \\ q=p+1 \ \& \ y_{pq}=1}} W^{TR}(d_i, d_j) t_{is} t_{js} X_{ip}^P X_{jq}^P \tag{4.4}$$

where $W^{TR}(d_i, d_j)$ is a matrix of penalty values. Note that it is not necessarily symmetric, e.g. to allow different preferences for 'easy then difficult' and 'difficult then easy' in the exam sequence.

Two Exams in a Day Penalty

Given the same condition with that in two exams in a row (C_s^{TR}) penalty but exam j and i are not scheduled in two consecutive periods. C_s^{TD} is defined as:

$$C_s^{TD} = \sum_{\substack{i,j \in E \\ i \neq j}} \sum_{\substack{p,q \in P \\ q>p+1 \ \& \ y_{pq}=1}} W^{TD}(d_i, d_j) t_{is} t_{js} X_{ip}^P X_{jq}^P \tag{4.5}$$

Period Spread Penalty

Given that student s enrolled in both exam i and j (two distinct exams) and g is the period gap between i and j, C_s^{PS} is defined as:

$$C_s^{PS} = \sum_{\substack{i,j \in E \\ i \neq j}} \sum_{\substack{p,q \in P \\ p<q \leq p+g}} W^{PS}(d_i, d_j) t_{is} t_{js} X_{ip}^P X_{jq}^P \tag{4.6}$$

with associated matrices $W^{TD}(d_i, d_j)$ and $W^{PS}(d_i, d_j)$ of penalty parameters.

4.2.2 An Extension to Take Fairness Issues into Account

In term of objective function, in ITC 2007, the objective function is to minimise the total of soft constraint violation penalty over all students. In this sense, we don't know

how penalties are distributed over individual students. In other words, it could be very possible case that some students have very good timetable i.e. the exams are spread out evenly while some other students have the bad one i.e. the exams are scheduled too close to each other. Therefore, the fairness issue among students, as confirmed by our survey finding, becomes important to consider.

In order to promote fairness, a useful measure can be the Jain Fairness Index (JFI [159]). For a set C of students, with their associated penalties $P(C) = \{p_i\}$, the JFI is simply a convenient non-linear combination of the variance on the penalties, σ_P^2, and their mean value, $\bar{P}$:

$$J(C) = \left(1 + \sigma_P^2/\bar{P}^2\right)^{-1} = \frac{\left(\sum_{i \in C} p_i\right)^2}{|C| \sum_{i \in C} p_i^2} \tag{4.7}$$

By this definition, in order to incorporate fairness among the entire student body within ITC 2007 problem formulation, an objective function could be defined to maximise:

$$J(C) = \left(1 + \sigma_P^2/\bar{P}^2\right)^{-1} = \frac{\left(\sum_{i \in C} p_i\right)^2}{|C| \sum_{i \in C} p_i^2}, \text{where } p_i = C_i^{TR} + C_i^{TD} + C_i^{PS} \tag{4.8}$$

Note that only three soft constraints from ITC 2007 examination problem formulations that related to each individual student, namely, two in row, two in day, and period spread penalty; instead of all penalties as given in Equation 2.8 in Chapter 2. Thus, only these constraints must be considered in defining fairness among students within examination timetabling problem.

Referring to equation 4.8, since we know that $\sigma_P/\bar{P}$ is coefficient of variation (CV), and $\sigma_P/|\bar{P}|$ is relative standard deviation (RSD), the objective function could be defined also to minimise:

$$(CV(C))^2 = (RSD(C))^2 \tag{4.9}$$

The nice property of JFI is the ability to measure fairness quantitatively within the range $1/N$ and 1, in which N is the total number of entities, e.g. students. The value $1/N$ is achieved when the penalty distribution is completely unfair, e.g. the penalty is associated to one student only, conversely 1 is achieved when the penalty distribution is completely fair, e.g. each student is associated with exactly equal penalty.

Since the JFI should be maximised while the standard objective function should be

minimised, in order to turn the second objective into minimisation, the second objective is redefined as unfairness measure, let us say, $AJFI$. $AJFI$ is defined in Equation 4.10 below.

$$AJFI = 1 - JFI \tag{4.10}$$

4.2.2.1 Weighted Fairness Objective Function

Referring to equation 4.8 it is clear that to be completely fair each student must be associated with exactly equal penalty. However, in practice students might have different number of examinations. From above definition of soft penalty constraints (see equation 4.4, 4.5, and 4.6), it can be observed that students with more examinations are more likely to be associated with more penalties. Thus, forcing students to have the same penalty is unfair.

To tackle this problem, a generalised definition of JFI so-called weighted JFI [95] was used. Suppose x_i, w_i are total penalty and the number of enrolled exams of student i, weighted JFI is defined in equation 4.11

$$\begin{aligned} &J^*(x_1, x_2, x_3, ..., x_N; w_1, w_2, w_3, ..., w_N) = \\ &\left(n \sum_{i=1}^{n} \left(\frac{x_i}{\sum_{j=1}^{n} x_j} - \frac{w_i}{\sum_{j=1}^{n} w_j} \right)^2 + 1 \right)^{-1} \end{aligned} \tag{4.11}$$

Interestingly, as proven in [95] the weighted JFI also poses all the nice property that standard JFI have. Thus, the value is also between 1/N and 1. The complete fair will be achieved if only the penalty associated to each student divided by the number of examinations that the student has is equal.

4.2.2.2 Fairness Within a Cohort Objective Function

As discussed in Appendix B, our survey finding shows that 'fairness within a course' or 'fairness within a cohort' as a more general term, is more important than fairness within the entire student body. In real world, a cohort could refer to a set of students enrolled in the same school/faculty/year rather than in the same course.

To promote 'fairness within a cohort' another objective function is introduced. Suppose C_i is a set of students in cohort i of a total number k cohort, and $J(C_i)$ is Jain fairness

index over C_i then the candidate function could be:

$$(maximise)JFIC = 1/k \sum_{i=1}^{k} J(C_i) \tag{4.12}$$

Then, to transform into a minimisation problem, $AJFIC$ is defined in Equation 4.13 below.

$$AJFIC = 1 - JFIC \tag{4.13}$$

To illustrate how the extended formulation works, let suppose we have two cases as shown in Table 4.1 and Table 4.2.

In the first case, suppose there are two cohorts with 2 students each and 2 solutions with associated penalties. Whereas `P1` and `P2` refer to penalty associated to each student in cohort 1 and cohort 2, `J1` and `J2` refer to `JFI` on cohort 1 and cohort 2 respectively.

Table 4.1: Illustration Example 1

Soln	**P1**	**P2**	**avg(P)**	**Jall**	**J1**	**J2**	**avg(J)**
S1	{4,4}	{2,2}	**3**	0.9	1.0	1.0	**1.0**
S2	{4,2}	{4,2}	**3**	0.9	0.9	0.9	0.9

In the illustration example 1, as we can observe from Table 4.1, both `S1` and `S2` have the same average total penalty. Further, in term of fairness, treating it over entire student body also gives both solutions the same `JFI` (`Jall`) i.e. `0.9`. However, `S1` has average `JFI` per cohort, i.e. avg(J), greater than `S2`, i.e. `1` against `0.9`. It indicates that `S1` is fairer within each cohort than `S2`. Therefore, in this case the first solution (`S1`), we believe, would be preferred by the students.

The illustration example 2, as shown by Table 4.2, is a more complicated case in which there are three students per cohort and three differing solutions instead. Comparing `S1` and `S2`, it can be seen that although `S2` has a slightly lower overall penalty, `S1` is much fairer within cohort, hence `S1` might be preferred over `S2`. On the other hand, compared with the other solutions, `S3` is the fairest solution overall, but it is not quite as good on the overall penalty or the fairness within each cohort. It suggests that there should be trade-off between overall total penalty, overall fairness, and fairness within cohort in deciding which one is the best solution overall.

Obviously, if this were the only factor then some courses might have high penalties, but the intent is that this 'fairness within course' is another objective in the overall multi-objective problem, and there would be a trade-off.

Table 4.2: Illustration Example 2

Soln	P1	P2	avg(P)	Jall	J1	J2	avg(J)
S1	{8,8,9}	{2,2,2}	5.2	0.725	0.997	1.0	**0.998**
S2	{8,8,2}	{8,2,2}	**5.0**	0.735	0.82	0.67	0.742
S3	{7,7,9}	{4,3,3}	5.5	**0.852**	0.985	0.980	0.983

Regarding cohort, since in the existing benchmark dataset there is no information regarding cohort information, i.e. course and year, we extend the existing dataset with cohort information. For Carter and ITC 2007 dataset, we cluster the students based on the similarity of exams/course enrolled by the students using standard clustering technique in data mining, i.e. k-means algorithm [152]. Whereas, for Yeditepe and Nottingham, we determined the cohort based on faculty and the year of student from the real data. We called this cohort as "specified cohort". These extended datasets can be accessed online [1].

In order to take different number of exams enrolled by the students, as an alternative to weighted fairness, we also proposed to define cohorts based on the number of exams enrolled by the students. We named this as 'exam-count-based cohort'. Students with the same number of enrolled exams are put in the same cohort. Table 4.3 summarised the number of cohort for each problem instance, either based on the specified cohort or number of exam cohort.

4.2.3 Viewing the objective function of examination timetabling problem from multiple stakeholders

Basically, this work is inspired by the work of [69]. In which rather than consider the objective function as single aggregate function (see Section 2.1.2 in Chapter 2), the objective functions are grouped into four. These groups represent the differing stake holders of examination timetable: students, invigilator, markers, and estates (see Figure 4.1). Note that *2R*, *2D*, *PS*, *NMD*, and *FL* stand for 'Two-In-Row', 'Two-In-Day', 'Period Spread','Non-Mixed Duration', and 'Front Load' penalties respectively. Therefore, from single objective optimisation problem, now, it could be divided into four objective problem. More formally, it could be formulated as a minimisation problem of:

$$f(x) = [f_1(x), f_2(x), f_3(x), f_4(x)] \tag{4.14}$$

Where,

[1] http://cs.nott.ac.uk/∼abm/research/exam-fairness-data/examproblemwithfairness/

Table 4.3: The number of cohort of the problem instances based on the specified cohort and exam-count-based Cohort

Dataset	Problem Instance	Number of Cohorts	
		Exam-count-based Cohort	Specified Cohort
NOTT	NOTT-I	9	9
	NOTT-II-32	9	9
	NOTT-II-31	9	9
	NOTT-II-30	9	9
	NOTT-II-26	9	9
	NOTT-II-23	9	9
CARTER	CAR-S-91	9	3
	CAR-F-92	7	3
	EAR-F-83	10	3
	HEC-S-92	7	3
	KFU-S-93	8	3
	LSE-F-91	8	3
	PUR-S-93	9	3
	RYE-S-93	10	3
	STA-F-83	4	3
	TRE-S-92	6	3
	UTA-S-92	7	3
	UTE-S-92	6	3
	YOR-F-83	9	3
ITC 2007	EXAM1	11	3
	EXAM2	10	3
	EXAM3	9	3
	EXAM4	8	3
	EXAM5	8	3
	EXAM6	7	3
	EXAM7	12	3
	EXAM8	10	3
	EXAM9	11	3
	EXAM10	18	3
	EXAM11	9	3
	EXAM12	6	3
YEDITEPE	YUE20011	9	4
	YUE20012	9	4
	YUE20013	4	4
	YUE20021	11	5
	YUE20022	11	6
	YUE20023	4	6
	YUE20031	9	6
	YUE20032	10	6

1. $f_1 = C^{stu} = \sum_{s \in S}(w^{2R}C_s^{2R} + w^{2D}C_s^{2D} + w^{PS}C_s^{PS})$
2. $f_2 = C^{inv} = w^{NMD}C^{NMD}$
3. $f_3 = C^{mar} = w^{FL}C^{FL}$
4. $f_4 = C^{est} = C^P + C^R$

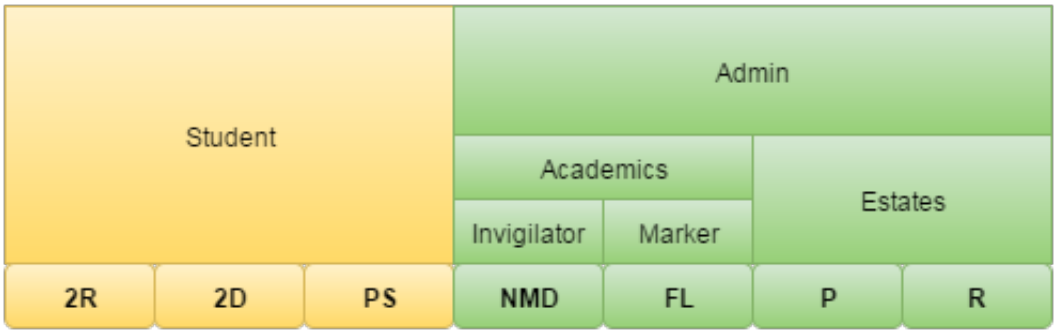


Figure 4.1: Grouping the examination timetable's objective functions based on its Stakeholders.

We believe that this grouping does make sense. Because, in real-world scenario, it is most likely that each of these stakeholder has differing, or even conflicting interest. Students expect that their exams are spread out evenly along the exam period to allow them having enough revision' time for each exam. On the other hand, the markers, or lecturers of the exam expect to have enough time for marking, especially for exams with big number of students. Furthermore, invigilators prefer to invigilate an exam in a room where the students will finish at the same time. Whereas, the estates, the one who will be responsible with the room, expect to have no exam at particular room and time.

However, dividing the problems into four objective functions based on the stakeholders of exam timetable could be very complicated. It is because visualising and then analysing the trade-off between more than two objectives is not trivial. It might be also harder for the algorithms to solve the problem. As an alternative, in order to make it simpler, the objective functions could be grouping into two objective functions instead. For instance, as shown by Figure 4.1, the examination's stakeholders could be grouping into student and admin. Therefore, the bi-objective variant of ITC 2007 examination problem model could be defined as a minimisation problem of:

$$f(x) = [f_1(x), f_2(x)] \tag{4.15}$$

Where,

1. $f_1 = C^{stu} = \sum_{s \in S}(w^{2R}C_s^{2R} + w^{2D}C_s^{2D} + w^{PS}C_s^{PS})$

2. $f_2 = C^{admin} = w^{NMD}C^{NMD} + w^{FL}C^{FL} + C^P + C^R$

In this study, we will treat the problem as both a bi-objective problem and as a four objective problem.

4.3 'Sum of Powers' Approach to Enforce Fairness

This section evaluates 'Sum of Powers' as an objective function within a single stage approach that is proposed to optimise both the standard objective function and fairness concurrently. Beforehand, we evaluate different fairness quality indicators to measure fairness. We tested our proposed approach over the Carter dataset.

4.3.1 'Sum of Powers' Method

The algorithm used in this approach is self-adaptive great deluge hyper-heuristic which is exactly the same with the algorithm used to optimise standard benchmark objective function discussed in Chapter 3 (see Section 3.2.1). The only difference is the objective function. Instead of the standard objective function, 'sum of powers' objective function is used to enforce fairness over all students, that henceforth, it will be referred to as global fairness.

The original objective function formulation of Carter dataset examination timetabling problem (see Section 2.1.1 in Chapter 2) can be rewritten differently as follows. Suppose P_s is total penalty associated to an individual student s and K is the total number of students, the objective function is a linear summation of P_s as given by Equations 4.16-4.18. This objective function is to be minimised. The variables in the equations are summarised in Table 4.4.

$$LS = \frac{1}{K}\sum_{s=1}^{K} P_s \tag{4.16}$$

Where,

$$P_s = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} P(s,i,j) \tag{4.17}$$

$$P(s,i,j)) = \begin{cases} W_{|t_j - t_i|} & \text{if student s takes exams i and j} \\ 0 & \text{otherwise} \end{cases} \tag{4.18}$$

Table 4.4: Variable Description for Equation 4.16- 4.18

Variable	Description
$P(s,i,j)$	Penalty associated to student s as taking both examination i and j.
t_i,t_j	Timeslots in which examination i and j are scheduled respectively.
$W_{\|t_j-t_i\|}$	The penalty weight whenever a student who has to sit for both examinations are scheduled $\mid t_j - t_i \mid$ apart equal to $2^{5-\|t_j-t_i\|}$ if $\mid t_j - t_i \mid \in \{1,2,3,4,5\}$ and equal to 0 if $\mid t_j - t_i \mid > 5$.
N	The total number examinations

It is clear from the formulation that the distribution of penalties among students is not taken into consideration, whereas it is very possible that the distribution is unfair. For example two solutions with different penalty distribution, i.e. A{3,3} and B{1,5} are equal in term of linear summation, however, since A is fairer than B, we prefer A to B. Within the original linear summation objection function, we can not do that. Therefore, in order to enforce fairness and in the same time also minimise the standard objective function, we evaluate, "sum of powers" as given in Equation 4.19. If q in Equation 4.19 is 2, then it will be "sum of square", whereas if q=∞ it will be min-max function, i.e. Equation 4.20. Respectively, within Equation 4.19, the original standard objective function, i.e. linear summation (LS) as shown by Equation 4.16, is 'Sum of Powers' with `q=1`.

$$SOP = \frac{1}{K}\sum_{s=1}^{K}(P_s)^q \tag{4.19}$$

$$Max = \max_{s_1,\ldots s_K} P_s \tag{4.20}$$

By sum of powers, either with $q = 2$ or sum of square or with q=∞, solution A, which is preferable, has lower objective function than solution B. It is illustrated in Table 4.5.

Table 4.5: An Illustrating Example of two solutions with different penalty distribution and their respective different objective functions: Linear summation (LS), Sum of Squares (SOS), and min-max (Max)

Solution	Objective Function Value		
	LS	SOS	Max
A {3,3}	3	9	3
B {1,5}	3	13	5

4.3.2 Experimental Results

We conducted several experiments reported in the following sections. Firstly, we evaluate different fairness objective function in Section 4.3.2.1. Secondly, we investigate the correlation between the number of iterations of search in the algorithm, standard objective function, and fairness objective function in Section 4.3.2.2. Thirdly, we evaluate 'Sum of Powers' function with different exponent numbers in Section 4.3.2.3. Lastly, in Section 4.3.2.4 we tested 'Sum of Powers' approach over all the problem instances of Carter dataset.

4.3.2.1 Evaluating Different Fairness Objective Functions

As discussed in literature review, in Section 2.8 of Chapter 2, there are different quality indicators used in the scientific literature to measure fairness. Among of them are Jain fairness index, variance, minmax, GINI index, and relative standard deviation. In order to know the correlation between these different fairness quality indicators and also their correlation with the original standard benchmark objective function, i.e. cost, we evaluated 1024 random feasible solutions for eight randomly chosen problem instances. Over these solutions, we measured their quality indicators' average value and conducted a standard statistical correlation test (we used Microsoft excel in this case). The correlation between Anti Jain Fairness Index (AJFI)=1-jfi, Variance (VAR), anti MinMax(aMM)=1-MM, GINI index, and relative standard deviation (RSD); and the standard benchmark objective function (cost) is shown in Table 4.6.

Table 4.6: Correlation between different fairness quality indicators and the standard benchmark objective function.

Instance	Corr	cost	AJFI	VAR	aMM	GINI	RSD	Instance	Corr	cost	AJFI	VAR	aMM	GINI	RSD
NOTT-I	cost	X	-0.22	**0.91**	0.00	0.36	-0.22	EXAM1	cost	X	-0.27	**0.88**	0.00	0.08	-0.25
	AJFI	X	X	0.16	0.00	**0.71**	**0.99**		AJFI	X	X	0.15	0.00	**0.75**	**0.99**
	VAR	X	X	X	0.00	**0.63**	0.16		VAR	X	X	X	0.00	0.41	0.16
	aMM	X	X	X	X	0.00	0.00		aMM	X	X	X	X	0.00	0.00
	GINI	X	X	X	X	X	**0.70**		GINI	X	X	X	X	X	**0.75**
CAR-S-F-91	cost	X	**-0.74**	**0.89**	0.00	**-0.72**	**-0.74**	EXAM12	cost	X	**-0.95**	**0.98**	0.00	**-0.81**	**-0.93**
	AJFI	X	X	-0.35	0.00	**0.96**	**1.00**		AJFI	X	X	**-0.87**	0.00	**0.92**	**0.99**
	VAR	X	X	X	0.00	-0.36	-0.35		VAR	X	X	X	0.00	**-0.70**	**-0.85**
	aMM	X	X	X	X	0.00	0.00		aMM	X	X	X	X	0.00	0.00
	GINI	X	X	X	X	X	**0.96**		GINI	X	X	X	X	X	**0.89**
YOR-F-83-I	cost	X	0.03	**0.79**	0.00	0.13	0.03	YUE20011	cost	X	-0.12	0.00	0.00	-0.12	-0.11
	AJFI	X	X	0.64	0.00	**0.97**	**1.00**		AJFI	X	X	-0.27	0.00	**1.00**	**0.99**
	VAR	X	X	X	0.00	0.69	0.64		VAR	X	X	X	0.00	-0.21	-0.27
	aMM	X	X	X	X	0.00	0.00		aMM	X	X	X	X	0.00	0.00
	GINI	X	X	X	X	X	**0.97**		GINI	X	X	X	X	X	**0.99**
PUR-S-93-I	cost	X	**-0.76**	**0.74**	0.00	**-0.74**	**-0.76**	YUE20032	cost	X	0.23	-0.02	0.00	0.23	0.23
	AJFI	X	X	-0.13	0.00	**0.97**	**1.00**		AJFI	X	X	-0.12	0.00	**1.00**	**1.00**
	VAR	X	X	X	0.00	-0.12	-0.13		VAR	X	X	X	0.00	-0.06	-0.12
	aMM	X	X	X	X	0.00	0.00		aMM	X	X	X	X	0.00	0.00
	GINI	X	X	X	X	X	**0.97**		GINI	X	X	X	X	X	**0.99**

From Table 4.6, it can be observed that there is a very strong positive correlation

between AJFI, GINI, and RSD. This almost perfect positive correlation between AJFI and RSD confirmed the reformulation of jain fairness index discussed in Section 4.2.2 of Chapter 4. The table also shows that for some problem instances,i.e. CAR-S-91, PUR-S-93, EXAM12 there is a strong negative correlation between cost and the three fairness quality indicators.

Since AJFI, GINI, and RSD have a strong positive correlation, these fairness indices can be used interchangeably. However, we should consider their computational times. As shown by Table 4.7, our empirical experimental data on computing these fairness indices over 1024 solutions for each 8 problem instances given in Table 4.6 showed that AJFI index is slightly more efficient than the other fairness indices. Thus, AJFI might be the best option to measure fairness. However, the computational times given in Table 4.7 is arguable because, of course, it depends on the implementation.

Table 4.7: Average computing time of different quality indicator of fairness index.

Fairness Index	Avg. Time (miliseconds)
AJFI	**61.86**
GINI	64.03
RSD	63.17
VAR	63.35

4.3.2.2 Investigating correlation between number of iterations of search in the algorithm, standard objective function, and fairness objective function

Firstly, before testing the proposed objective function we investigate the correlation between the number of iterations of search in SA-GD hyper-heuristic Algorithm, i.e. Algorithm 3 in Chapter 3, the standard objective function and unfairness objective function measured with AJFI in each iteration of the search. Note that we used AJFI instead of JFI to transform the JFI, which is to be maximised, to be a minimisation problem. To get this correlation, we run the algorithm with standard benchmark objective function over 13 problem instances (since the focus of study in this section is uncapacitated examination timetabling problem, the capacitated problem instances are excluded) within `1,500,000` iterations for each instance. Every `100,000` iterations, the values of standard objective function and unfairness objective function were recorded.

The standard objective function values as well as the unfairness objective function values during the iteration for each problem instance are visualised by Figure 4.2. Furthermore, in order to see the correlation clearer, Figures C.1 - C.2 in Appendix C present the normalised values (between 0-1) of standard objective function and Anti Jain fairness

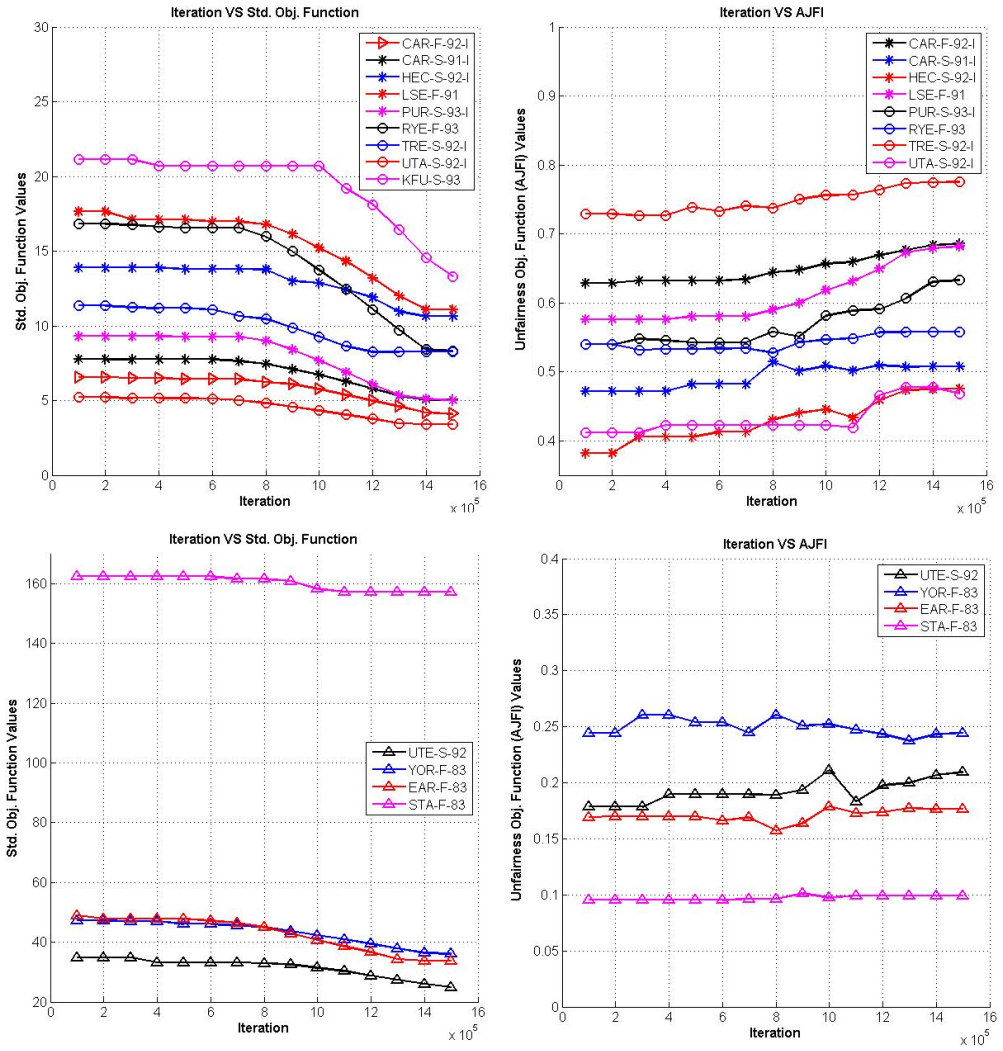


Figure 4.2: Correlation between the number of iteration, standard objective function (OF) values, and anti Jain fairness index (AJFI) values.

Index (AJFI). We also conducted correlation test between iteration, standard objective function value, AJFI value shown in Table 4.8.

As shown by Figure 4.2 it can be observed that the more number of iterations the standard objective function value is decreasing and contrarily the unfairness objective function is increasing. From Figures C.1 - C.2, and Table 4.8 it can be observed that there is strong negative correlation between number of iterations and the standard objective function values. It indicates that the more number of iteration, the better (less) the standard objective function values. Except for two problem instances, i.e. `EAR-F-83-I and YOR-F-83-I`, it also can be observed that there is strong positive correlation between the number of iteration and unfairness; and negative correlation between standard objective function values and unfairness. It indicates that the better standard objective

Table 4.8: Correlation between number of iteration, standard objective function values, and Anti Jain fairness Index (AJFI) values.

Instance	Iteration VS Std. OF	Iteration VS AJFI	Std. OF VS AJFI
CAR-F-92-I	-0.9195	0.9116	-0.9892
CAR-S-91-I	-0.9249	0.952	-0.9923
EAR-F-83-I	-0.9568	0.4781	-0.6189
HEC-S-92-I	-0.9047	0.8872	-0.7109
KFU-S-93	-0.8275	0.8314	-0.9076
LSE-F-91	-0.9241	0.975	-0.9306
PUR-S-93-I	-0.9134	0.926	-0.9981
RYE-F-92	-0.9186	0.9098	-0.9857
STA-F-83-I	-0.9149	0.8374	-0.8031
TRE-S-92	-0.9553	0.7831	-0.8989
UTA-S-92-I	-0.9496	0.9618	-0.9804
UTE-S-92	-0.9367	0.8156	-0.7907
YOR-F-83-I	-0.9546	-0.4614	0.5997

function, the worse its unfairness index. In other words, except in `EAR-F-83-I and YOR-F-83-I` problem instances, the standard objective function and fairness objective function are conflicting each other; if we minimise the standard objective function it tends to worsen its fairness.

4.3.2.3 Evaluating 'Sum of Powers' function with different exponent numbers

Secondly, we evaluated three different objective functions i.e. 'Sum of Powers', SOP (see Equation 4.19) with exponent number, `q=2` or 'sum of square', `q=16`, and `q=`∞ or MinMax (MM) in order to optimise both standard objective function and fairness concurrently. Initially, we tested the objective functions within Self-Adaptive-Late Acceptance (SA-LA) hyper-heuristic (see Section 3.2.1 of Chapter 3) over problem instance `HEC-S-92-I`. SA-LA hyper-heuristic is chosen because in this approach, no parameter setting is required and the performance is good enough. Whereas `HEC-S-92-I` is chosen because it is one of less complex problem problem instance. Being run within `3,000,000` iterations, the standard objective function values and unfairness objective function values, i.e. AJFI, during the iterations are visualised in Figure 4.3.

From Figure 4.3 we can observe that SOP objective function with `q=16` and `q=`∞ can make the solution much fairer, however it compensates to worsen the standard objective function very significantly. On the other hand, with `q=2` can make the solution fairer enough with slightly increase on standard objective function. In other words, we can say that the higher exponent number, the fairer solution and the worse final solution will be. It might suggest that SOP with `p=2` or 'Sum of Squares' is the best option

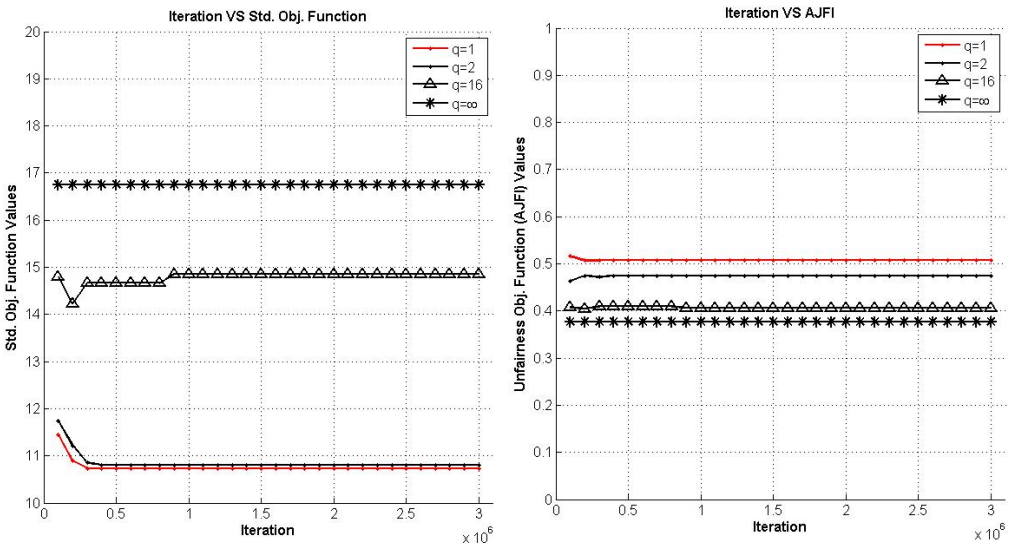


Figure 4.3: The standard objective function values and unfairness objective function values during the iterations, with 'Sum of Powers' (SOP) objective function using different exponent numbers (`q`)

to choose. However, the final decision is on the decision maker, e.g. the user of exam timetabling software.

4.3.2.4 Experimental Results Over 13 Problem Instances of Carter Dataset

Furthermore, to study more comprehensively on 'Sum of Powers', we tested SA-GD hyper-heuristic approach (see Section 3.2.1 of Chapter 3) over all 13 problem instances of Carter dataset. We run the algorithm with both standard objective function and 'Sum of Powers' using `q=2` or 'Sum of Square', within `30,000,000` iterations, and recorded the standard objective function and unfairness objective function during the iteration. The comparison of the objective function values during the iterations between standard objective function and SOP (`q=2`) over all 13 problem instances of Carter dataset is given in Figures C.3 - C.7 in Appendix C.

From the figures we can conclude that generally, compared to standard objective function, by using 'Sum of Powers' the algorithm ends up with less efficient i.e. worse in term of standard objective function, but fairer solution. In other words, the standard objective function is increasing while the unfairness objective function is decreasing. However, there is an exception that for some instances, i.e. `EAR-F-83, LSE-F-91, RYE-F-92, STA-F-83, UTE-S-92`, the algorithm ends up with very competitive result, even better efficiency and fairer solution. As studied in [179] pushing fairness objective function might cause escape from local minima. Therefore, the standard objective function is

even better for some problem instances.

On average, as shown by Table 4.9, the increase of `3.29%` in standard objective function causes the decrease of `6.20%` unfairness. Table 4.9 also shows that the trade-off between efficiency and unfairness in each problem instance is different. It might due to the different structure of the problem instances.

Table 4.9: The trade-off between efficiency, i.e. standard objective function (std.OF) and unfairness (AJFI) of the solutions produced with standard objective function vs 'Sum of Square'.

Instances	std. OF		AJFI		Delta (%)	
	std	sop	std	sop	std	sop
CAR-F-92-I	4.04	4.33	0.73	0.70	7.23	-4.19
CAR-S-91-I	4.92	5.08	0.69	0.66	3.37	-4.03
EAR-F-83-I	33.68	33.83	0.18	0.16	0.44	-12.84
HEC-S-92-I	10.56	11.04	0.50	0.47	4.57	-6.06
KFU-S-93	13.22	13.86	0.47	0.44	4.91	-7.42
LSE-F-91	10.79	10.67	0.49	0.45	**-1.10**	**-8.87**
PUR-S-93-I	4.77	5.36	0.69	0.63	12.50	-8.91
RYE-F-92	8.47	8.65	0.62	0.60	2.14	-3.22
STA-F-83-I	157.17	157.18	0.10	0.10	0.01	-0.11
TRE-S-92	8.18	8.53	0.56	0.54	4.30	-4.94
UTA-S-92-I	3.36	3.52	0.78	0.75	4.83	-2.85
UTE-S-92	25.39	25.19	0.21	0.20	**-0.81**	**-7.29**
YOR-F-83-I	36.83	36.98	0.27	0.24	0.41	-9.86
Average					3.29	-6.20

Furthermore, to get more accurate data, instead of single run, for each instance in Carter datasets we conduct `11` times testing runs, within `360` second per run. The experimental results are given in Table 4.10. Delta in the table is the different of objective function values between 'SOF' and 'SOP'. Table 4.10 presents the objective functions, i.e. standard objective function and global unfairness measured in AJFI of the best solution in term of standard objective function of 11 runs, and also the average objective function values of 11 runs. From Table 4.10, we can observe that the trade-off between standard objective function value and unfairness objective function is generally consistent with the single run experiment.

Since it is obvious that 'Sum of Powers' requires more computation time, in order to make fairer comparison, we repeated the experiment, by running the algorithm within the same number of iterations, i.e. `3,000,000`, instead of the same running time as the previous experiment. In addition, we also conducted another experiment using 'Sum of Powers' with `q=16`. The results are given in Table 4.11 for the objective function values of the best solution (in terms of standard objective function) and Table 4.12 for the objective function values average. Note that in 4.11, SOF is benchmark standard objective function, AJFI is anti Jain fairness index [0-1], d.SOF is delta 'SOF' and

Table 4.10: The trade-off between efficiency, i.e. standard objective function (SOF) and unfairness (AJFI) of the solutions produced with SA-GD hyper-heuristics using different objective functions: standard objective function vs 'Sum of Square'.

Instance	The Best Solution*						The Average Objective Function Values					
	SOF		SOP		delta (%)		SOF		SOP		delta (%)	
	SOF	AJFI	SOF	AJFI	SOF	AJFI	SOF	AJFI	SOF	AJFI	SOF	AJFI
CAR-F-92-I	4.72	0.71	4.84	0.69	2.48	-2.67	4.87	0.70	5.04	0.69	3.57	-2.40
CAR-S-91-I	5.77	0.67	5.90	0.65	2.27	-3.98	6.01	0.66	6.21	0.64	3.42	-3.24
EAR-F-83-I	34.89	0.18	36.06	0.17	3.36	-6.61	37.20	0.17	37.45	0.16	0.68	-8.44
HEC-S-92-I	10.84	0.53	11.00	0.50	1.44	-5.64	11.25	0.51	11.53	0.47	2.42	-8.19
KFU-S-93	14.02	0.49	14.25	0.45	1.59	-6.85	14.80	0.46	14.92	0.43	0.80	-5.63
LSE-F-91	11.49	0.50	11.95	0.44	4.01	-13.29	11.81	0.48	12.33	0.43	4.33	-10.90
PUR-S-93-I	7.82	0.61	7.82	0.60	-0.04	-1.60	8.00	0.60	8.14	0.58	1.77	-3.14
RYE-F-92	9.04	0.63	9.09	0.61	0.48	-4.11	9.31	0.62	9.59	0.60	3.03	-3.98
STA-F-83-I	157.34	0.10	157.75	0.10	0.26	-2.18	157.34	0.10	157.75	0.10	0.26	-2.17
TRE-S-92	8.82	0.55	8.86	0.54	0.47	-2.55	8.96	0.55	9.19	0.53	2.56	-3.25
UTA-S-92-I	3.83	0.76	3.99	0.74	3.97	-2.29	3.97	0.76	4.16	0.74	4.71	-2.65
UTE-S-92	25.99	0.21	25.71	0.20	-1.10	-0.68	26.29	0.21	26.01	0.20	-1.06	-4.39
YOR-F-83-I	38.28	0.24	38.88	0.24	1.57	-0.17	39.22	0.25	39.67	0.24	1.15	-6.28
				Average	**1.60**	**-4.05**				Average	**2.13**	**-4.97**

* in terms of standard objective function value

d.AJFI is delta 'AJFI'.

Table 4.11: The trade-off between efficiency and unfairness of the solutions produced with SA-GD hyper-heuristics using three different objective functions.

Instance	SOF		SOP (q=2)				SOP (q=16)			
	SOF	AJFI	SOF	d.SOF(%)	AJFI	d.AJFI(%)	SOF	d.SOF(%)	AJFI	d.AJFI(%)
CAR-F-92-I	3.93	0.73	4.19	6.67	0.70	-2.96	5.45	38.67	0.67	-8.27
CAR-S-91-I	4.78	0.70	5.08	6.26	0.66	-5.21	6.64	38.99	0.63	-10.05
EAR-F-83-I	33.03	0.18	33.39	1.09	0.16	**-9.50**	37.42	13.28	0.14	**-20.50**
HEC-S-92-I	10.39	0.51	10.63	2.33	0.49	**-3.73**	12.49	20.26	0.44	-13.49
KFU-S-93	13.24	0.47	13.50	1.92	0.45	**-4.99**	15.95	20.42	0.43	-9.93
LSE-F-91	10.07	0.50	10.69	6.15	0.45	**-11.24**	13.92	38.24	0.37	-26.61
RYE-F-92	8.27	0.64	8.55	3.38	0.60	**-6.07**	11.23	35.73	0.55	-13.80
STA-F-83-I	157.08	0.10	157.18	0.07	0.10	**-0.65**	157.39	0.20	0.10	**-1.48**
TRE-S-92	8.08	0.55	8.05	-0.36	0.53	**-3.29**	9.57	18.51	0.50	-8.36
UTA-S-92-I	3.28	0.78	3.43	4.48	0.76	-2.93	4.33	31.93	0.72	-7.43
UTE-S-92	24.81	0.21	25.11	1.21	0.20	**-4.66**	26.72	7.70	0.18	-13.81
YOR-F-83-I	35.35	0.25	36.78	4.04	0.24	-3.81	40.05	13.30	0.22	-11.48
	Avg. Delta			3.10		-4.92		23.10		-12.10

From Table 4.11 and Table 4.12, we can observe that over all problem instances 'Sum of Powers' objective function could produce fairer solutions than solutions produced with standard objective function. However, except over problem instance `TRE-S-92`, it compensates with worse standard objective functions. The trade-off between standard objective function and fairness are varied amongst problem instances. For example, for the best solution of problem instance `EAR-F-83-I`, the solution could be made 9.50% fairer by compensating the standard objective function only 1.08% worse. Contrarily, for the best solution of problem instance `CAR-F-92-I`, to get the solution 2.96% fairer, it compensates with 6.67% worse standard objective function.

From Table 4.11 and Table 4.12, we can also observe that within 'Sum Of Powers' objective function approach, we can make the solution much fairer with higher exponent,

Table 4.12: The average objective function values: the trade-off between efficiency and unfairness of the solutions produced with SA-GD Hyper-heuristics using three different objective functions.

Instance	SOF		SOP (q=2)				SOP (q=16)			
	SOF	AJFI	SOF	d.SOF(%)	AJFI	d.AJFI(%)	SOF	d.SOF(%)	AJFI	d.AJFI(%)
CAR-F-92-I	4.05	0.73	4.22	4.35	0.70	-3.58	5.65	39.54	0.66	-8.81
CAR-S-91-I	4.87	0.69	5.13	5.38	0.66	-4.84	6.89	41.44	0.62	-10.35
EAR-F-83-I	33.56	0.18	33.84	0.81	0.17	**-7.07**	38.97	16.09	0.14	**-24.50**
HEC-S-92-I	10.64	0.51	10.87	2.15	0.48	**-6.50**	13.41	26.07	0.41	-20.90
KFU-S-93	13.50	0.47	13.75	1.81	0.44	**-5.99**	17.15	27.01	0.37	-21.41
LSE-F-91	10.44	0.50	10.97	5.06	0.44	**-12.28**	14.58	39.61	0.35	-30.72
RYE-F-92	8.37	0.63	8.71	4.02	0.60	**-4.47**	11.62	38.83	0.55	-12.60
STA-F-83-I	157.08	0.10	157.18	0.07	0.10	**-0.65**	157.39	0.20	0.10	**-1.48**
TRE-S-92	8.21	0.56	8.34	1.67	0.53	**-4.81**	9.83	19.77	0.51	-9.36
UTA-S-92-I	3.35	0.78	3.51	4.77	0.75	-3.64	4.49	34.10	0.72	-7.68
UTE-S-92	25.01	0.21	25.30	1.16	0.20	**-5.60**	27.24	8.88	0.18	**-14.91**
YOR-F-83-I	36.25	0.25	37.06	2.22	0.24	**-6.20**	40.94	12.93	0.21	**-15.10**
	Avg.			2.79		**-5.47**		25.37		**-14.82**

i.e. `q=16`. However, it also compensates much higher worse in standard objective function value. It suggest that, within 'Sum of Powers' objective function approach, we can control the level of fairness by the parameter of exponent.

Since we should compute each individual student penalty, the drawback of applying 'Sum of Powers' objective function approach is higher computational time. The computational time comparison between the standard linear summation, 'Sum of Powers' with `q=2` and `q=16` over problem instance `STA-F-83-I` (1 run with `3,000,000` iterations) is given in Table 4.13. It shows that 'Sum of Powers' approach requires approximately 28 times higher computational times.

Let n is the total number of students, p_s is penalty associated to individual student s, and q is the exponent number. Asymptotically, the comparison between standard linear summation (LS) and 'sum of power' (SOP) in terms of the number of primitive operations, i.e. addition and multiplication, is shown by Equation 4.21 and Equation 4.22. From the equations, it is clear that the number of primitive operations for LS is n, whereas for SOP is $q.n$. In other words, the complexity of LS and SOP is $\theta(n)$ and $\theta(q.n)$. It makes sense that compared to LS, SOP is more expensive operation.

$$p_1 + p_2 + p_3 + ... + p_n \tag{4.21}$$

$$\underbrace{p_1 \text{ x } p_1 \text{ x } ... \text{ x } p_1}_{q} + \underbrace{p_2 \text{ x } p_2 \text{ x } ... \text{ x } p_2}_{q} + ... + \underbrace{p_n \text{ x } p_n \text{ x } ... \text{ x } p_n}_{q} \tag{4.22}$$

Another drawback of this approach is that it results in only a single solution. Therefore, the decision maker could not make trade-off amongst some alternatives of solutions to be considered in order to decide the best trade-off between standard objective function and

Table 4.13: Comparison of computation time between standard objective function and 'Sum of Powers' with `q=2` and `q=16`.

Ob. Function	Computational Time	Ratio
sof	69321	1.00
sop (q=2)	1960197	28.28
sop (q=16)	1929239	27.83

fairness objective function. Multiple runs with different powers could be an alternative to produce multiple solutions. However it is impractical, because as shown in Table 4.13, 'sum of powers' required significant increase in computational times.

4.4 Conclusion

An extended examination timetabling problem model, which is intended to be more matching to student preferences investigated in Chapter B, was proposed in this chapter. The new examination timetabling problem model is basically an extension of examination timetabling problem formulation used in international timetabling competition 2007 (ITC 2007) [207]. Firstly, in order to consider the different difficulty level of the exam, rather than treating the exams equally, a difficulty index associated to each exam was introduced in this study. This difficulty index represents relative difficulty level of the respective exam compared to the other exams. Secondly, an extension of the examination timetabling problem model to cope with fairness. The new model is inherently multi-objectives with fairness as the new objective function in addition the original standard objective function. Several notions of fairness were proposed. In addition to fairness between all students, fairness amongst students within a 'specified' cohort was proposed. The specified cohort could be interpreted as students within the same course or school and the same year of entrance.

Moreover, since the students though within the same cohort more than likely have different number of exams, more sensible fairness formulation, i.e. weighted fairness and 'exam-count-based cohort' were also proposed. Lastly, inspired by the work of [69], this chapter discussed a multi-objective examination timetabling formulation with multiple examination timetabling problem stakeholders' perspective that split the original aggregated objective function into several separated objective functions based on these stakeholders. In addition to new examination timetabling problem formulation, the works in this chapter also contribute for initial work towards building a public repository that extends current benchmark instances with the information needed to build cohorts. Thus, allowing methods on our formulation to be studied by the communities.

In this chapter, a study in attempt to enforce fairness in examination timetabling problems with so-called 'Sum of Powers' objective function approach over Carter dataset also has been carried out. The experimental results of 'Sum of Powers' approach suggest that, this approach could produce fairer solutions for all problem instances. However, it also shows that fairer solution compensates on worsening the standard objective function that varied amongst different problem instances. The study also suggests that the level of fairness could be controlled by the exponent within 'Sum of Powers'. The higher exponent, the fairer the solution. But, of course, it also compensates much worse in term of standard objective function. In addition to higher computation time, the drawback of this approach is that this result only results a single solution that makes a decision maker has no alternative solutions to decide the best trade-off between standard objective function and fairness.

Chapter 5

Multi-Stage Approach to Enforcing Fairness in Examination Timetabling Problems

5.1 Introduction

In Section 4.3 of Chapter 4, the 'Sum of Powers' approach has been studied to enforce fairness in the examination timetabling problem. Within this approach, both the standard objective function and the fairness objective functions are optimised altogether as a single objective function. The deficiency of this approach is the significant increase in computation time that might make this approach impractical. To make an efficient computation time, rather than optimising both objective function concurrently, this chapter studies a two-stage approach that optimises the two objective functions one after the other as will be discussed in Section 5.2. In this approach, the standard objective function (see Equations 2.7-2.8) is optimised first in the first stage, and the fairness objective function is optimised in second stage. Moreover, in this chapter we tested our proposed approach over four different problem instances: Carter, ITC 2007, Yeditepe, and Nottingham dataset, whereas in the Section 4.3 of Chapter 4, the proposed approach was tested over the Carter dataset only.

In this Chapter, some experimentations will be discussed to evaluate the different notions

of fairness. First, in Section 5.3.1, the initial solutions of the Stage 2 of the two-stage algorithm, which are results from optimisation in terms of standard objective function in the Stage 1 of the algorithm, are evaluated by their different objective functions. These objective functions include different notions of fairness, i.e. fairness between all students or global fairness, specified cohort fairness, and exam-count-based fairness; as well as different fairness metrics, i.e. Jain fairness index, weighted Jain fairness index, and GINI index. Further Section 5.3.2.1, Section 5.3.2.2, and Section 5.3.2.3 will present experimentation on enforcing fairness between all students or global fairness, specified cohort fairness, and exam-count-based fairness respectively as a part of Stage 2 of the Two-Stage Algorithm. In each experimentation with different notions of fairness, three different move acceptance strategies of Hyper-heuristics, i.e. lexicographic, hill climbing, and great deluge, are evaluated.

5.2 Method: Two-Stage Approach

The algorithm devised in this approach is basically a hyper-heuristic approach similar to the algorithm discussed in Chapter 3. The different is that in this two-stage approach the algorithm is divided into two stages, with different objective function in each stage.

Stage 1 aims at optimising the standard objective function. Therefore, the algorithm is exactly the same as the algorithm used in Chapter 3, i.e. reinforcement learning - extended great deluge algorithm (RL-XGD) hyper-heuristic. Subsequently, stage 2 aims at optimising fairness objective function, by changing the objective function from the standard objective function to fairness objective function, i.e. Jain Fairness Index (JFI) or Anti Jain Fairness Index (AJFI) to make a minimisation problem instead. As discussed in Chapter 4, in Stage 2, a new hard constraint is added, i.e. the standard objective function is not allowed to be worse at all or allowed to be worse but no more than a given percentage from the initial solution.

In Stage 2, three different move acceptance mechanism strategy, namely lexicographic [212], hill climbing (HC), and great deluge (GD) are evaluated within reinforcement-learning hyper-heuristic. Within lexicographic move acceptance, during the iteration, the new solution is accepted to replace the current solution if it is better than the current solution in term of standard objective function. The new solution is also accepted, if it is not worse than the current solution in term of standard objective function but better than the current solution in term of fairness objective function. Whereas hill climbing and great deluge move acceptance strategies have been discussed in Chapter 3.

In addition to optimise global fairness as studied in Section 4.3 of Chapter 4, in this chapter, we also study to optimise 'fairness within cohort' (see Section 4.2.2.2 of Chapter 4). Both notions of cohort, i.e. specified cohort, which is based on school/faculty and year of student, and 'exam-count-based cohort' are studied.

5.3 Experimental Results

Since the algorithm used in stage 1 of this two-stage approach is basically the same as the algorithm used in Chapter 3, in this experiment we focus on Stage 2. We used the best result from Stage 1, which is taken from the experiment in Chapter 3, as initial solutions for Stage 2.

5.3.1 Evaluating The Objective Functions Values of The Initial Solutions

Since the aim of Stage 2 is to optimise fairness objective function, it is worth to know the objective functions values, i.e. standard objective function, global fairness, and cohort fairness, of the initial solutions. It should be kept in mind that these initial solutions are result from standard objective function optimisation, in which the fairness objective function is ignored. The objective functions' values of the initial solutions are given in Table 5.1. This initial solutions are obtained from reinforcement learning - extended great deluge algorithm discussed in Section 3.3.2.2 of Chapter 3.

From Table 5.1 we can observe that generally, in term of global fairness, the initial solutions are very unfair, except for some instances from Carter dataset, i.e. `EAR-F-83, STA-F-83, UTE-S-92, and YOR-F-83` that are quite fair already. However, in term of cohort fairness, especially exam-count-based cohort fairness, the initial solutions are fairer than when they are measured in terms of global fairness. Though, the unfairness index is still quite high and should be made fairer.

The thing that should be pondered in this fairness measurement is the fact that each student enrolled different number of exams. Logically, with respect to objective function in Equation 4.17, the student with higher number of exams is more likely to get higher penalty and its vice versa. In this Carter objective function formulation, a student with only one exam will always get zero penalty. Whereas, to get optimal fairness in term of the standard Jain Fairness Index(JFI), each student has to get equal number of penalties. In this sense, fairness measure that consider the number of exams makes more sense. such as by dividing the penalty of individual students with the number of the enrolled exams.

Table 5.1: The Objective Functions' Values, i.e. Standard Objective Function (STD.OF), Global Unfairness, Specified Cohort Unfairness, and Exam-Count-Based Cohort Unfairness, of The Initial Solutions.

Dataset	Problem Instance	std.of	Global Unfairness			Cohort Unfairness			
						Specified		Exam-count-based	
			AJFI	W.AJFI*	GINI	AJFI	W.AJFI	AJFI	W.AJFI
NOTT	NOTT-I	11923	0.7944	0.0001	0.6995	0.8105	0.0026	0.5382	0.0019
	NOTT-II-32	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	NOTT-II-31	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	NOTT-II-30	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	NOTT-II-26	8	0.9993	0.0001	0.9994	0.2215	0.0004	0.2198	0.0011
	NOTT-II-23	68	0.9925	0.0001	0.9924	0.7697	0.0020	0.6466	0.0045
CARTER	CAR-S-91	5.30	0.6769	0.0000	0.6895	0.6648	0.0000	0.3729	0.0097
	CAR-F-92	4.51	0.7070	0.0000	0.7166	0.6850	0.0000	0.4289	0.0015
	EAR-F-83	36.73	0.1739	0.0000	0.2543	0.1586	0.0002	0.0996	0.0041
	HEC-S-92	10.91	0.5090	0.0002	0.5262	0.2431	0.0001	0.2928	0.0008
	KFU-S-93	14.36	0.4546	0.0001	0.4838	0.3166	0.0004	0.3334	0.0012
	LSE-F-91	11.02	0.4887	0.0026	0.4996	0.4768	0.0035	0.2612	0.0048
	PUR-S-93	5.03	0.6709	0.0000	0.6787	0.6974	0.0001	0.4173	0.0010
	RYE-S-93	9.01	0.6266	0.0001	0.6488	0.4091	0.0003	0.2977	0.0002
	STA-F-83	157.12	0.0999	0.0001	0.1806	0.0043	0.0000	0.0021	0.0000
	TRE-S-92	8.75	0.5640	0.0001	0.5902	0.5551	0.0001	0.3726	0.0006
	UTA-S-92	3.60	0.7708	0.0000	0.7761	0.7403	0.0000	0.4441	0.0011
	UTE-S-92	25.20	0.2098	0.0000	0.2780	0.1427	0.0000	0.2070	0.0125
	YOR-F-83	38.03	0.2498	0.0001	0.3178	0.2384	0.0006	0.2305	0.0088
ITC 2007	EXAM1	6856	0.8173	0.0002	0.7876	0.7803	0.0014	0.4745	0.0049
	EXAM2	632	0.9996	0.0000	0.9996	0.6664	0.0000	0.4895	0.0096
	EXAM3	11659	0.9286	0.0000	0.8762	0.8968	0.0001	0.5773	0.0013
	EXAM4	16325	0.7343	0.0002	0.7467	0.7820	0.0004	0.5027	0.0054
	EXAM5	3837	0.8592	0.0000	0.8642	0.4551	0.0012	0.6099	0.0015
	EXAM6	27370	0.7878	0.0000	0.7206	0.6054	0.0001	0.2866	0.0004
	EXAM7	5528	0.8363	0.0001	0.8355	0.6208	0.0002	0.3602	0.0012
	EXAM8	9798	0.6486	0.0000	0.6433	0.4377	0.0003	0.2936	0.0008
	EXAM9	1246	0.7459	0.0004	0.6974	0.6824	0.0028	0.2412	0.0040
	EXAM10	14556	0.6436	0.0002	0.6133	0.5040	0.0003	0.0821	0.0206
	EXAM11	36810	0.9065	0.0000	0.8120	0.8633	0.0001	0.5489	0.0014
	EXAM12	5300	0.8843	0.0001	0.8290	0.7543	0.0111	0.5655	0.0078
YEDITEPE	YUE20011	56	0.9175	0.0017	0.9190	0.8982	0.0073	0.4531	0.0146
	YUE20012	122	0.8585	0.0021	0.8627	0.8503	0.0091	0.5663	0.0244
	YUE20013	29	0.8841	0.0047	0.8843	0.8961	0.0234	0.3652	0.0119
	YUE20021	76	0.9223	0.0020	0.9215	0.7300	0.0491	0.5550	0.0489
	YUE20022	160	0.8730	0.0022	0.8753	0.7331	0.0093	0.5449	0.0226
	YUE20023	56	0.8902	0.0007	0.8922	0.5915	0.0807	0.4665	0.0902
	YUE20031	143	0.9039	0.0000	0.9053	0.9155	0.0256	0.6005	0.0077
	YUE20032	434	0.7683	0.0009	0.7791	0.6377	0.0039	0.5521	0.0449

*W.A JFI: Weighted Anti Jain Fairness Index, see Section 4.2.2.1 of Chapter 4

However, our experimental results show that dividing the penalty of individual students gave no useful difference. The correlation between JFI with and without dividing student penalties with its respective number of the exams is `0.999`.

As discussed in Chapter 4, another option to account for different numbers of exams enrolled by the students is by using weighted Jain fairness index (see Equation 4.11 in Section 4.2.2.1 of Chapter 4) instead of the original Jain fairness index. However, if we used weighted JFI as shown by Table 5.1, the values of weighted Jain fairness index over both global fairness and cohort fairness, are always almost equal to `1`, or weighted AJFI almost equal to `0`. It indicates that measured with weighted Jain fairness index, the solutions are perfectly fair already. However, it does not make sense, since the values are always the same.

Moreover, another option to consider differing number of exams enrolled by the students in fairness quality indicator, 'exam-count-based cohort' fairness might be more appropriate. In, 'exam-count-based cohort', students with the same number of enrolled exams are considered in the same cohort. In other words, the fairness is measured amongst student with the same total number of exams. The number of 'exam-count-based cohort' for each problem instance was given in Table 4.3 in Section 4.2.2.2 of Chapter 4. Using, this fairness notions, as shown by Table 5.1, unfairness amongst students in the same 'exam-count-based cohort' is still high. It indicates that by optimising on standard objective function only, the fairness amongst students even with the same number of exam, is still bad. Therefore, fairness should be considered in addition to the standard objective function.

5.3.2 Enforcing Fairness

We divided the experiment into three folds. First, experiment to optimise global fairness. Second, experiment to optimise specified cohort fairness. Third, experiment to optimised exam-count-based cohort fairness. Further, each experiment consists two scenarios: First, strictly prohibit to make the standard objective function worse as a hard constraint, in other words deficiency =0%. Second, prohibit to make standard objective function worse than 5% as a hard constraint, in other words deficiency <=5%. The value of 5% is based on our assumption that 5% is still tolerable decrease on the quality of standard objective function.

5.3.2.1 Enforcing Global Fairness

Having global fairness as an objective function, in this experiment, we evaluated three different move acceptance strategies within Reinforcement-Learning Hyper-heuristic, i.e. great deluge, hill climbing, and lexicographic. Henceforth they will be referred to as RL-GD-HH, RL-HC-HH, and RL-LEX-HH respectively. For each problem instance, the algorithm was run once with `10,000` iterations.

Experimental Results With RL-GD-HH

In scenario 1, which is the deficiency=0%, the experimental results showed that both standard objective functions and global fairness could not be improved any further. Within this scenario, it is also shown that the cohort fairness, including specified and exam-count-based cohort, could not be improved too. In scenario 2, which is the deficiency<=5%, the experimental results are given in Table 5.2. Note that STD.OF, d.STD.OF, and d.AJFI stand for standard objective function, deficiency STD.OF and deficiency AJFI respectively.

Table 5.2 shows that even though the standard objective function is allowed to be worse up to 5%, the global fairness could not be improved significantly. Only problem instance `STA-F-83` shows significant improvement in global fairness, though actually, its initial solution (see Table 5.1) is fair already. The table also shows that by optimising global fairness is not by product optimising cohort fairness by product. For example, in problem instance `STA-F-83`, the global fairness could be improved by 18% but the specified cohort fairness is getting worse by 32%.

Experimental Results With RL-HC-HH

The experimental results with RL-HC-HH for scenario 1, and scenario 2, are given Table 5.3 and Table 5.4 respectively. From Table 5.3, we can observe that, over few problem instances, the global fairness could be improved very slightly. Moreover, as shown by Table 5.4, having deficiency of standard objective function not more than 5%, the global fairness could be improved over all problem instances. Especially, over Carter dataset, six of the problem instances, their global fairness could be improved quite significantly, i.e. between 5-31%. Compared to the results from RL-GD-HH shown in Table 5.2, these results show that, RL-HC-HH performs better than RL-GD-HH.

Table 5.2: Experimental Results From Enforcing Global Fairness With RL-GD-HH and Deficiency <=5%.

Dataset	Problem Instance	STD.OF		Global Unfairness		Cohort Unfairness			
						Specified		Exam-count-based	
		STD.OF	d.STD.OF (%)	AJFI	d.AJFI (%)	AJFI	d.AJFI (%)	AJFI	d.AJFI (%)
NOTT	NOTT-I	12505	4.88	0.7914	**-0.38**	0.8094	-0.14	0.5436	-1.01
	NOTT-II-32	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-31	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-30	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-26	8	0.00	0.9993	0.00	0.2215	0.00	0.2198	0.00
	NOTT-II-23	68	0.00	0.9925	0.00	0.7697	0.00	0.6466	0.00
CARTER	CAR-S-91	5.55	4.77	0.6722	**-0.69**	0.6597	-0.77	0.3687	1.11
	CAR-F-92	4.72	4.58	0.7012	**-0.82**	0.6791	-0.86	0.4224	1.51
	EAR-F-83	37.54	2.23	0.1739	**-0.02**	0.1596	0.62	0.1066	-7.03
	HEC-S-92	10.91	0.00	0.5090	0.00	0.2431	0.00	0.2928	0.00
	KFU-S-93	14.36	0.00	0.4546	0.00	0.3166	0.00	0.3334	0.00
	LSE-F-91	11.39	3.39	0.4841	**-0.96**	0.4747	-0.43	0.2586	0.97
	PUR-S-93	5.28	4.94	0.6662	**-0.71**	0.6931	-0.62	0.4143	0.71
	RYE-S-93	9.01	0.00	0.6266	0.00	0.4091	0.00	0.2977	0.00
	STA-F-83	163.11	3.81	0.0820	**-17.97**	0.0056	32.02	0.0033	-56.80
	TRE-S-92	8.75	0.00	0.5640	0.00	0.5551	0.00	0.3726	0.00
	UTA-S-92	3.65	1.29	0.7672	**-0.46**	0.7353	-0.67	0.4421	0.43
	UTE-S-92	25.32	0.47	0.2072	**-1.24**	0.1414	-0.91	0.2070	0.01
	YOR-F-83	39.64	4.24	0.2448	**-2.00**	0.2321	-2.65	0.2266	1.70
ITC 2007	EXAM1	7179	4.71	0.8111	**-0.76**	0.7495	-3.95	0.4699	0.97
	EXAM2	632	0.00	0.9996	0.00	0.6664	0.00	0.4895	0.00
	EXAM3	11729	0.60	0.9282	**-0.04**	0.8962	-0.06	0.5773	0.01
	EXAM4	16325	0.00	0.7343	0.00	0.7820	0.00	0.5027	0.00
	EXAM5	4021	4.80	0.8527	**-0.75**	0.4529	-0.48	0.6043	0.91
	EXAM6	27370	0.00	0.7878	0.00	0.6054	0.00	0.2866	0.00
	EXAM7	5789	4.72	0.8331	**-0.38**	0.6224	0.26	0.3592	0.28
	EXAM8	10187	3.97	0.6379	**-1.64**	0.4306	-1.62	0.2904	1.10
	EXAM9	1253	0.56	0.7435	**-0.33**	0.6809	-0.22	0.2395	0.70
	EXAM10	15008	3.11	0.6324	**-1.74**	0.4999	-0.82	0.0776	5.46
	EXAM11	38239	3.88	0.9006	**-0.65**	0.8594	-0.45	0.5458	0.56
	EXAM12	5558	4.87	0.8805	**-0.44**	0.7529	-0.18	0.5621	0.61
YEDITEPE	YUE20011.6.450	56	0.00	0.9175	0.00	0.8982	0.00	0.4531	0.00
	YUE20012.6.450	126	3.28	0.8524	**-0.71**	0.8464	-0.46	0.5633	0.54
	YUE20013.1.150	29	0.00	0.8841	0.00	0.8961	0.00	0.3652	0.00
	YUE20021.7.550	76	0.00	0.9223	0.00	0.7300	0.00	0.5550	0.00
	YUE20022.7.550	161	0.63	0.8720	**-0.12**	0.7321	-0.13	0.5444	0.09
	YUE20023.1.150	56	0.00	0.8902	0.00	0.5915	0.00	0.4665	0.00
	YUE20031.6.550	143	0.00	0.9039	0.00	0.9155	0.00	0.6005	0.00
	YUE20032.6.550	436	0.46	0.7675	-0.10	0.6372	-0.08	0.5516	0.09

Table 5.3: Experimental Results From Enforcing Global Fairness With RL-HC-HH and Deficiency <=0%.

Dataset	Problem Instance	STD.OF		Global Unfairness		Cohort Unfairness			
						Specified		Exam-count-based	
		STD.OF	d.STD.OF (%)	AJFI	d.AJFI (%)	AJFI	d.AJFI (%)	AJFI	d.AJFI (%)
NOTT	NOTT-I	11923	0.00	0.7944	0.00	0.8105	0.00	0.5382	0.00
	NOTT-II-32	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-31	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-30	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-26	8	0.00	0.9993	0.00	0.2215	0.00	0.2198	0.00
	NOTT-II-23	68	0.00	0.9925	0.00	0.7697	0.00	0.6466	0.00
CARTER	CAR-S-91	5.30	0.00	0.6769	**-0.01**	0.6648	-0.01	0.3729	0.00
	CAR-F-92	4.51	0.00	0.7069	**-0.01**	0.6849	-0.01	0.4288	-0.02
	EAR-F-83	36.73	0.00	0.1739	0.00	0.1586	0.00	0.0996	0.00
	HEC-S-92	10.91	0.00	0.5090	0.00	0.2431	0.00	0.2928	0.00
	KFU-S-93	14.36	0.00	0.4546	0.00	0.3166	0.00	0.3334	0.00
	LSE-F-91	11.02	0.00	0.4884	**-0.07**	0.4765	-0.06	0.2612	0.01
	PUR-S-93	5.03	-0.04	0.6705	**-0.07**	0.6972	-0.03	0.4169	-0.09
	RYE-S-93	9.01	0.00	0.6266	0.00	0.4091	0.00	0.2977	0.00
	STA-F-83	157.12	0.00	0.0999	0.00	0.0043	0.00	0.0021	0.00
	TRE-S-92	8.75	0.00	0.5640	0.00	0.5551	0.00	0.3726	0.00
	UTA-S-92	3.60	0.00	0.7708	0.00	0.7403	-0.01	0.4441	0.00
	UTE-S-92	25.20	0.00	0.2098	0.00	0.1426	0.00	0.2065	-0.22
	YOR-F-83	38.03	0.00	0.2498	0.00	0.2384	0.00	0.2305	0.00
ITC 2007	EXAM1	6856	0.00	0.8170	**-0.03**	0.7802	-0.02	0.4745	-0.02
	EXAM2	632	0.00	0.9996	0.00	0.6664	0.00	0.4895	0.00
	EXAM3	11659	0.00	0.9285	**-0.01**	0.8967	-0.01	0.5773	0.00
	EXAM4	16325	0.00	0.7343	0.00	0.7820	0.00	0.5027	0.00
	EXAM5	3837	0.00	0.8592	0.00	0.4551	0.00	0.6099	0.00
	EXAM6	27370	0.00	0.7878	0.00	0.6054	0.00	0.2866	0.00
	EXAM7	5528	0.00	0.8356	**-0.07**	0.6196	-0.18	0.3594	-0.21
	EXAM8	9798	0.00	0.6486	0.00	0.4377	0.00	0.2936	0.00
	EXAM9	1246	0.00	0.7459	0.00	0.6824	0.00	0.2412	0.00
	EXAM10	14556	0.00	0.6436	0.00	0.5040	0.00	0.0821	0.00
	EXAM11	36810	0.00	0.9063	**-0.01**	0.8633	0.00	0.5488	-0.02
	EXAM12	5300	0.00	0.8843	0.00	0.7543	0.00	0.5655	0.00
YEDITEPE	YUE20011.6.450	56	0.00	0.9175	0.00	0.8982	0.00	0.4531	0.00
	YUE20012.6.450	122	0.00	0.8585	0.00	0.8503	0.00	0.5663	0.00
	YUE20013.1.150	29	0.00	0.8841	0.00	0.8961	0.00	0.3652	0.00
	YUE20021.7.550	76	0.00	0.9205	**-0.19**	0.7286	-0.19	0.5447	-1.87
	YUE20022.7.550	160	0.00	0.8730	0.00	0.7331	0.00	0.5449	0.00
	YUE20023.1.150	56	0.00	0.8902	0.00	0.5915	0.00	0.4665	0.00
	YUE20031.6.550	143	0.00	0.9039	0.00	0.9155	0.00	0.6005	0.00
	YUE20032.6.550	434	0.00	0.7683	0.00	0.6377	0.00	0.5521	0.00

Table 5.4: Experimental Results From Enforcing Global Fairness With RL-HC-HH and Deficiency <=5%.

Dataset	Problem Instance	STD.OF		Global Unfairness		Cohort Unfairness			
						Specified		Exam-count-based	
		STD.OF	d.STD.OF (%)	AJFI	d.AJFI (%)	AJFI	d.AJFI (%)	AJFI	d.AJFI (%)
NOTT	NOTT-I	12514	4.96	0.7850	**-1.17**	0.7976	-1.59	0.5383	0.03
	NOTT-II-32	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-31	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-30	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-26	8	0.00	0.9993	0.00	0.2215	0.00	0.2198	0.00
	NOTT-II-23	71	4.41	0.9921	**-0.04**	0.7695	-0.03	0.6464	-0.03
CARTER	CAR-S-91	5.57	4.98	0.6634	**-2.00**	0.6494	-2.32	0.3555	-4.65
	CAR-F-92	4.73	4.99	0.6932	**-1.95**	0.6698	-2.22	0.4183	-2.48
	EAR-F-83	38.56	4.99	0.1477	**-15.06**	0.1381	-12.92	0.1040	4.42
	HEC-S-92	11.45	4.93	0.4898	**-3.76**	0.2346	-3.48	0.2775	-5.23
	KFU-S-93	15.08	5.00	0.4247	**-6.57**	0.2883	-8.93	0.2693	-19.23
	LSE-F-91	11.57	5.00	0.4659	**-4.68**	0.4597	-3.58	0.2556	-2.13
	PUR-S-93	5.28	5.00	0.6577	**-1.98**	0.6855	-1.71	0.4080	-2.22
	RYE-S-93	9.46	4.99	0.6058	**-3.32**	0.3991	-2.43	0.2826	-5.06
	STA-F-83	164.98	5.00	0.0688	**-31.11**	0.0036	-15.86	0.0024	14.74
	TRE-S-92	9.18	5.00	0.5346	**-5.22**	0.5200	-6.33	0.3478	-6.67
	UTA-S-92	3.78	5.00	0.7597	**-1.44**	0.7299	-1.41	0.4356	-1.91
	UTE-S-92	26.46	4.98	0.1731	**-17.50**	0.1234	-13.51	0.1658	-19.89
	YOR-F-83	39.93	5.00	0.2264	**-9.36**	0.2164	-9.25	0.2202	-4.47
ITC 2007	EXAM1	7198	4.99	0.8045	**-1.56**	0.7559	-3.13	0.4623	-2.57
	EXAM2	663	4.91	0.9995	**-0.02**	0.6662	-0.03	0.5819	18.89
	EXAM3	12241	4.99	0.9245	**-0.44**	0.8927	-0.46	0.5750	-0.40
	EXAM4	17141	5.00	0.7163	**-2.45**	0.7610	-2.68	0.4906	-2.42
	EXAM5	4028	4.98	0.8445	**-1.70**	0.4023	-11.61	0.5985	-1.87
	EXAM6	28730	4.97	0.7842	**-0.45**	0.6053	-0.02	0.2962	3.34
	EXAM7	5804	4.99	0.8275	**-1.05**	0.6099	-1.75	0.3545	-1.59
	EXAM8	10287	4.99	0.6360	**-1.95**	0.4413	0.81	0.2884	-1.75
	EXAM9	1308	4.98	0.7302	**-2.11**	0.6703	-1.77	0.2274	-5.74
	EXAM10	15283	4.99	0.6226	**-3.27**	0.4923	-2.33	0.0716	-12.85
	EXAM11	38650	5.00	0.9009	**-0.61**	0.8584	-0.57	0.5453	-0.65
	EXAM12	5565	5.00	0.8786	**-0.65**	0.7904	4.79	0.5603	-0.93
YEDITEPE	YUE20011.6.450	58	3.57	0.9140	**-0.38**	0.8917	-0.72	0.4517	-0.31
	YUE20012.6.450	128	4.92	0.8493	**-1.07**	0.8377	-1.48	0.5625	-0.69
	YUE20013.1.150	30	3.45	0.8798	**-0.48**	0.8888	-0.82	0.3637	-0.42
	YUE20021.7.550	79	3.95	0.9170	**-0.58**	0.9247	26.67	0.5438	-2.03
	YUE20022.7.550	168	5.00	0.8647	**-0.96**	0.7253	-1.06	0.5317	-2.41
	YUE20023.1.150	58	3.57	0.8856	**-0.52**	0.5874	-0.70	0.4599	-1.42
	YUE20031.6.550	150	4.90	0.9001	**-0.42**	0.9102	-0.57	0.5984	-0.36
	YUE20032.6.550	455	4.84	0.7543	**-1.82**	0.7863	23.31	0.5446	-1.35

Experimental Results With RL-LEX-HH

In order to be consistent with the notion of lexicographic optimality, we conducted experiment for RL-LEX-HH algorithm with scenario 1. The experimental results with this algorithm is given in Table 5.5. From Table 5.5, we can observed that no problem instance could be improved its global fairness objective function. The advantage of this approach is that the standard objective function still could be improved, though it could compensate on worsening the global fairness.

Table 5.5: Experimental Results From Enforcing Global Fairness With RL-LEX-HH Deficiency <=0%.

Dataset	Problem Instance	STD.OF		Global Unfairness		Cohort Unfairness			
						Specified		Exam-count-based	
		STD.OF	d.STD.OF (%)	AJFI	d.AJFI (%)	AJFI	d.AJFI (%)	AJFI	d.AJFI (%)
NOTT	NOTT-I	11803	**-1.01**	0.7944	0.00	0.8105	0.00	0.5382	0.00
	NOTT-II-32	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-31	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-30	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-26	8	0.00	0.9993	0.00	0.2215	0.00	0.2198	0.00
	NOTT-II-23	68	0.00	0.9925	0.00	0.7697	0.00	0.6466	0.00
CARTER	CAR-S-91	5.30	0.00	0.6769	0.00	0.6648	0.00	0.3729	0.00
	CAR-F-92	4.51	0.00	0.7070	0.00	0.6850	0.00	0.4289	0.00
	EAR-F-83	36.73	0.00	0.1739	0.00	0.1586	0.00	0.0996	0.00
	HEC-S-92	10.91	0.00	0.5090	0.00	0.2431	0.00	0.2928	0.00
	KFU-S-93	14.36	0.00	0.4546	0.00	0.3166	0.00	0.3334	0.00
	LSE-F-91	11.02	0.00	0.4887	0.00	0.4768	0.00	0.2612	0.00
	PUR-S-93	5.03	0.00	0.6709	0.00	0.6974	0.00	0.4173	0.00
	RYE-S-93	9.01	0.00	0.6266	0.00	0.4091	0.00	0.2977	0.00
	STA-F-83	157.12	0.00	0.0999	0.00	0.0043	0.00	0.0021	0.00
	TRE-S-92	8.75	0.00	0.5640	0.00	0.5551	0.00	0.3726	0.00
	UTA-S-92	3.60	0.00	0.7708	0.00	0.7403	0.00	0.4441	0.00
	UTE-S-92	25.20	0.00	0.2098	0.00	0.1427	0.00	0.2070	0.00
	YOR-F-83	38.03	0.00	0.2498	0.00	0.2384	0.00	0.2305	0.00
ITC 2007	EXAM1	6854	**-0.03**	0.8174	0.01	0.7804	0.01	0.4746	0.02
	EXAM2	632	0.00	0.9996	0.00	0.6664	0.00	0.4895	0.00
	EXAM3	11655	**-0.03**	0.9286	0.01	0.8968	0.01	0.5773	0.01
	EXAM4	16325	0.00	0.7343	0.00	0.7820	0.00	0.5027	0.00
	EXAM5	3837	0.00	0.8592	0.00	0.4551	0.00	0.6099	0.00
	EXAM6	27370	0.00	0.7878	0.00	0.6054	0.00	0.2866	0.00
	EXAM7	5498	**-0.54**	0.8363	0.00	0.6208	0.00	0.3602	0.00
	EXAM8	9798	0.00	0.6486	0.00	0.4377	0.00	0.2936	0.00
	EXAM9	1246	0.00	0.7459	0.00	0.6824	0.00	0.2412	0.00
	EXAM10	14556	0.00	0.6436	0.00	0.5040	0.00	0.0821	0.00
	EXAM11	36810	0.00	0.9065	0.00	0.8633	0.00	0.5489	0.00
	EXAM12	5300	0.00	0.8843	0.00	0.7543	0.00	0.5655	0.00
YEDITEPE	YUE20011.6.450	56	0.00	0.9175	0.00	0.8982	0.00	0.4531	0.00
	YUE20012.6.450	122	0.00	0.8585	0.00	0.8503	0.00	0.5659	-0.08
	YUE20013.1.150	29	0.00	0.8841	0.00	0.8961	0.00	0.3652	0.00
	YUE20021.7.550	76	0.00	0.9223	0.00	0.7300	0.00	0.5550	0.00
	YUE20022.7.550	160	0.00	0.8730	0.00	0.7331	0.00	0.5449	0.00
	YUE20023.1.150	56	0.00	0.8902	0.00	0.5915	0.00	0.4665	0.00
	YUE20031.6.550	143	0.00	0.9039	0.00	0.9155	0.00	0.6005	0.00
	YUE20032.6.550	434	0.00	0.7683	0.00	0.6377	0.00	0.5521	0.00

5.3.2.2 Enforcing Specified Cohort Fairness

With the same algorithms and scenarios, we conducted experiments as experiments to enforce specified cohort fairness. The only different is the objective function, i.e. specified cohort fairness instead of global fairness. Moreover, since it has been shown that RL-GD-HH is outperformed by RL-HC-HH, we did not conduct experiment with

RL-GD-HH.

Experimental Results With RL-HC-HH

The experimental results to enforce specified cohort fairness using RL-HC-HH algorithm with deficiency on standard objective function <=0% and <=5% are given in Table 5.6 and Table 5.7. Table 5.6 shows that similar with the experimental results to enforce global fairness, only few problem instances, i.e. 10 out 38, that can be improved their 'specified cohort' fairness without worsening their standard objective function value and the improvement is also very slightly. Meanwhile, if the standard objective function deficiency is allowed until 5%, as shown by Table 5.7, specified cohort fairness could be enforced over all problem instances, except for those are optimal already, e.g. `NOTT-II-32`. Compared to global fairness (see Table 5.2), specified cohort fairness could be enforced over more problem instances. For example, in ITC 2007 dataset, there is not a problem instance that can be improved its global fairness by more than 5%, whereas, there are three problem instances, i.e. `EXAM5, EXAM8, and EXAM12`, that can be improved their specified cohort fairness by more than 5%. From Table 5.7, we could also observe that over all problem instance, when the specified cohort is improved, the global fairness is also improved. However, the converse is not always true as shown in Table 5.2.

Experimental Results With RL-LEX-HH

The experimental results to enforce specified cohort fairness with RL-LEX-HH algorithm are given in Table 5.8. It can be observed from the table that with lexicographic optimality notion move acceptance, only two problem instances, i.e. `PUR-S-93, EXAM7` get their cohort fairness improved very slightly. The table also shows that the two problem instances also get their standard objective function improved.

5.3.2.3 Enforcing 'exam-count-based' Cohort Fairness

The experimental results for enforcing exam-count-based cohort fairness using RL-HC-GD, with scenario 1 and scenario 2 are given in Table 5.9 and Table 5.10 respectively. Whereas the experimental results with RL-LEX-GD are given in Table 5.11.

Table 5.9 shows that similar to the experiment to enforce global fairness and specified cohort fairness, the 'exam-count-based' cohort fairness could barely be improved without worsening the standard objective function value. However, if the standard objective function value is allowed to be worse by not more than 5%, compared to experiment to

Table 5.6: Experimental Results From Enforcing Specified Cohort Fairness With RL-HC-HH and Deficiency <=0%.

Dataset	Problem Instance	STD.OF		Global Unfairness		Cohort Unfairness			
						Specified		Exam-count-based	
		STD.OF	d.STD.OF (%)	AJFI	d.AJFI (%)	AJFI	d.AJFI (%)	AJFI	d.AJFI (%)
NOTT	NOTT-I	11923	0.00	0.7942	-0.02	0.8104	**-0.01**	0.5380	-0.02
	NOTT-II-32	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-31	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-30	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-26	8	0.00	0.9993	0.00	0.2215	0.00	0.2198	0.00
	NOTT-II-23	68	0.00	0.9925	0.00	0.7697	0.00	0.6466	0.00
CARTER	CAR-S-91	5.30	0.00	0.6768	-0.01	0.6647	**-0.02**	0.3728	-0.01
	CAR-F-92	4.51	0.00	0.7070	0.00	0.6850	0.00	0.4289	0.00
	EAR-F-83	36.73	0.00	0.1739	0.00	0.1586	0.00	0.0996	0.00
	HEC-S-92	10.91	0.00	0.5090	0.00	0.2431	0.00	0.2928	0.00
	KFU-S-93	14.36	0.00	0.4546	0.00	0.3166	0.00	0.3334	0.00
	LSE-F-91	11.02	0.00	0.4887	0.00	0.4768	0.00	0.2612	0.00
	PUR-S-93	5.03	-0.02	0.6699	-0.16	0.6966	**-0.12**	0.4159	-0.34
	RYE-S-93	9.01	0.00	0.6266	0.00	0.4091	0.00	0.2977	0.00
	STA-F-83	157.12	0.00	0.0999	0.00	0.0043	0.00	0.0021	0.00
	TRE-S-92	8.75	0.00	0.5640	0.00	0.5551	0.00	0.3726	0.00
	UTA-S-92	3.60	0.00	0.7708	0.00	0.7403	0.00	0.4441	0.00
	UTE-S-92	25.20	0.00	0.2098	0.00	0.1426	0.00	0.2065	-0.22
	YOR-F-83	38.03	0.00	0.2498	0.00	0.2384	0.00	0.2305	0.00
ITC 2007	EXAM1	6856	0.00	0.8170	-0.03	0.7802	**-0.02**	0.4743	-0.05
	EXAM2	632	0.00	0.9996	-0.01	0.6664	0.00	0.5876	20.06
	EXAM3	11659	0.00	0.9284	-0.02	0.8966	**-0.02**	0.5772	-0.03
	EXAM4	16325	0.00	0.7343	0.00	0.7820	0.00	0.5027	0.00
	EXAM5	3837	0.00	0.8592	0.00	0.4551	0.00	0.6099	0.00
	EXAM6	27370	0.00	0.7878	0.00	0.6054	0.00	0.2866	0.00
	EXAM7	5528	0.00	0.8360	-0.04	0.6193	**-0.23**	0.3598	-0.12
	EXAM8	9798	0.00	0.6482	-0.06	0.4376	**-0.03**	0.2931	-0.16
	EXAM9	1246	0.00	0.7459	0.00	0.6824	0.00	0.2412	0.00
	EXAM10	14556	0.00	0.6436	0.00	0.5040	0.00	0.0821	0.00
	EXAM11	36810	0.00	0.9064	-0.01	0.8631	**-0.03**	0.5489	0.00
	EXAM12	5300	0.00	0.8843	0.00	0.7543	0.00	0.5655	0.00
YEDITEPE	YUE20011.6.450	56	0.00	0.9175	0.00	0.8982	0.00	0.4531	0.00
	YUE20012.6.450	122	0.00	0.8585	0.00	0.8503	0.00	0.5663	0.00
	YUE20013.1.150	29	0.00	0.8841	0.00	0.8949	**-0.13**	0.3652	0.00
	YUE20021.7.550	76	0.00	0.9205	-0.19	0.7286	**-0.19**	0.5447	-1.87
	YUE20022.7.550	160	0.00	0.8730	0.00	0.7331	0.00	0.5449	0.00
	YUE20023.1.150	56	0.00	0.8902	0.00	0.5915	0.00	0.4665	0.00
	YUE20031.6.550	143	0.00	0.9039	0.00	0.9155	0.00	0.6005	0.00
	YUE20032.6.550	434	0.00	0.7683	0.00	0.6377	0.00	0.5521	0.00

Table 5.7: Experimental Results From Enforcing Specified Cohort Fairness With RL-HC-HH and Deficiency <=5%.

Dataset	Problem Instance	STD.OF		Global Unfairness		Cohort Unfairness			
						Specified		Exam-count-based	
		STD.OF	d.STD.OF (%)	AJFI	d.AJFI (%)	AJFI	d.AJFI (%)	AJFI	d.AJFI (%)
NOTT	NOTT-I	12519	5.00	0.7885	-0.74	0.8065	**-0.50**	0.5356	-0.49
	NOTT-II-32	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-31	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-30	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-26	8	0.00	0.9993	0.00	0.2215	0.00	0.2198	0.00
	NOTT-II-23	71	4.41	0.9921	-0.04	0.7695	**-0.03**	0.6464	-0.03
CARTER	CAR-S-91	5.57	4.99	0.6607	-2.40	0.6459	**-2.84**	0.3534	-5.23
	CAR-F-92	4.73	5.00	0.6948	-1.72	0.6722	**-1.87**	0.4186	-2.39
	EAR-F-83	38.56	5.00	0.1483	-14.71	0.1356	**-14.54**	0.0880	-11.64
	HEC-S-92	11.44	4.89	0.4910	-3.54	0.2183	**-10.20**	0.2819	-3.72
	KFU-S-93	15.08	5.00	0.4302	-5.35	0.2899	**-8.42**	0.2749	-17.54
	LSE-F-91	11.57	4.99	0.4598	-5.91	0.4516	**-5.29**	0.2552	-2.28
	PUR-S-93	5.28	5.00	0.6572	-2.04	0.6820	**-2.22**	0.4049	-2.96
	RYE-S-93	9.46	5.00	0.6132	-2.14	0.4036	**-1.34**	0.2883	-3.15
	STA-F-83	163.01	3.74	0.0877	-12.23	0.0009	**-79.96**	0.0006	-70.62
	TRE-S-92	9.18	5.00	0.5382	-4.57	0.5214	**-6.07**	0.3448	-7.47
	UTA-S-92	3.78	5.00	0.7589	-1.54	0.7263	**-1.89**	0.4351	-2.01
	UTE-S-92	26.45	4.96	0.1674	-20.19	0.1164	**-18.44**	0.1396	-32.53
	YOR-F-83	39.92	4.97	0.2319	-7.17	0.2181	**-8.55**	0.2322	0.71
ITC 2007	EXAM1	7198	4.99	0.8045	-1.56	0.7735	**-0.87**	0.4565	-3.79
	EXAM2	663	4.91	0.9994	-0.02	0.6663	**-0.01**	0.4893	-0.03
	EXAM3	12241	4.99	0.9232	-0.58	0.8915	**-0.59**	0.5733	-0.69
	EXAM4	17139	4.99	0.7188	-2.11	0.7685	**-1.72**	0.4753	-5.45
	EXAM5	4028	4.98	0.8430	-1.88	0.4018	**-11.73**	0.5962	-2.24
	EXAM6	28730	4.97	0.7844	-0.43	0.6032	**-0.38**	0.2932	2.30
	EXAM7	5802	4.96	0.8297	-0.79	0.6136	**-1.15**	0.3570	-0.88
	EXAM8	10287	4.99	0.6309	-2.73	0.4024	**-8.07**	0.2845	-3.10
	EXAM9	1308	4.98	0.7316	-1.92	0.6707	**-1.71**	0.2280	-5.46
	EXAM10	15283	4.99	0.6173	-4.09	0.4880	**-3.18**	0.0479	-41.65
	EXAM11	38647	4.99	0.8990	-0.83	0.8546	**-1.01**	0.5341	-2.68
	EXAM12	5563	4.96	0.8789	-0.62	0.5282	**-29.98**	0.5610	-0.80
YEDITEPE	YUE20011.6.450	58	3.57	0.9140	-0.38	0.8928	**-0.61**	0.5598	23.54
	YUE20012.6.450	128	4.92	0.8493	-1.07	0.8402	**-1.18**	0.6555	15.74
	YUE20013.1.150	30	3.45	0.8798	-0.48	0.8876	**-0.95**	0.3637	-0.42
	YUE20021.7.550	79	3.95	0.9188	-0.38	0.7197	**-1.42**	0.5445	-1.90
	YUE20022.7.550	168	5.00	0.8647	-0.96	0.7253	**-1.06**	0.5420	-0.52
	YUE20023.1.150	58	3.57	0.8856	-0.52	0.5856	**-1.01**	0.4648	-0.38
	YUE20031.6.550	150	4.90	0.8991	-0.53	0.9124	**-0.33**	0.5949	-0.94
	YUE20032.6.550	455	4.84	0.7557	-1.64	0.6252	**-1.95**	0.5891	6.71

Table 5.8: Experimental Results From Enforcing Specified Cohort Fairness With RL-LEX-HH and Deficiency <=5%.

Dataset	Problem Instance	STD.OF		Global Unfairness		Cohort Unfairness			
						Specified		Exam-count-based	
		STD.OF	d.STD.OF (%)	AJFI	d.AJFI (%)	AJFI	d.AJFI (%)	AJFI	d.AJFI (%)
NOTT	NOTT-I	11891	**-0.27**	0.7947	0.04	0.8106	**0.01**	0.5384	0.05
	NOTT-II-32	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-31	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-30	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-26	8	0.00	0.9993	0.00	0.2215	0.00	0.2198	0.00
	NOTT-II-23	68	0.00	0.9925	0.00	0.7697	0.00	0.6466	0.00
CARTER	CAR-S-91	5.30	0.00	0.6769	0.00	0.6648	0.00	0.3729	0.00
	CAR-F-92	4.51	0.00	0.7070	0.00	0.6850	0.00	0.4289	0.00
	EAR-F-83	36.73	0.00	0.1739	0.00	0.1586	0.00	0.0996	0.00
	HEC-S-92	10.91	0.00	0.5090	0.00	0.2431	0.00	0.2928	0.00
	KFU-S-93	14.36	0.00	0.4546	0.00	0.3166	0.00	0.3334	0.00
	LSE-F-91	11.02	0.00	0.4887	0.00	0.4768	0.00	0.2612	0.00
	PUR-S-93	5.03	**-0.03**	0.6709	0.00	0.6972	**-0.03**	0.4174	0.03
	RYE-S-93	9.01	0.00	0.6266	0.00	0.4091	0.00	0.2977	0.00
	STA-F-83	157.12	0.00	0.0999	0.00	0.0043	0.00	0.0021	0.00
	TRE-S-92	8.75	0.00	0.5640	0.00	0.5551	0.00	0.3726	0.00
	UTA-S-92	3.60	0.00	0.7708	0.00	0.7403	0.00	0.4441	0.00
	UTE-S-92	25.20	0.00	0.2098	0.00	0.1427	0.00	0.2070	0.00
	YOR-F-83	38.03	0.00	0.2498	0.00	0.2384	0.00	0.2305	0.00
ITC 2007	EXAM1	6856	0.00	0.8173	0.00	0.7803	0.00	0.4745	0.00
	EXAM2	632	0.00	0.9996	0.00	0.6664	0.00	0.4895	0.00
	EXAM3	11616	**-0.37**	0.9288	0.03	0.8970	**0.02**	0.5777	0.06
	EXAM4	16325	0.00	0.7343	0.00	0.7820	0.00	0.5027	0.00
	EXAM5	3837	0.00	0.8592	0.00	0.4551	0.00	0.6099	0.00
	EXAM6	27370	0.00	0.7878	0.00	0.6054	0.00	0.2866	0.00
	EXAM7	5508	**-0.36**	0.8364	0.01	0.6204	**-0.07**	0.3603	0.03
	EXAM8	9798	0.00	0.6486	0.00	0.4377	0.00	0.2936	0.00
	EXAM9	1246	0.00	0.7459	0.00	0.6824	0.00	0.2412	0.00
	EXAM10	14556	0.00	0.6436	0.00	0.5040	0.00	0.0821	0.00
	EXAM11	36810	0.00	0.9065	0.00	0.8633	0.00	0.5489	0.00
	EXAM12	5300	0.00	0.8843	0.00	0.7543	0.00	0.5655	0.00
YEDITEPE	YUE20011.6.450	56	0.00	0.9175	0.00	0.8982	0.00	0.4531	0.00
	YUE20012.6.450	122	0.00	0.8585	0.00	0.8503	0.00	0.5663	0.00
	YUE20013.1.150	29	0.00	0.8841	0.00	0.8961	0.00	0.3652	0.00
	YUE20021.7.550	76	0.00	0.9223	0.00	0.7300	0.00	0.5550	0.00
	YUE20022.7.550	160	0.00	0.8730	0.00	0.7331	0.00	0.5449	0.00
	YUE20023.1.150	56	0.00	0.8902	0.00	0.5915	0.00	0.4665	0.00
	YUE20031.6.550	143	0.00	0.9039	0.00	0.9155	0.00	0.6005	0.00
	YUE20032.6.550	434	0.00	0.7683	0.00	0.6377	0.00	0.5521	0.00

enforce global fairness and specified cohort fairness, there are more problem instances that can be improved their 'exam-count-based' cohort fairness significantly. For example, a problem instance `YUE20032.6.550` from Yeditepe dataset, its 'exam-count-based'-cohort fairness could be improved by 9.57%, whereas its global fairness and specified cohort fairness could be improved by only 1.82% and 1.95% respectively. On the other hand, Table 5.11 shows that using RL-LEX-HH only two problem instances that can be improved their 'exam-count-based' - cohort fairness with improvement very slightly. It is similar with the case of enforcing global fairness and specified cohort fairness.

Table 5.9: Experimental Results From Enforcing Exam-count-based Cohort Fairness With RL-HC-HH and Deficiency <=0%.

Dataset	Problem Instance	STD.OF		Global Unfairness		Cohort Unfairness			
						Specified		Exam-count-based	
		STD.OF	d.STD.OF (%)	AJFI	d.AJFI (%)	AJFI	d.AJFI (%)	AJFI	d.AJFI (%)
NOTT	NOTT-I	11923	0.00	0.7943	-0.01	0.8105	0.00	0.5380	**-0.03**
	NOTT-II-32	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-31	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-30	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-26	8	0.00	0.9993	0.00	0.2215	0.00	0.2198	0.00
	NOTT-II-23	68	0.00	0.9925	0.00	0.7697	0.00	0.6466	0.00
CARTER	CAR-S-91	5.30	0.00	0.6768	-0.01	0.6647	-0.02	0.3728	**-0.01**
	CAR-F-92	4.51	0.00	0.7070	0.00	0.6850	0.00	0.4289	0.00
	EAR-F-83	36.73	0.00	0.1739	0.00	0.1586	0.00	0.0996	0.00
	HEC-S-92	10.91	0.00	0.5090	0.00	0.2431	0.00	0.2928	0.00
	KFU-S-93	14.36	0.00	0.4545	-0.01	0.3165	-0.01	0.3329	**-0.16**
	LSE-F-91	11.02	0.00	0.4886	-0.02	0.4767	-0.01	0.2611	**-0.01**
	PUR-S-93	5.03	0.00	0.6703	-0.09	0.6970	-0.05	0.4167	**-0.15**
	RYE-S-93	9.01	0.00	0.6265	-0.02	0.4090	-0.01	0.2975	**-0.06**
	STA-F-83	157.12	0.00	0.0999	0.00	0.0043	0.00	0.0021	0.00
	TRE-S-92	8.75	0.00	0.5640	0.00	0.5551	0.00	0.3726	0.00
	UTA-S-92	3.60	0.00	0.7708	0.00	0.7403	0.00	0.4440	0.00
	UTE-S-92	25.20	0.00	0.2097	-0.02	0.1426	-0.02	0.2062	**-0.37**
	YOR-F-83	38.03	0.00	0.2498	0.00	0.2384	0.00	0.2305	0.00
ITC 2007	EXAM1	6856	0.00	0.8170	-0.03	0.7805	0.01	0.4743	**-0.05**
	EXAM2	632	0.00	0.9996	-0.01	0.6664	0.00	0.4894	**-0.02**
	EXAM3	11659	0.00	0.9280	-0.06	0.8962	-0.07	0.5761	**-0.21**
	EXAM4	16325	0.00	0.7343	0.00	0.7820	0.00	0.5027	0.00
	EXAM5	3837	0.00	0.8592	0.00	0.4551	0.00	0.6099	0.00
	EXAM6	27370	0.00	0.7878	0.00	0.6054	0.00	0.2866	0.00
	EXAM7	5528	0.00	0.8358	-0.06	0.6200	-0.12	0.3594	**-0.22**
	EXAM8	9798	0.00	0.6486	0.00	0.4377	0.00	0.2936	0.00
	EXAM9	1246	0.00	0.7459	0.00	0.6824	0.00	0.2412	0.00
	EXAM10	14556	0.00	0.6436	0.00	0.5040	0.00	0.0821	0.00
	EXAM11	36810	0.00	0.9064	-0.01	0.8633	0.00	0.5469	**-0.36**
	EXAM12	5300	0.00	0.8843	0.00	0.7543	0.00	0.5655	0.00
YEDITEPE	YUE20011.6.450	56	0.00	0.9175	0.00	0.8982	0.00	0.4531	0.00
	YUE20012.6.450	122	0.00	0.8585	0.00	0.8503	0.00	0.5659	**-0.08**
	YUE20013.1.150	29	0.00	0.8841	0.00	0.8961	0.00	0.3652	0.00
	YUE20021.7.550	76	0.00	0.9223	0.00	0.7300	0.00	0.5550	0.00
	YUE20022.7.550	160	0.00	0.8730	0.00	0.7331	0.00	0.5449	0.00
	YUE20023.1.150	56	0.00	0.8902	0.00	0.5915	0.00	0.4665	0.00
	YUE20031.6.550	143	0.00	0.9039	0.00	0.9155	0.00	0.6005	0.00
	YUE20032.6.550	434	0.00	0.7683	0.00	0.6377	0.00	0.5521	0.00

5.4 Conclusion

In this chapter, a study of enforcing fairness in examination timetabling problem with two-stage approach was carried out. While the stage 1 aims at optimising standard objective function, the stage 2 aims at enforcing fairness. Three notions of fairness,

Table 5.10: Experimental Results From Enforcing Exam-count-based Cohort Fairness With RL-HC-HH and Deficiency <=5%.

Dataset	Problem Instance	STD.OF		Global Unfairness		Cohort Unfairness			
						Specified		Exam-count-based	
		STD.OF	d.STD.OF (%)	AJFI	d.AJFI (%)	AJFI	d.AJFI (%)	AJFI	d.AJFI (%)
NOTT	NOTT-I	12514	4.96	0.7844	-1.25	0.8034	-0.87	0.5278	**-1.93**
	NOTT-II-32	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-31	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-30	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-26	8	0.00	0.9993	0.00	0.2215	0.00	0.2198	0.00
	NOTT-II-23	71	4.41	0.9921	-0.04	0.7695	-0.03	0.6464	**-0.03**
CARTER	CAR-S-91	5.57	4.99	0.6674	-1.40	0.6548	-1.50	0.3383	**-9.27**
	CAR-F-92	4.73	4.99	0.6958	-1.58	0.6714	-1.99	0.3822	**-10.88**
	EAR-F-83	38.56	5.00	0.1601	-7.93	0.1427	-10.05	0.0696	**-30.15**
	HEC-S-92	11.45	4.94	0.4823	-5.25	0.2326	-4.32	0.2621	**-10.51**
	KFU-S-93	15.08	5.00	0.4457	-1.95	0.2972	-6.13	0.2910	**-12.70**
	LSE-F-91	11.57	5.00	0.4763	-2.55	0.4680	-1.84	0.2381	**-8.84**
	PUR-S-93	5.28	5.00	0.6601	-1.61	0.6889	-1.22	0.4022	**-3.61**
	RYE-S-93	9.46	5.00	0.6137	-2.05	0.3886	-4.99	0.2818	**-5.35**
	STA-F-83	161.77	2.96	0.0831	-16.86	0.0021	-49.81	0.0006	**-70.85**
	TRE-S-92	9.18	4.99	0.5481	-2.82	0.5341	-3.78	0.3444	**-7.58**
	UTA-S-92	3.78	4.99	0.7601	-1.39	0.7301	-1.38	0.4260	**-4.06**
	UTE-S-92	26.44	4.94	0.1854	-11.63	0.1297	-9.11	0.1302	**-37.07**
	YOR-F-83	39.93	4.98	0.2429	-2.76	0.2311	-3.06	0.1984	**-13.92**
ITC 2007	EXAM1	7190	4.87	0.8052	-1.47	0.7700	-1.32	0.4655	**-1.90**
	EXAM2	663	4.91	0.9995	-0.02	0.6661	-0.05	0.4893	**-0.04**
	EXAM3	12241	4.99	0.9252	-0.36	0.8940	-0.31	0.5713	**-1.05**
	EXAM4	17141	5.00	0.7242	-1.37	0.7732	-1.13	0.4907	**-2.39**
	EXAM5	4028	4.98	0.8420	-2.00	0.4493	-1.29	0.5958	**-2.32**
	EXAM6	28035	2.43	0.7922	0.55	0.6144	1.48	0.2844	**-0.75**
	EXAM7	5804	4.99	0.8272	-1.08	0.6135	-1.16	0.3539	**-1.74**
	EXAM8	10287	4.99	0.6317	-2.61	0.4259	-2.70	0.2795	**-4.79**
	EXAM9	1308	4.98	0.7243	-2.90	0.6634	-2.79	0.2142	**-11.19**
	EXAM10	15281	4.98	0.6235	-3.13	0.4949	-1.82	0.0675	**-17.77**
	EXAM11	38650	5.00	0.9006	-0.65	0.8574	-0.69	0.5415	**-1.35**
	EXAM12	5565	5.00	0.8786	-0.65	0.6099	-19.15	0.5213	**-7.82**
YEDITEPE	YUE20011.6.450	58	3.57	0.9140	-0.38	0.8956	-0.30	0.4309	**-4.90**
	YUE20012.6.450	128	4.92	0.8477	-1.26	0.8402	-1.19	0.5565	**-1.73**
	YUE20013.1.150	30	3.45	0.8798	-0.48	0.8924	-0.42	0.3544	**-2.98**
	YUE20021.7.550	79	3.95	0.9188	-0.38	0.9261	26.86	0.5527	**-0.43**
	YUE20022.7.550	168	5.00	0.8669	-0.70	0.8885	21.21	0.5254	**-3.57**
	YUE20023.1.150	58	3.57	0.8856	-0.52	0.5893	-0.39	0.4551	**-2.46**
	YUE20031.6.550	150	4.90	0.8948	-1.00	0.9119	-0.39	0.5911	**-1.57**
	YUE20032.6.550	455	4.84	0.7584	-1.29	0.7878	23.55	0.4992	**-9.57**

Table 5.11: Experimental Results From Enforcing Exam-count-based Cohort Fairness With RL-LEX-HH and Deficiency <=0%.

Dataset	Problem Instance	STD.OF		Global Unfairness		Cohort Unfairness			
						Specified		Exam-count-based	
		STD.OF	d.STD.OF (%)	AJFI	d.AJFI (%)	AJFI	d.AJFI (%)	AJFI	d.AJFI (%)
NOTT	NOTT-I	11911	**-0.10**	0.7939	-0.06	0.8099	-0.07	0.5378	**-0.08**
	NOTT-II-32	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-31	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-30	0	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00
	NOTT-II-26	8	0.00	0.9993	0.00	0.2215	0.00	0.2198	0.00
	NOTT-II-23	68	0.00	0.9925	0.00	0.7697	0.00	0.6466	0.00
CARTER	CAR-S-91	5.30	0.00	0.6769	0.00	0.6648	0.00	0.3729	0.00
	CAR-F-92	4.51	0.00	0.7070	0.00	0.6850	0.00	0.4289	0.00
	EAR-F-83	36.73	0.00	0.1739	0.00	0.1586	0.00	0.0996	0.00
	HEC-S-92	10.91	0.00	0.5090	0.00	0.2431	0.00	0.2928	0.00
	KFU-S-93	14.36	0.00	0.4546	0.00	0.3166	0.00	0.3334	0.00
	LSE-F-91	11.02	0.00	0.4887	0.00	0.4768	0.00	0.2612	0.00
	PUR-S-93	5.03	0.00	0.6709	0.00	0.6974	0.00	0.4173	0.00
	RYE-S-93	9.01	0.00	0.6266	0.00	0.4091	0.00	0.2977	0.00
	STA-F-83	157.12	0.00	0.0999	0.00	0.0043	0.00	0.0021	0.00
	TRE-S-92	8.75	0.00	0.5640	0.00	0.5551	0.00	0.3726	0.00
	UTA-S-92	3.60	0.00	0.7708	0.00	0.7403	0.00	0.4441	0.00
	UTE-S-92	25.20	0.00	0.2097	-0.02	0.1426	-0.01	0.2066	**-0.16**
	YOR-F-83	38.03	0.00	0.2498	0.00	0.2384	0.00	0.2305	0.00
ITC 2007	EXAM1	6856	0.00	0.8173	0.00	0.7803	0.00	0.4745	0.00
	EXAM2	632	0.00	0.9996	0.00	0.6664	0.00	0.4895	0.00
	EXAM3	11599	**-0.51**	0.9290	0.05	0.8972	0.05	0.5778	**0.09**
	EXAM4	16325	0.00	0.7343	0.00	0.7820	0.00	0.5027	0.00
	EXAM5	3837	0.00	0.8592	0.00	0.4551	0.00	0.6099	0.00
	EXAM6	27370	0.00	0.7878	0.00	0.6054	0.00	0.2866	0.00
	EXAM7	5510	**-0.33**	0.8364	0.02	0.6207	-0.01	0.3604	**0.05**
	EXAM8	9798	0.00	0.6486	0.00	0.4377	0.00	0.2936	0.00
	EXAM9	1246	0.00	0.7459	0.00	0.6824	0.00	0.2412	0.00
	EXAM10	14556	0.00	0.6436	0.00	0.5040	0.00	0.0821	0.00
	EXAM11	36810	0.00	0.9065	0.00	0.8633	0.00	0.5489	0.00
	EXAM12	5300	0.00	0.8843	0.00	0.7543	0.00	0.5655	0.00
YEDITEPE	YUE20011.6.450	56	0.00	0.9175	0.00	0.8982	0.00	0.4531	0.00
	YUE20012.6.450	122	0.00	0.8585	0.00	0.8503	0.00	0.5663	0.00
	YUE20013.1.150	29	0.00	0.8841	0.00	0.8961	0.00	0.3652	0.00
	YUE20021.7.550	76	0.00	0.9223	0.00	0.7300	0.00	0.5550	0.00
	YUE20022.7.550	160	0.00	0.8730	0.00	0.7331	0.00	0.5449	0.00
	YUE20023.1.150	56	0.00	0.8902	0.00	0.5915	0.00	0.4665	0.00
	YUE20031.6.550	143	0.00	0.9039	0.00	0.9155	0.00	0.6005	0.00
	YUE20032.6.550	434	0.00	0.7683	0.00	0.6377	0.00	0.5521	0.00

i.e. global fairness, specified or specified cohort fairness, and 'exam-count-based cohort' fairness are enforced separately in stage 2.

The experimental results suggests that compared to great deluge, and lexicographic move acceptance, hill climbing move acceptance is the most effective method used to enforce fairness. From the analysis of search trajectory from different algorithms as given in Chapter 3 (see Figure 3.1), we can observe that since during the search within hill climber algorithm , only better solution will be accepted, it is very fast in finding good solution. Within the same number iterations, it could find much better solutions compared to the solutions from other algorithm that accept worse solution during the search. Though, within longer run, hill climbing could be trapped into local optima very soon. We believe it also the case in optimising fairness objective function discussed in this chapter. On the other hand, since great deluge and lexicographic move acceptance accept worse solution during the search, we believe it is the case that within the given time, these algorithms failed to improve the solution.

The study also shows that without worsening the standard objective function, all notions of fairness could not be significantly improved. Moreover, if the deficiency of standard objective function is allowed up to 5%, all notions of fairness could be improved over all problem instances that are varies on its significance. There are more problem instances that can be enforced fairer significantly by specified cohort fairness notion rather than by global fairness notion and there are even more problem instances that can be enforced fairer significantly by 'exam-count-based cohort' fairness notion than by specified cohort fairness notion. In addition, the experimental results also showed that fairness amongst students within 'specified cohort' could be improved more significantly than fairness between all students, and fairness amongst student within 'exam-count-based cohort' could be improved more significantly than fairness amongst students within 'specified cohort'

In conclusion, since almost over all problem instances the all notions of fairness could not be improved without worsening the standard objective function. It shows that there is an intrinsic trade-off between the standard objective function and fairness objective function. From this study it is clear that we require an approach that can make best trade-off between the standard objective function and fairness objective function.

Chapter 6

Multi-objective Approach with Scalarisation Techniques

6.1 Introduction

As discussed in Chapter 5, within two-stage approach, the fairness barely could be improved without worsening the standard objective function. Therefore, a multi-objective optimisation method is required to find optimal trade-off between standard objective function and fairness objective function. This chapter investigates hyper-heuristic approaches based on classical scalarisation techniques to cope with the multi-objective examination timetabling problems (MOETP); and also evaluates the effect of 'multi-objectivisation' on the benchmark objective function. The proposed hyper-heuristic approaches are tested over two variants of the proposed multi-objective examination timetabling problems, i.e. MOETP with fairness and MOETP with multiple stakeholder perspectives as discussed in Chapter 4:

The experiments are conducted over two variants of proposed multi-objective examination timetabling problems:

1. Multi-objective examination timetabling problem with different notions of fairness:
   (a) Bi-objective functions: standard objective function and global unfairness objective function
   (b) Bi-objective functions: standard objective function and 'specified cohort unfairness' objective function.
   (c) Bi-objective functions: standard objective function and 'count-exam-based

cohort' unfairness objective function.

(d) Three objective functions: standard objective function, global unfairness objective function and 'specified cohort unfairness' objective function.

(e) Three objective functions: standard objective function, global unfairness objective function and 'exam-count-based cohort unfairness' objective function.

2. Multi-objective examination timetabling problem with multiple stakeholders' perspective:

(a) Bi-objective : Students objective function and Administrators objective function.

(b) Four-objective : students objective function, markers objective function, invigilators objective function, and estates objective function

The contents of this chapter are organised as follows: Section 6.2 presents the proposed hyper-heuristic approach. Section 6.3 reports the preliminary experimentations to determine best parameter setting of the proposed hyper-heuristic approach. The experimental results over multi-objective examination timetabling problem with different notion of fairness are discussed in Section 6.4 whereas the results over multi-objective examination timetabling problem with multiple stakeholder perspectives are discussed in Section 6.5. Further, Section 6.6 evaluates the effect of multi-objectivisation on the benchmark objective function and Section 6.7 analyses the nature of relationship between objectives in the problems. Lastly, Section 6.8 summarises and discusses the research works reported in this chapter.

6.2 Method: Multi-objective Hyper-heuristic With Classical Scalarisation Techniques

Basically, the method used in this chapter is similar with method used in Chapter 3 - 5, i.e. hyper-heuristic within HyFlex framework. However, different from the hyper-heuristic used in the previous chapters that is dedicated for single objective problem, in this chapter we proposed hyper-heuristic approaches to cope with multi-objective problems. Instead of generating a single solution as in the case of a single objective optimisation problem, in a multi-objective problem, a set of Pareto optimal solution is generated.

In order to generate a set of Pareto optimal solutions, we employed two classical scalarisation techniques, i.e. weighted sum and weighted Tchebycheff [212]. These scalarisa-

tion techniques are combined with Pareto sorting technique within great deluge based hyper-heuristic. Within these scalarisation techniques, we have to set the weights for each objective function. Suppose $f_1, f_2, ...f_N$ are the the N objective functions and their respective weights are $w_1, w_2, ..., w_N$ in which$\sum_{i=0}^{i=N} w_i = 1$, the weighted sum and weighted Tchebycheff objective functions are defined in Equation 6.1 and Equation 6.2 respectively. In Equation 6.2, $r_1, r_2, ..., r_N$ are reference points or ideal points for $f_1, f_2, ..., f_N$ respectively.

$$\text{minimise} \sum_{i=1}^{i=N} f_i(x)w_i \tag{6.1}$$

$$\text{minimise}[\max(|f_1(x) - r_1(x)|w_1, |f_2(x) - r_2(x)|w_2), ..., |f_N(x) - r_N(x)|w_N)] \tag{6.2}$$

Similar to the earlier chapters, the proposed hyper-heuristic approaches employed in this chapter are implemented in HyFlex, a hyper-heuristic framework. However, since the existing version of HyFlex only support for single objective optimisation problem, extension of the existing version is required. The following sections explain our extension work over the existing version of HyFlex in order to cope with multi-objective optimisation problems as well as the hyper-heuristic strategy.

6.2.1 Extension of HyFlex

Since the current version of HyFlex only support for single objective optimisation problems, we extended the problem domain interface to cope with multi-objective optimisation problem. Table 6.1 lists methods added to problem domain interface of HyFlex to deal with multi-objective problems with scalarisation techniques.

Different from single objective optimisation problems, in multi-objective optimisation problems, multiple objective function values are maintained instead of a single objective function value. Therefore, to cope with the multi-objective problems two problem domain methods, namely, `getNumberOfObjectives()` and `getFunctionValues(int i)` are added to get the number of objective functions and their values respectively. In addition, since its objective function value has different range, a problem domain method `normaliseObjectiveFunctionValue (boolean s)` is added to normalise the objective function values into a value between `0-1`.

Moreover, different from single objective problem, in which only one solution is

Table 6.1: List of methods added to `ProblemDomain` interface in HyFlex to deal with multi-objective problems with scalarisation technique

Method	Description
`getNumberOfObjectives()`	Return the number of objective functions.
`normaliseObjectiveFunctionValue (boolean s)`	A method for setting whether normalisation over the objective function values is used.
`getFunctionValues(int i)`	Return an array of objective functions' values (`double`) instead of a single objective value from the solution located in index `i` of solution memory.
`setWeightVector(double[] w)`	Set a weight vector for each the objective function with weight vector `w`.
`setReferenceVector(double[] r)`	Set a reference vector for the objective function values.
`addSolutionToParetoSet(int i)`	Add the solution located in index `i` of solution memory to the Pareto set.
`getWeightedFunctionValue(int i)`	Get aggregate objective function of the solution located in index `i`. This method corresponds to the Equation 6.1 and Equation 6.2 for weighted sum and weighted Tchebycheff respectively.
`sortParetoSet()`	Sort solutions in Pareto set using Algorithm 10 to exclude dominated solutions.
`getParetoSet()`	Return the solutions in Pareto set.

maintained, in a multi-objective problem, a set of non-dominated solutions, i.e. Pareto set, are maintained. Therefore, three methods for maintaining the Pareto optimal set solution, i.e. `addSolutionToParetoSet(int i)`, `sortParetoSet()`, and `getParetoSet()` are also added to the problem domain interface. The first method is dedicated for adding a new solution to the Pareto set whereas the second method for excluding the dominated solutions from the Pareto set and maintaining the non-dominated solutions. Within method, `getParetoSet()`, we used Algorithm 10 adopted from [180]. In Algorithm 10, it is assumed that there are only two objective functions. Refer to [180] for more than two objective functions. Lastly, the third method is dedicated to get the set of Pareto solutions.

The last methods added to the problem domain interface of HyFlex are methods dedicated for scalarisation techniques. For this purpose, two methods, i.e `setReferenceVector(double[] r)` and `getWeightedFunctionValue(int i)` are added. The first method is for setting the reference vector whereas the second method is for getting the objective function value with the scalarisation techniques, weighted sum or weighted Tchebycheff given in Equation 6.1 and Equation 6.2 respectively. Noteworthy here is that the methods:`setWeightVector(double[] w)`, `setReferenceVector(double[] r)`, `addSolutionToParetoSet(int i)`, `getWeightedFunctionValue(int i)`, `sortParetoSet()`, `getParetoSet()`, instead of being part of Problem domain, they could also be part of the hyper-heuristic strategy. However, to determine which option is more effective is out of the scope of this study.

Algorithm 10 Pseudo-code of Pareto Optimal Sorting With Two Objectives (based on [180])

```
1: procedure sortParetoSet(SolutionSet S)
2:     // Initiate Pareto optimal solution set
3:     P_opt ← ∅
4:     //Sort (ascending) solutions by the second objective function value
5:     S* ← sortAscending(S,2)
6:     //Sort (ascending) solutions by the first objective function value
7:     S** ← sortAscending(S,1)
8:     b ← ∞
9:     for i ← 1, sizeOf(S**) do
10:        //e_i is i^th element of S**
11:        //f_1(e_i) is the first objective function value of solution e_i
12:        if f_1(e_i) < b then
13:            P_opt ←P_opt∪e_i
14:            b ←f_1(e_i)
15:        end if
16:    end for
17:    return P_opt
18: end procedure
```

6.2.2 Hyper-heuristic Strategy

There is no change in the legacy hyper-heuristic interface needed to implement our proposed hyper-heuristic approach to deal with our proposed multi-objective examination

timetabling problems. It should be kept in mind that within a multi-objective problem a set of Pareto solutions are maintained instead of a single solution as in single objective problem. Our proposed hyper-heuristic is given in Algorithm 11. The methods used in Algorithm 11 are explained in Table 6.2.

Basically, Algorithm 11 is similar to reinforcement learning - great deluge hyper-heuristic for a single objective problem given in Algorithm 7 in Chapter 3. The only difference is the mechanism of Pareto solutions archiving and sorting that are absent in a single objective problem.

As shown by Algorithm 11, the algorithm consists of outer iterations (line 4) and inner iterations (line 19). In each outer iteration, the weight vector is generated randomly. In other words, a different weight vector will be used in each outer iteration. In each inner iteration, whenever a new solution is accepted it will be added to Pareto set. Finally, in the end of outer iteration the Pareto Set are sorted to exclude any dominated solutions using Algorithm 10.

Table 6.2: List of methods as part of Hyper-heuristic strategy given in Algorithm 11

Method	Description
`genRandVec(n)`	Generate a weight vector of size `n` consist of random numbers between [0-1], in which the sum of the numbers is 1.
`genRefPoint(nof, of)`	Generate reference points, initially we set to 80 percent of the initial solution objective function values.
`initiateReward(R)`	A method to give the same initial reward for each low-level heuristics. We set to `10`.
`upgradeReward(R[`ι`])`	Update the reward for low-level heuristic `R[`ι`]` by adding the value with `reward` (we set to `1`) if the value of `R[`ι`]` does not reach the `upperbound` (we set to `20`).
`downgradeReward(R[`ι`])`	Update the reward for low-level heuristic `R[`ι`]` by decreasing the value with `reward` (we set to `1`) if the value of `R[`ι`]` does not reach the `lowerbound` (we set to `-20`).
`P.initiateSolution(i)`	Generate an initial feasible solution and stored it in solution memory index `i`. Note it is part of the legacy problem domain method.
`P.copySolution(1,0)`	Copy solution in solution memory index `i` to solution memory index `j`. Note it is part of the legacy problem domain method.

6.3 Preliminary Experiments

In this section, we investigate our proposed hyper-heuristic based on reinforcement learning and great deluge algorithm with scalarisation techniques given in Algorithm 11 over bi-objective examination timetabling problems with fairness, aiming at optimising standard objective function and global unfairness to get the best parameter settings. The

Algorithm 11 Pseudo-code of reinforcement learning - great deluge hyper-heuristic with scalarisation technique for multi-objective examination timetabling problems. Refer to Table 6.1 and Table 6.2 for detailed explanations of the methods.

```
1: procedure SOLVE(ProblemDomain P)
2:     Set number of iteration, N
3:     Set stopping condition, St
4:     for i← 1, N do
5:         nof ← P.getNumberOfObjectives()
6:         w ← genRandVec(nof)
7:         P.setWeightVector(w)
8:         P.initialiseSolution(0)
9:         of ← P.getFunctionValues(0)
10:        ref ← genRefPoint(nof, of)
11:        P.setReferenceVector(ref)
12:        P.addSolutionToParetoSet(0)
13:        //set boundary level
14:        B ← P.getWeightedFunctionValue(0)
15:        Set decay rate α
16:        Set M ← P.getNumberofHeuristics()
17:        Set R ← new int[M]
18:        initiateReward(R)
19:        while St is not met do
20:            //get the index of low-level heuristic (LLH) with highest score
21:            ι ← selectLLH(R)
22:            P.applyLLH(ι,0,1)
23:            newFunctionVal ← P.getWeightedFunctionValue(1)
24:            if (newFunctionVal ≤ S_curr OR newFunctionVal ≤ B) then
25:                //accept the new solution
26:                P.copySolution(1,0)
27:                P.addSolutionToParetoSet(1)
28:                upgradeReward(R[ι])
29:            else
30:                downgradeReward(R[ι])
31:            end if
32:            B ← B-α
33:        end while
34:    end for
35:    P.sortParetoSet()
36:    P_a ← P.getParetoSet()
37:    //return Pareto Solution
38:    return P_a
39: end procedure
```

experiments are reported in Appendix E and include:

1. Evaluating Pareto scanning strategy.
2. Evaluating the initial solution.
3. Evaluating number of iterations.
4. Evaluating normalisation.
5. Evaluating reference points.

Moreover, with the best parameter settings from earlier preliminary experiments, in Section 6.3.1, the best parameter settings are tested over all problem instances from four datasets, i.e. Nottingham, Carter, ITC 2007, and Yeditepe. Furthermore, in Section 6.3.2 as opposed to reinforcement learning, two different low-level heuristic selection strategies, i.e. random selection and self-adaptive learning are evaluated. Lastly, in Section 6.3.3 will be evaluated two different scalarisation techniques, i.e. weighted sum and weighted Tchebycheff.

As quality indicators, or in other words quality metric of Pareto solutions, we used three quality indicators commonly used in the scientific literature to measure the performance of algorithms over multi-objective optimisation problems, i.e. coverage, hypervolume, and spread (see Section 2.7 in Chapter 2 for the detailed elaboration).

6.3.1 Experimental Results with The Best Parameter Setting over All Problem Instances

Using the best parameter settings evaluated in the preliminary experiments reported in [219], this section evaluates Algorithm 11 over all problem instances from four datasets, i.e. Nottingham, Carter, ITC 2007, and Yeditepe. Again, we tested Algorithm 11 with the best parameter settings under two different scenarios given in Table 6.4. Though, it has been proven in preliminary experiments that Scenario 2, in which any new solutions generated by applying the chosen low-level heuristic is added to Pareto solution set, outperforms Scenario 1, in which only new solution accepted by move acceptance mechanism is added to the Pareto set; this section evaluates whether it is consistent that Scenario 2 outperforms Scenario 1 if the number of iteration is changed from 21 to 100 and the initial solution in each iteration after the first iteration is changed by choosing randomly from the existing Pareto set rather than always initiate with the same initial solution. In addition, in order to prevent the standard objective function from getting too worse, any solutions with standard objective function worse than 5% from the initial

solution standard objective function value is not added to the Pareto solution set.

Table 6.3: The best parameter settings for the experiments over the bi-objective problem using Algorithm 11.

Paremeter	Value
Number of iterations (N)	100 iterations
Stopping condition (St)	10000 iterations
Ref.Points	80% of initial solutions
Normalisation	The objective function values are converted to the value between 0-1.
Initial Solution	In the first iteration, the initial solution is the solution generated by constructor with improvement phase over the standard objective function as discussed in Chapter 3. In the following iteration, the initial solution is a solution randomly chosen from the current Pareto solution set.
Decay rate (α)	0.0001

Table 6.4: Experimental Scenarios of Pareto Scanning

Scenario	Pareto Scanning
S1	A new solution generated by applying the chosen low-level heuristic is added to Pareto solution set if it is accepted by move acceptance mechanism and the standard objective function is not worsen more than 5% from the initial solution.
S2	Any new solutions generated by applying the chosen low-level heuristic with standard objective function not worsen more than 5% from the initial solution is added to Pareto solution set.

Table 6.5 compares the range of objective function values of Pareto solution set produced with both scenarios. The 'NA' indicates that over the respective problem instance the Algorithm with default memory heap size, results 'out of memory size' status due to the size of Pareto solution set. From Table 6.5 we can observe that on average, Scenario 2 results Pareto solution set with wider objective function values range.

Table 6.6 and Table 6.7 present the best solution in Pareto solution set in terms of standard objective function and global fairness respectively. From Table 6.6, we can observe that over some problem instances, e.g. EAR83, EXAM2, and YUE20022, the first objective function, i.e. standard objective function still could be improved quite significantly and in the same time the second objective, i.e. global fairness, is slightly worse or constant. Even, over several problem instances, e.g. UTE92, both objective functions could be improved. Regarding enforcing global fairness, as shown by Table 6.7, though the standard objective function is prevented from getting worse more than 5% of the initial solution, over some problem instances, the global fairness objective function could be improved quite significantly, for instance, under Scenario 2, problem instances EAR83, STA83, UTE92, and EXAM9 their global fairness could be improved by 17.11%, 31.35%, 19.47, and 8.77% respectively.

Table 6.5: The range of objective function values of the Pareto solution set from the experiments with different Pareto scanning scenarios.

P. Instance	Init. Solution		Scenario 1 (S1)						Scenario 2 (S2)					
			STD.OF			AJFI			STD.OF			AJFI		
	STD.OF	AJFI	min	max	range	min	max	range	min	max	range	min	max	range
NOTT-I	11923	0.77	11746	12512	766	0.77	0.79	0.02	11715	12468	753	0.75	0.78	0.03
NOTT-II-32	0	0.00	0	0	0	0.00	0.00	0.00	0	0	0	0.00	0.00	0.00
NOTT-II-31	0	0.00	0	0	0	0.00	0.00	0.00	0	0	0	0.00	0.00	0.00
NOTT-II-30	0	0.00	0	0	0	0.00	0.00	0.00	0	0	0	0.00	0.00	0.00
NOTT-II-26	8	1.00	8	8	0	1.00	1.00	0.00	8	8	0	1.00	1.00	0.00
NOTT-II-23	68	0.99	68	71	3	0.99	0.99	0.00	68	71	3	0.99	0.99	0.00
CAR91	5.30	0.68	5.29	5.44	0.14	0.67	0.68	0.01	5.30	5.55	0.25	0.66	0.67	0.02
CAR92	4.51	0.71	4.49	4.63	0.14	0.70	0.71	0.01	4.47	4.73	0.26	0.69	0.71	0.02
EAR83	36.73	0.17	36.04	38.53	2.49	0.15	0.17	0.02	36.36	38.49	2.13	0.14	0.17	0.03
HEC92	10.91	0.51	10.90	11.38	0.48	0.48	0.51	0.03	10.72	11.45	0.73	0.46	0.51	0.05
KFU93	14.36	0.45	14.36	14.97	0.62	0.43	0.45	0.02	14.35	15.04	0.68	0.41	0.45	0.04
LSE91	11.02	0.49	10.94	11.56	0.62	0.45	0.48	0.03	10.92	11.56	0.65	0.43	0.49	0.05
PUR93	5.03	0.67			NA						NA			
RYE92	9.01	0.63	9.01	9.42	0.42	0.61	0.63	0.02	8.99	9.45	0.46	0.60	0.63	0.03
STA83	157.12	0.10	157.12	164.84	7.72	0.07	0.10	0.03	157.12	164.95	7.83	0.07	0.10	0.03
TRE92	8.75	0.56	8.75	9.14	0.40	0.53	0.56	0.04	8.69	9.17	0.48	0.53	0.56	0.03
UTA92	3.60	0.77	3.60	3.74	0.14	0.76	0.77	0.01	3.60	3.78	0.18	0.75	0.77	0.02
UTE92	25.20	0.21	25.10	26.45	1.35	0.18	0.20	0.02	25.06	26.30	1.25	0.17	0.21	0.04
YOR83	38.03	0.25	38.03	39.82	1.79	0.22	0.25	0.03	37.40	39.72	2.31	0.21	0.25	0.04
EXAM1	6856	0.60	6749	7184	435	0.80	0.82	0.03	6721	7194	473	0.78	0.82	0.04
EXAM2	632	0.92	597	663	66	1.00	1.00	0.00			NA			
EXAM3	11659	0.91			NA						NA			
EXAM4	16325	0.71	16273	17129	856	0.72	0.73	0.01	16324	17122	798	0.71	0.73	0.03
EXAM5	3837	0.64			NA						NA			
EXAM6	27370	0.76	27355	28680	1325	0.78	0.79	0.00	27365	28710	1345	0.78	0.79	0.01
EXAM7	5528	0.79	5384	5740	356	0.82	0.84	0.02			NA			
EXAM8	9798	0.58	9726	10168	442	0.62	0.64	0.02	9686	10272	586	0.59	0.63	0.04
EXAM9	1246	0.63	1242	1293	51	0.71	0.74	0.04	1199	1304	105	0.68	0.74	0.06
EXAM10	14556	0.63	14556	15277	721	0.62	0.64	0.02	14556	15276	720	0.60	0.64	0.04
EXAM11	36810	0.89			NA						NA			
EXAM12	5300	0.88	5286	5532	246	0.88	0.88	0.01	5286	5565	279	0.88	0.88	0.01
YUE20011	56	0.92	51	58	7	0.91	0.92	0.01	51	58	7	0.91	0.92	0.01
YUE20012	122	0.86	118	128	10	0.85	0.86	0.02	121	128	7	0.84	0.86	0.02
YUE20013	29	0.88	29	30	1	0.88	0.88	0.00	29	30	1	0.88	0.88	0.00
YUE20021	76	0.92	76	79	3	0.92	0.92	0.00	76	79	3	0.92	0.92	0.00
YUE20022	162	0.87	150	168	18	0.86	0.87	0.02	151	170	19	0.85	0.88	0.02
YUE20023	56	0.89	56	58	2	0.87	0.87	0.00	56	58	2	0.87	0.87	0.00
YUE20031	143	0.90	141	150	9	0.90	0.90	0.01	138	150	12	0.89	0.90	0.01
YUE20032	434	0.77	433	455	22	0.75	0.77	0.02	423	455	32	0.74	0.77	0.03

Finally, Table 6.8 compares the quality metric or quality indicator of the Pareto solution set generated with the two scenarios. Again, it confirms that Pareto scanning by adding any new solution generated by the chosen low-level heuristic in each iteration to the Pareto solution set results a better Pareto set than by adding only solutions which are accepted the under move acceptance mechanism, i.e. great deluge algorithm, to the Pareto solution set.

Table 6.6: The solution with the best standard objective function values in the Pareto solution set.

P. Instance	Init. Solution		Scenario 1 (S1)				Scenario 2 (S2)			
	STD.OF	AJFI	STD.OF	d (%)	AJFI	d(%)	STD.OF	d (%)	AJFI	d(%)
NOTT-I	11923	0.79	11746	**-1.48**	0.79	**-0.18**	11715	**-1.74**	0.78	**-1.53**
NOTT-II-32	0	0.00	0	0.00	0.00	0.00	0	0.00	0.00	0.00
NOTT-II-31	0	0.00	0	0.00	0.00	0.00	0	0.00	0.00	0.00
NOTT-II-30	0	0.00	0	0.00	0.00	0.00	0	0.00	0.00	0.00
NOTT-II-26	8	1.00	8	0.00	1.00	0.00	8	0.00	1.00	0.00
NOTT-II-23	68	0.99	68	0.00	0.99	0.00	68	0.00	0.99	0.00
CAR91	5.30	0.68	5.29	-0.15	0.68	0.04	5.30	**-0.11**	0.67	**-0.44**
CAR92	4.51	0.71	4.49	**-0.36**	0.71	**-0.06**	4.47	**-0.90**	0.71	**-0.14**
EAR83	36.73	0.17	36.04	**-1.87**	0.17	**-2.27**	36.36	-1.00	0.17	0.57
HEC92	10.91	0.51	10.90	-0.10	0.51	-0.15	10.72	**-1.72**	0.51	**-0.38**
KFU93	14.36	0.45	14.36	-0.02	0.45	0.08	14.35	**-0.04**	0.45	**-0.02**
LSE91	11.02	0.49	10.94	-0.68	0.48	-1.60	10.92	**-0.93**	0.49	**-0.42**
PUR93	5.03	0.67		NA				NA		
RYE92	9.01	0.63	9.01	0.00	0.63	0.00	8.99	-0.15	0.63	0.09
STA83	157.12	0.10	157.12	0.00	0.10	0.00	157.12	0.00	0.10	0.01
TRE92	8.75	0.56	8.75	0.00	0.56	0.00	8.69	**-0.68**	0.56	**-1.17**
UTA92	3.60	0.77	3.60	**-0.03**	0.77	**-0.16**	3.60	0.00	0.77	0.00
UTE92	25.20	0.21	25.10	**-0.38**	0.20	**-2.34**	25.06	**-0.57**	0.21	**-1.06**
YOR83	38.03	0.25	38.03	0.00	0.25	0.23	37.40	**-1.65**	0.25	**-1.29**
EXAM1	6856	0.82	6749	-1.56	0.82	0.70	6721	-1.97	0.82	0.28
EXAM2	632	1.00	597	**-5.54**	1.00	**-0.01**		NA		
EXAM3	11659	0.93		NA				NA		
EXAM4	16325	0.73	16273	**-0.32**	0.73	**-0.14**	16324	**-0.01**	0.73	**-0.02**
EXAM5	3837	0.86		NA				NA		
EXAM6	27370	0.79	27355	-0.05	0.79	0.00	27365	**-0.02**	0.79	**-0.03**
EXAM7	5528	0.84	5384	-2.60	0.84	0.23		NA		
EXAM8	9798	0.65	9726	**-0.73**	0.64	**-0.78**	9686	**-1.14**	0.63	**-2.45**
EXAM9	1246	0.75	1242	**-0.32**	0.74	**-0.23**	1199	**-3.77**	0.74	**-0.86**
EXAM10	14556	0.64	14556	0.00	0.64	0.00	14556	0.00	0.64	0.00
EXAM11	36810	0.91		NA				NA		
EXAM12	5300	0.88	5286	-0.26	0.88	0.02	5286	-0.26	0.88	0.02
YUE20011	56	0.92	51	-8.93	0.92	0.40	51	-8.93	0.92	0.40
YUE20012	122	0.86	118	-3.28	0.86	0.71	121	**-0.82**	0.86	**-0.01**
YUE20013	29	0.88	29	0.00	0.88	0.00	29	0.00	0.88	0.00
YUE20021	76	0.92	76	0.00	0.92	-0.19	76	0.00	0.92	-0.19
YUE20022	162	0.87	150	**-7.41**	0.87	**-0.02**	151	-6.79	0.88	0.41
YUE20023	56	0.89	56	0.00	0.87	-2.13	56	0.00	0.87	-2.13
YUE20031	143	0.90	141	**-1.40**	0.90	**-0.04**	138	-3.50	0.90	0.12
YUE20032	434	0.77	433	**-0.23**	0.77	**-0.26**	423	-2.53	0.77	0.06

Table 6.7: The best solution in terms of global fairness function values in the Pareto solution set.

P. Instance	Init. Solution		Scenario 1 (S1)				Scenario 2 (S2)			
	STD.OF	AJFI	STD.OF	d (%)	AJFI	d(%)	STD.OF	d (%)	AJFI	d(%)
NOTT-I	11923	0.79	12512	4.94	0.77	-2.62	12468	4.57	0.75	**-5.03**
NOTT-II-32	0	0.00	0	0.00	0.00	0.00	0	0.00	0.00	0.00
NOTT-II-31	0	0.00	0	0.00	0.00	0.00	0	0.00	0.00	0.00
NOTT-II-30	0	0.00	0	0.00	0.00	0.00	0	0.00	0.00	0.00
NOTT-II-26	8	1.00	8	0.00	1.00	0.00	8	0.00	1.00	0.00
NOTT-II-23	68	0.99	71	4.41	0.99	-0.04	71	4.41	0.99	-0.06
CAR91	5.30	0.68	5.44	2.54	0.67	-1.72	5.55	4.67	0.66	-3.03
CAR92	4.51	0.71	4.63	2.75	0.70	-1.48	4.73	4.92	0.69	-3.09
EAR83	36.73	0.17	38.53	4.92	0.15	**-15.23**	38.49	4.79	0.14	**-17.11**
HEC92	10.91	0.51	11.38	4.30	0.48	**-5.99**	11.45	4.95	0.46	**-9.50**
KFU93	14.36	0.45	14.97	4.27	0.43	-4.33	15.04	4.72	0.41	**-8.79**
LSE91	11.02	0.49	11.56	4.92	0.45	**-8.14**	11.56	4.94	0.43	**-11.29**
PUR93	5.03	0.67		NA				NA		
RYE92	9.01	0.63	9.42	4.62	0.61	-2.99	9.45	4.95	0.60	-4.48
STA83	157.12	0.10	164.84	4.91	0.07	**-31.11**	164.95	4.98	0.07	**-31.35**
TRE92	8.75	0.56	9.14	4.56	0.53	**-6.91**	9.17	4.82	0.53	**-6.73**
UTA92	3.60	0.77	3.74	3.96	0.76	-1.13	3.78	4.96	0.75	-2.24
UTE92	25.20	0.21	26.45	4.96	0.18	**-14.26**	26.30	4.38	0.17	**-19.47**
YOR83	38.03	0.25	39.82	4.71	0.22	**-10.87**	39.72	4.43	0.21	**-16.89**
EXAM1	6856	0.82	7184	4.78	0.80	-2.49	7194	4.93	0.78	-4.42
EXAM2	632	1.00	663	4.91	1.00	-0.16		NA		
EXAM3	11659	0.93		NA				NA		
EXAM4	16325	0.73	17129.00	4.92	0.72	-2.09	17122	4.88	0.71	-3.81
EXAM5	3837	0.86		NA				NA		
EXAM6	27370	0.79	28680	4.79	0.78	-0.38	28710	4.90	0.78	-0.72
EXAM7	5528	0.84	5740	3.84	0.82	-1.67		NA		
EXAM8	9798	0.65	10168	3.78	0.62	-3.86	10272	4.84	0.59	**-8.63**
EXAM9	1246	0.75	1293	3.77	0.71	-5.00	1304	4.65	0.68	**-8.77**
EXAM10	14556	0.64	15277	4.95	0.62	-3.86	15276	4.95	0.60	**-6.85**
EXAM11	36810	0.91		NA				NA		
EXAM12	5300	0.88	5532	4.38	0.88	-0.60	5565	5.00	0.88	-0.92
YUE20011	56	0.92	58	3.57	0.91	-0.95	58	3.57	0.91	-0.95
YUE20012	122	0.86	128	4.92	0.85	-1.46	128	4.92	0.84	-2.29
YUE20013	29	0.88	30	3.45	0.88	-0.48	30	3.45	0.88	-0.48
YUE20021	76	0.92	79	3.95	0.92	-0.58	79	3.95	0.92	-0.58
YUE20022	162	0.87	168	3.70	0.86	-1.81	170	4.94	0.85	-2.05
YUE20023	56	0.89	58	3.57	0.87	-2.66	58	3.57	0.87	-2.66
YUE20031	143	0.90	150	4.90	0.90	-0.65	150	4.90	0.89	-1.52
YUE20032	434	0.77	455	4.84	0.75	-2.28	455	4.84	0.74	-3.33

Table 6.8: The quality metric: coverage, hypervolume, and Spread of the Pareto solution set.

Dataset	P. Instance	Coverage		Hypervolume		Spread	
		$\mathcal{C}$(S1,S2)	$\mathcal{C}$(S2,S1)	S1	S2	S1	S2
NOTT	NOTT-I	0.00	**1.00**	0.07	**0.55**	1.09	**0.76**
	NOTT-II-32	**1.00**	**1.00**	**0.00**	**0.00**	**1.00**	**1.00**
	NOTT-II-31	**1.00**	**1.00**	**0.00**	**0.00**	**1.00**	**1.00**
	NOTT-II-30	**1.00**	**1.00**	**0.00**	**0.00**	**1.00**	**1.00**
	NOTT-II-26	**1.00**	**1.00**	**0.00**	**0.00**	**1.00**	**1.00**
	NOTT-II-23	0.25	**1.00**	0.22	**0.45**	**0.23**	**0.23**
CARTER	CAR-S-91	0.00	**0.88**	0.35	**0.66**	1.18	**0.83**
	CAR-F-92	0.00	**1.00**	0.32	**0.54**	0.98	**0.81**
	EAR-F-83	**0.88**	0.01	**0.69**	0.47	1.22	**0.89**
	HEC-S-92	0.00	**1.00**	0.27	**0.62**	0.77	**0.66**
	KFU-S-93	0.00	**1.00**	0.26	**0.50**	1.26	**1.06**
	LSE-F-91	0.00	**1.00**	0.36	**0.64**	1.17	**0.96**
	PUR-S-93			NA			
	RYE-S-93	0.00	**1.00**	0.35	**0.56**	1.11	**0.77**
	STA-F-83	0.01	**0.98**	0.52	**0.55**	0.88	**0.83**
	TRE-S-92	0.06	**0.85**	0.47	**0.55**	1.00	**0.82**
	UTA-S-92	0.16	**0.75**	0.33	**0.54**	1.12	**0.89**
	UTE-S-92	0.11	**0.86**	0.39	**0.58**	1.04	**0.85**
	YOR-F-83	0.00	**1.00**	0.23	**0.63**	0.98	**0.91**
ITC 2007	EXAM1	0.00	**1.00**	0.23	**0.54**	0.90	**0.76**
	EXAM2			NA			
	EXAM3			NA			
	EXAM4	**0.38**	0.34	0.35	**0.43**	1.03	**0.70**
	EXAM5			NA			
	EXAM6	0.00	**0.96**	0.28	**0.53**	0.72	**0.54**
	EXAM7			NA			
	EXAM8	0.00	**1.00**	0.06	**0.52**	0.95	**0.64**
	EXAM9	0.00	**1.00**	0.16	**0.58**	0.84	**0.50**
	EXAM10	0.00	**0.99**	0.27	**0.65**	1.05	**0.94**
	EXAM11			NA			
	EXAM12	0.11	**0.72**	0.46	0.61	0.97	0.45
YEDITEPE	YUE20011	**1.00**	**1.00**	**0.43**	**0.43**	**0.00**	**0.00**
	YUE20012	0.38	**0.70**	0.32	**0.35**	**0.36**	0.53
	YUE20013	**1.00**	**1.00**	**0.00**	**0.00**	**0.00**	**0.00**
	YUE20021	**1.00**	**1.00**	**0.33**	**0.33**	**0.00**	**0.00**
	YUE20022	**0.82**	0.21	**0.40**	0.36	0.39	0.37
	YUE20023	**1.00**	**1.00**	**0.25**	**0.25**	**0.00**	**0.00**
	YUE20031	0.00	**1.00**	0.19	**0.48**	0.61	**0.29**
	YUE20032	0.00	**1.00**	0.30	**0.50**	0.61	**0.27**

6.3.2 Comparing Reinforcement Learning and Self Adaptive learning as Low-level heuristic Selection Strategies

With the best parameter settings from previous experiment in Section 6.3.1 summarised in Table 6.9, this section evaluates a different low-level heuristic strategy, namely, self adaptive learning shown by Algorithm 12, instead of reinforcement learning (see Scenario in Table 6.10).

Algorithm 12 Self Adaptive - Great Deluge (SA-GD) Based Hyper-heuristics Hyper-heuristic With Scalarisation Technique for Multi-objective Examination Timetabling Problems. Refer to Table 6.1 and Table 6.2 for detailed explanation of the methods.

```
1: procedure SOLVE(ProblemDomain P)
2:     Set number of iteration N
3:     Set stopping condition St
4:     for i← 1, N do
5:         nof ← P.getNumberOfObjectives()
6:         w ← genRandVec(nof)
7:         P.setWeightVector(w)
8:         P.initialiseSolution(0)
9:         of ← P.problem.getFunctionValues(0)
10:        ref ← genRefPoint(nof, of)
11:        P.setReferenceVector(ref)
12:        P.addSolutionToParetoSet(0)
13:        //set boundary level
14:        B ← P.getWeightedFunctionValue(0)
15:        Set decay rate α
16:        Set M ← P.getNumberOfHeuristics()
17:        Set a list of low-level heuristic indices, LH of size S
18:        Set a list of Well-performed low-level heuristic indices, WH
19:        // Fill LH with the index of low-level heuristics predefined in P chosen randomly
20:        for i=0 to S-1 do
21:            LH.add(getRand(M))
22:        end for
23:        while (St is not met) do
24:            if LH is not empty then
25:                // get the earliest element of LH and delete it from LH
26:                ι ← LH.poll()
27:            else
28:                fill 75% elements of LH with elements from WH
29:                fill 25% elements of LH with the index of low-level heuristics predefined in P chosen randomly.
30:                Clear the element of WH
31:                ι ← LH.poll()
32:            end if
33:            newFunctionVal ← P.applyLLH(ι,0,1)
34:            if newFunctionVal≤ S_curr OR newFunctionVal ≤ B then
35:                //the new solution is accepted
36:                P.copySolution(1,0)
37:                P.addSolutionToParetoSet(1)
38:                WH.add(ι)
39:            end if
40:            B ← B-α
41:        end while
42:    end for
43:    P.sortParetoSet()
44:    P_a ← P.getParetoSet()
45:    //return Pareto Solution
46:    return P_a
47: end procedure
```

The comparison of the Pareto solution set from both strategies is given in Table 6.11. Table 6.11 suggests that on average, in terms of coverage, hypervolume and spread as quality indicators, reinforcement learning outperforms self-adaptive learning.

Table 6.9: The parameter settings for the experiments to compare reinforcement learning and self adaptive learning, over bi-objective examination timetabling problem with fairness.

Paremeter	Value
Pareto Scanning	In each iteration, any solution, which is generated by applying the chosen low-level heuristic, if its standard objective function value not worsened more than 5% from the initial solution is added to Pareto solution set.
Other parameters	The same as in Table 6.3

Table 6.10: Experimental Scenarios of Low-level heuristic (LLH) selection strategy

Scenario	LLH selection Strategy
S1	Self adaptive learning, as shown by Algorithm 12.
S2	Reinforcement learning, as shown by Algorithm 11.

6.3.3 Comparing Two Different Scalarisation Techniques : Weighted Tchebycheff and Weighted Summation

Further, with the best parameter settings from previous sections summarised in Table 6.12, we compared the two scalarisation techniques, i.e. weighted sum and weighted Tchebycheff (see Scenario given in Table 6.13).

The comparison of the Pareto solution set from the experiments of the hyper-heuristic given in Algorithm 11 with weighted sum and weighted Tchebycheff over the problem instances are given in Table 6.14. Table 6.14 suggests that on overage, in terms of coverage, hypervolume, and spread, 'weighted summation' outperforms 'weighted Tchebycheff'.

6.3.4 Comparing Reinforcement Learning and Simple Random as Low-level Selection Strategy

Lastly, with the best parameter settings from previous experiment given in Table 6.15, we compared reinforcement learning with simple random as low-level selection strategies (see Scenarios given in Table 6.16). The experimental results are given Table 6.17. From Table 6.17, we can observe that between reinforcement learning and simple random are very competitive in which, in terms of coverage, hypervolume and spread reinforcement learning is slightly better than simple random.

Table 6.11: Coverage, hypervolume, and Spread of the Pareto solution set from the experiments of a hyper-heuristic with reinforcement learning (S1) and self-adaptive learning (S2) using 'Weighted Tchebycheff'.

Dataset	P. Instance	Coverage		Hypervolume		Spread	
		$\mathcal{C}$(S1,S2)	$\mathcal{C}$(S2,S1)	S1	S2	S1	S2
NOTT	NOTT-I	0.00	**1.00**	0.39	**0.55**	0.97	**0.76**
	NOTT-II-32	1.00	1.00	0.00	0.00	1.00	1.00
	NOTT-II-31	1.00	1.00	0.00	0.00	1.00	1.00
	NOTT-II-30	1.00	1.00	0.00	0.00	1.00	1.00
	NOTT-II-26	1.00	1.00	0.00	0.00	1.00	1.00
	NOTT-II-23	1.00	1.00	0.45	0.45	0.23	0.23
CARTER	CAR-S-91	**0.74**	0.13	**0.59**	0.57	1.04	**0.82**
	CAR-F-92	0.00	**1.00**	0.39	**0.54**	1.04	**0.81**
	EAR-F-83	0.00	**1.00**	0.50	**0.61**	0.92	**0.89**
	HEC-S-92	0.00	**1.00**	0.50	**0.62**	0.66	0.66
	KFU-S-93	0.07	**0.97**	0.48	**0.50**	1.06	1.06
	LSE-F-91	0.00	**1.00**	0.33	**0.64**	**0.89**	0.96
	PUR-S-93			NA			
	RYE-S-93	0.00	**0.98**	0.51	**0.56**	0.79	**0.77**
	STA-F-83	**0.60**	0.27	0.55	0.55	**0.82**	0.83
	TRE-S-92	0.00	**1.00**	0.45	**0.58**	1.01	**0.82**
	UTA-S-92	**0.45**	0.43	0.54	**0.58**	1.03	**0.89**
	UTE-S-92	0.00	**1.00**	0.45	**0.58**	0.97	**0.85**
	YOR-F-83	0.00	**1.00**	0.33	**0.63**	1.04	**0.91**
ITC 2007	EXAM1	**0.76**	0.10	**0.55**	0.54	0.79	**0.76**
	EXAM2			NA			
	EXAM3			NA			
	EXAM4	**0.58**	0.37	**0.49**	0.47	**0.67**	0.69
	EXAM5			NA			
	EXAM6	0.00	**1**	0.45	**0.52**	**0.39**	0.52
	EXAM7			NA			
	EXAM8	0.00	**1**	0.16	**0.52**	0.86	**0.64**
	EXAM9	0.00	**1**	0.50	**0.58**	0.63	**0.50**
	EXAM10	0.01	**0.97**	0.44	**0.65**	0.99	**0.94**
	EXAM11			NA			
	EXAM12	0.24	**0.73**	0.60	0.61	0.61	**0.45**
YEDITEPE	YUE20011	0.88	**1.00**	0.43	0.43	0.14	**0.00**
	YUE20012	0.50	**1.00**	0.29	**0.34**	0.34	0.34
	YUE20013	1.00	1.00	0.00	0.00	0.00	0.00
	YUE20021	1.00	0.50	**0.43**	0.33	**0.17**	0.61
	YUE20022	0.00	**1.00**	0.25	**0.47**	0.70	**0.28**
	YUE20023	0.00	**1.00**	0.00	**0.25**	0.85	**0.00**
	YUE20031	0.00	**1.00**	0.25	**0.48**	0.57	**0.29**
	YUE20032	0.13	**0.97**	0.46	**0.51**	**0.11**	0.30

Table 6.12: The parameter settings for the experiments to compare 'weighted sum' and 'weighted Tchebycheff'.

Paremeter	Value
Pareto Scanning	In each iteration, any solution, which is generated by applying the chosen low-level heuristic, if its standard objective function value not worsened more than 5% from the initial solution is added to Pareto solution set.
LLH Selection	Reinforcement Learning.
Other parameters	The same as in Table 6.3

Table 6.13: Experimental Scenarios Comparing two Scalarisation Techniques

Scenario	Scalarisation Technique
S1	Weighted Summation (WS).
S2	Weighted Tchebycheff (WT).

6.4 Experimental Results Over The Multi-objective Optimisation Examination Timetabling Problem with Different Notions of Fairness

From the preliminary experiments discussed in previous section, i.e. Section 6.3, we obtain the best parameters settings and algorithm strategies for our proposed hyper-heuristic algorithms over bi-objective examination timetabling problem: standard objective function and global fairness, as summarised in Table 6.18. Though these can only be good default and maybe would need to re-tune for each of the notions, but this would be too expensive and long. Therefore, with this parameter settings we conduct experiment over five variants of multi-objective examination timetabling problem with different notions of fairness:

1. Bi-objective functions: standard objective function and global unfairness objective function
2. Bi-objective functions: standard objective function and 'specified cohort unfairness' objective function.
3. Bi-objective functions: standard objective function and 'exam-count-based cohort' unfairness objective function.
4. Three objective functions: standard objective function, global unfairness objective function and 'specified cohort unfairness' objective function.
5. Three objective functions: standard objective function, global unfairness objective function and 'exam-count-based cohort unfairness' objective function.

Table 6.14: Coverage, hypervolume and Spread of the Pareto solution set from the experiments using 'Weighted Sum' (S1) and 'Weighted Tchebycheff' (S2).

Dataset	P. Instance	Coverage		Hypervolume		Spread	
		C(S1,S2)	C(S2,S1)	S1	S2	S1	S2
NOTT	NOTT-I	**1**	0	**0.53**	0.11	0.96	**0.86**
	NOTT-II-32	1.00	1.00	0.00	0.00	1.00	1.00
	NOTT-II-31	1.00	1.00	0.00	0.00	1.00	1.00
	NOTT-II-30	1.00	1.00	0.00	0.00	1.00	1.00
	NOTT-II-26	1.00	1.00	0.00	0.00	1.00	1.00
	NOTT-II-23	1.00	1.00	0.45	0.45	0.23	0.23
CARTER	CAR-S-91	**1**	0	**0.57**	0.12	**0.79**	0.90
	CAR-F-92	**0.84**	0	**0.63**	0.34	**0.83**	0.86
	EAR-F-83	**0.72**	0.00	**0.59**	0.47	**0.84**	0.93
	HEC-S-92	0.00	**1.00**	0.40	**0.62**	0.73	**0.66**
	KFU-S-93	**0.94**	0.02	**0.59**	0.49	1.25	**1.06**
	LSE-F-91	0.13	**0.86**	0.51	**0.60**	**0.84**	0.96
	PUR-S-93			NA			
	RYE-S-93	**0.76**	0.05	**0.58**	0.39	**0.76**	0.82
	STA-F-83	0.29	**0.48**	0.54	0.54	**0.82**	0.83
	TRE-S-92	**0.76**	0.00	**0.61**	0.43	**0.77**	0.85
	UTA-S-92	**1.00**	0.00	**0.61**	0.25	**0.90**	0.92
	UTE-S-92	0.30	**0.33**	**0.56**	0.52	**0.78**	0.87
	YOR-F-83	0.00	**1.00**	0.38	**0.63**	**0.90**	0.91
ITC 2007	EXAM1	**0.84**	0.14	**0.55**	0.39	0.81	**0.79**
	EXAM2			NA			
	EXAM3			NA			
	EXAM4	**1.00**	0.00	**0.50**	0.44	0.72	**0.70**
	EXAM5			NA			
	EXAM6	0.14	**0.66**	0.51	**0.52**	**0.43**	0.57
	EXAM7			NA			
	EXAM8	**1.00**	0.00	**0.52**	0.23	**0.67**	0.76
	EXAM9	**0.52**	0.32	0.50	0.50	0.63	**0.54**
	EXAM10	**1.00**	0.00	**0.56**	0.44	**0.69**	0.97
	EXAM11			NA			
	EXAM12	**0.99**	0.03	**0.59**	0.56	**0.43**	0.50
YEDITEPE	YUE20011	1.00	1.00	0.43	0.43	0.00	0.00
	YUE20012	**1.00**	0.00	**0.43**	0.11	**0.14**	0.61
	YUE20013	1.00	1.00	0.00	0.00	0.00	0.00
	YUE20021	1.00	1.00	0.33	0.33	0.00	0.00
	YUE20022	**1.00**	0.00	**0.47**	0.21	**0.00**	0.50
	YUE20023	0.00	**1.00**	0.00	**0.25**	0.85	**0.00**
	YUE20031	**1.00**	0.77	**0.51**	0.48	0.29	0.29
	YUE20032	**1.00**	0.00	**0.48**	0.42	**0.20**	0.40

Table 6.15: The parameter settings for the experiments to compare reinforcement learning and simple random.

Paremeter	Value
Pareto Scanning	In each iteration, any solution, which is generated by applying the chosen low-level heuristic, if its standard objective function value not worsened more than 5% from the initial solution is added to Pareto solution set.
Scalarisation technique	Weighted Summation.

Table 6.16: Experimental Scenarios Comparing Reinforcement Learning and Simple Random

Scenario	Scalarisation Technique
S1	Reinforcement Learning (RL).
S2	Simple Random (SR).

Table 6.17: Coverage, hypervolume, and Spread of the Pareto solution set from the experiments of a hyper-heuristic with reinforcement learning (S1) and simple random (S2) using 'Weighted Summation'.

Dataset	P. Instance	Coverage		Hypervolume		Spread	
		C(S1,S2)	C(S2,S1)	S1	S2	S1	S2
NOTT	NOTT-I	**0.72**	0.06	**0.61**	0.47	**0.96**	1.01
	NOTT-II-32	1.00	1.00	0.00	0.00	1.00	1.00
	NOTT-II-31	1.00	1.00	0.00	0.00	1.00	1.00
	NOTT-II-30	1.00	1.00	0.00	0.00	1.00	1.00
	NOTT-II-26	1.00	1.00	0.00	0.00	1.00	1.00
	NOTT-II-23	1.00	1.00	0.45	0.45	0.23	0.23
CARTER	CAR-S-91	**1**	0	**0.57**	0.42	**0.79**	0.97
	CAR-F-92	0.37	**0.54**	**0.54**	0.53	0.82	**0.77**
	EAR-F-83	0.02	**0.86**	0.48	**0.58**	**0.83**	0.99
	HEC-S-92	0.00	**1.00**	0.38	**0.55**	0.72	**0.61**
	KFU-S-93	0.00	**1.00**	0.42	**0.50**	**1.22**	1.26
	LSE-F-91	0.42	**0.48**	**0.55**	0.53	**0.82**	1.09
	PUR-S-93			NA			
	RYE-S-93	**0.79**	0.14	**0.54**	0.53	0.77	**0.65**
	STA-F-83	**0.75**	0.15	**0.54**	0.53	**0.82**	0.96
	TRE-S-92	**0.91**	0.04	**0.61**	0.56	**0.74**	0.91
	UTA-S-92	**1.00**	0.00	**0.61**	0.50	**0.90**	0.91
	UTE-S-92	**0.72**	0.12	**0.54**	0.52	**0.76**	0.86
	YOR-F-83	0.01	**0.93**	0.55	**0.69**	**0.88**	0.93
ITC 2007	EXAM1	0.00	**1.00**	0.41	**0.52**	0.84	**0.69**
	EXAM2			NA			
	EXAM3			NA			
	EXAM4	0.41	**0.66**	0.49	0.49	**0.73**	0.77
	EXAM5			NA			
	EXAM6	0.33	**0.57**	0.51	**0.53**	**0.43**	0.47
	EXAM7			NA			
	EXAM8	**0.87**	0.03	**0.52**	0.48	**0.67**	0.77
	EXAM9	**0.60**	0.32	**0.50**	0.49	0.63	**0.32**
	EXAM10	**0.68**	0.18	**0.56**	0.43	**0.70**	0.91
	EXAM11			NA			
	EXAM12	**0.59**	0.43	0.52	**0.54**	0.48	**0.37**
YEDITEPE	YUE20011	1.00	1.00	0.43	0.43	0.00	0.00
	YUE20012	**0.73**	0.38	**0.50**	0.41	0.46	**0.36**
	YUE20013	1.00	1.00	0.00	0.00	0.00	0.00
	YUE20021	1.00	1.00	0.33	0.33	0.00	0.00
	YUE20022	0.00	**1.00**	0.31	**0.48**	0.34	**0.28**
	YUE20023	1.00	1.00	0.25	0.25	0.00	0.00
	YUE20031	1.00	1.00	0.51	0.51	0.29	0.29
	YUE20032	0.00	1.00	0.40	**0.55**	0.43	**0.20**

The experiments were conducted over four datasets, i.e. Nottingham, Carter [89], ITC 2007 [205], Yeditepe [233, 241]. In the following sections, only the experimental results over four arbitrarily chosen problem instances, i.e. `STA-F-83`, `EXAM4`, `EXAM9` and `YUE20011` are presented. The complete experimental results over all problem instances are given in Appendix F. Note that within this experiment, as discussed in the earlier sections, the standard objective functions are prevented from getting worse by more than 5% from the initial solutions. The d(%) values in the following tables presenting the experimental results are the percentage of the respective objective function values' change from the initial solutions values.

Table 6.18: The best parameter settings for experiments over bi-objective examination timetabling problem: (1) Standard objective function, (2) Global fairness using Algorithm 11

Paremeter	Value
Pareto Scanning	In each iteration, any solution, which is generated by applying the chosen low-level heuristic, if its standard objective function value not worsened more than 5% from the initial solution is added to Pareto solution set.
Scalarisation technique	Weighted Summation.
LLH Selection	Reinforcement learning.
Other parameters	The same as in Table 6.3

6.4.1 Experimental results over multi-objective examination timetabling problem with bi-objective functions: standard objective function and global unfairness objective function

From the experiments of a hyper-heuristic given in Algorithm 11 using 'Weighted Summation' as the scalarisation technique under the parameter settings given in Table 6.18 tested, Table 6.19 presents the range of objective functions value of Pareto solution set from the experiments over multi-objective examination timetabling with two objectives, i.e. standard objective function and global unfairness, which is measured by anti Jain fairness index (AJFI). Accordingly, Table 6.20 presents the objective functions value of the solutions in the Pareto solution set with the best individual objective function value, i.e. the solution with the best standard objective function value and the solution with the best global unfairness function value. Lastly, Figure 6.1 visualises the trade-off between the standard objective function and global fairness, of solutions in the Pareto solution set.

From Table 6.19 and Table 6.20 we can observe that by preventing the standard objective from getting worse more than 5%, over all the problem instances, the global fairness

Table 6.19: The objective function values range of the Pareto solution set from the experiments over the bi-objective problem with global unfairness.

P. Instance	Init Solution		STD.OF			Global Unfairness		
	STD.OF	AJFI	min	max	range	min	max	range
STA-F-83	157.12	0.10	157.12	164.97	7.85	0.07	0.10	0.03
EXAM4	16325	0.73	16287	17141	854	0.70	0.73	0.03
EXAM9	1246	0.75	1221	1308	87	0.67	0.74	0.07
YUE20011	56	0.92	51	58	7	0.91	0.92	0.01

Table 6.20: The objective function values of the solution with the best standard objective function and the best global fairness in the Pareto solution set from the experiments over the bi-objective problem with global unfairness.

P. Instance	Init. Solution		Sol. With Best STD.OF				Sol. With Best Global Fairness			
			STD.OF		AJFI		STD.OF		AJFI	
	STD.OF	AJFI	Val.	d (%)	Val.	d(%)	Val.	d (%)	Val.	d(%)
STA-F-83	157.12	0.10	157.12	0.00	0.10	0.00	164.97	4.99	0.07	-31.49
EXAM4	16325	0.73	16287	-0.23	0.73	-0.11	17141.00	5.00	0.70	-4.01
EXAM9	1246	0.75	1221	-2.01	0.74	-1.30	1308	4.98	0.67	-10.32
YUE20011	56	0.92	51	-8.93	0.92	0.40	58	3.57	0.91	-0.95

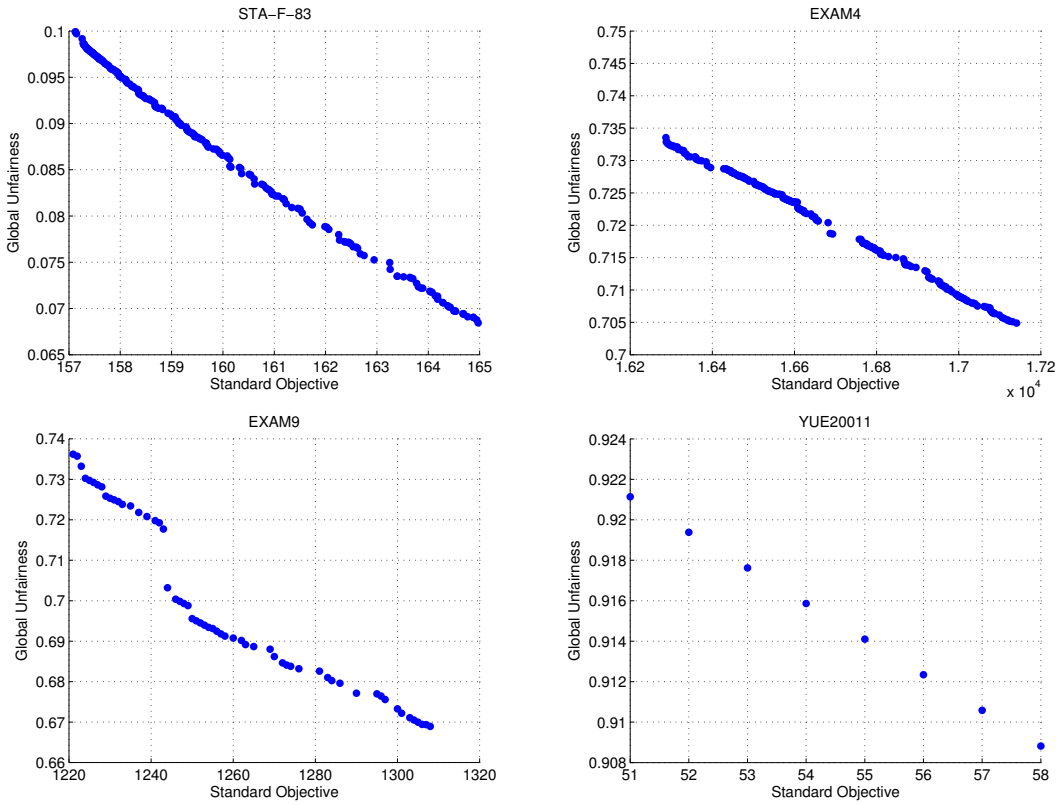


Figure 6.1: The scatter plot of the Pareto solution set from the experiments over the bi-objective problem with global unfairness.

could be improved. However, the significance of the improvement varies amongst different problem instances. For instance, over `STA-F-83`, the global fairness can be improved by `31.49%` whereas over `YUE20011` only by `0.95%`. From Table 6.20, we can also observe that over problem instance `EXAM4` and `EXAM9`, both standard objective function and global fairness could be improved slightly in which within two-stage approach, it was failed to be improved (see Table 5.4). The trade-off between standard objective function and global fairness of each individual within Pareto solution set could be evaluated from scatter plot presented in Figure 6.1. It is obvious that by giving a set of Pareto solutions instead of single solution as in Chapter 5, the decision maker could have more options to be considered as final decision.

6.4.2 Experimental results over multi-objective examination timetabling problem with bi-objective functions: standard objective function and 'specified cohort unfairness' objective function

Instead of global fairness, in this section another notion of fairness, i.e. 'specified cohort fairness' as discussed in earlier chapters, is taken as the second objective function. From the experiments of a hyper-heuristic given in Algorithm 11 using 'Weighted Summation' as the scalarisation technique under the parameter settings given in Table 6.18, Table 6.21 presents the range of objective functions value of Pareto solution set from the experiments over multi-objective examination timetabling with two objectives, i.e. standard objective function and 'specified cohort' unfairness, which is measured by averaged anti Jain fairness index within specified cohort(AJFIC1). Further, Table 6.22 presents the objective functions value of the solutions in the Pareto solution set with the best individual objective function value, i.e. the solution with the best standard objective function value and the solution with the best 'specified cohort unfairness' function value. Lastly, Figure 6.2 visualises the trade-off between the standard objective function and 'specified cohort unfairness', of solutions in the Pareto solution set.

Table 6.21: The objective function values range of the Pareto solution set from the experiments over the bi-objective problem with 'specified cohort unfairness'.

P. Instance	Init Solution		STD.OF			Specified Cohort Unfairness		
	STD.OF	AJFIC1	min	max	range	min	max	range
STA-F-83	157.12	0.00	157.12	161.54	4.41	0.0005	0.0043	0.0038
EXAM4	16325	0.78	16056	17136	1080	0.56	0.79	0.22
EXAM9	1246	0.68	1241	1304	63	0.58	0.66	0.08
YUE20011	56	0.90	54	58	4	0.89	0.90	0.01

Table 6.22: The objective function values of the solution with the best standard objective function and the solution with the best global fairness in the Pareto solution set from the experiments over the bi-objective problem with 'specified cohort unfairness'.

P. Instance	Init Solution		Sol. With Best STD.OF				Sol. With Best Spec. Cohort Unfairness			
			STD.OF		AJFIC1		STD.OF		AJFIC1	
	STD.OF	AJFIC1	Val.	d (%)	Val.	d(%)	Val.	d (%)	Val.	d(%)
STA-F-83	157.12	0.0043	157.12	0.00	0.0043	0.00	161.54	2.81	0.0005	-88.47
EXAM4	16325	0.78	16056	-1.65	0.79	0.47	17136	4.97	0.56	-28.26
EXAM9	1246	0.68	1241	-0.40	0.66	-3.28	1304	4.65	0.58	-14.49
YUE20011	56	0.90	54	-3.57	0.90	0.07	58	3.57	0.89	-1.39

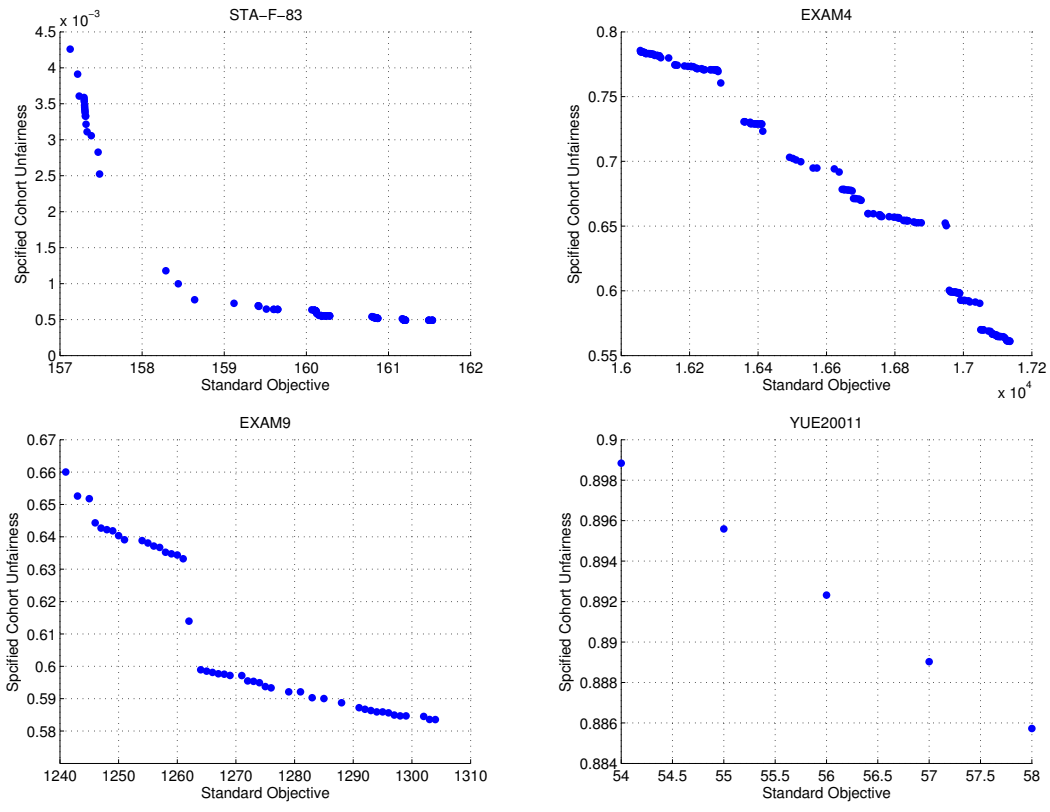


Figure 6.2: The scatter plot: standard objective function and 'specified cohort unfairness'.

From Table 6.21 and Table Table 6.22, we can observe a key result that 'specified cohort fairness' over all the problem instances could be improved much more significantly than the global fairness. For example, over problem instance `EXAM9`, the 'specified cohort fairness' could be improved by `14.49%` whereas its global fairness could be improved by only `5.00%`. From Table 6.21 we can also observe that the 'specified cohort fairness' in Problem instance `STA-F-83` is almost perfectly fair already. It is shown by its specified cohort unfairness value that is less than `0.00`. Moreover, from Figure 6.2, we can see that over problem instances `STA-F-83`, `EXAM4`, `EXAM9` the pattern of the trade-off between standard objective function and 'specified cohort unfairness' is different from the pattern of the the trade-off between standard objective function and 'global unfairness' shown in Figure 6.1. It suggests that between standard objective function and 'specified cohort unfairness' is not so strongly conflicting as standard objective function and global unfairness.

6.4.3 Experimental results over multi-objective examination timetabling problem with bi-objective functions: standard objective function and 'exam-count-based cohort' unfairness objective function

Within the experiments reported in this section another notion of fairness, namely, 'exam-count-based unfairness' is taken as the second objective. From the experiments of a hyper-heuristic given in Algorithm 11 using 'Weighted Summation' as the scalarisation technique under the parameter settings given in Table 6.18, Table 6.23 presents the range of objective functions value of Pareto solution set from the experiments over multi-objective examination timetabling with two objectives, i.e. standard objective function and 'exam-count-based cohort' (ECB) unfairness, which is measured by averaged anti Jain fairness index within exam-count-based cohort(AJFC2). Accordingly, Table 6.24 presents the objective functions value of the solutions in the Pareto solution set with the best individual objective function value, i.e. the solution with the best standard objective function value and the solution with the best 'exam-count-based cohort unfairness' function value. Lastly, Figure 6.3 visualises the trade-off between the standard objective function and exam-count-based cohort unfairness', of solutions in the Pareto solution set.

From Table 6.23 and TableTable 6.24, we can observe that 'exam-count-based cohort' fairness could also be improved much more significantly than the global fairness. Even, over the majority of the Problem instances, the 'exam-count-based cohort fairness' could

Table 6.23: The objective function values range of the Pareto solution set from the experiments over the bi-objective problem with 'exam-count-based cohort' unfairness.

P. Instance	Init Solution		STD.OF			ECB Cohort Unfairness		
	STD.OF	AJFI.C2	min	max	range	min	max	range
STA-F-83	157.12	0.0021	157.12	161.96	4.83	0.0003	0.0021	0.0018
EXAM4	16325	0.50	16325	17134	809	0.43	0.50	0.07
EXAM9	1246	0.24	1241	1302	61	0.18	0.22	0.04
YUE20011	56	0.45	54	58	4	0.41	0.45	0.04

Table 6.24: The objective function values of the solution with the best standard objective function and the best global fairness in the Pareto solution set from the experiments over the bi-objective problem with 'exam-count-based cohort' unfairness.

P. Instance	Init Solution		Sol. With Best STD.OF				Sol. With Best ECB Cohort Unfairness			
			STD.OF		AJFIC2		STD.OF		AJFIC2	
	STD.OF	AJFIC2	Val.	d (%)	Val.	d(%)	Val.	d (%)	Val.	d(%)
STA-F-83	157.12	0.0021	157.1243863	0.00	0.0021	0.00	161.96	3.07	0.0003	-84.57
EXAM4	16325	0.50	16325	0.00	0.50	0.00	17134.00	4.96	0.43	-14.63
EXAM9	1246	0.24	1241	-0.40	0.22	-8.12	1302	4.49	0.18	-26.31
YUE20011	56	0.45	54	-3.57	0.45	0.27	58	3.57	0.41	-9.54

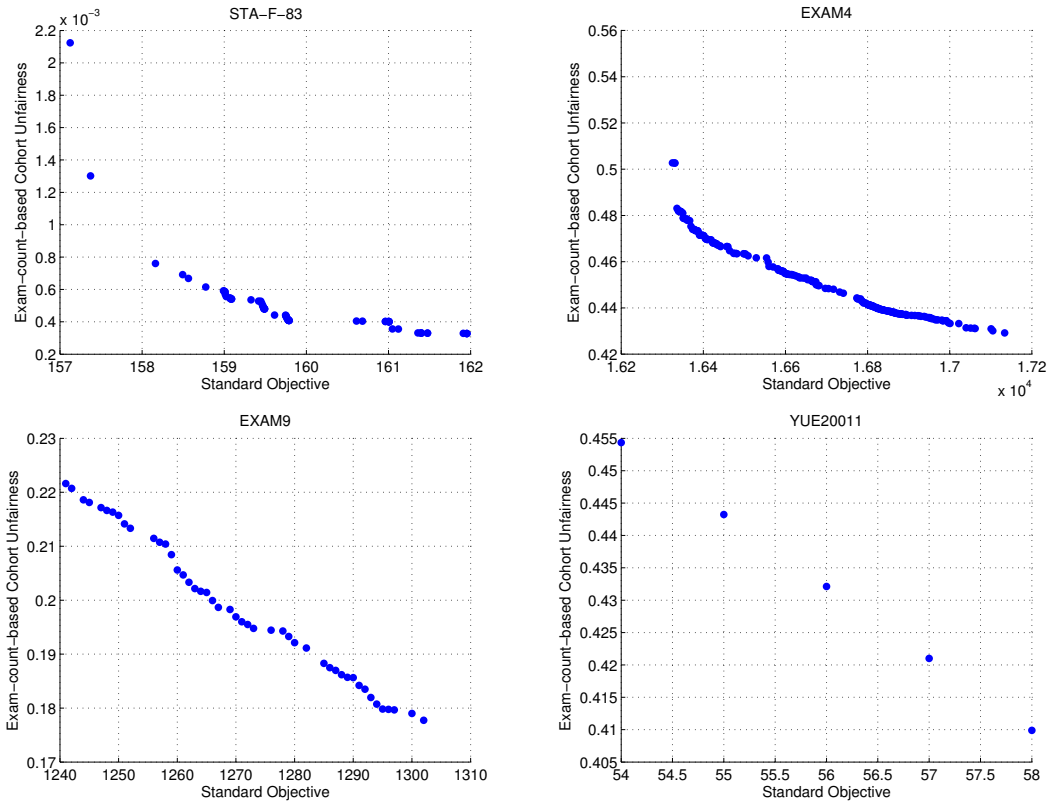


Figure 6.3: The scatter plot: standard objective function and 'exam-count-based cohort' unfairness.

be improved more significantly than 'specified cohort fairness'. For example, over problem instance `EXAM9`, in terms of global fairness, 'specified cohort fairness', and 'exam-count-based fairness' the fairness could be improved by `10.32%, 14.49%, and 26.31%` respectively.

6.4.4 Experimental results over multi-objective examination timetabling problem with three objective functions: standard objective function, global unfairness objective function and 'specified cohort unfairness' objective function

From the experiments of a hyper-heuristic given in Algorithm 11 using 'Weighted Summation' as the scalarisation technique under the parameter settings given in Table 6.18, Table 6.25 presents the range of objective functions value of Pareto solution set from the experiments over multi-objective examination timetabling with three objectives, i.e. standard objective function, global fairness, and 'specified cohort' unfairness, which is measured by averaged anti Jain fairness index within specified cohort(AJFIC1). Accordingly, Tables 6.26 - 6.28 presents the objective functions value of the solutions in the Pareto solution set with the best individual objective function value, i.e. the solution with the best standard objective function value, the solution with best global unfairness, and the solution with the best 'specified cohort unfairness' function value respectively. Lastly, Figure 6.4 visualises the trade-off between the standard objective function and global fairness, and 'specified cohort unfairness' of solutions in the Pareto solution set with Parallel coordinates.

Table 6.25: The objective function values range of the Pareto solution set from the experiments over three-objective examination timetabling problem: standard objective function, global unfairness and 'specified cohort unfairness'.

P. Instance	Init Solution			STD.OF			Global Fairness			Spec. Cohort Unfairness		
	STD.OF	AJFI	AJFIC1	min	max	range	min	max	range	min	max	range
STA-F-83	157.12	0.10	0.0043	157.12	164.97	7.84	0.07	0.10	0.03	0.0012	0.0043	0.0031
EXAM4	16325	0.73	0.78	16325	17130	805.00	0.71	0.73	0.03	0.68	0.78	0.10
EXAM9	1246	0.75	0.68	1246	1308	62.00	0.71	0.75	0.04	0.63	0.68	0.06
YUE20011	56	0.92	0.90	54	58	4.00	0.91	0.92	0.01	0.88	0.89	0.01

From Table 6.25 - 6.28, we can observe that the trade-off between standard objective function, global unfairness, and 'specified cohort unfairness' objective function varies amongst different problem instances. Generally, similar to over the bi-objective examination timetabling problem, over the three-objective examination timetabling problem, 'specified cohort fairness' could be improved more significantly than the global fairness. In addition, as shown by Table 6.26, over problem instance `YUE20011` the three objective functions could be improved concurrently.

Table 6.26: The objective function values of the solution with the best standard objective function in the Pareto solution set from the experiments over the three-objective problem.

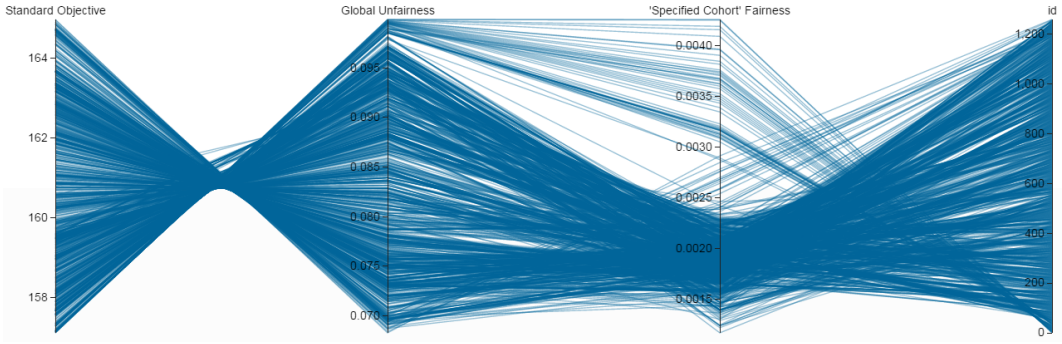
P. Instance	Init. Solution			Sol. With Best STD.OF					
				STD.OF		AJFI		AJFIC1	
	STD.OF	AJFI	AJFIC1	Val.	d (%)	Val.	d(%)	Val.	d(%)
STA-F-83	157.12	0.10	0.0043	157.12	0.00	0.10	0.00	0.0043	0.00
EXAM4	16325	0.73	0.78	16325	0.00	0.73	0.00	0.78	0.00
EXAM9	1246	0.75	0.68	1246	0.00	0.75	0.00	0.68	0.00
YUE20011	56	0.92	0.90	54	-3.57	0.92	-0.18	0.89	-0.50

Table 6.27: The objective function values of the solution with the best global fairness in the Pareto solution set from the experiments over the three-objective problem.

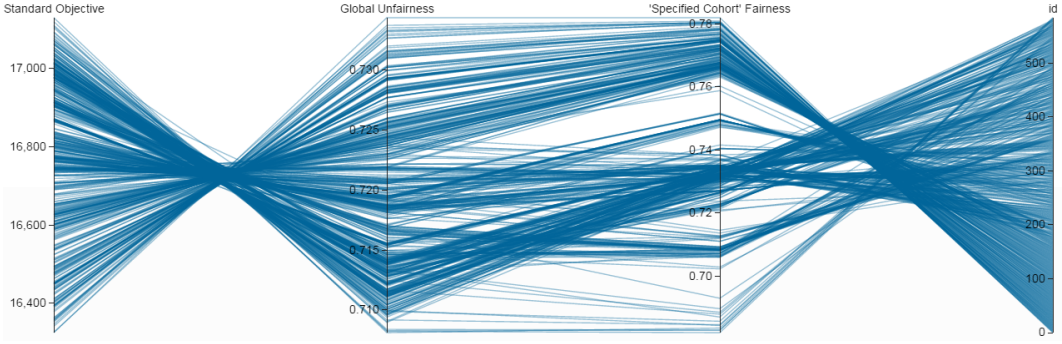
P. Instance	Init. Solution			Sol. With Best Global Unfairness					
				STD.OF		AJFI		AJFIC1	
	STD.OF	AJFI	AJFIC1	Val.	d (%)	Val.	d(%)	Val.	d(%)
STA-F-83	157.12	0.10	0.0043	164.97	4.99	0.07	-31.69	0.0029	-31.61
EXAM4	16325	0.73	0.78	17130	4.93	0.71	-3.57	0.68	-12.59
EXAM9	1246	0.75	0.68	1308	4.98	0.71	-5.42	0.63	-7.09
YUE20011	56	0.92	0.90	58	3.57	0.91	-0.95	0.88	-2.06

Table 6.28: The objective function values of the solution with the best specified cohort fairness in the Pareto solution set from the experiments over the three-objective problem.

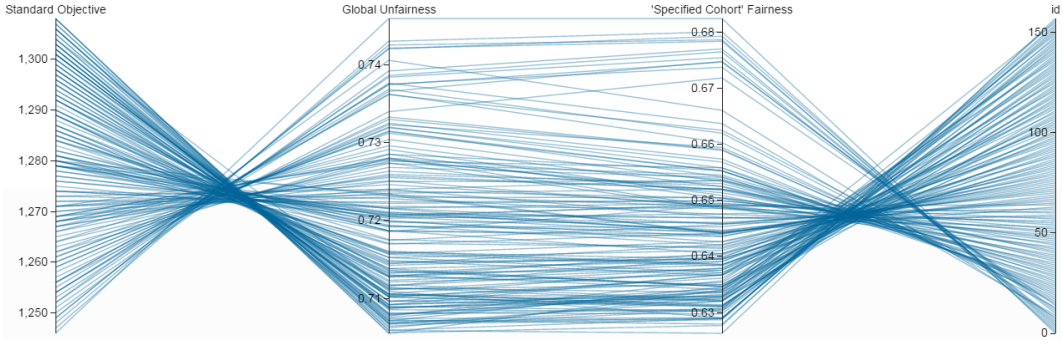
P. Instance	Init. Solution			Sol. With Best Specified Cohort Unfairness					
				STD.OF		AJFI		AJFIC1	
	STD.OF	AJFI	AJFIC1	Val.	d (%)	Val.	d(%)	Val.	d(%)
STA-F-83	157.12	0.10	0.0043	159.62	1.59	0.09	-6.51	0.0012	-72.62
EXAM4	16325	0.73	0.78	17120	4.87	0.71	-3.56	0.68	-12.77
EXAM9	1246	0.75	0.68	1308	4.98	0.71	-5.37	0.63	-8.23
YUE20011	56	0.92	0.90	58	3.57	0.91	-0.95	0.88	-2.06



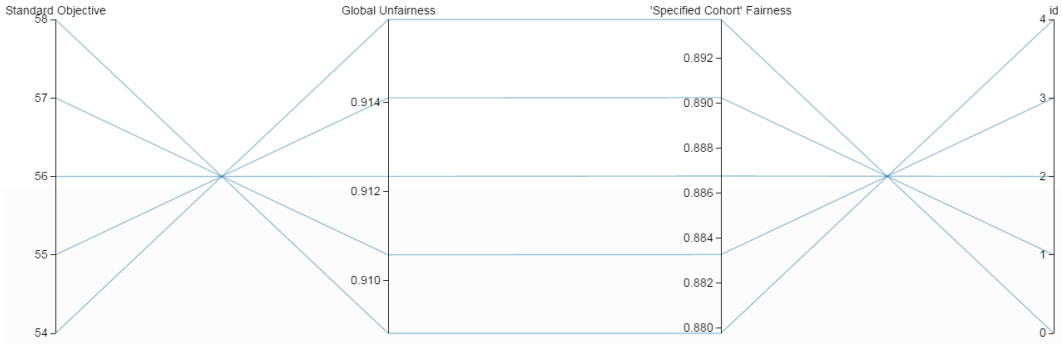
(a) STA-F-83



(b) EXAM4



(c) EXAM9



(d) YUE20011

Figure 6.4: Parallel coordinate of the Pareto solution set from the experiments over the three-objective problem.

Further, from Figure 6.4, we can observe that that there is obvious reverse-correlation between standard penalty and global unfairness, i.e. AJFIA. In this sense, decreasing the standard penalty will increase unfairness. However, the correlation between global unfairness and specified cohort unfairness is not so obvious. It varies amongst different problem instances. For example, over problem instance `YUE20011`, global fairness and 'specified cohort fairness' are positively correlated whereas over the other problem instances it is not so obvious.

From users or decision makers point of view, they most probably prefer solution that in term of standard penalty slightly worse than the best one, has reasonable overall fairness, but still have very good fairness within a cohort. For illustration, Figure 6.5 and Figure 6.6 visualise different options of selected solutions from Pareto solution set in Problem instance `STA-F-83` and `EXAM9` respectively. Accordingly, the objective function values are given in Table 6.29 and Table 6.31, whereas the change of the objective function values from `Solution 0`, the initial solution, to the other solutions are given in Table 6.30 and Table 6.32. Moreover, Figure 6.7 and Figure 6.8 depicts the penalty associated to each individual student.

Over problem instance `STA-F-83`, the decision maker may prefer `Solution 534` to `Solution 0`, in which the global fairness and 'specified cohort fairness' could be improved by around `6.5%` and `73%` respectively with compensation on worsening the standard objective function by only around `1.6%` only. Whereas over Problem instance `EXAM9`, the decision maker may prefer `Solution 22`, in which the specified cohort fairness and global fairness could be improved by `3.90%` and `1.87%` respectively with compensation on worsening the standard objective function value by `1.12%` only. How the improvement of fairness affect penalties associated to each individual students over problem instance `STA-F-83` and `EXAM9` could be observed in Figure 6.7 and Figure 6.8.

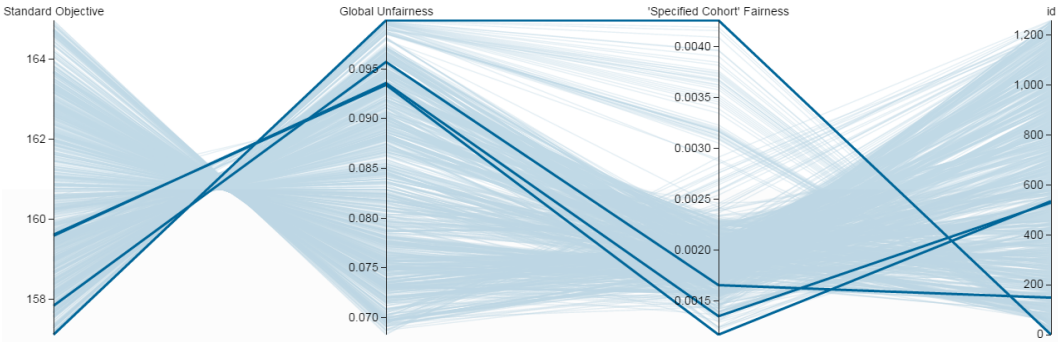
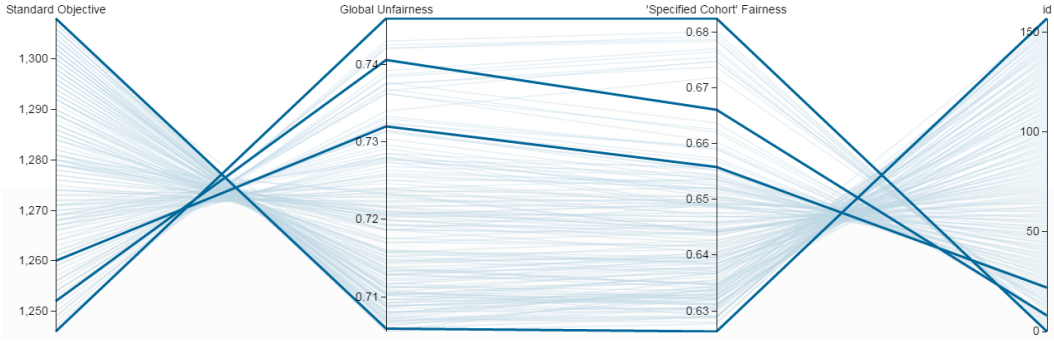


Figure 6.5: Parallel coordinate of the selected solution in the Pareto solution set from the experiments over the three-objective problem of problem instance `STA-F-83`.



Solution 0, 8, 22, and 157

Figure 6.6: The parallel coordinate of problem instance `EXAM9`.

Table 6.29: The objective function values of the selected solution over problem instance `STA-F-83` from the Pareto solution set given in Figure 6.5

Solution #	Obj. Function Values		
	STD.OF	AJFI	AJFIC1
0	157.1244	0.0999	0.0043
148	157.8462	0.0957	0.0017
528	159.5908	0.0936	0.0013
534	159.6187	0.0934	0.0012

Table 6.30: The changes in objective function values (Δ) from solution `0` to the other solutions of selected solutions over problem instance `STA-F-83` given in Table 6.29

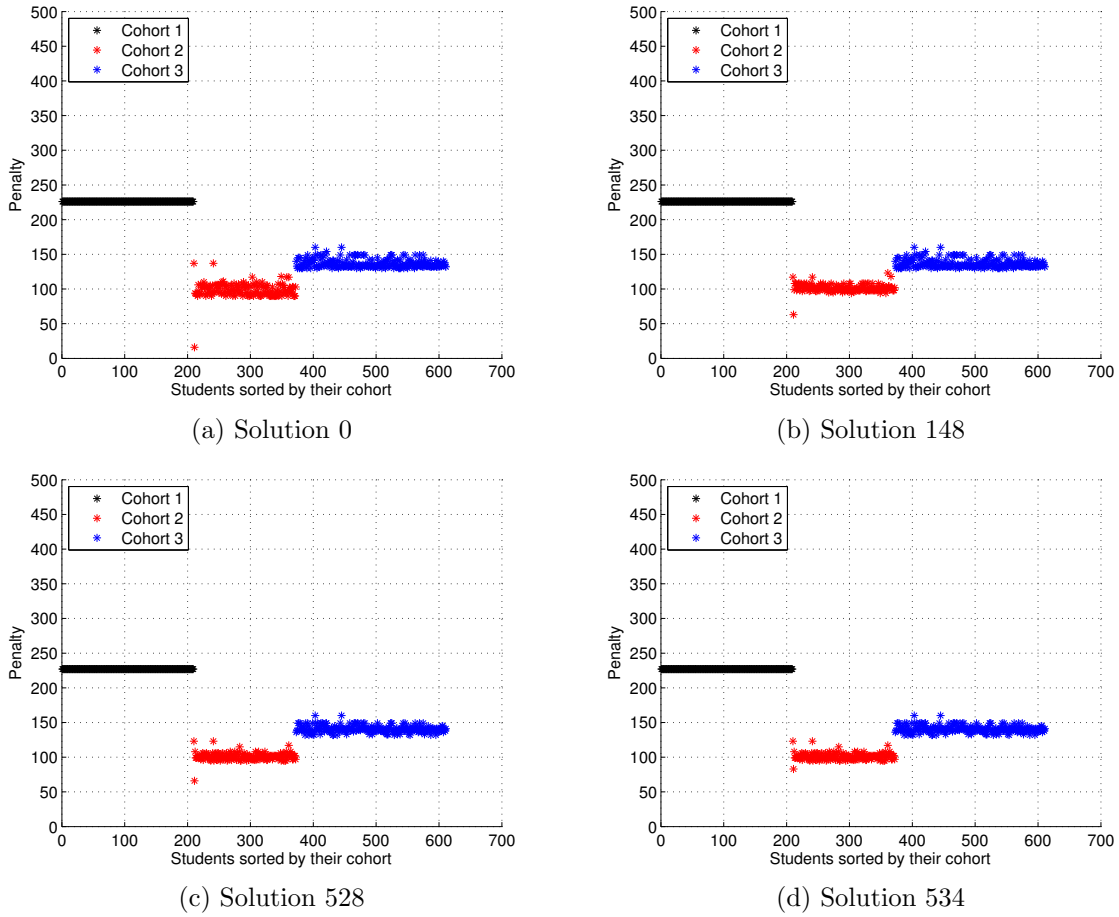
Solution #	Delta, Δ (%) of Obj. Function Values		
	Δ (%) STD.OF	Δ (%) AJFI	Δ (%) AJFIC1
148	0.4594	-4.1948	-61.1781
528	1.5697	-6.3254	-68.3584
534	1.5875	-6.5079	-72.6167

Table 6.31: The Objective Function Values of selected solution over problem instance `EXAM9` from Pareto solution set given in Figure 6.6

Solution #	Obj. Function Values		
	STD.OF	AJFI	AJFIC1
0	1246	0.75	0.68
8	1252	0.74	0.67
22	1260	0.73	0.66
157	1308	0.71	0.63

Table 6.32: The changes in objective function values (Δ) from solution `0` to the other solutions of selected solutions over problem instance `EXAM9` given in Table 6.31

Solution #	Delta, Δ (%) of Obj. Function Values		
	Δ (%) STD.OF	Δ (%) AJFI	Δ (%) AJFIC1
8	0.48	-0.72	-2.40
22	1.12	-1.87	-3.90
157	4.98	-5.37	-8.23



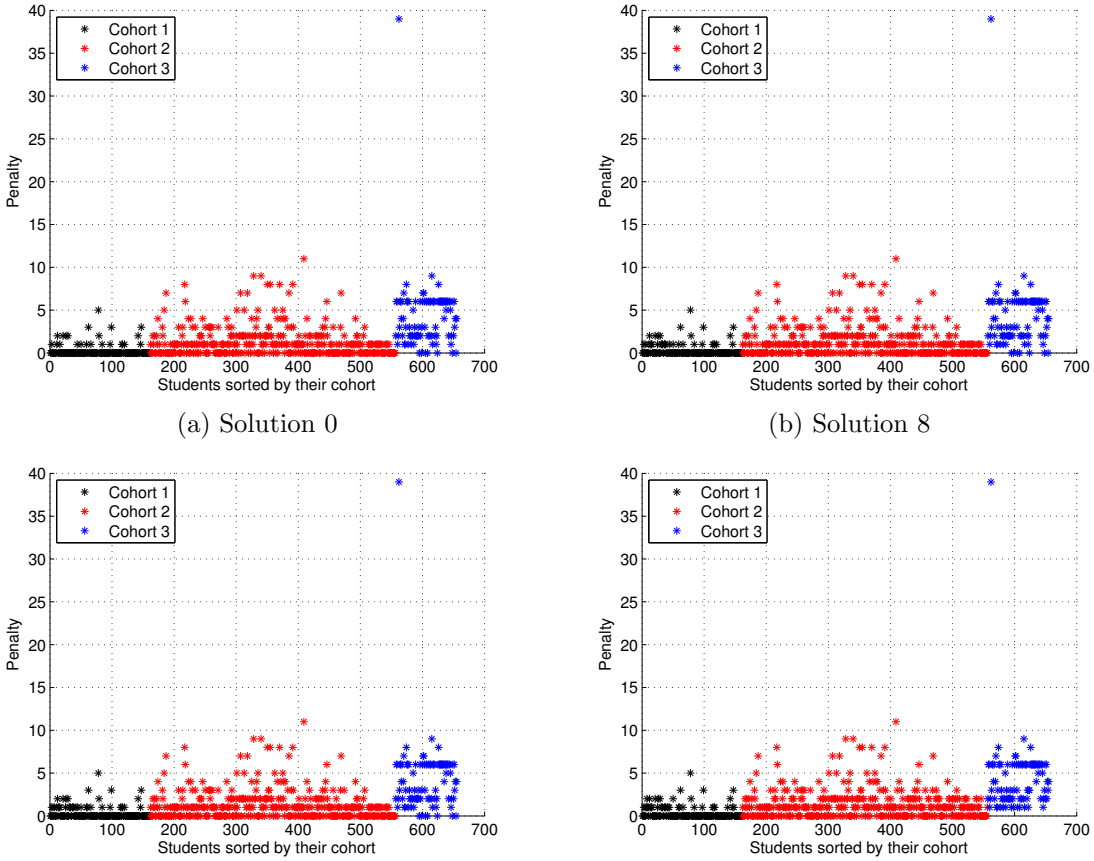
(a) Solution 0

(b) Solution 148

(c) Solution 528

(d) Solution 534

Figure 6.7: Penalty associated to each individual students of selected solution from the Pareto solution set with the three-objective functions over problem instance `STA-F-83`.



(a) Solution 0

(b) Solution 8

(c) Solution 22

(d) Solution 157

Figure 6.8: The penalty associated to each individual students over problem instance `EXAM9`.

From Figure 6.7, we can observe that in cohort 1 the situation is totally fair already. We can also observe that, from `Solution 0` to `Solution 534`, the fairness within cohort 2 and cohort 3 still could be improved by making penalties over some students slightly better and in the same time penalties over the other students slightly worse. It can be pointed out from the figure that in Solution `Solution 0` there are two students with penalty significantly above the average, then in other solutions to make fairer, the penalties of these two students become closer to the average. It also can be pointed out that in cohort 2 from `Solution 0`, there is an outlier, i.e. a student with penalty significantly lower than the average, then in other solutions it is forced to be closed to the average.

On the other hand, over problem instance `EXAM9` as shown by Figure 6.8, a quite large number of students got zero penalty. Further, from `Solution 0` to `Solution 22`, the fairness within a specified cohort could be improved by making penalty over students with zero penalty slightly worse.

6.4.5 Experimental results over multi-objective examination timetabling problem with three objective functions: standard objective function, global unfairness objective function and 'exam-count-based cohort unfairness' objective function

From the experiments of a hyper-heuristic given in Algorithm 11 using 'Weighted Summation' as the scalarisation technique under the parameter settings given in Table 6.18, Table 6.33 presents the range of objective functions value of Pareto solution set from the experiments over multi-objective examination timetabling with three objectives, i.e. standard objective function, global unfairness and 'exam-count-based cohort' unfairness, which is measured by averaged anti Jain fairness index within specified cohort(AJFIC2). Accordingly, Tables 6.34 - 6.36 present the objective functions value of the solutions in the Pareto solution set with the best individual objective function value, i.e. the solution with the best standard objective function value, the solution with the best global unfairness, and the solution with the best 'exam-count-based cohort unfairness' function value. Lastly, Figure 6.9 visualises the trade-off between the standard objective function, global unfairness, and 'exam-count-based cohort unfairness' of solutions in the Pareto solution set with parallel coordinate.

From Table 6.33- 6.36, we can observe that generally, the pattern is similar over multi-objective examination timetabling problem with three objective functions: standard objective function, global unfairness objective function and 'specified cohort unfairness'

Table 6.33: The objective function values range of the Pareto solution set from the experiments over three-objective examination timetabling problem: standard objective function, global unfairness and 'exam-count-based cohort unfairness'.

P. Instance	Init Solution			STD.OF			Global Fairness			ECB Cohort Unfairness		
	STD.OF	AJFI	AJFIC1	min	max	range	min	max	range	min	max	range
STA-F-83	157.12	0.10	0.0021	157.12	164.96	7.84	0.07	0.10	0.03	0.0006	0.0021	0.0015
EXAM4	16325	0.73	0.50	16325	17137	812.00	0.71	0.73	0.03	0.44	0.51	0.06
EXAM9	1246	0.75	0.24	1246	1308	62.00	0.70	0.75	0.04	0.19	0.24	0.05
YUE20011	56	0.92	0.45	56	58	2.00	0.91	0.92	0.01	0.43	0.45	0.02

Table 6.34: The objective function values of the solution with the best standard objective function in the Pareto solution set from the experiments over the three-objective problem with 'exam-count-based cohort unfairness'.

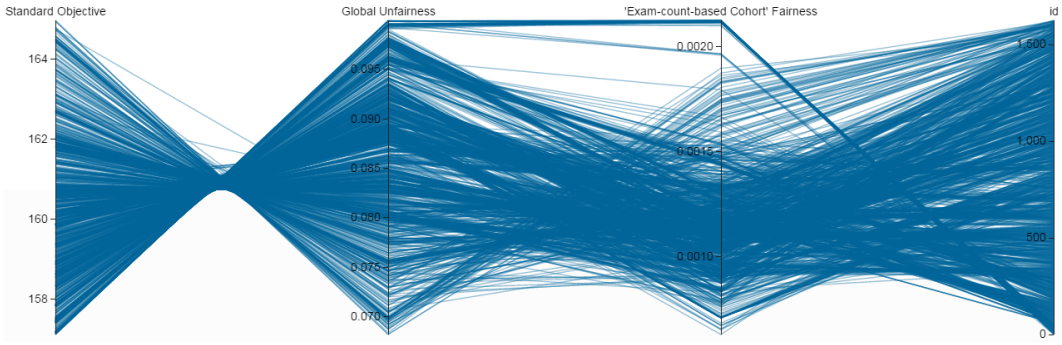
P. Instance	Init. Solution			Sol. With Best STD.OF					
				STD.OF		AJFI		AJFIC2	
	STD.OF	AJFI	AJFIC2	Val.	d (%)	Val.	d(%)	Val.	d(%)
STA-F-83	157.12	0.10	0.0021	157.12	0.00	0.10	0.04	0.0020	-7.54
EXAM4	16325	0.73	0.50	16325	0.00	0.73	0.00	0.50	0.00
EXAM9	1246	0.75	0.24	1246	0.00	0.75	0.00	0.24	0.00
YUE20011	56	0.92	0.45	56	0.00	0.92	0.00	0.45	-0.23

Table 6.35: The objective function values of the solution with the best global fairness in the Pareto solution set from from the experiments over the three-objective problem with 'exam-count-based cohort unfairness'.

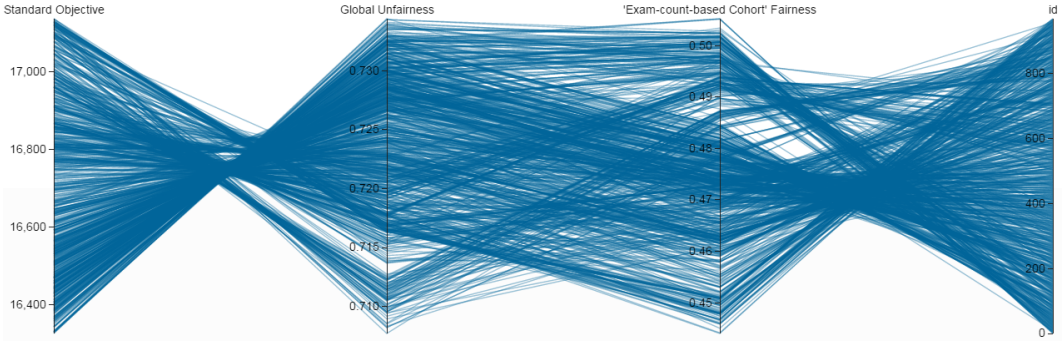
P. Instance	Init. Solution			Sol. With Best Global Unfairness					
				STD.OF		AJFI		AJFIC2	
	STD.OF	AJFI	AJFIC2	Val.	d (%)	Val.	d(%)	Val.	d(%)
STA-F-83	157.12	0.10	0.0021	164.96	4.99	0.07	-31.74	0.0016	-23.69
EXAM4	16325	0.73	0.50	17128	4.92	0.71	-3.63	0.48	-3.64
EXAM9	1246	0.75	0.24	1307	4.90	0.70	-5.55	0.19	-19.38
YUE20011	56	0.92	0.45	58	3.57	0.91	-0.65	0.44	-3.13

Table 6.36: The objective function values of the solution with the best exam-count-based cohort fairness in the Pareto solution set from the experiments over the three-objective problem with 'exam-count-based cohort unfairness'.

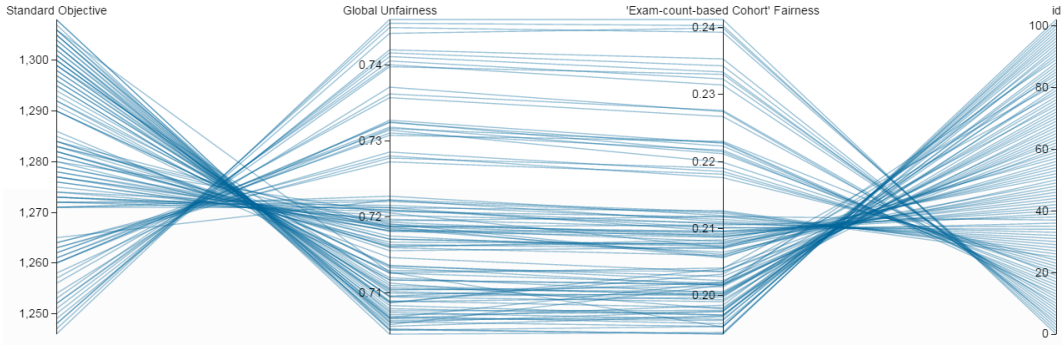
P. Instance	Init. Solution			Sol. With Best ECB Cohort Unfairness					
				STD.OF		AJFI		AJFIC2	
	STD.OF	AJFI	AJFIC2	Val.	d (%)	Val.	d(%)	Val.	d(%)
STA-F-83	157.12	0.10	0.0021	160.89	2.40	0.09	-12.84	0.0006	-70.42
EXAM4	16325	0.73	0.50	17112	4.82	0.72	-2.30	0.44	-11.67
EXAM9	1246	0.75	0.24	1308	4.98	0.71	-5.45	0.19	-19.48
YUE20011	56	0.92	0.45	58	3.57	0.91	-0.38	0.43	-5.13



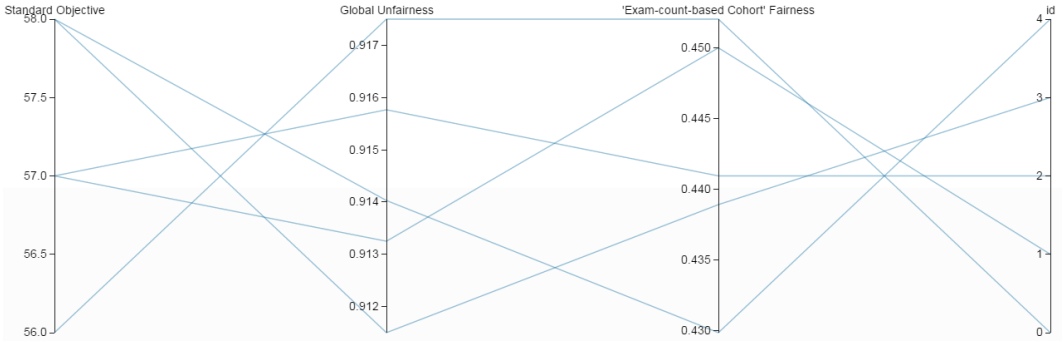
(a) STA-F-83



(b) EXAM4



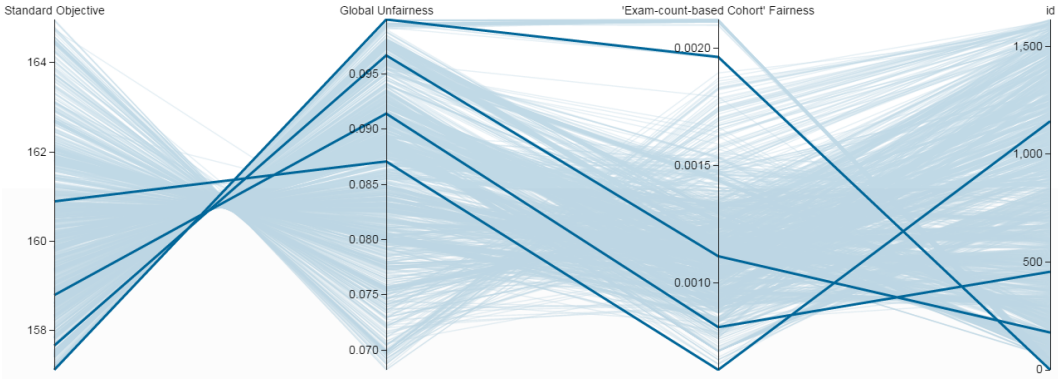
(c) EXAM9



(d) YUE20011

Figure 6.9: Parallel coordinate representation of the Pareto solution set from the experiments over the three-objective problem with 'exam-count-based cohort unfairness' on four selected problem instances.

objective function. However, compared to 'specified cohort unfairness' , the 'exam-count-based cohort unfairness' could be improved more significantly. In addition, over problem instance `YUE20011` that is in the three objective problem with 'specified cohort unfairness' the correlation between global unfairness and specified cohort unfairness is positively correlated, in the the three objective problem with 'exam-count-based cohort unfairness' as shown by Figure 6.9, the correlation between global unfairness and exam-count-based cohort unfairness is not obvious.



Solution 0, 173, 455, and 1153

Figure 6.10: The parallel coordinate for problem instance `STA-F-83`.

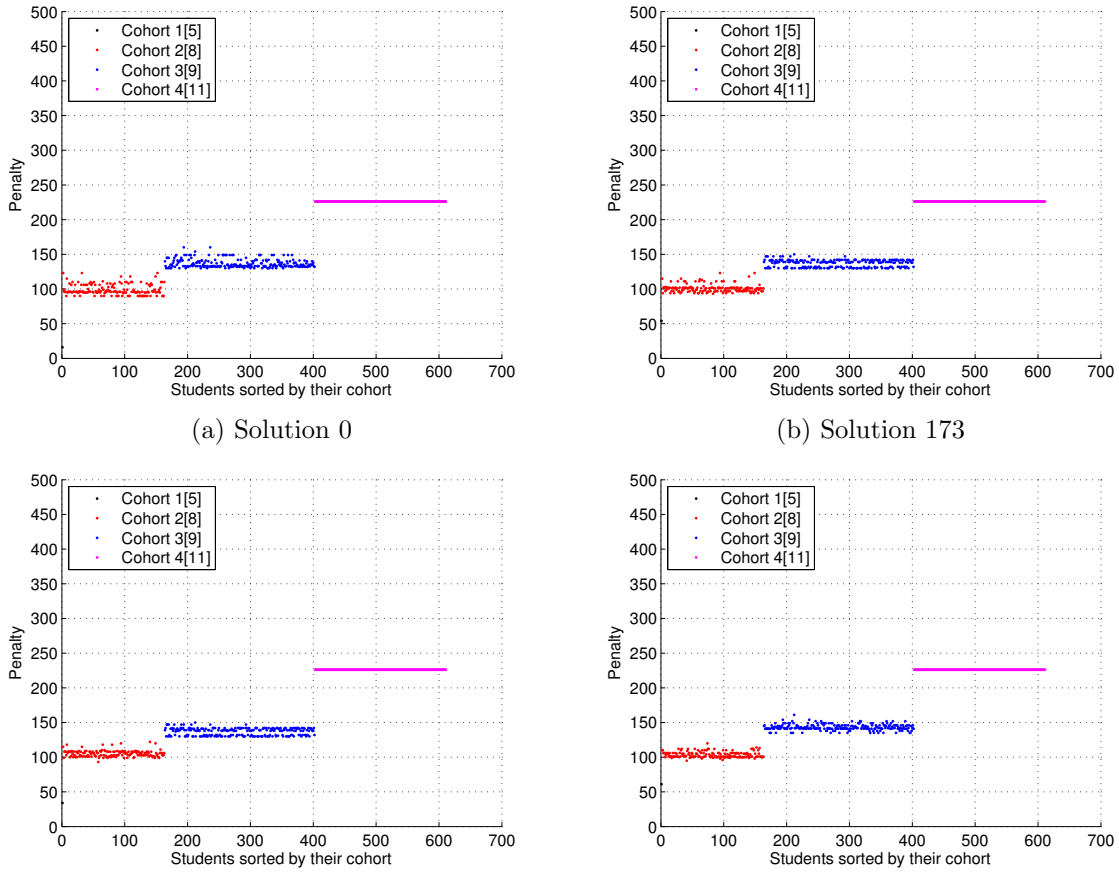
Table 6.37: The objective function values of the selected solution over problem instance `STA-F-83` with the three-objective functions with 'exam-count-based cohort unfairness', from Pareto solution set given in Figure 6.10

Solution #	Obj. Function Values		
	STD.OF	AJFI	AJFIC2
0	157.1211	0.0999	0.0020
173	157.6645	0.0967	0.0011
455	158.7938	0.0914	0.0008
1153	160.8920	0.0871	0.0006

Table 6.38: The changes in objective function values (Δ) from solution `0` to the other solutions of selected solutions over problem instance `STA-F-83` with the three-objective functions with 'exam-count-based cohort unfairness', given in Table 6.37

Solution #	Delta,Δ(%) of Obj. Function Values		
	Δ(%) STD.OF	Δ(%) AJFI	Δ(%) AJFIC2
173	0.3458	-3.2547	-43.3060
455	1.0646	-8.5343	-58.7395
1153	2.4000	-12.8726	-68.0041

As an illustration, Figure 6.10 presents the selected solutions in Pareto solution set over problem instance `STA-F-83`. Given four selected solutions given in Figure 6.10, the decision maker may prefer `Solution 455` to `Solution 0`. In which as shown by Table 6.37 and Table 6.38, the global fairness and 'exam-count-based fairness' could be improved



(a) Solution 0

(b) Solution 173

(c) Solution 455

(d) Solution 1153

Figure 6.11: Penalty associated to each individual students of selected solution from the Pareto solution set with the three-objective functions over problem instance `STA-F-83`.

by around 8.5% and 58.7% respectively with compensation on worsening standard objective function by around 1% only. The penalties associated to each individual student are depicted by Figure 6.11.

6.5 Experimental Results Over The Multi-objective Optimisation Problem with Multiple Stakeholder Perspectives

In addition to multi-objective examination timetabling problem with fairness as discussed in Section 6.4, we also carried out experiment over multi-objective examination timetabling problem with multiple stakeholder perspectives as formulated in Section 4.2.3 of Chapter 4. As discussed in Section 4.2.3 of Chapter 4, there are two variants of the problem:

1. Bi-objective examination timetabling problem, in which the original single objective function examination timetabling problem of ITC 2007 model is divided into students and administrators(see Equation 4.15 in Section 4.2.3 of Chapter 4).
2. Four-objective examination timetabling problem, in which the original single objective function examination timetabling problem of ITC 2007 model is divided into students, markers, invigilators, and estates (see Equation 4.14 in Section 4.2.3 of Chapter 4).

To solve both problems, we employed the same hyper-heuristic approach used over multi-objective examination timetabling with fairness as discussed in in Section 6.4. Four different strategies summarised in Table 6.39 were tested over the problems. The strategies are the combination of two low-level heuristic selection strategies, i.e. reinforcement learning and self-adaptive learning and two different scalarisation techniques, i.e. weighted sum and weighted Tchebychef within great deluge based hyper-heuristic. Revisit Algorithm 11 and Algorithm 12 for great deluge based hyper-heuristic with reinforcement learning and self adaptive learning respectively.

The proposed hyper-heuristic with strategies summarised in Table 6.39 and the parameter settings given in Table 6.40 are tested over our proposed multi-objective version of ITC 2007 dataset. To evaluate the performance of the algorithm with the four strategies, we compared the Pareto solution set produced by each strategy using three quality indicators, i.e. coverage, hypervolume, and spread as shown by Table 6.42 and Table 6.43 for the two objective and four objective problems respectively.

Table 6.39: Four variants of hyper-heuristics scenarios

No	Hyper-heuristic	Description
S1	RL-WSGD-HH	Reinforcement Learning + Weighted sum scalarisation technique + Great Deluge
S2	RL-WTGD-HH	Reinforcement Learning + weighted Tchebycheff scalarisation technique + Great Deluge
S3	SA-WSGD-HH	Self adaptive + Weighted sum scalarisation technique + Great Deluge
S4	SA-WTGD-HH	Self adaptive + weighted Tchebycheff scalarisation technique + Great Deluge

Table 6.40: The parameter settings for experiment with Hyper-heuristics presented in Algorithm 11 and Algorithm 12 over multi-objective examination timetabling problem with multiple stakeholders' perspective.

Paremeter	Value
`Number of iterations (N)`	100
`Stopping condition (St)`	10,000 iterations
Initial Solution	In the first iteration, the initial solution is generated by constructor without improvement phase with Algorithm 6 (See Chapter 3). In the following iteration, the initial solution is a solution randomly chosen from the current Pareto solution set.
`Ref.Points`	`50%` of initial solutions
`Normalisation`	The objective function values are converted to the value between `0-1`.
Pareto Scanning	In each iteration, any solution, which is generated by applying the chosen low-level heuristic, if its standard objective function value not worsened more than 5% from the initial solution is added to Pareto solution set.
`Decay rate` (α)	0.0001

Table 6.41: Coverage Definition given in Table 6.42 and Table 6.43

Variable	Definition
a	Pareto Set generated with RL-WSGD-HH (S1)
b	Pareto Set generated with RL-WTGD-HH (S2)
c	Pareto Set generated with SA-WSGD-HH (S3)
d	Pareto Set generated with SA-WTGD-HH (S4)
$\mathcal{C}$(P1)	$\mathcal{C}$(a,b)+$\mathcal{C}$(a,c)+$\mathcal{C}$(a,d)
$\mathcal{C}$(P2)	$\mathcal{C}$(b,a)+$\mathcal{C}$(b,c)+$\mathcal{C}$(b,d)
$\mathcal{C}$(P3)	$\mathcal{C}$(c,a)+$\mathcal{C}$(c,b)+$\mathcal{C}$(c,d)
$\mathcal{C}$(P4)	$\mathcal{C}$(d,a)+$\mathcal{C}$(d,b)+$\mathcal{C}$(d,c)

Table 6.42: The quality indicators of the Pareto solutions from the experiment of hyper-heuristics with different strategies tested over the bi-objective examination timetabling problem with multiple stakeholder perspective.

P. Instance	Coverage				Hypervolume				Spread			
	$\mathcal{C}$(P1)	$\mathcal{C}$(P2)	$\mathcal{C}$(P3)	$\mathcal{C}$(P4)	S1	S2	S3	S4	S1	S2	S3	S4
EXAM1	**3.00**	0.00	1.80	1.16	**0.71**	0.00	0.13	0.07	**0.70**	0.85	0.86	1.06
EXAM2	**2.93**	0.49	0.00	2.00	**0.97**	0.00	0.00	0.86	0.95	**0.90**	1.11	1.21
EXAM3	**3.00**	0.13	1.90	0.90	**0.86**	0.00	0.37	0.16	**0.67**	0.92	0.98	0.97
EXAM4	**3.00**	1.09	0.62	0.33	**0.93**	0.56	0.44	0.38	0.74	0.98	0.95	**0.83**
EXAM5	**2.21**	0.07	1.45	0.72	**0.83**	0.56	0.70	0.57	**0.80**	1.09	0.89	0.82
EXAM6	0.45	**0.92**	0.68	0.81	0.63	**0.67**	0.57	0.66	0.91	0.78	1.05	**0.78**
EXAM7	2.00	1.00	**3.00**	0.00	0.25	0.00	**0.72**	0.00	1.12	1.17	**0.63**	1.10
EXAM8	1.39	0.00	1.54	**2.94**	**0.91**	0.75	0.87	0.94	1.17	1.23	**0.96**	1.15
EXAM9	2.00	0.71	**2.54**	0.08	0.77	0.00	**0.94**	0.00	1.04	**0.86**	1.04	0.98
EXAM10	2.12	0.59	**2.64**	0.43	0.94	0.75	**0.96**	0.45	0.95	0.97	0.94	**0.90**
EXAM11	1.14	**2.15**	0.88	0.76	0.66	**0.81**	0.59	0.54	**0.83**	1.05	0.90	0.93
EXAM12	1.20	0.20	**2.20**	0.37	0.31	**0.81**	0.43	0.43	0.83	**0.63**	0.85	0.98

Table 6.43: The quality indicators of the Pareto solutions from the experiment of hyper-heuristics with different strategies tested over the four-objective examination timetabling problem with multiple stakeholder perspective.

P. Instance	Coverage				Hypervolume				Spread			
	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4
EXAM1	**1.07**	1.45	2.03	0.01	0.32	**0.39**	**0.39**	0.20	**0.51**	0.76	0.53	**0.51**
EXAM2	1.00	**2.97**	1.97	0.00	0.43	**0.54**	0.48	0.11	0.60	0.58	**0.57**	**0.57**
EXAM3	**3.00**	0.02	0.80	1.21	**0.56**	0.00	0.00	0.00	**0.58**	0.78	0.64	0.81
EXAM4	0.37	1.79	0.41	**2.26**	0.00	0.00	0.00	0.00	NaN	NaN	NaN	NaN
EXAM5	**1.95**	0.00	0.00	1.00	0.00	0.00	0.00	0.00	NaN	NaN	NaN	NaN
EXAM6	2.16	0.15	0.60	**2.24**	0.00	0.00	0.00	0.00	NaN	NaN	NaN	NaN
EXAM7	1.91	0.00	**2.57**	0.26	0.40	0.04	**0.69**	0.33	0.63	0.66	**0.60**	0.81
EXAM8	1.80	0.39	0.74	**2.95**	0.35	0.32	0.04	**0.57**	0.58	**0.51**	0.63	0.79
EXAM9	**2.17**	0.87	2.11	0.06	**0.68**	0.42	0.67	0.40	0.50	**0.46**	0.55	0.52
EXAM10	**2.04**	0.24	1.34	0.57	**0.68**	0.61	0.56	0.58	0.49	0.48	**0.47**	0.53
EXAM11	**2.93**	0.89	0.00	1.97	**0.84**	0.44	0.15	0.73	**0.53**	0.71	0.59	0.76
EXAM12	0.18	0.09	**1.29**	0.69	0.62	0.73	0.65	0.82	0.84	0.45	0.87	0.36

Table 6.42 and Table 6.43 suggest that that overall the hyper-heuristic with RL-WSGD-HH strategy slightly outperforms the other strategies, although it is not always the case for each problem instance. Furthermore, in order to give a visual illustration of the Pareto optimal solutions, the scatter plot shown by Figure 6.12 visualises the objective functions values of the Pareto solutions of problem instance EXAM9 from the experiments over the bi-objective problem. Similarly, the parallel coordinate shown by Figure 6.13 visualises the trade-off of the objective functions of the Pareto solutions of problem instance EXAM9 from the experiments over the four-objective problem.

From Figure 6.12, we can observe that the Pareto solutions generated with reinforcement learning and the ones generated with self-adaptive learning are not significantly different, they are both competitive. We can also observe that almost all Pareto solutions generated with weighted Tchebychef are dominated by Pareto solutions generated with weighted summation. However, Pareto solutions generated with weighted Tchebychef spread out more evenly in the objective function space than the Pareto solutions generated with weighted summation. On other hand, as shown by Figure 6.13 it is quite difficult to see visually the different of Pareto solutions generated by the different algorithms over the four objective problem.

Furthermore, to evaluate the effect of stopping condition (parameter St in Algorithm 11), testing over problem instance EXAM9, we repeated the experiment of best-performing algorithms (SA-WSGD-HH and RL-WSGD-HH for the two objective problem and the four objective problems respectively) with longer stopping condition, St=3,000,000 iterations instead of St=10,000 and decay rate, α=3.33 $.10^{-7}$ instead of 0.0001. Figure 6.14 and Figure 6.15 presents the comparison. It is clear from the Figure 6.14 that by making the stopping condition longer from 10,000 to 3,000,000, the Pareto solutions are improved quite significantly. Since it is difficult to see the different over the four objective problem from parallel coordinate shown by Figure 6.14, we evaluate the four objectives by per two objectives presented in scatter plot as shown by Figure 6.16.

In addition, to make the Pareto solutions of the four objective problem more readable, we highlight some selected solutions from the Pareto solutions from the experiment of RL-WSGD-HH hyper-heuristic given in Algorithm 11 with two stopping condition, St=3,000,000 iterations and decay rate, α=3.33 $.10^{-7}$ in Figure 6.17. It is clear from Figure 6.17, that a solution, i.e. an examination timetable, could be very good from the perspective of one stakeholder but in the same time, it is very bad for the other stakeholders. For example, we can observe that Solution 0 is the most preferable from the perspective of students, i.e. has the minimum students' penalty. However, it is the worst choice from the perspective of Invigilator. Contrarily, Solution 509 is the

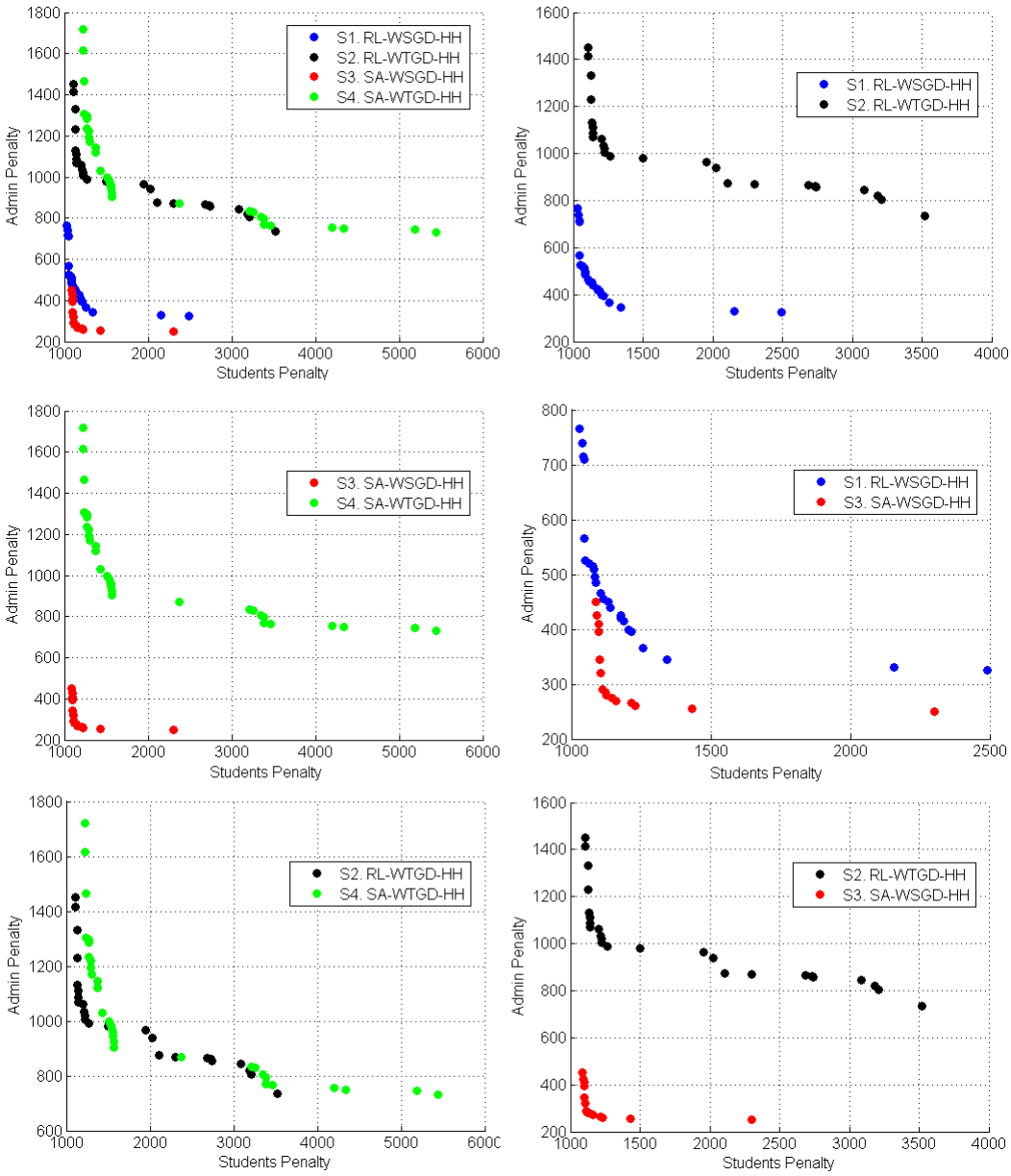
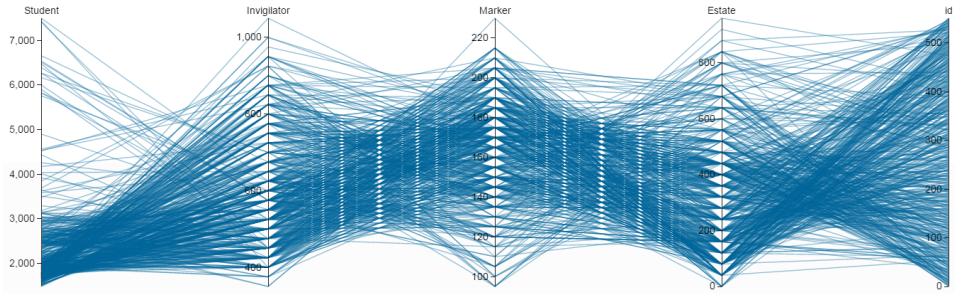
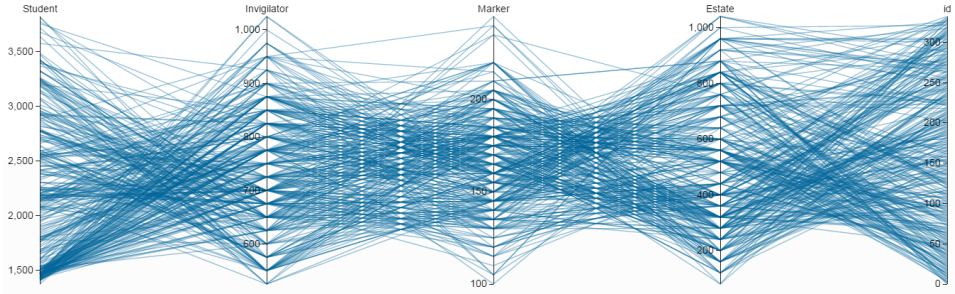


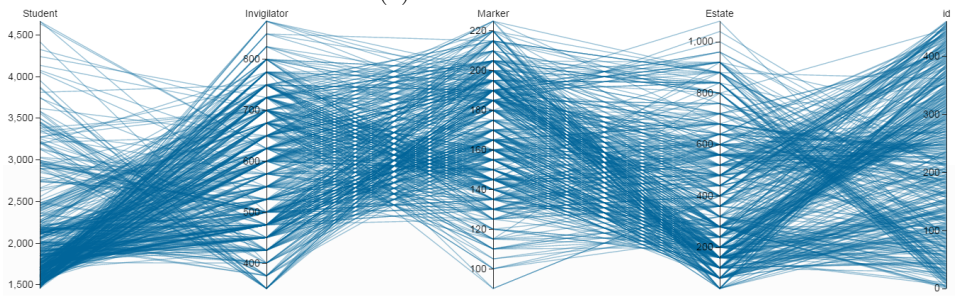
Figure 6.12: The scatter plot of the Pareto solutions of problem instance `EXAM9` from the experiment of hyper-heuristics with different strategies tested over the bi-objective examination timetabling problem with multiple stakeholder perspective.



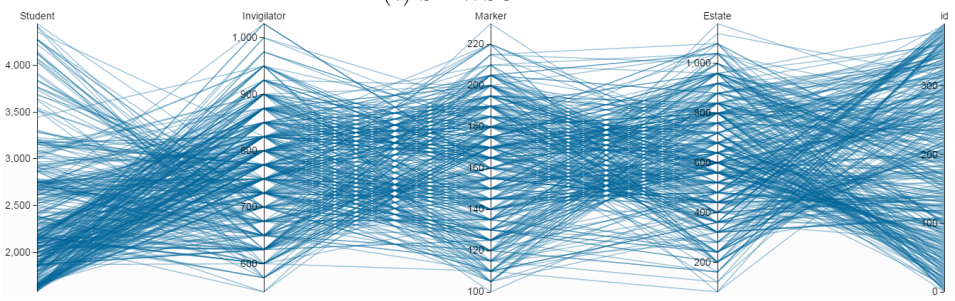
(a) RL-WSGD-HH



(b) RL-WTGD-HH



(c) SA-WSGD-HH



(d) SA-WTGD-HH

Figure 6.13: The parallel coordinate of the Pareto solutions from the experiment over the four-objective problem.

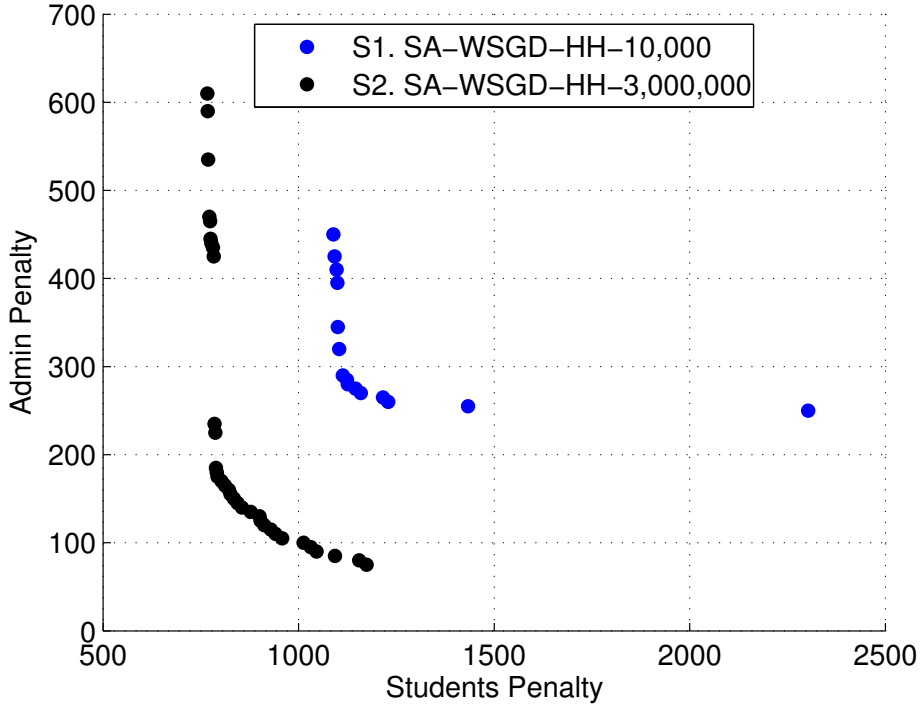
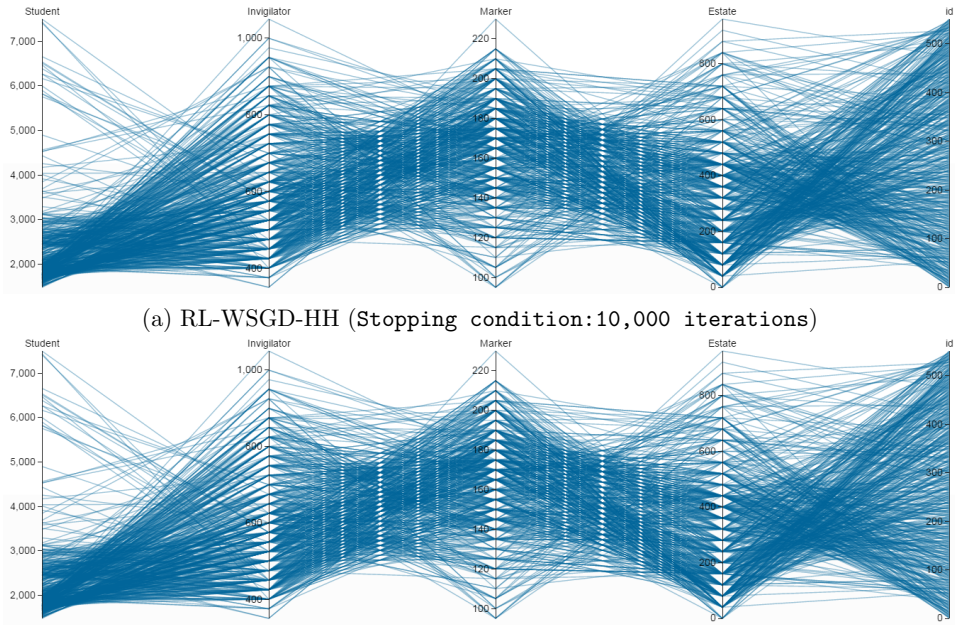


Figure 6.14: The scatter plot of the Pareto solutions from the experiment of SA-WSGD-HH with two different stopping conditions over the bi-objective problem.



(a) RL-WSGD-HH (`Stopping condition:10,000 iterations`)

(b) RL-WSGD-HH (`Stopping condition:3,000,000 iterations`)

Figure 6.15: The parallel coordinate of the Pareto solutions from the experiment of RL-WSGD-HH with two different stopping conditions over the four-objective problem.

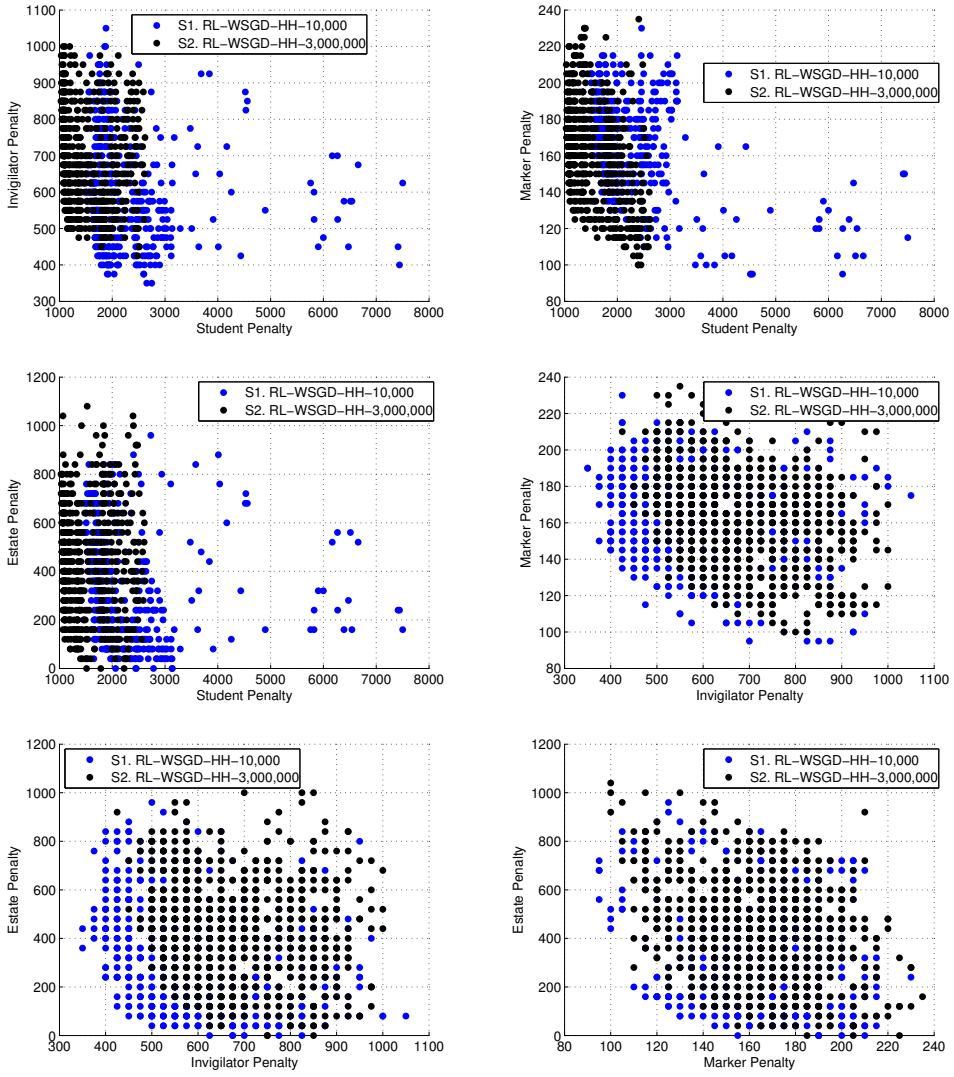
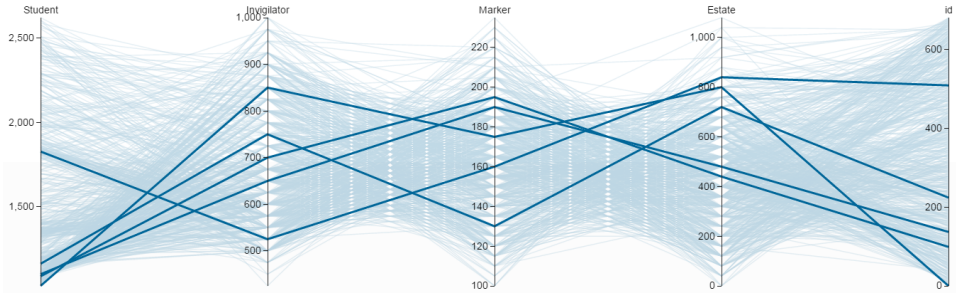


Figure 6.16: The scatter plot of the Pareto solutions from the experiment of RL-WSGD-HH with two different stopping conditions over the four-objective problem.

most preferable solution from Invigilator's point of view but it is the least preferable from students point of view. Therefore, choosing a solution that giving more sensible trade-off such as `Solution 99` or `Solution 224` might be more preferable.

It is obvious that the final decision is on the decision maker of the examination timetabling system. However, we believe that by presenting the quality of examination timetabling in detail how it will affect its each single stakeholder, rather than aggregate quality of the timetable, will result in a better decision. Though analysing many dimensional data is complex and complicated.



Solutions 0, 99, 137, 224, and 509

Figure 6.17: The selected solutions of the Pareto solutions from the experiment of RL-WSGD-HH hyper-heuristic with two different stopping conditions over the four-objective problem.

6.6 The Effect of Multi-objectivisation On The Benchmark Objective function

In [179],the effect of multi-objective approach to single objective problem was studied. The study over travelling salesman problems showed that 'multi-objectivizing a single-objective optimisation problem could help escape local optima. It is interesting to evaluate whether it also works for examination timetabling problems. Thus, we conducted experiments to investigate it.

Rather than using multi-phase approach as discussed in the previous experiment, where the best solution from improving standard quality phase was used as the initial solution in improving fairness phase, in this experiment the two objectives (standard objective function and fairness) are optimised simultaneously with initial solution generated from scratch. Given 360 seconds per run, the solver was run 21 times including for generating feasible initial solution. The results for the best solution in terms of the standard objective function are shown in Table 6.44 and Table 6.45.

In Table 6.44 and Table 6.45, the result from experiment over bi-objective, i.e. standard

Table 6.44: Comparing the best solutions from different approaches: single objective with RL-GD/XGD/MXGD-HH versus two Objectives with RRL-WSGD/WTGD-HH.

Dataset	Instance	RL-GD-HH			RL-XGD-HH			RL-MXGD-HH			RL-WTGD-HH			RL-WSGD-HH		
		Best	Avg	Med	Best	Avg	Med	Best	Avg	Med	Best	Avg	Med	Best	Avg	Med
CARTER	CAR91	**5.24**	5.44	5.44	5.30	5.44	5.41	5.30	5.53	5.55	5.40	5.65	5.65	5.42	5.64	5.62
	CAR92	**4.42**	4.65	4.66	4.51	4.66	4.62	4.49	4.65	4.64	4.67	4.93	4.88	4.70	4.96	4.97
	EAR83	36.90	38.53	38.11	36.73	38.27	38.23	36.74	38.69	38.34	**35.95**	38.55	38.01	36.96	38.85	38.81
	HEC92	10.87	11.56	11.43	10.91	11.43	11.35	10.92	11.42	11.46	**10.53**	11.21	11.23	10.55	11.22	11.19
	KFU93	**14.27**	15.07	15.07	14.36	15.08	15.12	14.14	15.02	14.96	16.01	16.85	16.79	16.13	17.07	17.00
	LSE91	**11.29**	12.32	12.39	11.02	12.00	12.09	11.29	12.20	12.27	11.92	12.54	12.58	11.78	12.54	12.61
	PUR93	5.02	5.20	5.21	5.03	5.20	5.22	**5.00**	5.16	5.13	5.14	5.25	5.23	5.07	5.21	5.20
	RYE92	**9.27**	9.77	9.79	9.01	9.53	9.58	9.08	9.73	9.73	11.33	12.89	13.02	11.85	12.54	12.38
	STA83	157.09	157.35	157.33	157.12	157.39	157.32	157.06	157.32	157.37	**157.05**	157.31	157.29	157.06	157.37	157.36
	TRE92	8.97	9.28	9.27	8.75	9.19	9.13	8.74	9.10	9.06	**8.72**	9.44	9.39	8.96	9.41	9.36
	UTA92	**3.58**	3.73	3.72	3.60	3.72	3.72	3.63	3.78	3.79	3.73	3.93	3.96	3.76	3.94	3.90
	UTE92	25.56	26.62	26.5	25.20	26.39	26.4	25.42	26.30	26.28	**25.43**	26.08	25.98	25.83	27.10	26.46
	YOR83	38.64	40.36	40.28	38.03	39.72	39.56	38.35	39.73	39.77	**37.27**	39.19	39.39	38.14	39.72	39.38
ITC 2007	EXAM1	**6848**	7666	7631	6856	7146	7176	7363	7937	7954	14941	16766	16488	14449	17264	17305
	EXAM2	663	769	759	**632**	720	724	646	796	780	1614	11310	10875	3463	12387	11080
	EXAM3	11822	12463	12312	11659	12574	12429	11806	12424	12344	12314	14697	13028	**11690**	15293	13344
	EXAM4	17785	19179	19271	**16325**	18874	18991	16464	18726	18682	21078	26354	25001	23399	27677	27032
	EXAM5	3896	4272	4288	3837	4157	4050	**3786**	4280	4181	4595	6121	5880	4687	5989	5291
	EXAM6	27575	28656	28930	**27370**	28440	28250	27655	28875	29020	27615	29482	29710	27965	29380	29230
	EXAM7	**5561**	5860	5803	5528	6006	5848	5662	6207	5956	5957	6337	6189	5904	6293	6233
	EXAM8	10078	10553	10560	**9798**	10272	10178	9998	10380	10349	20054	39744	37820	19852	38903	36146
	EXAM9	1363	1504	1492	**1246**	1320	1320	1322	1438	1421	1649	2657	2583	1574	2700	2675
	EXAM10	**15124**	15932	15855	14556	15218	15239	14745	15414	15369	15352	19947	19828	15704	20015	19331
	EXAM11	36878	40457	40095	**36810**	40752	40109	34412	39714	39500	40267	45437	45365	37550	43591	44082
	EXAM12	5620	5966	5949	5300	5577	5581	5436	5968	5953	**5208**	5513	5487	5220	5501	5475
YEDITEPE	yue20011	74	111	112	56	71	68	64	84	82	62	98	90	**57**	74	69
	yue20012	135	187	185	122	159	161	127	158	156	**115**	153	136	120	153	143
	yue20013	**29**	33	33	**29**	31	29	**29**	32	30	**29**	29	29	**29**	29	29
	yue20021	115	148	142	76	113	111	96	139	128	126	220	171	**109**	160	146
	yue20022	203	261	268	162	209	212	189	257	260	205	259	257	**174**	207	203
	yue20023	**56**	64	62	**56**	62	61	**56**	64	61	**56**	59	58	**56**	57	56
	yue20031	**204**	273	270	143	254	206	183	293	249	337	1727	1226	639	1367	1053
	yue20032	461	519	518	434	673	479	433	553	491	**392**	1061	993	427	1032	912

objective function and global unfairness, examination timetabling problem, with classical scalarisation approaches, i.e. Reinforcement Learning - Weighted Sum - Great Deluge - Hyper-heuristic (RL-WSGD-HH) and Reinforcement Learning - Weighted Tchebycheff - Great Deluge - Hyper-heuristic (RL-WTGD-HH) are compared with single-objective-approaches discussed in Chapter 3, i.e. Reinforcement Learning - Great Deluge based Hyper-heuristic (RL-GD-HH), Reinforcement Learning - Extended Great Deluge based Hyper-heuristic (RL-XGD-HH), and Reinforcement Learning - Modified Extended Great Deluge based Hyper-heuristic (RL-MXGD-HH)(See Table 3.20 in Chapter 3) as well as the benchmark. As shown by Table 6.44 and Table 6.45, two-objective-function with Tchebycheff sum approach (RL-WTGD-HH) outperforms the single-objective-function approaches for some instances. It shows that RL-WTGD-HH could help escape from local optima and confirms that the idea of 'multi-objectivizing' [179] also works in examination timetabling problems, especially over Carter and Yeditepe datasets.

In addition, with the same initial solution, number of iterations, and number of runs, we compared the best solution in term of standard benchmark objective function from experiment over examination timetabling problem with single objective problem, and the best standard objective function from experiment over examination timetabling problem with multi-objective functions discussed in earlier sections:

1. Bi-objective functions: standard objective function and global unfairness objective

Table 6.45: Comparing the best solutions from different approaches: single objective with RL-GD/XGD/MXGD-HH versus two Objectives with RL-WSGD/WTGD-HH.

Instance	Our Solver					Benchmark
	RL-GD-HH	RL-XGD-HH	RL-MXGD-HH	RL-WTGD-HH	RL-WSGD-HH	(Burke, 2010)
CAR91	5.24	5.30	5.30	5.40	5.42	**5.03**
CAR92	4.42	4.51	4.49	4.67	4.70	**4.22**
EAR83	36.90	36.73	36.74	**35.95**	36.96	36.06
HEC92	10.87	10.91	10.92	**10.53**	10.55	11.71
KFU93	14.27	14.36	**14.14**	16.01	16.13	16.02
LSE91	11.29	**11.02**	11.29	11.92	11.78	11.15
PUR93	5.02	5.03	**5.00**	5.14	5.07	N.A
RYE92	9.27	**9.01**	9.08	11.33	11.85	9.42
STA83	157.09	157.12	157.06	**157.05**	157.06	158.86
TRE92	8.97	8.75	8.74	8.72	8.96	**8.37**
UTA92	3.58	3.60	3.63	3.73	3.76	**3.37**
UTE92	25.56	**25.20**	25.42	25.43	25.83	27.99
YOR83	38.64	38.03	38.35	**37.27**	38.14	39.53
						(Muller, 2007)
EXAM1	6848	6856	7363	14941	14449	**4370**
EXAM2	663	632	646	1614	3463	**400**
EXAM3	11822	11659	11806	12314	11690	**10049**
EXAM4	17785	**16325**	16464	21078	23399	18141
EXAM5	3896	3837	3786	4595	4687	**2988**
EXAM6	27575	27370	27655	27615	27965	**26585**
EXAM7	5561	5528	5662	5957	5904	**4213**
EXAM8	10078	9798	9998	20054	19852	**7742**
EXAM9	1363	1246	1322	1574	1649	**1030**
EXAM10	15124	**14556**	14745	15352	15704	16682
EXAM11	36878	36810	34412	40267	37550	34129
EXAM12	5620	5300	5436	**5208**	5220	5535
						(Muller, 2007)
yue20011	74	**56**	64	57	62	62
yue20012	135	122	127	120	**115**	125
yue20013	**29**	**29**	**29**	**29**	**29**	**29**
yue20021	115	76	96	126	109	**70**
yue20022	203	**162**	189	205	174	170
yue20023	**56**	**56**	**56**	**56**	**56**	70
yue20031	204	**143**	183	337	639	223
yue20032	461	434	433	**392**	427	440

function (BO:P1).

2. Bi-objective functions: standard objective function and 'specified cohort unfairness' objective function (BO:P2).

3. Bi-objective functions: standard objective function and 'exam-count-based cohort' unfairness objective function (BO:P3).

4. Three objective functions: standard objective function, global unfairness objective function and 'specified cohort unfairness' objective function (TO:P4).

5. Three objective functions: standard objective function, global unfairness objective function and 'exam-count-based cohort unfairness' objective function (TO:P5).

Table 6.46: Comparing the best standard objective function value from the experiment with `RL-WSGD-HH` over the single objective problem and the multiple objective problems.

Dataset	P. Instance	Init Solution	SO	BO: P1	BO: P2	BO: P3	TO: P4	TO:P5
NOTT	NOTT-I	11923	11872	**11600**	11920	11918	11920	11923
	NOTT-II-32	0	0	0	0	0	0	0
	NOTT-II-31	0	0	0	0	0	0	0
	NOTT-II-30	0	0	0	0	0	0	0
	NOTT-II-26	8	8	8	8	8	8	8
	NOTT-II-23	68	68	68	68	68	68	68
CARTER	CAR-S-91	5.30	5.30	**5.28**	**5.28**	5.30	5.30	5.30
	CAR-F-92	4.51	4.50	4.48	**4.45**	4.51	4.51	4.51
	EAR-F-83	36.73	36.73	36.61	36.73	**36.59**	36.73	36.70
	HEC-S-92	10.91	10.91	**10.88**	10.90	10.91	10.91	10.91
	KFU-S-93	14.36	14.36	14.36	14.36	14.36	14.36	14.36
	LSE-F-91	11.02	**11.00**	**11.00**	11.01	11.02	11.02	11.02
	PUR-S-93	5.03	**5.01**	5.03	5.02	5.03	5.02	5.02
	RYE-S-92	9.01	**8.99**	9.01	9.01	9.01	9.00	9.01
	STA-F-83	157.12	157.12	157.12	157.12	157.12	157.12	157.12
	TRE-S-92	8.75	8.74	**8.73**	8.74	8.75	8.75	8.74
	UTA-S-92	3.60	3.60	**3.57**	**3.57**	3.59	3.60	3.60
	UTE-S-92	25.20	25.20	25.18	**25.11**	25.20	25.20	25.20
	YOR-F-83	38.03	38.03	38.03	38.03	38.03	38.03	38.03
ITC 2007	EXAM1	6856	6856	**6743**	6856	6784	6846	6825
	EXAM2	632	627	**569**	607	627	627	627
	EXAM3	11659	11577	**11405**	11422	11615	11636	11639
	EXAM4	16325	16324	16287	**16056**	16325	16325	16325
	EXAM5	3837	**3823**	3837	3837	3837	3837	3836
	EXAM6	27370	27345	27265	**27080**	27370	27370	27370
	EXAM7	5528	5494	5467	5528	**5439**	5518	5526
	EXAM8	9798	9723	**9515**	9790	9798	9784	9792
	EXAM9	1246	1244	**1221**	1241	1241	1246	1246
	EXAM10	14556	14556	**14519**	14556	14556	14556	14556
	EXAM11	36810	36430	36536	**36000**	36759	36777	36750
	EXAM12	5300	5300	**5286**	5300	5287	5300	5300
YEDITEPE	YUE20011	56	56	**51**	54	54	54	56
	YUE20012	122	122	121	120	**117**	122	122
	YUE20013	29	29	29	29	29	29	29
	YUE20021	76	76	76	**69**	76	76	72
	YUE20022	162	162	**150**	155	157	162	162
	YUE20023	56	56	56	56	56	56	56
	YUE20031	143	143	**138**	142	**138**	143	143
	YUE20032	434	434	**417**	433	434	434	434

The comparison is given in Table 6.46, it is clear from Table 6.46 that overall, bi-objective problem with standard objective function and global fairness results most of

the best solutions, i.e. 17 out 38 problem instances. It is because as seen in previous sections, between standard objective function and global fairness is strictly conflicting, thus optimising both objective function concurrently could help escaping from local optima.

In addition, from the experiment of `SA-GDWS-HH` algorithm (number of iterations: `100`, stopping condition: 4.10^6, and decay rate: $2.5\ 10^{-7}$) over the bi-objective examination timetabling problem with multiple stakeholder perspectives, we got solutions beating the best-known solutions over two problem instances, i.e.`EXAM9, EXAM12` as shown by Table 6.47. See Table A.3 in Appendix A for the complete best-known solutions.

Table 6.47: Solutions from our multi-objective approach compared to the best-known solutions

Problem Instance	The Best Known Solutions	Our New Solutions
EXAM9	964 [146]	**935**
EXAM12	5197[146]	**5170**

6.7 Deeper Understanding of the Multi-objective Nature of Examination Timetabling Problems

In order to get a deeper understanding of the nature of the multi-objective problems, this section analyses the relationships between objectives within multi-objective examination timetabling problem (MOETP) using a technique proposed in [252]. The analysis was carried out over two variants of MOETP discussed earlier in this chapter: MOETP with fairness consisting of three objectives ($Z_1 - Z_3$) - standard objective function, global fairness, and 'specified cohort fairness'; and MOETP with multiple stakeholder perspectives, consisting of four objectives ($Z_1 - Z_4$) - students, invigilators, markers, and estates. The data used for analysis were Pareto solutions as results from experiments over five randomly selected problem instances with `SA-WSGD-HH` algorithm. The five instances are `EXAM4, EXAM6, EXAM9, EXAM10, and EXAM12`, simplified as problem instance 1 (I_1) - problem instance 5 (I_5).

6.7.1 Multi-objective Examination Timetabling Problems With Fairness

As proposed in [252], the analysis consists of four steps:

Step 1 - Global Pairwise Relationship Analysis

To identify the global pairwise relationships between objectives. The Kendall correlation coefficients between each pair of objectives are provided in Table 6.48. The values 1.0, -1.0, and 0.0 indicate completely harmonious, conflicting and independent relationship respectively. From Table 6.48, we can observe that generally there are strong conflicting relationships between standard objective function and global fairness; and between standard objective function and specified cohort fairness. Contrarily, between global fairness and specified fairness is quite harmonious. This finding indicates that decomposition technique is not suitable for this problem and multi-objective approach should be employed instead.

Table 6.48: The results for the global pairwise relationship analysis over MOETP with fairness

Instance	I_1	I_2	I_3	I_4	I_5	Mean
$Z_1 - Z_2$	-0.93	-0.72	-1.00	-0.98	-0.69	**-0.86**
$Z_1 - Z_3$	-0.89	-0.60	-0.67	-0.82	-0.69	**-0.73**
$Z_2 - Z_3$	0.82	0.32	0.67	0.77	0.33	**0.58**

Step 2 - Objective Range Analysis

Table 6.49 shows that only the standard objective function has a large enough range, which indicates that there must be solutions with good values for the standard objective function, but poor value for global fairness or specified cohort fairness. It also indicates that the standard objective function is interesting to be analysed further, whereas the other objective could be ignored.

Table 6.49: The results of objective range analysis over MOETP with fairness

Instance		I_1	I_2	I_3	I_4	I_5	Mean
Z_1	Max	68555	61620	1245	14566	5304	30258
	Min	16325	27370	1222	14556	5300	12955
	Range	**76.19%**	**55.58%**	**1.85%**	**0.07%**	**0.08%**	**57.19%**
Z_2	Max	0.7086	0.7631	0.6427	0.6343	0.8793	0.7256
	Min	0.2395	0.6864	0.6293	0.6339	0.8790	0.6136
	Range	**66.20%**	**10.05%**	**2.09%**	**0.06%**	**0.03%**	**15.43%**
Z_3	Max	0.7240	0.5775	0.5053	0.4941	0.7224	0.6047
	Min	0.1768	0.4615	0.4878	0.4927	0.7222	0.4682
	Range	**75.58%**	**20.10%**	**3.46%**	**0.27%**	**0.02%**	**22.57%**

Step 3 - Trade-Off Regions Analysis

Let ↑ and ↓ indicate good and bad respectively-for example $Z_1^{\uparrow}$ means good value for objective function 1. Good means the respective objective function value is better, i.e. less than the average value. Figure 6.18 presents the Karnaugh map indicating the trade-off region analysis. Note that the grey scale in each region indicates the number of objectives with good value. The regions with the same number of good objective function have the same grey scale in such a way that lighter tones represents a higher number of good objective functions and vice-versa for the darker tones. The value in the each region indicates the percentage of solutions with the respective objective function value criteria from the total solutions.

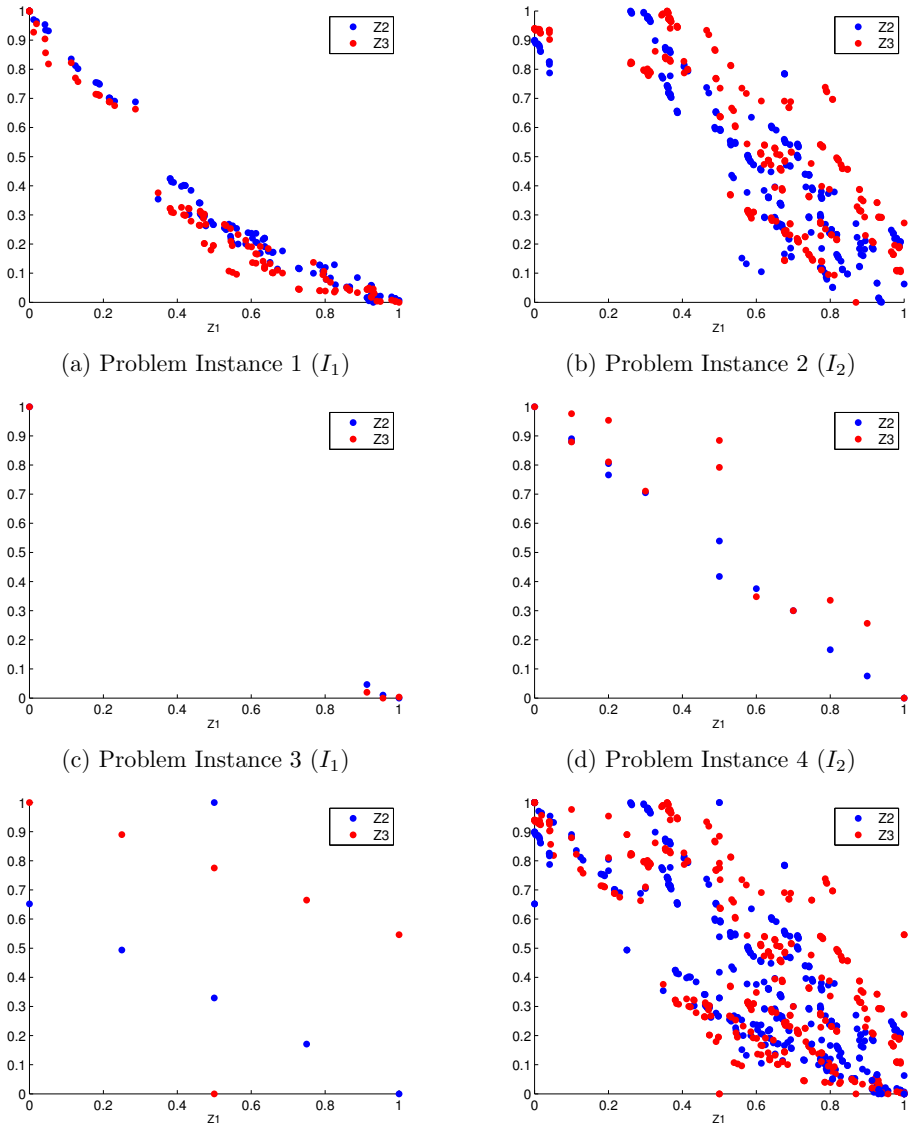
From Figure 6.18 we can observe that the majority of solutions have good global fairness and cohort fairness, but bad standard objective function; or conversely good standard objective function but bad global fairness and cohort fairness. Though there are solutions with acceptable values in all the objectives, the number is not significant.

	$Z_2^{\uparrow}$		$Z_2^{\downarrow}$	
$Z_3^{\uparrow}$	1.98%	44.75%	5.33%	1.77%
$Z_3^{\downarrow}$	1.76%	9.81%	1.54%	33.05%
	$Z_1^{\uparrow}$	$Z_1^{\downarrow}$		$Z_1^{\uparrow}$

Figure 6.18: The results of trade-off regions analysis over MOETP with fairness

Step 4 - Multi-Objective Scatter-plot Analysis

As the last step, Figure 6.19 presents the scatter plot between the standard objective function and the other objectives. This scatter plot helps to identify local rather than global relationship between objective functions. However, as shown by Figure 6.19, no local relationship could be identified. This means that the relationship behaviour between each pair of objectives is global, as identified in step 1. The scatter plot also strongly indicates that there is a conflicting relationship between the standard objective function and fairness objective functions.



(a) Problem Instance 1 (I_1)

(b) Problem Instance 2 (I_2)

(c) Problem Instance 3 (I_1)

(d) Problem Instance 4 (I_2)

(e) Problem Instance 5 (I_5)

(f) All Instances (Merged)

Figure 6.19: The results of multiobjective scatter-plot analysis over MOETP with fairness

6.7.2 Multi-Objective Examination Timetabling Problems with Multiple Stakeholder Perspectives

Step 1 - Global Pairwise Relationship Analysis

Unlike the case of MOETP with fairness, in which there is a strong correlation between each pair of the objectives, the Kendall coefficient in Table 6.50 shows that the objectives are generally independent of each other within MOETP with multiple stakeholder perspectives, however there is a quite strong conflicting relationship between students and markers, which might suggest that the independent objectives could be merged.

Table 6.50: The results for the global pairwise relationship analysis over MOETP with multiple stakeholder perspectives

Instance	I_1	I_2	I_3	I_4	I_5	Mean
$Z_1 - Z_2$	0.00	0.15	-0.23	0.12	-0.20	**-0.03**
$Z_1 - Z_3$	-0.58	0.00	-0.58	-0.60	-0.93	**-0.54**
$Z_1 - Z_4$	-0.13	-0.74	-0.20	0.29	0.00	**-0.16**
$Z_2 - Z_3$	0.00	0.00	0.07	-0.35	0.00	**-0.06**
$Z_2 - Z_4$	0.00	-0.43	-0.33	0.29	0.00	**-0.09**
$Z_3 - Z_4$	-0.34	0.00	0.00	-0.61	0.00	**-0.19**

Step 2 - Objective Range Analysis

Though generally the objectives are independent of each other, as shown in Table 6.51, the range of their values is sufficiently large, indicating that there are solutions with high discrepancy on one of the objective functions.

Table 6.51: The results of objective range analysis over MOETP with multiple stakeholder perspectives

Instance		I_1	I_2	I_3	I_4	I_5	Mean
Z_1	Max	35513	38580	3956	51088	8648	27557
	Min	22154	24300	816	14171	5493	13386.8
	Range	**37.62%**	**37.01%**	**79.37%**	**72.26%**	**36.48%**	**51.42%**
Z_2	Max	0	1675	750	250	10	537
	Min	0	25	0	0	0	5
	Range	**0.00%**	**98.51%**	**100.00%**	**100.00%**	**100.00%**	**99.07%**
Z_3	Max	150	375	195	215	110	209
	Min	100	375	25	0	60	112
	Range	**33.33%**	**0.00%**	**87.18%**	**100.00%**	**45.45%**	**46.41%**
Z_4	Max	7450	4490	800	1950	0	2938
	Min	2050	1415	0	0	0	693
	Range	**72.48%**	**68.49%**	**100.00%**	**100.00%**	**0.00%**	**76.41%**

Step 3 - Trade-Off Regions Analysis

The Karnough map in Figure 6.20 shows that the solutions of greatest significance are those with good values in terms of students, invigilators and estates objective function, but bad value in terms of marker objective function. Compared to the case of MOETP with fairness, within MOETP with multiple stakeholder perspectives solutions are distributed more fairly among different regions, similar to the case of MOETP with fairness, though there are insignificant number of solutions with acceptable values for all the objectives.

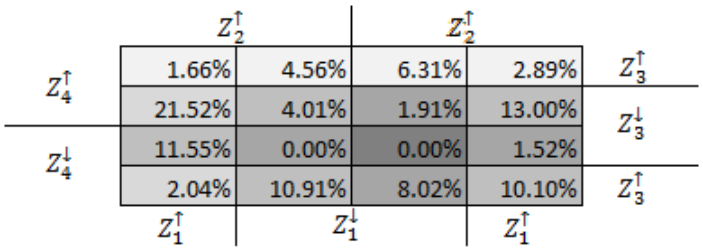


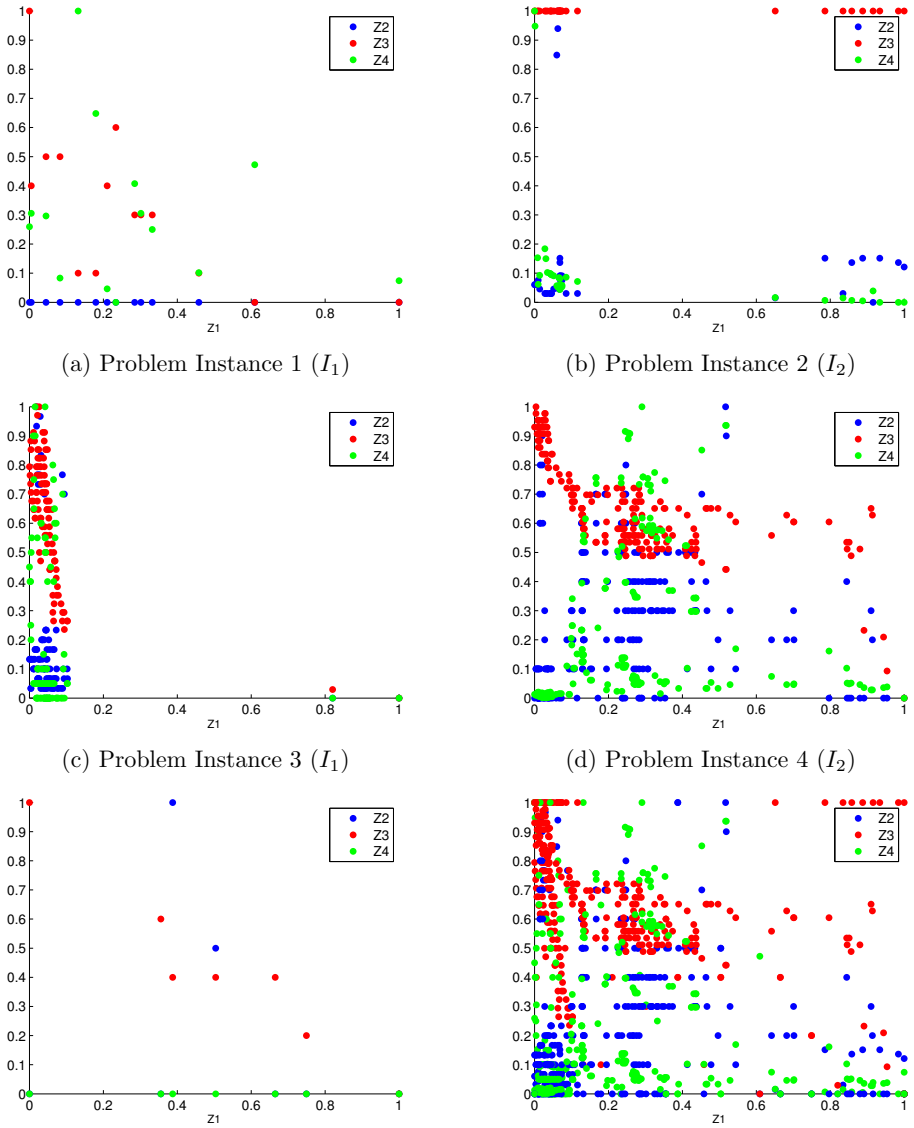
Figure 6.20: The results of trade-off regions analysis over MOETP with multiple stakeholder perspectives

Step 4 - Multi-Objective Scatter-Plot Analysis

Similar to the case of MOETP with fairness, as shown in Figure 6.21, no local relationship between objectives could be identified. The relationship between students and markers objectives that is identified as conflicting in step 1 is scattered evenly over the objective function space. From Figure 6.21, we can identify that in some instances one of the objective function is perfectly optimal already (see I_1 and I_5) or constant (see I_2).

6.8 Conclusion

This chapter discusses hyper-heuristic approaches with reinforcement learning and self-adaptive learning as low-level selection strategy and great deluge as move acceptance strategy combined with two classical scalarisation techniques, i.e. weighted summation and weighted Thcebycheff to solve two variants of multi-objective examination timetabling problems, i.e Multi-objective examination timetabling problem with fairness and Multi-objective examination timetabling problem with multiple stakeholder perspective. Experimental results over the two proposed variants of multi-objective



(a) Problem Instance 1 (I_1)

(b) Problem Instance 2 (I_2)

(c) Problem Instance 3 (I_1)

(d) Problem Instance 4 (I_2)

(e) Problem Instance 5 (I_5)

(f) All Instances (Merged)

Figure 6.21: The results of multiobjective scatter-plot analysis over MOETP with multiple stakeholder perspectives

examination timetabling problems show good results. In multi-objective examination timetabling problem with fairness, the experimental results show that the trade-off between standard objective function and fairness objective function varies among problem instances. For some instances, by preventing the standard objective function from getting worse more than 5% of its initial solution, the fairness objective function could be improved quite significantly, i.e. by greater than 10%.

The experimental results also show that by treating examination timetabling as multi-objective problem, the standard objective function could be improved even better, in which under single objective function approach, our proposed algorithm was stuck. Even, our multi-objective approach could generate best-known solutions over two problem instances of ITC 2007 dataset. It suggests that the 'multi-objectivisation' could help the algorithm from being trapped in local optimum. Since between standard objective function and global fairness is strictly conflicting, thus optimising both objective function concurrently could help escaping from local optima trap. Overall, the advantage of this approach is that rather than result in a single solution, in this approach a decision maker is given a set of approximation of Pareto optimal solutions presenting the trade-off among the might be conflicting objective functions.

Chapter 7

Multi-objective Population-based Algorithm Hyper-heuristic Approach

7.1 Introduction

In order to solve the multi-objective examination timetabling problem, in Chapter 6, the iterated scalarisation or objective weighting based method within a great-deluge based hyper-heuristic was studied. However, as demonstrated in [293], the classical methods (including weighted objective, distance function or goal programming, min-max formulation methods) have some limitations. In particular, they result in a single point solution per iteration; they may not work effectively if the objectives are noisy or discontinuous; they also require prior knowledge to determine the reference points, and their sensitivity to weight vector. Therefore, this chapter studies multi-objective methods, and does so using hyper-heuristic approaches inspired by some state-of-the-art population based algorithms to solve two variants of the proposed multi-objective examination timetabling problems: multi-objective examination timetabling problem with fairness and multi-objective examination timetabling with multiple stakeholder perspectives as discussed in Chapter 4. Within this chapter, Section 7.2 presents the proposed algorithm and Section 7.3 presents and discusses the experimental results.

7.2 Method: Population-based Hyper-heuristics for Multi-objective Optimisation Problem

Implemented within the hyper-heuristic framework, we employ two well-known population based algorithms that were reported in the scientific literature as effective methods to solve many multi-objective optimisation problems: non-dominated sorting genetic algorithm II (NSGA-II) [121] and a hybridisation of artificial bee colony (ABC) [168, 169] and strength pareto evolutionary algorithm 2 (SPEA2) [333]. However, some adaptation and modification over the algorithms discussed in the following sections are required in order to make the algorithm work well over the multi-objective examination timetabling problems.

7.2.1 NSGA-II based Algorithm Hyper-heuristic

In [121], a fast elitist multi-objective evolutionary algorithm was presented, namely non-dominated sorting genetic algorithm II (NSGA-II). The basic idea of NSGA-II algorithm is shown in Algorithm 13. The distinctive features of NSGA-II are the Pareto non-dominated sorting and crowding distance. The detail procedure of ranking based on Pareto - Fast non-dominated sort as well as the crowding distance in Algorithm 13 can be observed in [121]. Our proposed NSGA-II based hyper-heuristic is given in Algorithm 14.

Algorithm 13 NSGA-II Algorithm

```
1: procedure NSGA-II
2:     Set mutation rate, Mr
3:     Set cross-over rate, Cr
4:     Set number of generations, G
5:     Set the population size, N
6:     // Initialise Population
7:     Generate N solutions and add to the population, P.
8:     // Main loop
9:     while not exceed G do
10:        // Generate offspring population F of size N
11:        for i ← 1, N/2 do
12:            Select parents
13:            Based on Cr execute cross-over operator to generate two new individuals.
14:            Based on Mr execute mutation operator on each of the two new individuals.
15:            Add the two new individuals to offspring population F
16:        end for
17:        Combine population P and offspring population F to current population C
18:        Assign_Pareto_ranking_and_crowding_distance(C)
19:        P ← Select_Best_Individuals(C)
20:    end while
21:    return P
22: end procedure
```

In addition to some adaptations of the original NSGA-II algorithm to the HyFlex framework as shown in Algorithm 14, we also made some better modifications from the original version of NSGA-II algorithm to make the algorithm work well over the examination timetabling problem. The modifications that we made are described below.

Our experimentation showed that the standard mutation and crossover operators with integer chromosome representation within Genetic Algorithm work very poorly over the examination timetabling problems. We suspect that the highly constrained nature of the examination timetabling problems prevents the direct integer representation based mutation and cross-over operators from easily generating feasible solution as in unconstrained problems. In addition, the set of decision variables in the problems is too big. For example, as shown by Table 2.2, the smallest problem instance from ITC dataset has 78 exams that have to be signed to period and room. It means that the smallest problem instance in ITC 2007 problem dataset has 156 decision variables, i.e. 78 decisions of timeslot allocation and 78 of room allocation.

Some prior works for dealing with the difficulty on chromosome over timetabling problems have been carried out. For instance, in [86] to solve a real-world bi-objective class/teacher timetabling problem, within NSGA algorithm, the chromosome is represented by an `R x T` matrix, in which `R` is the number of rooms, `T` is the number of timeslots, and the element of matrix is a label representative for a lesson (class/teacher/subject) assigned to the respective room (row) and timeslot (column). A modified (from the traditional genetic operator) crossover and mutation operator were also proposed. The modified operators are generally the movement of lessons from one element to another element of the matrix. The experimental results showed that the timetable generated by the algorithm is better than the timetable created manually. However, in our experimentation over ITC 2007 examination timetabling problem, the results by using these operators is inferior compared to the results by choosing randomly the low-level heuristic (see Table 3.15) in each iteration.

Similarly, to solve the standard single objective ITC 2007 examination timetabling problem, in [8], an `R x T` matrix is also used as a chromosome representation within a memetic algorithm. For the crossover operator, period-exchange crossover as proposed in [97] is adapted with pre-conditions to maintain the feasibility of solution. In addition, the mutation operator is replaced by applying a chosen neighbourhood structure from a list of pre-defined neighbourhood structures over the chosen individual. However, our preliminary experimentation showed that employing the same cross-over operator as proposed in [8] into our proposed hyper-heuristic given in Algorithm 14 produced inferior Pareto solution set compared to the Pareto solution set produced by the same algorithm without cross-over operator, as shown by Figure 7.1.

Therefore, in our proposed hyper-heuristic given in Algorithm 14, we substitute the mutation operator by randomly choosing the low-level heuristic list given in Table 3.15 and no crossover operator, choosing an operator from a list of local search operators

given in Table 7.1 instead. Being substituted by applying a chosen local search low-level heuristic and local search low-level heuristic respectively, the algorithm can be denoted as 'Non-dominated sorting memetic II' (NSMA-II) algorithm instead of NSGA-II. In Algorithm 14, we adopted the implementation of fast and elitist non-dominated sorting and crowding distance based on NSGA-II provided by jMetal [130], a framework for solving multi-objective problems. Algorithm 15 detailed the implementation of standard ranking and crowding distance assignment in NSGA-II algorithm [121]. Briefly,in Algorithm 15, firstly all the solutions are ranked into different fronts. Furthermore, within each front, for each solution a crowding distance is counted.

Table 7.1: List of local search operators.

Local Search Operator	Description
LS-1	Hill climbing local search operator. Move a random exam to new timeslot and room; accept the move if it results a solution that is better (in term of aggregated objective function) than the current solution until predefined number of non-improving moves is reached.
LS-2	Similar to LS-1, except the objective function is only the first objective.
LS-3	Similar to LS-1, except the objective function is only the second objective.
LS-4	Great deluge local search operator. Move a random exam to new timeslot and room; accept the move based on great deluge mechanism until predefined number of non-improving moves is reached.
LS-5	Late acceptance local search operator. Move a random exam to new timeslot and room; accept the move based on late acceptance mechanism [48] until the predefined number of non-improving moves is reached.

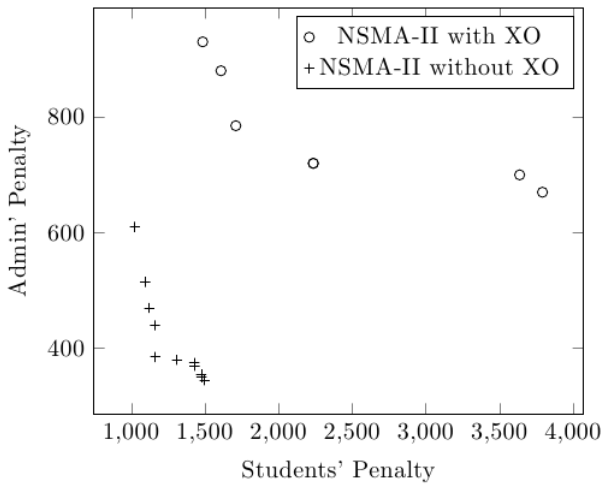


Figure 7.1: The scatter plot of the approximated Pareto optimal solution set from the experimentation of NSMA-II Algorithm with and without cross-over (XO) operator, over problem instance `EXAM9` over the bi-objective ITC examination timetabling problem.

Algorithm 14 NSMA-II Hyper-heuristics. Revisit Table 6.1 for the methods defined in `ProblemDomain` P

```
1: procedure SOLVE(ProblemDomain P)
2:     Set number of generation, G
3:     Set population size, N
4:     Set offspring size, O
5:     Set the mutation rate, Mr
6:
7:     // Get the number of objectives
8:     nob ← P.getNumberOfObjectives()
9:
10:    // Set the solution memory size of the problem domain
11:    P.setMemorySize(N+O)
12:
13:    // Get the predefined mutational low-level heuristics from the problem domain P
14:    M ← P.getHeuristicsOfType('mutation')
15:
16:    // Get the predefined local search low-level heuristics
17:    L ← P.getHeuristicsOfType('local_search')
18:
19:    // Initialise population of size N
20:    jmetal.SolutionSet Pop ← new jmetal.SolutionSet(N)
21:
22:    // Initiate crowding distance
23:    jmetal.distance dist ← new jmetal.distance()
24:    for i ← 1, N do
25:       // Initiate a new solution and stored it in the solution memory index i-1
26:       P.initialiseSolution(i-1)
27:       double [] of ← P.getFunctionValues(i-1)
28:       jmetal.Solution indv ← new jmetal.Solution(nob)
29:       for j ← 1, nob do
30:          indv.setObjectives(j-1,of[j-1])
31:       end for
32:       Pop.add(indv)
33:    end for
34:
35:    // The main loop
36:    for i ← 1, G do
37:       // Initiate offspring
38:       jmetal.SolutionSet Osp ← new jmetal.SolutionSet(O)
39:       // For each individual in the population
40:       for j ← 0, N-1 do
41:          // Generate a real random number, r with value between 0-1
42:          r ← randomNumber()
43:          if r < Mr then
44:             // Select a mutational low-level heuristic randomly
45:             mh ← selectMutationHeur(M)
46:             // Apply low-level heuristic mh, over solution stored in solution memory with index j
47:             // And stored the new solution to solution memory index with index j+N
48:             s1 ← P.applyLLH(mh, j, j+N)
49:             P.copySolution(j+N,j)
50:             // Select a local search low-level heuristic randomly
51:             lh ← selectLocalsearchHeur(L)
52:             s2 ← P.applyLLH(lh, j, j+N)
53:             if s2 ≤ s1 then
54:                P.copySolution(j+N,j)
55:                jmetal.Solution off ← new jmetal.Solution(nob)
56:                for z ← 1, nob do
57:                   off.setObjectives(z-1,of[z-1])
58:                end for
59:                Osp.add(off)
60:             end if
61:          end if
62:       end for
63:       jmetal.SolutionSet UnionPop ← Pop.union(Osp)
64:
65:       // Pareto Ranking and Crowd Distance Assignment (See Algorithm 15)
66:       RANK_AND_CROWDDISTANCEASSIGNMENT(UnionPop)
67:    end for
68:    jmetal.Ranking ranking ← new Ranking(population)
69:    double[][] Pareto ← ranking.getSubfront(0).writeObjectivesToMatrix()
70:    Print(Pareto)
71: end procedure
```

Algorithm 15 Pareto and Crowd Distance Assignment Method

```
1: procedure RANK_AND_CROWDDISTANCEASSIGNMENT(jmetal.SolutionSet UnionPop)
2:     jmetal.Ranking rank ← new jmetal.Ranking(UnionPop)
3:     remain ← N
4:     index ← 0
5:     Pop.clear()
6:     j.metal.SolutionSet front ← rank.getSubfront(index)
7:     while (remain > 0 && remain ≥ front.size()) do
8:        dist.crowdingDistanceAssignment(front, nob)
9:        for  k ← 0, front.size()-1 do
10:           Pop.add(front.get(k))
11:       end for
12:       remain ← front.size()-1
13:       index++
14:       if remain > 0 then
15:          front ← rank.getSubfront(index)
16:       end if
17:    end while
18:    if  remain > 0 then
19:       dist.crowdingDistanceAssignment(front, nob) front.sort(new jmetal.CrowdingComparator())
20:       for k ← 0, remain-1 do
21:          population.add(front.get(k)
22:          remain ← 0
23:       end for
24:    end if
25: end procedure
```

7.2.2 Artificial Bee Colony-Based Hyper-heuristic (ABC-SPEA2)

The second population-based hyper-heuristic approach studied in this thesis is based on the hybridisation of two algorithms: artificial bee colony (ABC) [168, 169] and strength pareto evolutionary algorithm 2 (SPEA2) [333]. The original ABC and SPEA2 algorithms are shown in Algorithm 16 and 17 respectively, whereas our proposed hyper-heuristic based on the hybridisation of both algorithms is given in Algorithm 18. This hybridisation is called ABC-SPEA2.

The main loop of ABC-SPEA2 algorithm is between line `40` and line `55` in Algorithm 18. Briefly, we hybridise ABC and SPEA2 algorithms by substituting the standard crossover and mutation operators in SPEA2 algorithm with employed bee (see line `51` in Algorithm 18; and 19) and scout bee (see line `53` in Algorithm 18; and 20) operators from ABC algorithm respectively. The other procedures remain the same as in the original SPEA2 algorithm.

In this work we also adopted the artificial bee colony algorithm implemented for single objective examination timetabling problem in [20, 18] and hybridised with SPEA2 Algorithm adopted from jMetal framework to cope with the multi-objective examination timetabling problem instead.

We tested the proposed hyper-heuristic over two variants of the multi-objective examination timetabling problems (i.e. multi-objective examination timetabling problem with fairness and multi-objective examination timetabling problem with multiple stakeholder perspectives). The experimental results are discussed in the following sections.

Algorithm 16 The pseudocode of the original artificial bee colony algorithm

```
1: procedure ABC
2:     // Population , employed bee, and onlooker bee size are equal
3:     Set Population Size, P
4:     Set Scout Bee Size, S
5:     Set Initial Solutions, I[P] ← (s_0,s_1,...,s_P)
6:     repeat
7:         //Exploration search conducted by employed bee
8:         Neighbourhood search upon each initial solution.
9:         //Exploitation search conducted by onlooker bee
10:        Select solution from previous solutions and undergo further neighbourhood search.
11:        //Conducted by scout bee
12:        Replace some previous solutions with new randomly generated solution.
13:        record the best solution.
14:    until Termination Condition Met
15: end procedure
```

Algorithm 17 The pseudocode of original SPEA2 Algorithm

```
1: procedure SPEA2
2:     Initialise population P (internal population) at random
3:     Create empty external set E (external population)
4:     while Stop condition is false do
5:         Compute fitness of each individual in P and and E
6:         Copy all nondominated individuals in P and E to E
7:         Use the truncation operator to remove elements from E when the capacity has been exceeded
8:         if the capacity of E has not been exceeded then
9:             Use dominated individuals in P to fill E
10:        end if
11:        Perform binary tournament selection with replacement to filll the mating pool
12:        Apply crossover and mutation to the mating pool
13:    end while
14: end procedure
```

7.3 Experimental Results over Multi-objective examination timetabling problem with Fairness

The proposed hyper-heuristics are tested over bi-objective examination timetabling problem with fairness. As discussed in Section 4.2.2 of Chapter 4, the objective functions are: standard objective function and global unfairness. Compared to the single-solution based hyper-heuristics, the population based hyper-heuristic requires more memory size and computation time significantly. Therefore, in the experiment discussed in the following sections, we reported the proposed hyper-heuristic over problem instance `EXAM9` of ITC 2007 dataset, which is in term of the number of exams it is the second smallest problem instance.

Note that we already tested the proposed algorithm over all other problem instances greater than `EXAM9`. However, the experimental results showed that our proposed NSMA-II was impractical over the multi-objective problems with fairness, for larger problem instances. Given the maximum memory allocation of our machine, the testing came out with 'run out memory'. We suspect that in addition to large number of population and offspring that should be maintained within the algorithm, it also because when computing fairness the penalties of each individual student is maintained too. We believe more efficient data structures will help coping with this problem, but it is beyond

Algorithm 18 ABC-SPEA2-based Hyper-heuristic

```
1: procedure SOLVE(ProblemDomain P)
2:     //Population , employed bee, and onlooker bee size are equal
3:     Set population size, N
4:     int [] trial ← new int[N]
5:     Set trial limit, Tl
6:     jmetal.SolutionSet internalPopulation ← new jmetal.SolutionSet(N)
7:     jmetal.SolutionSet externalPopulation ← new jmetal.SolutionSet(N)
8:
9:     // Get the predefined mutational low-level heuristics from the problem domain P
10:    M ← P.getHeuristicsOfType('mutation')
11:
12:    // Get the predefined local search low-level heuristics
13:    L ← P.getHeuristicsOfType('local_search')
14:
15:    // Get the number of objectives
16:    nob ← P.getNumberOfObjectives()
17:
18:    // Set the solution memory size of the problem domain
19:    P.setMemorySize(2.N)
20:    // Initialise internal population
21:    for i ← 1, N do
22:       // Initiate a new solution and stored it in the solution memory index i-1
23:       P.initialiseSolution(i-1)
24:       double [] of ← P.getFunctionValues(i-1)
25:       jmetal.Solution newSolution ← new jmetal.Solution(nob)
26:       for j ← 1, nob do
27:          newSolution.setObjectives(j-1,of[j-1])
28:       end for
29:       internalPopulation.add(newSolution)
30:    end for
31:    // Initialise employed bees
32:    double[] employedBee ← new double[N]
33:    double[] employedBeeP ← new double[N]
34:    double[] employedBeeF ← new double[N]
35:    for i ← 1, N do
36:       employedBee[i] ← P.getFunctionValues(i-1)[0]
37:    end for
38:    Set a list of low-level heuristic indices, LH of size S
39:    Set a list of Well-performed low-level heuristic indices, WH
40:    while (Stopping condition is not met) do
41:       jmetal.SolutionSet unionPopulation ← internalPopulation.union(externalPopulation)
42:       jmetal.Spea2Fitness spea ← new jmetal.Spea2Fitness(unionPopulation)
43:       spea.fitnessAssign()
44:       externalPopulation ← spea.environmentalSelection(N)
45:       jmetal.SolutionSet offSpringSolutionSet ← new jmetal.SolutionSet(N)
46:       // Fill LH with the index of mutational low-level heuristics predefined in P chosen
   randomly
47:       for i=0 to S-1 do
48:          LH.add(selectMutationHeur(M))
49:       end for
50:       //Sending employed bees
51:       SENDING_EMPLOYED_BEES()
52:       //Sending scout bees
53:       SENDING_SCOUT_BEES()
54:       internalPopulation ← offSpringSolutionSet
55:    end while
56:    return getFront(Pop)
57: end procedure
```

Algorithm 19 Sending employed bees method of Algorithm 18

```
1: procedure SENDING_EMPLOYED_BEES()
2:     for i ← 0, N-1 do
3:         repeat
4:             // get the earliest element of LH and delete it from LH
5:             mh ← LH.poll()
6:             newFunctionVal ← P.applyLLH(mh,i,i+N)
7:             if newFunctionVal < employedBee[i] then
8:                 employedBee[i] ← newFunctionVal
9:                 P.copySolution(i+N, i)
10:                WH.add(mh)
11:                trial[i]← 0
12:            else
13:                trial[i]++
14:            end if
15:        until (LH is empty)
16:        fill 75% elements of LH with elements from WH
17:        fill 25% elements of LH with the index of low-level heuristics predefined in P chosen randomly.
18:        Clear the element of WH
19:    end for
20:    sumBee ← 0.0
21:    for all s ∈ employedBee do
22:        sumBee ←sumBee+s
23:    end for
24:     avgBee ← sumBee / this.popSize
25:    for i ← 0, i ← N-1 do
26:        employedBeeF[i] ← Math.abs(employedBee[i] - avgBee)
27:    end for
28:    sumBeeF ← 0.0
29:    for all s ∈ employedBeeF do
30:        sumBeeF ← sumBeeF+s
31:    end for
32:    for i ← 0, i ← N-1 do
33:        employedBeeF[i] ← employedBeeF[i]/sumBeeF
34:    end for
35:    for i ← 0, i ← N-1 do
36:        // Select a local search low-level heuristic randomly
37:        lh ← selectLocalsearchHeur(L)
38:        //Select Onlooker bees base on roulette-wheel selection mechanism
39:        ob ← SELECT_ONLOOKER_BEE(employedBeeP)
40:        newFunctionVal ← P.applyLLH(lh,ob,ob+N)
41:        if newFunctionVal < employedBee[ob] then
42:            employedBee[ob] ← newFunctionVal
43:            P.copySolution(ob+N, ob)
44:            trial[ob]← 0
45:        else
46:            trial[ob]++
47:        end if
48:    end for
49: end procedure
```

Algorithm 20 Sending scout bees method of Algorithm 18

```
1: procedure SENDING_SCOUT_BEES()
2:     scout ← 0
3:     scoutV ← trial[0]
4:     for i ← 0, i ← N-1 do
5:         double [] of ← P.getFunctionValues(i)
6:         jmetal.Solution offspring ← new jmetal.Solution(nob)
7:         for j ← 1, nob do
8:             offspring.setObjectives(j-1,of[j-1])
9:         end for
10:        offSpringSolutionSet.add(newSolution)
11:        if trial[i] > scoutV then
12:            scout ← i
13:            scoutV ← trial[i]
14:        end if
15:    end for
16:    P.initialiseSolution(scout)
17:    employedBee[scout] ← P.getFunctionValues(scout)[0]
18:    jmetal.Ranking ranking ← new Ranking(externalPopulation)
19:    double[][] Pareto ← ranking.getSubfront(0).writeObjectivesToMatrix()
20:    Print(Pareto)
21: end procedure
```

the scope of this study.

7.3.1 The Experimental Results with NSMA-II based Hyper-heuristic

In order to evaluate the effect of the initial populations and the number of generation (parameter `G` in Algorithm 14), we tested our proposed NSMA-II based hyper-heuristic (see Algorithm 14) with three different scenarios summarised in Table 7.2 and under the other parameters setting given in Table 7.3. In scenario S1 the initial population is initiated with any feasible solutions generated by constructor (see Algorithm 6 in Chapter 3) without improvement phase over the standard objective function. Whereas in scenarios S2 and S3, the initial population is initiated with the same solution generated by the constructor with improvement phase over the standard objective function (see Algorithm 3 in Chapter 3). Scenarios S2 and S3 are different in the number of generation.

Table 7.2: The experimental scenarios over NSMA-II based hyper-heuristic to evaluate the effect of initial population and the number of generation.

Scenarios	Initial Population	Number of Generation (`N`)
S1	Any feasible solution generated by constructor (see Algorithm 6 in Chapter 3) without improvement phase over the standard objective function.	1000
S2	The solution generated by constructor with improvement phase over the standard objective function (see Algorithm 3 in Chapter 3).	1000
S3	The solution generated by constructor with improvement phase over the standard objective function (see Algorithm 3 in Chapter 3).	100

Table 7.3: The parameters setting of the experiment over NSMA-II based hyper-heuristic to evaluate the effect of initial population and the number of generation.

Parameter	Values
Population size (`N`)	256
Offspring size (`O`)	256
Mutation rate (`Mr`)	0.3

The comparison of objective function values of the Pareto solution set from the experiment with different scenarios given in Table 7.2 and parameter setting given in Table 7.3 over problem instance `EXAM9` is shown in Table 7.4 and Figure 7.2. Further, the coverage metric and the other multi-objective metrics, i.e. hypervolume (HV), additive epsilon, generational distance (GD), inverted generational distance (IGD), spread, and generalized spread are shown in Table 7.5 and Table 7.6. Revisit Section 2.5 in Chapter 2)

for the explanation of these metrics. Keep in mind that regarding the metrics values, except for coverage and hyper-volume, the lower the value the better it is.

Table 7.4: The objective function values of the Pareto optimal solution set from the experiment of NSMA-II over bi-objective examination timetabling problem: standard objective function and global fairness.

Scenarios	Init. Solution		f1			f2		
	f1	f2	min	max	avg	min	max	avg
S1	Random		1474	3716	1705	**0.489**	0.745	0.640
S2	1246	0.75	1246	1989	1447	0.614	0.746	0.692
S3	1246	0.75	**1241**	1731	1358	0.639	0.745	0.708

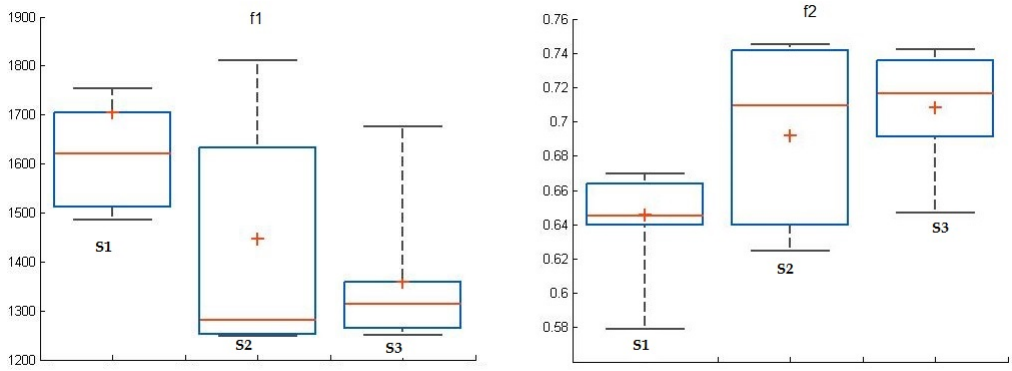


Figure 7.2: The box plot visualising the objective function values of the Pareto optimal solution set from the experiment of NSMA-II algorithm over the bi-objective problem.

Table 7.5: The coverage metric of the Pareto optimal solution set from the experiment of NSMA-II algorithm over the bi-objective problem.

$\mathcal{C}$ (x,y)	S1	S2	S3	sum
S1	X	0.21	0.16	0.37
S2	0.35	X	0.58	**0.93**
S3	0.41	0.33	X	0.75

Table 7.5 shows that in terms of coverage metric, Scenario 2 (S2) and Scenario 3 (S3) outperforms Scenario 1 (S1). It means that as might be expected better initial population results in better final population quality and similarly with the iterations.

Further we compare, the effectiveness of strategy for adding the new solutions produced after mutation and local search operator to the offspring population (see line 53 of Algorithm 14). In this case we compare two strategies. In the first strategy, 'improving or equal new solution only is added' (IO), the solution generated after applying local search operator is added to the offspring population if its standard objective function is not worse than before; whilst in the second strategy 'any new solution is added' the solution generated after applying local search operator after is always added to

Table 7.6: The other multi-objective metric of the Pareto optimal solution set from the experiment of NSMA-II algorithm over the bi-objective problem.

Scenario	Metric					
	HV	Epsilon	GD	IGD	Spread	Gen. Spread
S1	0.58	233	0.01	0.01	1.31	1.42
S2	0.46	5	0.01	0.04	1.00	1.00
S3	0.39	0	0.01	0.05	0.98	0.94

the offspring population whether its standard objective function is better or not. The experiment scenarios are given in Table 7.7. Before that, we carried out experiments to determine the effective mutation rate (`Mr`) rather than using the initial value, i.e `0.3` as in the previous experiments. Comparing four different values, i.e. `0.3,0.5,0.7,0.9`, our experimental results showed that the value `0.9` results in the best solution.

Table 7.7: The experimental scenarios over NSMA-II algorithm to evaluate the effectiveness of strategy for adding the new solutions produced after applying mutation and local search operator to the offspring population.

Scenarios	Offspring Population Strategy
S1 (IO)	The solution generated after applying local search operator is added to the offspring population if its standard objective function is not worse than before.
S2 (AA)	The solution generated after applying local search operator is always added to the offspring population whether its standard objective function is better or not.

Table 7.8: The parameters setting of the experiment over NSMA-II algorithm to evaluate the effectiveness of strategy for adding the new solutions produced after applying mutation and local search operator to the offspring population.

Parameter	Values
Population size (`N`)	256
Offspring size (`O`)	256
Mutation rate (`Mr`)	0.9
Number of generation (`G`)	1000
Initial population	The solution generated by constructor with improvement phase over the standard objective function (see Algorithm 3 in Chapter 3).

Under the parameter settings given in Table 7.8, the objective function values of the Pareto optimal solution set from the experiment are shown in Table 7.9 and Figure 7.3. Further, table 7.10 and Table 7.11 compare the Pareto optimal solution set in term of coverage metric and the other metrics respectively. It shown from the results that 'any new solution is added' strategy outperforms 'improving or equal new solution only is added' strategy.

Lastly, we compared the Pareto solution set from the previous experiment of NSMA-II based hyper-heuristic under parameter settings given in Table 7.7 and 'AA' offspring

Table 7.9: The objective function values of the Pareto optimal solution set from the experiment of NSMA-II algorithm over the bi-objective problem.

Scenario	f1			f2		
	min	max	avg	min	max	avg
S1	1241	**1731**	**1357.68**	0.64	**0.74**	0.71
S2	**1240**	14483	5716.98	**0.31**	0.75	**0.55**

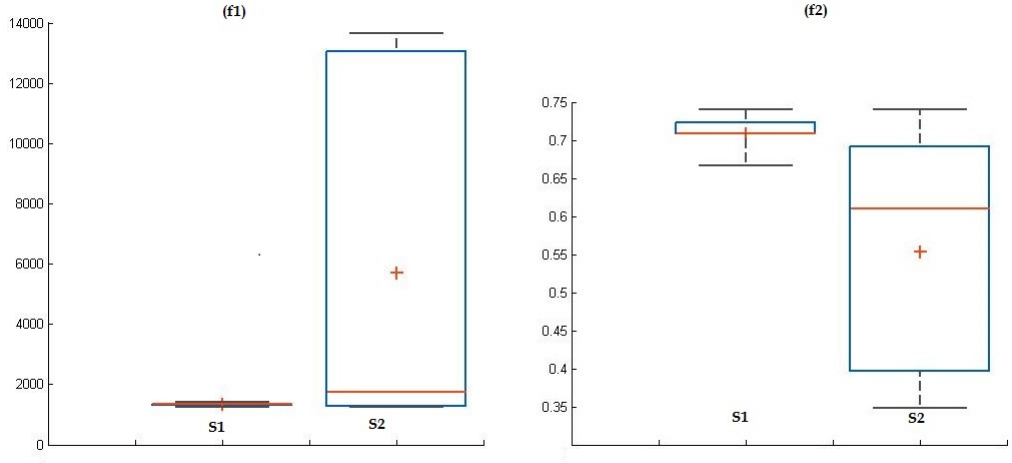


Figure 7.3: The box plot of the objective function values of the Pareto optimal solution set from the experiment of NSMA-II algorithm over the bi-objective problem.

Table 7.10: Coverage metric of the objective function values of the Pareto optimal solution set from the experiment of NSMA-II algorithm over the bi-objective problem.

P. Instance	$\mathcal{C}$(S1,S2)	$\mathcal{C}$(S2,S1)
EXAM9	0.11	**0.58**

Table 7.11: Other multi-objective Metric of The objective function values of the Pareto optimal solution set from the experiment of NSMA-II algorithm over the bi-objective problem.

Instance	HV		Epsilon		GD		IGD		Spread		Gen. Spread	
	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2
EXAM9	0.24	**0.75**	1	**0.00**	0.00	0.00	0.10	**0.00**	**0.98**	1.18	0.97	**0.91**

population strategy) with the Pareto solution set from the experiment of scalarisation based hyper-heuristics (see Algorithm 11 in Chapter 6): 'reinforcement learning - weighted sum great deluge - hyper-heuristic' (RL-WSGD-HH) and 'reinforcement learning - weighted Tchebycheff great deluge - hyper-heuristic' (RL-WTGD-HH) discussed in Chapter 6. Since in the experiments with the scalarisation techniques the standard objective function is prevented from getting worse by more than 5% of the initial solution, to make fair comparison, the solutions with standard objective function getting worse by more than 5% of the initial solution in the Pareto solution set from the NSMA-II are excluded. Note that the same initial solution is used in the experiments of these these approaches. The comparison of the objective function values of solutions in the Pareto solution set produced by each approach is summarised in Table 7.12 and visually compared by Figure 7.4. Further, the quality indicators are shown in Table 7.13 and Table 7.14.

Table 7.12: The objective function values of the Pareto optimal solution set from the experiment of three different hyper-heuristics based on: NSMA-II (S1), weighted sum scalarisation (S2), and weighted Tchebycheff scalarisation (S3) over the bi-objective problem.

Approach	Objective Function Values					
	f1			f2		
	min	max	avg	min	max	avg
S1	1240	**1285**	1261	0.69	0.75	0.74
S2	1221	1308	1260	**0.67**	**0.74**	**0.70**
S3	**1199**	1304	**1246**	0.68	**0.74**	0.71

Table 7.13: The coverage metric of the Pareto optimal solution set from the experiment with the three different hyper-heuristics.

$\mathcal{C}$(i,j)	S1	S2	S3	sum
S1	x	0.00	0.00	0.00
S2	1.00	x	0.52	**1.52**
S3	1.00	0.32	x	1.32

Table 7.14: The other metrics of the Pareto optimal solution set from the experiment of the three different hyper-heuristics.

HV			Spread			GD		
S1	S2	S3	S1	S2	S3	S1	S2	S3
0.1563	**0.4986**	0.4954	0.9708	0.6349	**0.5434**	0.1195	0.0091	**0.0075**

Table 7.13 and Table 7.14 suggest that the scalarisation technique based hyper-heuristic outperforms `NSMA-II` hyper-heuristic. Moreover, in terms of coverage and hypervolume the weighted summation outperforms the weighted Tchebycheff but in terms of spread and generational distance (GD), weighted Tchebycheff outperforms weighted summation.

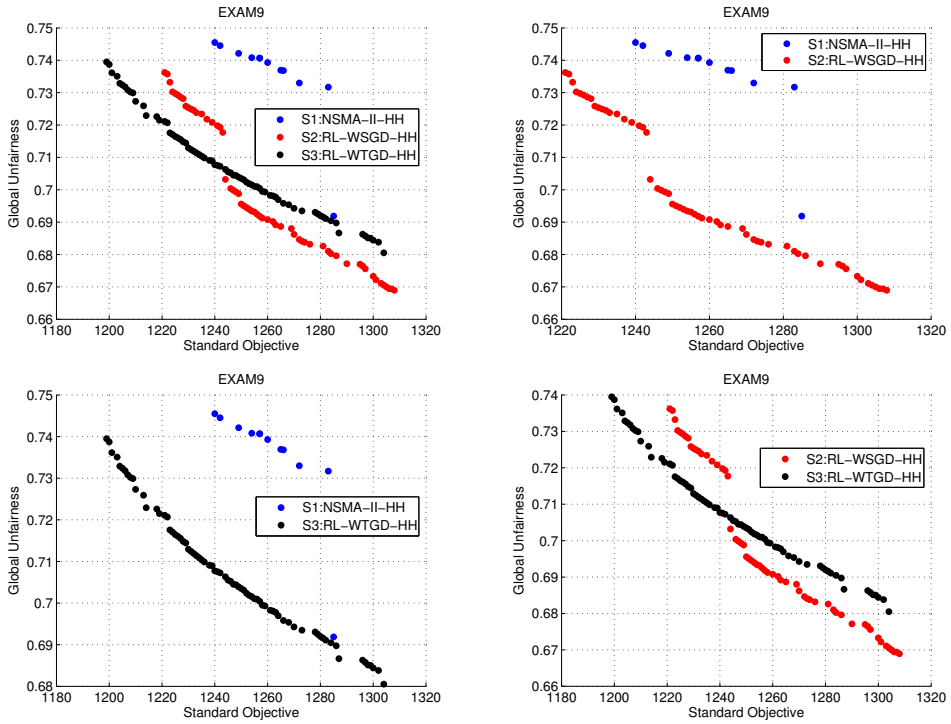


Figure 7.4: The scatter plot of the objective function values of the Pareto optimal solution set from the experiment of the three different hyper-heuristics.

7.3.2 Experimental Results with ABC-SPEA2 based Hyper-heuristic

In order to evaluate the effect of the initial population and stopping condition (see line 40 of Algorithm 18), which is equivalent to the number of generation in NSMA-II algorithm, to the final solution within ABC-SPEA2 based hyper-heuristic, we tested the algorithm with experimental scenarios summarised in Table 7.15. As shown by Table 7.15, three different initial population strategies are proposed. Firstly, the initial population is initiated with any random feasible solutions generated by the constructor (see Algorithm 6 in Chapter 3) without improvement phase over the standard objective function. Secondly, the initial population is initiated with the same solution generated by the constructor with improvement phase over the standard objective function (see Algorithm 3 in Chapter 3). Thirdly, one of the solutions in the initial population is a solution generated by the constructor with improvement phase over the standard objective function whereas the others are any random feasible solutions generated by the constructor without improvement phase over the standard objective function. Each strategy is tested with the number of generation being 100 and 1000.

Tested under parameter setting given in Table 7.16, the objective function values of the solutions in Pareto solution set from the experiment with the six scenarios are

Table 7.15: The experimental scenarios over ABC-SPEA2 based hyper-heuristic to evaluate the effect of initial population and the number of stopping condition.

Scenario	Intial Solution	Stopping Condition
S1	Random feasible solutions.	
S2	A good solution.	1000 iterations
S3	Hybrid: a good solution and random solutions.	
S4	Random feasible solutions.	
S5	A good solution.	100 iterations
S6	Hybrid: a good solution and random solutions.	

Table 7.16: The parameters setting of the experiment over ABC-SPEA2-based hyper-heuristic to evaluate the effect of initial population and the number of generation.

Parameter	Values
Population size (`N`)	50
Trial limit (`Tl`)	100
Length of low-level heuristic list(`S`), i.e. line `38` of Algorithm 18	200

shown in Table 7.17. Further, the quality indicators in terms of coverage metric and the other multi-objective algorithm metrics are given in Table 7.18 and Table 7.19 respectively.

Table 7.17: The objective function values of the Pareto optimal solution set from the experiment with ABC-SPEA2 over bi-objective examination timetabling problem: standard objective function and global fairness.

Scenarios	f1			f2		
	min	max	avg	min	max	avg
S1	1316	1727	1426	0.57	0.69	0.64
S2	1246	1246	1246	0.75	0.75	0.75
S3	1246	8546	2098	0.61	0.75	0.64
S4	1437	1783	1543	0.55	0.73	0.65
S5	1246	1246	1246	0.75	0.75	0.75
S6	1246	8546	1867	0.61	0.75	0.67

From Table 7.19, it is clear that in terms of coverage metric, initiating the solutions in the initial population with a good solution and random feasible solutions result in the best solutions. It is also shown that using a good solution as initial population does not perform well since the initial solution could not be improved further and more iterations is better. However, in terms of the other metric as quality indicators, it is shown that they are very competitive.

Secondly, given the same running time, we compared the performance of NSMA-II based hyper-heuristic and ABC-SPEA2 based hyper-heuristic. The objective function values of the Pareto solution set from the experiment with both hyper-heuristics over problem instance `EXAM9` are shown in Table 7.20 and Figure 7.5. While the coverage and other

Table 7.18: The coverage metric of the Pareto optimal solution set from the experiment with ABC-SPEA2 over the bi-objective problem.

$\mathcal{C}$)	S1	S2	S3	S4	S5	S6	sum
S1	X	0.00	0.55	0.89	0.00	0.94	2.38
S2	0.00	X	0.09	0.00	1.00	0.06	1.15
S3	0.05	1.00	X	0.63	1.00	0.82	**3.51**
S4	0.00	0.00	0.18	X	0.00	0.12	0.30
S5	0.00	1.00	0.09	0.00	X	0.06	1.15
S6	0.00	1.00	0.36	0.58	1.00	X	2.94

Table 7.19: The other metrics of the Pareto optimal solution set from the experiment with ABC-SPEA2 over the bi-objective problem.

Scenarios	HV	Epsilon	GD	IGD	Spread	Gen.Spread
S1	0.68	0.00	**0.05**	**0.01**	0.70	**0.64**
S2	0.00	0.00	NaN	NaN	NaN	NaN
S3	**0.99**	0.00	0.08	0.09	0.65	0.70
S4	0.80	0.00	0.09	0.06	0.77	0.78
S5	0.00	0.00	NaN	NaN	NaN	NaN
S6	**0.99**	0.00	0.07	0.06	0.65	0.70

multi-objective algorithm performance metrics are given in Table 7.21 and Table 7.22. As shown by Table 7.21 and Table 7.22, in terms of coverage metric and the other metrics, it is clear that NSMA-II based hyper-heuristic outperforms the ABC-SPEA2 based hyper-heuristic.

Table 7.20: The objective function values of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on ABC-SPEA2 (S1) and NSMA-II (S2) over the bi-objective problem.

Algorithm	f1			f2		
	min	max	avg	min	max	avg
S1	1246	8546	1832.55	0.6108	0.7459	0.6755
S2	1246	1989	1446.5	0.6138	0.7458	0.6921

7.4 Experimental Results for Multi-Objective Examination Timetabling Problem with Multiple Stakeholder Perspectives

The second problem over with our proposed population-based hyper-heuristics, i.e. NSMA-II based hyper-heuristic (see Algorithm 14) and ABC-SPEA2 based hyper-heuristic(see Algorithm 18), is the multi-objective examination timetabling problems with multiple stakeholder perspectives. As discussed in Section 4.2.3 of Chapter 4, there are two variants of the problem, i.e. with two objectives: (f1) Students, (f2) Admins; and with

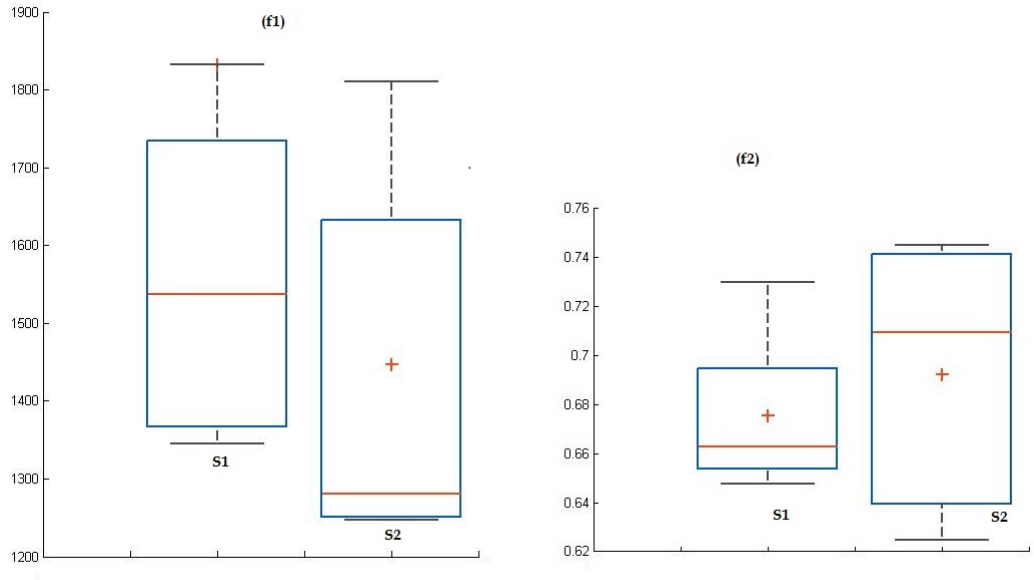


Figure 7.5: The box plot of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on ABC-SPEA2 (S1) and NSMA-II (S2) over the bi-objective problem.

Table 7.21: The coverage metric of of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on ABC-SPEA2 (S1) and NSMA-II (S2) over the bi-objective problem.

P. Instance	$\mathcal{C}$(S1,S2)	$\mathcal{C}$(S2,S1)
EXAM9	0.04	**0.55**

Table 7.22: The other metrics of of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on ABC-SPEA2 (S1) and NSMA-II (S2) over the bi-objective problem.

Instance	HV		Epsilon		GD		IGD		Spread		Gen. Spread	
	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2
EXAM9	0.71	**0.95**	0.06	**0.01**	0.01	**0.00**	**0.02**	0.03	1.36	**1.02**	1.55	**0.97**

four objectives: (f1) Students, (f2) Invigilators, (f3) Markers, (f4) Estates. The following sections reports and discusses the experiment of the proposed hyper-heuristics over the two variants of the multi-objective examination timetabling problems with multiple stakeholder perspectives.

7.4.1 Experimental Results with NSMA-II base Hyper-heuristic

Firstly, we conducted some preliminary experiments to determine a considerably good parameter setting of the NSMA-II based hyper-heuristic over the multi-objective examination timetabling problem with multiple multiple stakeholder perspectives. The parameter setting from the preliminary experiments is presented in Table 7.23. With the parameter setting given in Table 7.23 we tested the algorithm over five problem instances, i.e. `EXAM4, EXAM6, EXAM9, EXAM10, and EXAM10` of both variants of the multi-objective examination timetabling problems with multiple stakeholder perspectives. Furthermore, we compared the Pareto solution produced by the NSMA-II based hyper-heuristic with the scalarisation technique-based hyper-heuristic, i.e. self adaptive great deluge with weighted summation hyper-heuristic (SA-WSGD-HH) as shown in Algorithm 12. Table 7.24 and Table 7.25 present the comparison of the Pareto optimal solution from the experiment of the both hyper-heuristics in terms of multi-objective quality indicators widely used in the literature, over the two objective and the four objective problems respectively. In addition, the scatter plots in Figure 7.6 and Figure 7.7 visualise the objective function of the Pareto solution set from the experiment of both hyper-heuristics over bi-objective examination timetabling problem with multiple stakeholder perspectives.

Table 7.23: The parameters setting of the experiment over NSMA-II based hyper-heuristic over multi-objective examination timetabling problem with multiple stakeholder perspectives.

Parameter	Values
Population size (`N`)	256
Offspring size (`O`)	256
Mutation rate (`Mr`)	0.9
Number of generation (`G`)	10000
Initial population	The solution generated by constructor with improvement phase over the standard objective function (see Algorithm 3 in Chapter 3).

From Figures 7.6 and 7.7, it is clear that in the two problem problem, generally the SA-WSGD-HH outperforms NSMA-II. However, for spread and generalised spread metric, NSMA-II and SA-WSGD-HH are very competitive. Figures 7.6 and 7.7 also suggest that most solutions from NSMA-II are dominated by solutions from SA-WSGD-HH

approach. However, the solutions from NSMA-II are more widely spread out in the objective space.

Table 7.24: The quality metric of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on NSMA-II (S1) and SA-WSGD-HH (S2) over the bi-objective problem.

Instance	HV		Epsilon		GD		IGD		Spread		Gen. Spread	
	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2
EXAM4	0.48	**0.91**	1928.00	**0.00**	0.94	**0.00**	0.14	**0.00**	**0.86**	1.01	**0.94**	1.25
EXAM6	0.00	**0.65**	425.00	**0.00**	0.41	**0.00**	0.40	**0.00**	0.87	**0.64**	0.90	**0.71**
EXAM9	0.27	**0.79**	200.00	**0.00**	0.10	**0.00**	0.09	**0.00**	0.89	**0.86**	0.92	**0.88**
EXAM10	0.20	**0.30**	732.00	**0.00**	0.05	**0.00**	0.12	**0.00**	**0.91**	1.37	**0.85**	1.50
EXAM12	0.54	**0.68**	189.00	**0.00**	0.13	**0.00**	0.10	**0.00**	**0.76**	0.77	0.75	**0.73**

Table 7.25: The quality metric of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on NSMA-II (S1) and SA-WSGD-HH (S2) over the four-objective problem.

Instance	HV		Epsilon		GD		IGD		Spread		Gen. Spread	
	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2
EXAM4	**0.82**	0.34	**0.00**	3499.00	0.00	0.00	**0.01**	0.03	0.65	**0.60**	0.69	**0.68**
EXAM6	0.00	0.00	600.00	**0.00**	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
EXAM9	0.41	**0.81**	225.00	**0.00**	0.00	0.00	**0.01**	0.02	**0.44**	0.73	**0.61**	0.74
EXAM10	0.45	**0.49**	1676.00	**30.00**	0.00	0.00	0.01	**0.00**	**0.50**	0.75	**0.67**	0.70
EXAM12	**0.77**	0.42	144.00	**50.00**	0.00	0.00	0.10	**0.06**	0.60	**0.38**	0.72	**0.39**

7.4.2 Experimental Results with ABC-SPEA2 based Hyper-heuristic

In order to get good parameter settings of the ABC-SPEA based hyper-heuristic over multi-objective examination timetabling problem with multiple stakeholder perspectives, two preliminary experiments were conducted before the main experiment. Firstly, to evaluate the optimal number of scout bee that determine the number of current individual that will be replaced with newly randomly generated individual (see line `53` of Algorithm 18 and the method in Algorithm 20). For this purpose, we compared between using 1 scout bee as suggested in [18] and by using threshold. In threshold base, individuals that could not be improved for defined number of moves i.e. the parameter trial limit (`Tl`) are replaced with new randomly generated individuals. We set parameter `Tl` to `100`. Secondly, the preliminary experiment aims at evaluating the effect of local search low level heuristics within the ABC-SPEA2 based hyper-heuristic framework. For this purpose, we compared three scenarios: applying great deluge based local search only, applying late acceptance based local search only, and applying one local search randomly chosen from the five local search low-level heuristics given in Table 7.1.

Our preliminary experiment showed that the scout bee of size 1 combined with randomly chosen local search from the five local search heuristics are the best. Therefore, we used

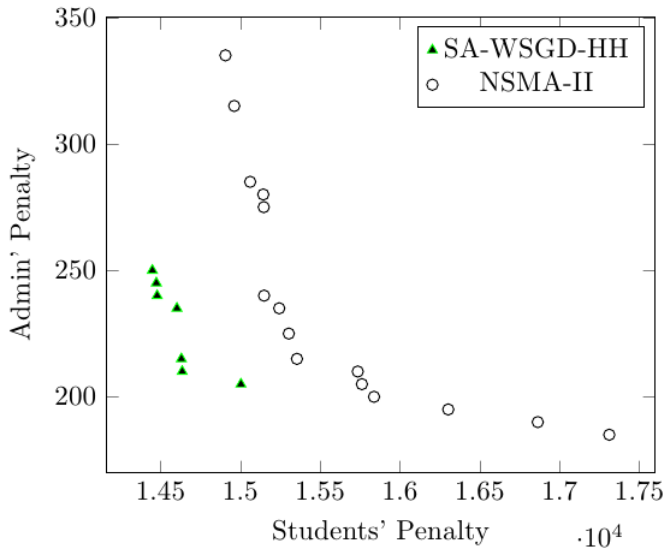


Figure 7.6: The scatter plot of the Pareto optimal solution set from the experiment of two different hyper-heuristics: NSMA-II (S1) and SA-WSGD-HH (S2) over Problem instance `EXAM10` of the bi-objective problem.

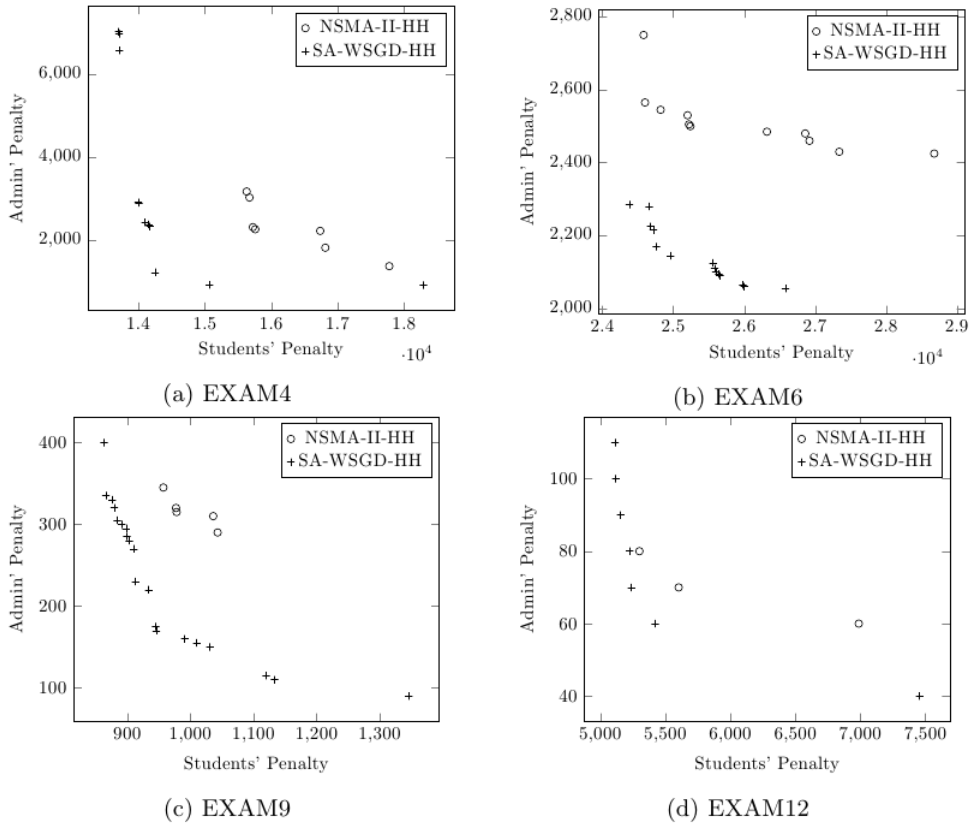


Figure 7.7: The scatter plot of the Pareto optimal solution set from the experiment of two different hyper-heuristics: NSMA-II (S1) and SA-WSGD-HH (S2) over the other four problem instances of the bi-objective problem.

Table 7.26: The parameters setting of the experiment of ABC-SPEA2-based hyper-heuristic over multi-objective examination timetabling problem with multiple stakeholder perspectives.

Parameter	Values
Population size (N)	50
Scout bee size	1
Length of low-level heuristic list(S), i.e. line 38 of Algorithm 18	200
Local search	A local search is chosen randomly from five local search low-level heuristics (see Table 7.1).

this setting (summarised in Table 7.26) in the main experiment over 5 problem instances of multi-objective examination timetabling problems with multiple stakeholder perspectives, either with two or the four objectives. We compared the Pareto optimal solution set from the experiment of ABC-SPEA2 and NSMA-II in terms of coverage metric in Table 7.27 and Table 7.28 for the two objective problem and the four objective examination problem respectively. Accordingly, results for the other metrics are presented in Table 7.29 and Table 7.30, respectively.

Table 7.27: The coverage metric of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on ABC-SPEA2 (S1) and NSMA-II(S2) over the bi-objective problem.

P.Instance	$\mathcal{C}$(S1,S2)	$\mathcal{C}$(S2,S1)
EXAM4	**1.00**	0.00
EXAM6	0.09	**0.67**
EXAM8	0.00	**1.00**
EXAM9	**0.40**	0.38
EXAM10	**0.47**	0.46
EXAM12	**1.00**	0.00

Table 7.28: The coverage metric of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on ABC-SPEA2 (S1) and NSMA-II (S2) over the four-objective problem.

P.Instance	$\mathcal{C}$(S1,S2)	$\mathcal{C}$(S2,S1)
EXAM4	**0.41**	0.00
EXAM6	**0.93**	0.00
EXAM8	**0.87**	0.12
EXAM9	**0.83**	0.04
EXAM10	**0.26**	0.16
EXAM12	**0.83**	0.00

Tables 7.27-7.30 suggest that generally, ABC-SPEA2 based hyper-heuristic (S1) outperforms NSMA-II based hyper-heuristic (S2). However, as shown by Tables 7.31-7.32, compared to the scalarisation technique-based hyper-heuristic (i.e. SA-WSGD-HH), ABC-SPEA2 is still inferior.

Furthermore, in order to evaluate the impact of extending the number of generations or

Table 7.29: The other quality metrics of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on ABC-SPEA2 (S1) and NSMA-II (S2) over the bi-objective problem.

Instance	HV		Epsilon		GD		IGD		Spread		Gen. Spread	
	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2
EXAM4	**0.77**	0.41	**0**	1174	**0.00**	0.11	**0.00**	0.08	0.74	**0.08**	**0.85**	0.88
EXAM6	0.60	**0.83**	130	**60**	0.06	**0.00**	0.07	**0.01**	0.88	**0.72**	0.90	**0.79**
EXAM8	**0.79**	0.00	**20**	943	**0.03**	1.53	**0.02**	1.43	**0.88**	0.97	**0.96**	
EXAM9	**0.60**	0.60	**21**	30	0.04	**0.03**	**0.03**	0.12	**0.58**	0.79	**0.55**	0.82
EXAM10	**0.79**	0.78	**20**	174	0.03	**0.02**	0.02	**0.02**	0.88	**0.64**	0.96	**0.68**
EXAM12	**0.76**	0.61	**0**	78	**0.00**	0.28	**0.00**	0.16	**0.58**	0.79	**0.64**	0.80

Table 7.30: The other quality metrics of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on ABC-SPEA2 (S1) and NSMA-II(S2) over the four-objective problem.

Instance	HV		Epsilon		GD		IGD		Spread		Gen. Spread	
	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2
EXAM4	**0.76**	0.77	**45**	2011	0.00	0.00	0.03	**0.01**	0.80	**0.66**	0.73	**0.70**
EXAM6	0.00	0.00	**50**	275	NaN	NaN	NaN	**NaN**	NaN	NaN	NaN	NaN
EXAM8	**0.76**	0.59	**50**	1500	0	0	0.019	**0.01**	**0.54**	0.59	0.78	**0.72**
EXAM9	**0.67**	0.72	**20**	143	0.00	0.00	0.01	**0.00**	0.46	**0.45**	**0.37**	0.59
EXAM10	**0.61**	0.72	**35**	1245	0.00	0.00	0.02	**0.00**	**0.36**	0.46	0.64	**0.57**
EXAM12	0.82	**0.59**	**10**	456	0.00	0.00	0.09	**0.05**	0.64	**0.53**	0.63	**0.49**

Table 7.31: The quality metric of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on ABC-SPEA2 (S1) and SA-WSGD-HH (S2) over the bi-objective problem.

Instance	Coverage		HV		GD		IGD		Spread		Gen. Spread	
	$\mathcal{C}$(S1,S2)	$\mathcal{C}$(S2,S1)	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2
EXAM4	0.00	**1.00**	0.74	**0.91**	0.07	**0.00**	0.08	**0.00**	**0.80**	1.01	**0.86**	1.25
EXAM6	0.00	**1.00**	0.00	**0.65**	0.60	**0.00**	0.49	**0.00**	0.98	**0.64**	0.98	**0.71**
EXAM9	0.00	**1.00**	0.28	**0.79**	0.08	**0.00**	0.08	**0.00**	0.86	0.86	0.90	**0.88**
EXAM10	0.00	**1.00**	0.17	**0.30**	0.06	**0.00**	0.12	**0.00**	**0.96**	1.37	**0.96**	1.50
EXAM12	0.00	**0.83**	**0.72**	0.68	0.03	**0.00**	0.08	**0.00**	**0.66**	0.77	**0.65**	0.73

Table 7.32: The quality metric of the Pareto optimal solution set from the experiment of two different hyper-heuristics based on ABC-SPEA2 (S1) and SA-WSGD-HH (S2) over the four-objective problem.

Instance	Coverage		HV		GD		IGD		Spread		Gen. Spread	
	$\mathcal{C}$(S1,S2)	$\mathcal{C}$(S2,S1)	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2
EXAM4	0.00	**1.00**	0.74	**0.91**	0.07	**0.00**	0.08	**0.00**	**0.80**	1.01	**0.86**	1.25
EXAM6	0.00	**1.00**	0.00	**0.65**	0.60	**0.00**	0.49	**0.00**	0.98	**0.64**	0.98	**0.71**
EXAM9	0.00	**1.00**	0.28	**0.79**	0.08	**0.00**	0.08	**0.00**	0.86	0.86	0.90	**0.88**
EXAM10	0.00	**1.00**	0.17	**0.30**	0.06	**0.00**	0.12	**0.00**	**0.96**	1.37	**0.96**	1.50
EXAM12	0.00	**0.83**	**0.72**	0.68	0.03	**0.00**	0.08	**0.00**	**0.66**	0.77	**0.65**	0.73

in other words the stopping condition in ABC-SPEA2 based hyper-heuristic (see line40 of Algorithm 18), we compared two scenarios. The first scenario uses 10000 generations as the stopping condition whereas the second scenario uses 20000 generations (double the amount in the first scenario) as the stopping condition. The results of the Pareto optimal solution set from the two scenarios are shown in Tables 7.33-7.36. It is obvious from Tables 7.33-7.36 that generally, as expected, the second scenario outperforms the first, suggesting that given more running time the solutions still could be improved.

Table 7.33: The coverage metric of the Pareto optimal solution set from the experiment of ABC-SPEA2 based hyper-heuristic with two scenarios of stopping generation over the bi-objective problem.

P.Instance	$\mathcal{C}$(S1,S2)	$\mathcal{C}$(S2,S1)
EXAM4	0.00	**0.95**
EXAM6	0.31	**0.70**
EXAM8	0.40	**0.60**
EXAM9	0.45	**0.75**
EXAM10	0.31	**0.73**
EXAM12	0.20	**0.80**

Table 7.34: The coverage metric of the Pareto optimal solution set from the experiment of ABC-SPEA2 based hyper-heuristic with two scenarios of stopping generation over the four-objective problem.

P.Instance	$\mathcal{C}$(S1,S2)	$\mathcal{C}$(S2,S1)
EXAM4	0.11	**0.68**
EXAM6	0.32	**0.50**
EXAM8	**0.40**	0.31
EXAM9	0.20	**0.47**
EXAM10	0.22	**0.34**
EXAM12	0.71	**1.29**

Table 7.35: The other metrics of the Pareto optimal solution set from the experiment of ABC-SPEA2 based hyper-heuristic with two scenarios of stopping generation over the bi-objective problem.

Instance	HV		Epsilon		GD		IGD		Spread		Gen. Spread	
	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2
EXAM4	0.59	**0.81**	717	**16**	0.04	**0.00**	**0.05**	0.06	0.78	**0.74**	**0.74**	0.82
EXAM6	0.82	**0.87**	**40**	80	**0.02**	0.04	0.03	**0.02**	0.87	**0.86**	0.91	**0.85**
EXAM8	0.01	**0.06**	113	**36**	**0.24**	0.86	0.28	**0.15**	**0.58**	0.96	**0.60**	1.08
EXAM9	0.76	**0.79**	**22**	25	0.03	**0.02**	0.08	**0.02**	0.74	**0.69**	0.78	**0.56**
EXAM10	0.70	**0.80**	368	**25**	0.04	**0.01**	0.04	**0.01**	**0.79**	0.88	0.87	**0.57**
EXAM12	0.50	**0.56**	10	10	0.03	**0.01**	0.03	**0.01**	**0.28**	0.41	**0.26**	0.39

Table 7.36: The other metrics of the Pareto optimal solution set from the experiment of ABC-SPEA2 based hyper-heuristic with two scenarios of stopping generation over the four-objective problem.

Instance	HV		Epsilon		GD		IGD		Spread		Gen. Spread	
	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2
EXAM4	0.68	**0.74**	**200**	365	0.00	0.00	0.01	0.01	**0.52**	0.59	0.53	**0.52**
EXAM6	0.00	**0.00**	75	**50**	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
EXAM8	**0.66**	0.65	116	**110**	0	0	0.01	0.01	0.49	**0.48**	0.64	**0.53**
EXAM9	0.62	**0.67**	**40**	50	0.00	0.00	0.01	0.01	0.43	0.43	**0.44**	0.57
EXAM10	0.72	**0.75**	121	**55**	0.00	0.00	0.01	0.01	0.45	**0.41**	0.52	**0.51**
EXAM12	**0.74**	0.70	**10**	52	0.00	0.00	0.02	0.02	**0.63**	0.64	**0.61**	0.71

7.5 Conclusion

This chapter studies hyper-heuristic approach with population based algorithms (`NSMA-II` and `ABC-SPEA2`) to solve the proposed new multi-objective examination timetabling problems. Some modification and hybridisation are carried out in order to make the standard algorithms work with the multi-objective examination timetabling problems as well as with hyper-heuristic framework. The proposed algorithms were tested over two multi-objective examination timetabling problems: (1) multi-objective examination timetabling problem with fairness and (2) multi-objective examination timetabling problem with multiple stakeholder perspectives as proposed in Chapter 4. Tested over a problem instance of multi-objective examination timetabling problem with fairness, the experimental results show good initial results and suggest that `NSMA-II` based hyper-heuristic outperforms `ABC-SPEA2` based hyper-heuristic, but it remains inferior to scalarisation technique based hyper-heuristic.

The limitation of the study over the multi-objective examination timetabling problem with fairness is that algorithms were tested over only one problem instance, which is a small problem instance. The proposed population-based algorithms (NSMA-II and ABC-SPEA2) resulted in run out memory for bigger problem instances. Studying on more efficient data structure to cope with this problem would be worthy of investigation in future work. Given the proposed proposed population-based algorithms were successful over all problem instances, we acknowledge that the results of comparing the algorithms NSMA-II. ABC-SPEA2, and scalarisation technique based approach, over the other problems might be different from the experimental results over multi-objective examination timetabling problem with fairness. The experimental results over several chosen problem instance of multi-objective examination timetabling problem with multiple stakeholder perspectives showed that ABC-SPEA2 based hyper-heuristic outperforms NSMA-II based hyper-heuristic.

However, similar to over multi-objective exam timetable with fairness, compared to the results from approach with classical techniques as discussed in Chapter 6, the population based hyper-heuristic is inferior. Of course, our proposed population based algorithm might well be improved and therefore it would be interesting to further investigate population based algorithm for highly constrained multi-objective combinatorial optimisation problems, such as the proposed multi-objective examination timetabling problems, as proposed in this study.

Chapter 8

Conclusions and Future Work

This chapter summarises all the research work investigating fairness in examination timetabling problems reported in this thesis. Section 8.1 highlights the main points of the overall research work as well as the main contributions, while Section 8.2 suggests some research directions for future work based on the main findings.

8.1 Research Summary

Overall, the main contribution of this thesis is that this study is the first study investigating fairness in examination timetabling problem, which can briefly be summarised as described below.

Before achieving the fairness objectives of the proposed examination timetabling, a study on solving the benchmark examination with standard objective function intensively studied in the scientific literature with hyper-heuristic approaches was carried out (Chapter 3). A hyper-heuristic approach was proposed to solve both benchmark uncapacitated examination timetabling problem instance (i.e. Carter dataset), and capacitated examination timetabling problem instances (i.e. ITC 2007, Yeditepe and Nottingham datasets). Chapter 3 reported three different initial feasible construction methods: sequential construction with maximal clique and saturation degree heuristic, adaptive ordering heuristic with hierarchical hybridisation of basic graph colouring heuristics, and adaptive linear combination ordering heuristic with heuristic modifier. These were studied over examination timetabling problem instances from four different

datasets.

The experimental results suggest that adaptive linear combination ordering heuristic with heuristic modifier is the most effective method capable of constructing a feasible solution for all problem instances from the four different datasets. Within adaptive linear combination ordering heuristic with heuristic modifier, the prediction of difficulty level (in terms of finding a feasible timeslot and room) is not only based on the standard graph-colouring notions as in the other approaches, but also adaptively changed based on how difficult to find feasible timeslot/room during the process of assignment. If no feasible timeslot/room can be found for an exam during an iteration, the difficulty index of the exam will be increased in the next iteration. We believe this dynamic difficulty prediction makes adaptive linear combination ordering heuristic with heuristic modifier more effective than the other approaches.

Moreover, to improve the initial feasible solution, a selection hyper-heuristic with different strategies combining different low-level heuristic selection strategies (i.e. simple random, reinforcement learning and self-adaptive learning) with different move acceptance strategies (i.e. great deluge algorithm, hill climbing, and late acceptance) were studied. The experimental results for uncapacitated examination timetabling problem suggest that the combination of self-adaptive learning and great deluge algorithm as low- level heuristic selection and move acceptance strategies (respectively) in SA-GD-HH outperforms the other combinations. Compared to results from other methods recently reported in the literature, `SA-GD-HH` produces very competitive results. It outperforms the other approaches over 8 of 13 problem instances. On other hand, over capacitated examination timetabling problem `SA-GD-HH` outperforms the other approaches, but over Carter datasets (which have been modified to the capacitated examination timetabling problem), `SR-GD-HH` surprisingly outperforms the other approaches. This suggests that the success of hyper-heuristic approach depends on the the quality of problem specific low-level heuristic in addition to the low-level heuristic selection and move acceptance strategy.

The main contribution of Chapter 3 was to present a hyper-heuristic method with self-adaptive learning as a low-level heuristic selection strategy and great deluge algorithm as move acceptance strategy to solve examination timetabling problem. Chapter 3 also showed that our proposed basic hyper-heuristic approach, over which the algorithm will be extended to solve examination timetabling with fairness, is very competitive with other approaches reported in the scientific literature. Although the improvements over the best known solutions (in terms of standard objective function) are of small numerical value, this does not mean the improvement is insignificant. The value is the average

timetable improvement of hundreds or even thousands of students. For example, the improvement of 0.01 over problem instance PUR93, given that the number of students is 30029 (see Table 2.1), could mean improvement over timetables of 300 students. Doing this improvement manually is impractical, since there are too many options of moving 2419 exams, some of which are conflicting, to 42 different timeslots. Choosing one of 2419 x 42 =101598 options to improve the timetable is almost impossible to do manually.

In order to know the difference between the current examination timetabling problem models widely studied in scientific literature and the real-world examination timetabling problems, especially regarding fairness issues, as reported in user preferences in Appendix B, a survey was carried out amongst undergraduate and taught-postgraduate students to investigate in more detail their own experiences and preferences for particular properties of exam timetables. The findings from the survey reveal that there are some critical issues lacking attention in the existing examination timetabling model in scientific literature. Firstly, regarding fairness, the findings from the survey confirm our thesis that fairness is indeed a concern to the students. In particular, it was apparent that students are mainly concerned about fairness within their 'cohort' which in this case refers to students in the same course and year. Secondly, each exam has a (perceived) different level of difficulty, however in the existing problem model all exams are assumed have the same difficulty level. For harder exams, students expect more preparation time or breaks.

To contribute in the effort of generating examination timetables that match student preferences and enhance their satisfaction, a new examination timetabling problem model that is believed more representative to the real-world examination timetabling problem structure is proposed in Chapter 4. Basically, the new model is an extension of the formulation of examination timetabling used in the International Timetabling Competition 2007 (ITC 2007) [207] by including objectives for fairness. The new formulation covers multiple objectives, including for fairness between all students and within specified cohorts.

To cope with exams with different levels of difficulty, we proposed ways to handle it by introducing a difficulty index associated with each exam. Initially we proposed the index as an integer with values 1 representing 'easy', 2 representing 'medium', and 3 representing 'hard'. This was not studied further in this thesis due to the lack of real-world data, but it would be useful for academic systems to try to collect needed data, e.g. by giving the students some form of vote on the exams for which they want the largest gap beforehand. The credits for exams could be taken into account too. However, we

believe that it is probably straightforward to merely change some weights of existent terms for existing algorithms.

Regarding the notion of fairness, in addition to the notion of fairness between all students or global fairness and fairness amongst students within specified cohorts, fairness amongst students within exam-count-based cohort is also proposed. The latest notion of fairness is intended to consider the fact that though students are in the same specified cohort, e.g. in the same school and in the same year, they most likely enrolled in a different number of exams.

In addition to the new examination timetabling problem formulation, the work in Chapter 4 also contributes initial work towards building a public repository[1] that extends current benchmark instances with the information needed to build cohorts. This allows the examination timetabling problems on our formulation to be studied by the community.

In the effort to enforce fairness in examination timetabling problems in addition to the standard objective functions within hyper-heuristic framework, four approaches are investigated: (1) 'sum of powers'; (2) two-stage; (3) multi-objective with iterated classical scalarisation; and (4) multi-objective with population-based algorithm.

The first attempt to enforce fairness with 'sum of powers' approach is discussed in Section 4.3 . `SA-GD-HH` hyper-heuristic approach was used for single objective examination timetabling problems, as employed in Chapter 3, with the difference of the objective function. Instead of linear summation function of penalty associated to each individual student, as in the original formulation, in sum of powers, the objective function is defined as summation of penalty associated to each individual student power exponent number `q`. Two values for the exponents are evaluated, i.e. `2` (later, it is named 'sum of squares') and `16`.

The experimental results in Section 4.3 showed that 'sum of powers' approach could optimise both the standard objective function and 'fairness between all students' objective function concurrently. Given the same running time, the higher exponential number makes the solution fairer. However, over almost all problem instances from Carter dataset, fairer solutions compensate worse standard objective function. The trade-off between the standard objective function and fairness objective function varies among different problem instances. On average, given the same number of iteration as stopping condition, 'sum of squares' could make the solutions fairer by `5.47%` and and compensates on worsening the standard objective function by `2.79%`; 'sum of powers'

[1] http://www.cs.nott.ac.uk/∼abm/research/exam-fairness-data/examproblemwithfairness/

with exponent number, q=16, could make the solutions fairer by 6.99% and compensates on worsening the standard objective function by 25.37%. This finding suggest that within 'sum of powers', exponent number, q=2 is more effective than exponent number, q=16.

The limitation of 'sum of powers' approach is that compared to the original linear summation it requires significantly higher computational time (approximately 28 higher times longer). Moreover, this approach ends with a single solution, therefore a decision maker could not evaluate multiple solutions with different trade-off between the standard objective function and fairness objective function, which might be essential in making a final decision.

Section 4.3 also evaluated different fairness quality indicators, comprising anti-Jain fairness index (AJFI), variance, anti-Minmax, GINI index, and relative standard deviation (RSD). The evaluation showed that AJFI, GINI index, and RSD are positively correlated, therefore these fairness quality indicators could be used inter-changeably.

In Chapter 5, two-stages approach was used for more efficient computational time compared to optimising the standard objective function and fairness objective function concurrently, as in Section 4.3. The first stage is the same as discussed in Chapter 3 and aims at optimising the standard objective function, whereas in the second stage the objective function is changed to fairness objective function (i.e. Jain fairness index), and we add not worsening the standard objective function as a hard constraint. Three different notions of fairness (fairness between all students, fairness amongst students within the same specified cohort, and fairness amongst students within 'exam-count-based cohort') underwent optimisation in the second stage, with reinforcement learning as low-level heuristic selection strategy to study three move acceptance strategies: lexicographic, hill climbing and great deluge.

The experimental results in Chapter 5 showed that global and cohort fairness could not be significantly improved over all problem instances without worsening the standard objective functions. Fairness could be negligibly improved over less than 1% of problem instances, revealing the intrinsic trade-off between the standard objective function and fairness objective function. On the other hand, if the standard objective function is allowed to deteriorate by less than or equal to 5%, over all problem instances, the fairness objective function could be improved. The significance of improvement varies amongst different problem instances, but over a few of them the fairness objective function could be improved by more than 5%.

The experimental results also showed that fairness amongst students within 'specified

cohort' could be improved more significantly than fairness between all students, and fairness amongst student within 'exam-count-based cohort' could be improved more significantly than fairness amongst students within 'specified cohort'. This finding nicely aligns with the students desires revealed in the survey reported in Appendix B: students are more concerned by fairness amongst students within the same cohort than fairness between all students.

Lastly, in terms of algorithm, compared to great deluge and lexicographic, hill climbing is the most effective to enforce fairness in the second stage of the algorithm. Since, during the search hill climbing only accepts a better solution, the trajectory search analysis suggests that within the same running time hill climbing could find much better solution than solutions from the other approaches. We believe it is the factor that make hill climbing outperforms the other approaches.

In Chapter 6, following up the experimental results from Chapter 5 where standard objective function and fairness objective function were conflicting, in the third attempt the problem was treated as a multi-objective optimisation problem. Two classical scalarisation techniques (weighted sum and weighted Tchebycheff combined with Pareto sorting algorithm) were employed within reinforcement learning/ self-adaptive learning - great deluge-hyper-heuristic approach. The proposed hyper-heuristics were tested over two variants of the multi-objective examination timetabling with fairness, i.e. bi-objectives: (1) standard objective function, (2) fairness between all students; and three-objectives: (1) standard objective function, (2) fairness between all students, (3) fairness amongst students within cohort. Moreover, the proposed hyper-heuristics were also tested over two variants of the multi-objective examination timetabling problem with multiple stakeholders perspectives, i.e. bi-objectives: (1) students, (2) administrators; and four-objectives: (1) students, (2) marker, (3) invigilators, and (4) estates. Different settings of algorithm were studied intensively this chapter.

The experimental results from Chapter 6 that the proposed hyper-heuristics were very effective to generate a set of (approximation) Pareto optimal solutions consisting of solutions with different trade-offs between objective functions, rather than single solution as in the earlier chapters. Within this approach, it was also shown that the standard objective function could be improved further, whereas in the earlier chapters it was not possible. As a side-effect, the proposed multi-objective approach also could generate best-known solutions of the benchmark problem over two problem instances of ITC 2007 datasets. In addition to the proposed hyper-heuristic, another contribution of research work from this chapter is the extension of the problem domain interface of the current version of Hyflex by adding new methods in order to cope with multi-objective

problems, especially with classical scalarisation techniques.

Chapter 7 investigated two population-based hyper-heuristic approaches (non-dominated sorting memetic algorithmic II (NSMA-II) algorithm and hybridisation of artificial bee colony with SPEA 2 algorithm (ABC-SPEA2)) to harness a population-based algorithm reported to be effective to solve many multi-objective optimisation problems in the scientific literature. Some modifications on the original version of the algorithms were made in order to cope with multi-objective examination timetabling problem. The proposed algorithms were tested over two multi-objective examination timetabling problems: (1) multi-objective examination timetabling problem with fairness, (2) multi-objective examination timetabling problem with multiple stakeholder perspectives.

The experimental results over the problem instances show good initial results. However, compared to the results from approach with classical techniques as discussed in Chapter 6, the population-based hyper-heuristic are inferior. This finding leaves further investigation of population-based algorithm for highly constrained multi-objective combinatorial optimisation problem, such as examination timetabling problem, as future work.

It is also note worthy that despite the improvements in terms of fairness over the best known solution (in terms of standard) manifesting small numerical values, this does not mean that such improvements are insignificant. The small magnitude of the values reflects that fairness is measured in Jain fairness index with values ranging between `1/N` to 1, in which `N` is the total number of students. The index is $\frac{k}{N}$ when k students shared equal penalty, and the rest, $n-k$ get zero penalty. In other words, given that the others get zero penalty, the index $\frac{k}{N}$ is achieved when it is fair over k students. Since there are thousands students in the problem instances, the improvement could be significant. In addition, doing the improvement manually is also impractical, since there are too many options of moving exams to different timeslots (for example over problem instance `EXAM4`, there are `21 x 273 =5733` options), subject to defined hard constraints that are hard to be checked manually.

8.2 Future Work

From the findings of the research work reported in this thesis, some suggestions for future research directions were identified, as presented below.

Regarding multi-objective examination timetabling problem model, this research work treated separately two variants of multi-objective examination timetabling (multi-objective

examination timetabling with fairness and multi-objective examination timetabling with multiple stakeholder perspectives). Future work could build more complex formulations to consider penalties associated with different stakeholders, such as different objective function in the first level, and consider fairness amongst the same stakeholders (e.g. between students and invigilators) in the second level.

In supporting the new more complex examination timetabling problem formulations, we strongly encourage people who share public dataset in the future to keep its meta-data. The meta-data could be information about student courses and years, and examination schools and faculties etc. Such meta-data would be useful to define 'cohorts'. Information about the difficulty of each exam gathered from students' opinions as well as information about the total credit of each exam could also be useful for defining the difficulty index.

In the study reported in this thesis, in addition to fairness between all students, two notions of fairness in examination timetabling problems were proposed: fairness amongst students within a 'specified cohort', and fairness amongst students within a 'exam-count-based cohort' were proposed. In the first notion of cohort students are clustered by their course and year of entrance, whereas in the second students are clustered by the number of their exams. However, we believe that more flexible cohort notions could be proposed in the future work. For instance, suppose N is a set of all students considered in the examination timetabling problem, P_{s_i} and P_{s_j} are penalties associated to students s_i and s_j respectively, and $S(s_i, s_j)$ is the similarity function between students s_i and s_j, in this context,'soft cohort' or 'fuzzy cohort' could be defined as follows:

$$\sum_{s_i \neq s_j, s_i \in N, s_j \in N} S(s_i, s_j) \left| P_{s_i} - P_{s_j} \right| \tag{8.1}$$

The similarity function, $S(s_i, s_j)$ could be a fuzzy function combining different notions of fairness. Furthermore, considering different notions of fairness, in the future work, a novel examination timetabling problem with multiple fairness objectives could be investigated. Future work could also look at interaction between fairness and the different difficulty level of each exam. Considering fairness and taking into account the difficulty index might increase student satisfaction.

In future work further investigation on the algorithms could be useful to investigate either approximation algorithm or exact algorithm extensions of IP models, e.g. branch-and-cut versions as implemented for course timetabling problems in [68]. The hybridisation of both scalarisation techniques would also be interesting to investigate.

Though NSMA-II algorithm was well-known and effective for solving many multi-objective optimisation problems, our investigation over multi-objective examination timetabling problem revealed that this algorithm is not effective, possibly because the structure of examination timetabling problem is a highly constrained combinatorial optimisation problem. It also suggests that different type of multi-objective problems require different types of evolutionary multi-objective algorithms. Therefore, further investigation on population-based algorithm for solving this difficult combinatorial optimisation problem could be interesting for future research work.

References

[1] Ariff Md Ab Malik, Masri Ayob, and Abdul Razak Hamdan. Iterated two-stage multi-neighbourhood tabu search approach for examination timetabling problem. In 2^{nd} *Conference on Data Mining and Optimization 2009*, pages 147–153, 2009.

[2] Syariza Abdul-Rahman, Andrzej Bargiela, Edmund K. Burke, Ender Özcan, and Barry McCollum. Construction of examination timetables based on ordering heuristics. In 24^{th} *International Symposium on Computer and Information Sciences, 2009. ISCIS 2009.*, pages 680–685, 2009.

[3] Syariza Abdul-Rahman, Andrzej Bargiela, Edmund K. Burke, Ender Özcan, Barry McCollum, and Paul McMullan. Adaptive linear combination of heuristic orderings in constructing examination timetables. *European Journal of Operational Research*, 232(2):287–297, 2014.

[4] Syariza Abdul-Rahman, Edmund K. Burke, Andrzej Bargiela, Barry McCollum, and Ender Özcan. A constructive approach to examination timetabling based on adaptive decomposition and ordering. *Annals of Operations Research*, 218(1):3–21, 2014.

[5] Salwani Abdullah, S. Ahmadi, Edmund K. Burke, M. Dror, and B. McCollum. A tabu-based large neighbourhood search methodology for the capacitated examination timetabling problem. *J Oper Res Soc*, 58(11):1494–1502, 2006.

[6] Salwani Abdullah, Samad Ahmadi, Edmund K. Burke, and Moshe Dror. Investigating Ahuja-Orlin's large neighbourhood search approach for examination timetabling. *OR Spectrum*, 29(2):351–372, 2007.

[7] Salwani Abdullah and Malek Alzaqebah. A hybrid self-adaptive bees algorithm for examination timetabling problems. *Applied Soft Computing*, 13(8):3608–3620, 2013.

[8] Salwani Abdullah and Hamza Turabieh. On the use of multi neighbourhood structures within a tabu-based memetic approach to university timetabling problems. *Information Sciences*, 191:146–168, 2012.

[9] Salwani Abdullah, Hamza Turabieh, and Barry McCollum. A hybridization of electromagnetic-like mechanism and great deluge for examination timetabling problems. In MaraJ. Blesa, Christian Blum, Luca Di Gaspero, Andrea Roli, Michael Sampels, and Andrea Schaerf, editors, *Hybrid Metaheuristics*, volume 5818 of *Lecture Notes in Computer Science*, pages 60–72. Springer Berlin Heidelberg, 2009.

[10] Salwani Abdullah, Hamza Turabieh, Barry McCollum, and Paul McMullan. A tabu-based memetic approach for examination timetabling problems. In Jian Yu, Salvatore Greco, Pawan Lingras, Guoyin Wang, and Andrzej Skowron, editors, *Rough Set and Knowledge Technology*, volume 6401 of *Lecture Notes in Computer Science*, pages 574–581. Springer Berlin Heidelberg, 2010.

[11] M. A. Ahandani and M. T. Vakil-Baghmisheh. Examination timetabling using a hill climbing with combined neighbourhood structure. In *Computer and Knowledge Engineering (ICCKE), 2011 1st International eConference on*, pages 45–48, Oct 2011.

[12] Samad Ahmadi, Rossano Barone, Peter Cheng, Edmund K. Burke, Peter Cowling, and Barry McCollum. Perturbation based variable neighbourhood search in heuristic space for examination timetabling problem. In *multidisciplinary international scheduling: theory and applications (MISTA 2003)*, pages 155–171, 2003.

[13] Mohammed Azmi Al-Betar, Ahamad Tajudin Khader, and Iyad Abu Doush. Memetic techniques for examination timetabling. *Annals of Operations Research*, pages 1–28, 2013.

[14] Mohammed Azmi Al-Betar, Ahamad Tajudin Khader, and Iman Yi Liao. A harmony search with multi-pitch adjusting rate for the university course timetabling. In Zong Woo Geem, editor, *Recent Advances In Harmony Search Algorithm*, volume 270 of *Studies in Computational Intelligence*, pages 147–161. Springer Berlin Heidelberg, 2010.

[15] Bashar A. Aldeeb, Norita Md Norwawi, Mohammed A. Al-Betar, and Mohd Zalisham Bin Jali. Solving university examination timetabling problem using intelligent water drops algorithm. In Bijaya Ketan Panigrahi, Ponnuthurai Nagaratnam Suganthan, and Swagatam Das, editors, *Swarm, Evolutionary, and Memetic*

Computing, volume 8947 of *Lecture Notes in Computer Science*, pages 187–200. Springer International Publishing, 2015.

[16] Eitan Altman, Konstantin Avrachenkov, and Sreenath Ramanath. Multiscale fairness and its application to resource allocation in wireless networks. *Computer Communications*, 35(7):820–828, 2012.

[17] M. Alzaqebah and S. Abdullah. Artificial bee colony search algorithm for examination timetabling problems. *International Journal of the Physical Sciences*, 6(17):4264–4272, 2011.

[18] M. Alzaqebah and S. Abdullah. An adaptive artificial bee colony and late-acceptance hill-climbing algorithm for examination timetabling. *Journal of Scheduling*, 17(3):249–262, 2014.

[19] M. Alzaqebah and S. Abdullah. Hybrid bee colony optimization for examination timetabling problems. *Computers & Operations Research*, 54:142–154, 2015.

[20] Malek Alzaqebah and Salwani Abdullah. Hybrid artificial bee colony search algorithm based on disruptive selection for examination timetabling problems. In Weifan Wang, Xuding Zhu, and Ding-Zhu Du, editors, *Combinatorial Optimization and Applications*, volume 6831 of *Lecture Notes in Computer Science*, pages 31–45. Springer Berlin Heidelberg, 2011.

[21] Khairul Anwar, Ahamad T. Khader, M. A. Al-Betar, and M. A. Awadallah. Harmony search-based hyper-heuristic for examination timetabling. In *IEEE 9th International Colloquium on Signal Processing and its Applications (CSPA), 2013*, pages 176–181, 2013.

[22] Khairul Anwar, Ahamad T. Khader, MohammedAzmi Al-Betar, and MohammedA. Awadallah. Development on harmony search hyper-heuristic framework for examination timetabling problem. In Ying Tan, Yuhui Shi, and CarlosA.Coello Coello, editors, *Advances in Swarm Intelligence*, volume 8795 of *Lecture Notes in Computer Science*, pages 87–95. Springer International Publishing, 2014.

[23] Hishammuddin Asmuni, Edmund K. Burke, Jonathan M. Garibaldi, and Barry McCollum. Fuzzy multiple heuristic orderings for examination timetabling. In Edmund Burke and Michael Trick, editors, *Practice and Theory of Automated Timetabling V*, volume 3616 of *Lecture Notes in Computer Science*, pages 334–353. Springer Berlin Heidelberg, 2005.

[24] Hishammuddin Asmuni, Edmund K. Burke, Jonathan M. Garibaldi, and Barry McCollum. Determining rules in fuzzy multiple heuristic orderings for constructing examination timetables. In *Proceeding 3rd Multidisciplinary International Scheduling: Theory and Applications Conference (MISTA 2007)*, page 5966, 2007.

[25] Hishammuddin Asmuni, Edmund K. Burke, Jonathan M. Garibaldi, and Barry McCollum. A novel fuzzy approach to evaluate the quality of examination timetabling. In Edmund K. Burke and Hana Rudová, editors, *Practice and Theory of Automated Timetabling VI*, volume 3867 of *Lecture Notes in Computer Science*, pages 327–346. Springer Berlin Heidelberg, 2007.

[26] Hishammudin Asmuni, Edmund K. Burke, Jonathan M. Garibaldi, Barry McCollum, and Andrew J. Parkes. An investigation of fuzzy multiple heuristic orderings in the construction of university examination timetables. *Computers & Operations Research*, 36(4):981 – 1001, 2009.

[27] M. Atsuta, K. Nonobe, and T. Ibaraki. ITC2007 track 1: An approach using a general csp solver. Technical report, International Timetabling Competition 2007, 2008.

[28] Masri Ayob, Salwani Abdullah, and Ariff Md Ab Malik. A practical examination timetabling problem at the Universiti Kebangsaan Malaysia. *International Journal of Computer Science and Network Security*, 7(9):198–204, 2007.

[29] Masri Ayob, Edmund K. Burke, and Graham Kendall. An iterative re-start variable neighbourhood search for the examination timetabling problem. In *Proceedings of the 6th International Conference on The Practice and Theory of Automated Timetabling PATAT*, pages 336–344, 2006.

[30] Masri Ayob and Graham Kendall. A monte carlo hyper-heuristic to optimise component placement sequencing for multi head placement machine. In *Placement Machine, INTECH03 Thailand*, pages 132–141, 2003.

[31] Zahra N. Azimi. Hybrid heuristics for examination timetabling problem. *Applied Mathematics and Computation*, 163(2):705–733, 2005.

[32] Ruibin Bai, Edmund K. Burke, Michel Gendreau, and Graham Kendall. A simulated annealing hyper-heuristic: Adaptive heuristic selection for different vehicle routing problems. In 3^{rd} *Multidisciplinary International Conference on Scheduling: Theory and Applications*, pages 67–70, 2007.

[33] Ruibin Bai and Graham Kendall. An investigation of automated planograms using a simulated annealing based hyper-heuristic. In *Metaheuristics: Progress as real problem solvers*, pages 87–108. Springer, 2005.

[34] James E. Baker. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*, pages 14–21, Hillsdale, NJ, USA, 1987. L. Erlbaum Associates Inc.

[35] Nagraj Balakrishnan, Abilio Lucena, and Richard T. Wong. Scheduling examinations to reduce second-order conflicts. *Computers & Operations Research*, 19(5):353 – 361, 1992.

[36] Sanghamitra Bandyopadhyay, Sriparna Saha, Ujjwal Maulik, and Kalyanmoy Deb. A simulated annealing-based multiobjective optimization algorithm: Amosa. *IEEE transactions on evolutionary computation*, 12(3):269–283, 2008.

[37] Z. Moshe Berrebi and Jacques Silber. Income inequality indices and deprivation: A generalization. *The Quarterly Journal of Economics*, pages 807–810, 1985.

[38] Z. Moshe Berrebi and Jacques Silber. Regional differences and the components of growth and inequality change. *Economics Letters*, 25(3):295–298, 1987.

[39] Dimitri Bertsekas and Robert Gallager. *Data Networks*. Prentice-Hall, Inc., New Jersey, USA, 1992.

[40] Dimitris Bertsimas, Vivek F. Farias, and Nikolaos Trichakis. The price of fairness. *Operations Research*, 59(1):17–31, 2011.

[41] Burak Bilgin, Ender Özcan, and Emin Erkan Korkmaz. An experimental study on hyper-heuristics and exam timetabling. In Edmund K. Burke and Hana Rudova, editors, *Practice and Theory of Automated Timetabling VI*, volume 3867 of *Lecture Notes in Computer Science*, chapter 25, pages 394–412. Springer Berlin Heidelberg, 2007.

[42] Tobias Blickle and Lothar Thiele. A mathematical analysis of tournament selection. In *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 9–16, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[43] Alex Bonutti, Fabio De Cesco, Luca Di Gaspero, and Andrea Schaerf. Benchmarking curriculum-based course timetabling: formulations, data formats, instances, validation, visualization, and results. *Annals of Operations Research*, 194(1):59–70, 2012.

[44] Sally C. Brailsford, Chris N. Potts, and Barbara M. Smith. Constraint satisfaction problems: Algorithms and applications. *European Journal of Operational Research*, 119(3):557 – 581, 1999.

[45] Daniel Brélaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256, April 1979.

[46] Bernd Bullnheimer. An examination scheduling model to maximize students study time. In Edmund Burke and Michael Carter, editors, *Practice and Theory of Automated Timetabling II*, volume 1408 of *Lecture Notes in Computer Science*, pages 78–91. Springer Berlin Heidelberg, 1998.

[47] Edmund K. Burke and Y. Bykov. Solving exam timetabling problems with the flex-deluge algorithm. In *6th International Conference on the Practice and Theory of Automated Timetabling*, pages 370–372, 2006.

[48] Edmund K. Burke and Y Bykov. A late acceptance strategy in hill-climbing for exam timetabling problems. In *7th international conference on the practice and theory of automated timetabling (PATAT:2008)*, August 2008 2008.

[49] Edmund K. Burke and Y. Bykov. The late acceptance hill-climbing heuristic. Technical report, Computing Science and Mathematics, University of Stirling, 2012.

[50] Edmund K. Burke, Yuri Bykov, James Newall, and Sanja Petrovic. A time-predefined local search approach to exam timetabling problems. *IIE Transactions*, 36(6):509–528, 2004.

[51] Edmund K. Burke, Yuri Bykov, and Sanja Petrovic. A multicriteria approach to examination timetabling. In Edmund Burke and Wilhelm Erben, editors, *Practice and Theory of Automated Timetabling III*, volume 2079 of *Lecture Notes in Computer Science*, pages 118–131. Springer Berlin Heidelberg, 2001.

[52] Edmund K. Burke, Tim Curtois, Matthew Hyde, Graham Kendall, Gabriela Ochoa, Sanja Petrovic, José A. Vázquez-Rodríguez, and Michel Gendreau. Iterated local search vs. hyper-heuristics: Towards general-purpose search algorithms. In *IEEE congress on evolutionary computation*, pages 1–8. IEEE, 2010.

[53] Edmund K. Burke, Moshe Dror, Sanja Petrovic, and Rong Qu. Hybrid graph heuristics within a hyper-heuristic approach to exam timetabling problems. In Bruce Golden, S. Raghavan, and Edward Wasil, editors, *The Next Wave in Computing, Optimization, and Decision Technologies*, volume 29 of *Operations Research/Computer Science Interfaces Series*, pages 79–91. Springer US, 2005.

[54] Edmund K. Burke, A. J. Eckersley, B. McCollum, S. Petrovic, and R. Qu. Hybrid variable neighbourhood approaches to university exam timetabling (technical report nottcs-tr-2006-2). Technical report, School of Computer Science, University of Nottingham, 2006.

[55] Edmund K. Burke, Adam J. Eckersley, Barry McCollum, Sanja Petrovic, and Rong Qu. Hybrid variable neighbourhood approaches to university exam timetabling. *European Journal of Operational Research*, 206(1):46–53, 2010.

[56] Edmund K. Burke, Dave Elliman, Peter Ford, and Rupert Weare. Examination timetabling in British universities: A survey. In *Practice and Theory of Automated Timetabling*, volume 1153 of *Lecture Notes in Computer Science*, pages 76–90. Springer Berlin Heidelberg, 1996.

[57] Edmund K. Burke, Dave Elliman, and Rupert F. Weare. A hybrid genetic algorithm for highly constrained timetabling problems, 1995.

[58] Edmund K. Burke, David Elliman, and Rupert Weare. A genetic algorithm based university timetabling system. In *East-West Conference on Computer Technologies in Education, Crimea, Ukraine*, pages 35–40, 1994.

[59] Edmund K. Burke, Michel Gendreau, Matthew Hyde, Graham Kendall, Gabriela. Ochoa, Ender Özcan, and Rong Qu. Hyper-heuristics: a survey of the state of the art. *J Oper Res Soc*, 64(12):1695–1724, Dec 2013.

[60] Edmund K. Burke, Matthew Hyde, Graham Kendall, Gabriela Ochoa, Ender Ozcan, and Rong Qu. A survey of hyper-heuristics. Technical report, University of Nottingham, 2009.

[61] Edmund K. Burke, Matthew Hyde, Graham Kendall, Gabriela Ochoa, Ender Özcan, and John R Woodward. A classification of hyper-heuristic approaches. In Michel Gendreau and Jean-Yves Potvin, editors, *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science*, chapter 15, pages 449–468. Springer US, 2010.

[62] Edmund K. Burke, Graham Kendall, Dario Landa-Silva, Ross O'Brien, and Eric Soubeiga. An ant algorithm hyperheuristic for the project presentation scheduling problem. In *The 2005 IEEE Congress on Evolutionary Computation, 2005*, volume 3, pages 2263–2270, 2005.

[63] Edmund K. Burke, Graham Kendall, Mustafa Misir, and Ender Özcan. Monte Carlo hyper-heuristics for examination timetabling. *Annals of Operations Research*, 196(1):73–90, 2012.

[64] Edmund K. Burke, Graham Kendall, Mustafa Mısır, Ender Özcan, EK Burke, G Kendall, E Özcan, and M Mısır. Applications to timetabling. In *Handbook of Graph Theory, chapter 5.6*, pages 445–474. Citeseer, 2004.

[65] Edmund K. Burke, Graham Kendall, Jim Newall, Emma Hart, Peter Ross, and Sonia Schulenburg. Hyper-Heuristics: An emerging direction in modern search technology. In Fred Glover and GaryA. Kochenberger, editors, *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research & Management Science*, pages 457–474. Springer US, 2003.

[66] Edmund K. Burke, Graham Kendall, and Eric Soubeiga. A tabu-search hyperheuristic for timetabling and rostering. *Journal of Heuristics*, 9(6):451–470, 2003.

[67] Edmund K. Burke and Dario Landa-Silva. The design of memetic algorithms for scheduling and timetabling problems. In William E. Hart, J.E. Smith, and N. Krasnogor, editors, *Recent Advances in Memetic Algorithms*, volume 166 of *Studies in Fuzziness and Soft Computing*, pages 289–311. Springer Berlin Heidelberg, 2005.

[68] Edmund K. Burke, Jakub Mareček, Andrew J. Parkes, and Hana Rudová. A branch-and-cut procedure for the udine course timetabling problem. *Annals of Operations Research*, 194(1):71–87, 2012.

[69] Edmund K. Burke, Barry McCollum, Paul McMullan, and Andrew J. Parkes. Multi-objective aspects of the examination timetabling competition track. In *Proceedings of PATAT*, pages 3119–3126, 2008.

[70] Edmund K. Burke, Barry McCollum, Amnon Meisels, Sanja Petrovic, and Rong Qu. A graph-based hyper-heuristic for educational timetabling problems. *European Journal of Operational Research*, 176(1):177–192, 2007.

[71] Edmund K. Burke and James Newall. A multistage evolutionary algorithm for the timetable problem. *IEEE Transactions on Evolutionary Computation*, 3(1):63–74, 1999.

[72] Edmund K. Burke and James P. Newall. Enhancing timetable solutions with local search methods. In Edmund K. Burke and Patrick De Causmaecker, editors, *Practice and Theory of Automated Timetabling IV*, volume 2740 of *Lecture Notes in Computer Science*, pages 195–206. Springer Berlin Heidelberg, 2003.

[73] Edmund K. Burke and James P. Newall. Solving examination timetabling problems through adaption of heuristic orderings. *Annals of Operations Research*, 129(1-4):107–134, 2004.

[74] Edmund K. Burke, James P. Newall, and Rupert F. Weare. A memetic algorithm for university exam timetabling. In Edmund Burke and Peter Ross, editors, *Practice and Theory of Automated Timetabling*, volume 1153 of *Lecture Notes in Computer Science*, pages 241–250. Springer Berlin Heidelberg, 1996.

[75] Edmund K. Burke, James P. Newall, and Rupert F. Weare. Initialization strategies and diversity in evolutionary timetabling. *Evolutionary Computation*, 6(1):81–103, 1998.

[76] Edmund K. Burke, James P. Newall, and Rupert F. Weare. A simple heuristically guided search for the timetable problem. In *In: Proceedings of the International ICSC Symposium on Engineering of Intelligent Systems, Univ. of La Laguna*, pages 574–579. Academic Press, 1998.

[77] Edmund K. Burke and T. C. Pais. Using differential evolution to identify fuzzy measures for the exam timetabling problem. In J. Fowler, G. Kendall, and B. McCollum, editors, *In proceedings of the 5th Multidisciplinary International Conference on Scheduling : Theory and Applications (MISTA 2011), 9-11 August 2011, Phoenix, Arizona, USA*, pages 335–351, 2011.

[78] Edmund K. Burke and Sanja Petrovic. Recent research directions in automated timetabling. *European Journal of Operational Research*, 140(2):266–280, 2002.

[79] Edmund K. Burke, Sanja Petrovic, and Rong Qu. Case-based heuristic selection for timetabling problems. *J. of Scheduling*, 9(2):115–132, April 2006.

[80] Edmund K. Burke, Nam Pham, Rong Qu, and Jay Yellen. Linear combinations of heuristics for examination timetabling. *Annals of Operations Research*, 194(1):89–109, 2012.

[81] Edmund K. Burke, Rong Qu, and A. Soghier. Adaptive selection of heuristics for improving constructed exam timetables. In *PATAT 2010*, page 136151, 2010.

[82] Edmund K. Burke, Rong Qu, and Amr Soghier. Adaptive selection of heuristics for improving exam timetables. *Annals of Operations Research*, 218(1):129–145, 2014.

[83] S. Cahon, N. Melab, and E. G. Talbi. ParadisEO: A framework for the reusable design of parallel and distributed metaheuristics. *Journal of Heuristics*, 10(3):357–380, 2004.

[84] Massimiliano Caramia, Paolo Dell'Olmo, and Giuseppe F. Italiano. New algorithms for examination timetabling. In Stefan Nher and Dorothea Wagner, edi-

tors, *Algorithm Engineering*, volume 1982 of *Lecture Notes in Computer Science*, pages 230–241. Springer Berlin Heidelberg, 2001.

[85] Massimiliano Caramia, Paolo Dell'Olmo, and Giuseppe F. Italiano. Novel local-search-based approaches to university examination timetabling. *INFORMS J. on Computing*, 20(1):86–99, January 2008.

[86] Marco P. Carrasco and Margarida V. Pato. A multiobjective genetic algorithm for the class/teacher timetabling problem. In Edmund Burke and Wilhelm Erben, editors, *Practice and Theory of Automated Timetabling III*, volume 2079 of *Lecture Notes in Computer Science*, pages 3–17. Springer Berlin Heidelberg, 2001.

[87] M. W. Carter and D. G. Johnson. Extended clique initialisation in examination timetabling. *Journal of the Operational Research Society*, 52(5):538–544, 2001.

[88] M. W. Carter, G. Laporte, and J. W. Chinneck. A general examination scheduling system. *Interfaces*, 24(3):109–120, 1994.

[89] Michael W. Carter, Gilbert Laporte, and Sau Y. Lee. Examination timetabling: Algorithmic strategies and applications. *Journal of the Operational Research Society*, 47(3):373–383, 1996.

[90] S. Casey and J. Thompson. GRASPing the examination scheduling problem. In E. Burke and P. DeCausmaecker, editors, *Practice and Theory of Automated Timetabling IV*, volume 2740 of *Lecture Notes in Computer Science*, pages 232–244, 2003.

[91] Carlos Castro and Sebastián Manzano. Variable and value ordering when solving balanced academic curriculum problems. In *6th Workshop of the ERCIM WG on Constraints*, 2001.

[92] Juan Castro-Gutierrez, Dario Landa-Silva, and José Moreno Pérez. Nature of real-world multi-objective vehicle routing with evolutionary algorithms. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 257–264. IEEE, 2011.

[93] Fatih Cavdur and Merve Kose. A fuzzy logic and binary-goal programming-based approach for solving the exam timetabling problem to create a balanced-exam schedule. *International Journal of Fuzzy Systems*, pages 1–11, 2015.

[94] Konstantin Chakhlevitch and Peter Cowling. Hyperheuristics: Recent developments. In Carlos Cotta, Marc Sevaux, and Kenneth Sörensen, editors, *Adaptive and Multilevel Metaheuristics*, volume 136 of *Studies in Computational Intelligence*, pages 3–29. Springer Berlin Heidelberg, 2008.

[95] Zhenmin Chen. A modified resource distribution fairness measure. *Journal of Mathematics*, 2013:6, 2013.

[96] C.Y. Cheong, K.C. Tan, and B. Veeravalli. Solving the exam timetabling problem via a multi-objective evolutionary algorithm - a more general approach. In *Computational Intelligence in Scheduling, 2007. SCIS '07. IEEE Symposium on*, pages 165–172, April 2007.

[97] C.Y. Cheong, K.C. Tan, and B. Veeravalli. A multi-objective evolutionary algorithm for examination timetabling. *Journal of Scheduling*, 12(2):121–146, 2009.

[98] Marco Chiarandini, Luca Di Gaspero, Stefano Gualandi, and Andrea Schaerf. The balanced academic curriculum problem revisited. *Journal of Heuristics*, 18(1):119–148, 2012.

[99] Carlos A. Coello Coello, David A. Van Veldhuizen, and Gary B. Lamont. *Evolutionary algorithms for solving multi-objective problems*, volume 242. Springer, 2002.

[100] A. W. Colijn and C. Layfield. Conflict reduction in examination schedules. In *1st international conference on the practice and theory of automated timetabling*, page 297307, 30 August1 September 1995 1995.

[101] Yann Collette and Patrick Siarry. *Multiobjective optimization: principles and case studies*. Springer Science & Business Media, 2013.

[102] Tim B. Cooper and Jeffrey H. Kingston. The complexity of timetable construction problems. In Edmund Burke and Peter Ross, editors, *Practice and Theory of Automated Timetabling*, volume 1153 of *Lecture Notes in Computer Science*, pages 281–295. Springer Berlin Heidelberg, 1996.

[103] Dave Corne, H. L. Fang, and C. Mellish. Solving the modular exam scheduling problem with genetic algorithms. In 6^{th} *International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE 93)*, Industrial and Engineering Applications of Artificial Intelligence and Expert Systems - Iea/Aie 93, pages 370–373, 1993.

[104] Dave Corne, Peter Ross, and Hsiao-Lan Fang. Evolutionary timetabling: Practice, prospects and work in progress. In *In Proceedings of the UK Planning and Scheduling SIG Workshop, Strathclyde*, 1994.

[105] David Corne, Peter Ross, and Hsiao-Lan Fang. Fast practical evolutionary timetabling. In *Selected Papers from AISB Workshop on Evolutionary Computing*, pages 250–263, London, UK, UK, 1994. Springer-Verlag.

[106] P. H. Corr, Barry McCollum, M. A. J. McGreevy, and Paul McMullan. A new neural network based construction heuristic for the examination timetabling problem. In *Proceedings of the 9th international conference on Parallel Problem Solving from Nature*, PPSN'06, pages 392–401, Berlin, Heidelberg, 2006. Springer-Verlag.

[107] Pascal Côté, Tony Wong, and Robert Sabourin. Application of a hybrid multi-objective evolutionary algorithm to the uncapacitated exam proximity problem. In *Proceedings of the 5th International Conference on Practice and Theory of Automated Timetabling (PATAT 2004)*, pages 151–167, 2004.

[108] Pascal Côté, Tony Wong, and Robert Sabourin. A hybrid multi-objective evolutionary algorithm for the uncapacitated exam proximity problem. In Edmund Burke and Michael Trick, editors, *Practice and Theory of Automated Timetabling V*, volume 3616 of *Lecture Notes in Computer Science*, pages 294–312. Springer Berlin Heidelberg, 2005.

[109] Peter Cowling, Graham Kendall, and L. M. Han. An investigation of a hyperheuristic genetic algorithm applied to a trainer scheduling problem. In *CEC'02: Proceedings of the 2002 Congress on Evolutionary Computation, Vols 1 and 2*, pages 1185–1190, 2002.

[110] Peter Cowling, Graham Kendall, and Naimah M. Hussin. A survey and case study of practical examination timetabling problems. In *Proceedings of the* 4^{th} *International Conference on the Practice and Theory of Automated Timetabling PATAT02*, pages 258–261, 2002.

[111] Peter Cowling, Graham Kendall, and Eric Soubeiga. A hyperheuristic approach to scheduling a sales summit. In Edmund Burke and Wilhelm Erben, editors, *Practice and Theory of Automated Timetabling III*, volume 2079 of *Lecture Notes in Computer Science*, pages 176–190. Springer Berlin Heidelberg, 2001.

[112] Peter Cowling, Graham Kendall, and Eric Soubeiga. A parameter-free hyperheuristic for scheduling a sales summit. In *4th Metaheuristic International Conference, MIC 2001*, pages 127–131, 2001.

[113] Peter Cowling, Graham Kendall, and Eric Soubeiga. Hyperheuristics: A robust optimisation method applied to nurse scheduling. In Juan Julián Merelo Guervós, Panagiotis Adamidis, Hans-Georg Beyer, Hans-Paul Schwefel, and José-Luis Fernández-Villacañas, editors, *Parallel Problem Solving from Nature PPSN VII*, volume 2439 of *Lecture Notes in Computer Science*, pages 851–860. Springer Berlin Heidelberg, 2002.

[114] Peter Cowling, Graham Kendall, and Eric Soubeiga. Hyperheuristics: A tool for rapid prototyping in scheduling and optimisation. In Stefano Cagnoni, Jens Gottlieb, Emma Hart, Martin Middendorf, and GntherR. Raidl, editors, *Applications of Evolutionary Computing*, volume 2279 of *Lecture Notes in Computer Science*, pages 1–10. Springer Berlin Heidelberg, 2002.

[115] Joseph C. Culberson and Feng Luo. Exploring the k-colorable landscape with iterated greedy. In *Dimacs Series in Discrete Mathematics and Theoretical Computer Science*, pages 245–284. American Mathematical Society, 1995.

[116] Piotr Czyzżak and Adrezej Jaszkiewicz. Pareto simulated annealing–a metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, 7(1):34–47, 1998.

[117] Philippe David. A constraint-based approach for examination timetabling using local repair techniques. In Edmund Burke and Michael Carter, editors, *Practice and Theory of Automated Timetabling II*, volume 1408 of *Lecture Notes in Computer Science*, pages 169–186. Springer Berlin Heidelberg, 1998.

[118] L. Davis. *Order-based genetic algorithms and the graph coloring problem*, chapter 6, page 7290. Reinhold, New York, 1991.

[119] Geoffrey De Smet. Itc 2007 examination track:drools-solver. Technical report, International Timetabling Competition 2007, 2008.

[120] Kalyanmoy Deb. Multi-objective genetic algorithms: Problem difficulties and construction of test functions. technical report no. ci-49/98. Technical report, Department of Computer Science/XI, University of Dortmund, Germany, 1998.

[121] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, Apr 2002.

[122] Peter Demeester, Burak Bilgin, Patrick De Causmaecker, and Greet Vanden Berghe. A hyperheuristic approach to examination timetabling problems: benchmarks and a new problem from practice. *Journal of Scheduling*, 15(1):83–103, 2012.

[123] Robert Denda, Albert Banchs, and Wolfgang Effelsberg. The fairness challenge in computer networks. In Jon Crowcroft, James Roberts, and MikhailI. Smirnov, editors, *Quality of Future Internet Services*, volume 1922 of *Lecture Notes in Computer Science*, pages 208–220. Springer Berlin Heidelberg, 2000.

[124] Luca Di Gaspero. Recolour, shake and kick: A recipe for the examination timetabling problem. In *Proceedings of the 4th international conference on practice and theory of automated timetabling*, pages 404–407, August 2002.

[125] Luca Di Gaspero and Andrea Schaerf. Tabu search techniques for examination timetabling. In Edmund Burke and Wilhelm Erben, editors, *Practice and Theory of Automated Timetabling III*, volume 2079 of *Lecture Notes in Computer Science*, pages 104–117. Springer Berlin Heidelberg, 2001.

[126] K.A. Dowsland and J.M. Thompson. Ant colony optimization for the examination scheduling problem. *Journal of the Operational Research Society*, 56:426–438, 2005.

[127] G. Dozier, J. Bowen, and D. Bahler. Solving small and large scale constraint satisfaction problems using a heuristic-based microgenetic algorithm. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, pages 306–311 vol.1, Jun 1994.

[128] Gunter Dueck. New optimization heuristics: The great deluge algorithm and the record-to-record travel. *Journal of Computational Physics*, 104(1):86–92, 1993.

[129] Tuan Anh Duong and Kim-Hoa Lam. Combining constraint programming and simulated annealing on university exam timetabling. In *Proceedings of the 2nd international conference in computer sciences, research, innovation & vision for the future (RIVF2004), Hanoï Vietnam, 2-5 Février 2004*, pages 205–210, 2004.

[130] J. J. Durillo and A.J. Nebro. jMetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 42(10):760–771, 2011.

[131] J. J. Durillo, A.J. Nebro, and E. Alba. The jMetal framework for multi-objective optimization: Design and architecture. In *CEC 2010*, pages 4138–4325, Barcelona, Spain, July 2010.

[132] R. W. Eglese and G. K. Rand. Conference seminar timetabling. *Journal of the Operational Research Society*, 38(7):591–598, 1987.

[133] Michael Eley. Ant algorithms for the exam timetabling problem. In EdmundK. Burke and Hana Rudov, editors, *Practice and Theory of Automated Timetabling VI*, volume 3867 of *Lecture Notes in Computer Science*, pages 364–382. Springer Berlin Heidelberg, 2007.

[134] Wilhelm Erben. A grouping genetic algorithm for graph colouring and exam timetabling. In Edmund Burke and Wilhelm Erben, editors, *Practice and Theory*

of Automated Timetabling III, volume 2079 of *Lecture Notes in Computer Science*, pages 132–156. Springer Berlin Heidelberg, 2001.

[135] Ersan Ersoy, Ender Özcan, and Ş Uyar. Memetic algorithms and hyperhill-climbers. In *Proc. of the 3rd Multidisciplinary Int. conf. on scheduling: theory and applications, P. Baptiste, G. Kendall, AM Kordon and F. Sourd, Eds*, pages 159–166, 2007.

[136] S. Even, A. Itai, and A. Shamir. On the complexity of time table and multi-commodity flow problems. *SIAM Journal on Computing*, 5(4):691–703, 1976.

[137] Duncan K. Foley. Resource allocation in the public sector. *Yale Econonomic Essays*, 7(1):45–98, 1967.

[138] C. W. Fong, H. Asmuni, and W. S. Lam. A hybrid swarm based approach in addressing university timetabling problems. Technical report, Universiti Teknologi Malaysia, 2013.

[139] Cheng Weng Fong, Hishammuddin Asmuni, Barry McCollum, Paul McMullan, and Sigeru Omatu. A new hybrid imperialist swarm-based optimization algorithm for university timetabling problems. *Information Sciences*, 283:1–21, 2014.

[140] C. M. Fonseca. *Multiobjective genetic algorithms with application to control engineering problems*. PhD thesis, Automatic Control and Systems Engineering, 1995.

[141] Eugene C. Freuder and Mark Wallace. Constraint programming. In Edmund K. Burke and Graham Kendall, editors, *Search Methodologies*, pages 239–272. Springer US, 2005.

[142] Luca Di Gaspero and Andrea Schaerf. Tabu search techniques for examination timetabling. In *Selected Papers from the Third International Conference on Practice and Theory of Automated Timetabling III*, PATAT '00, pages 104–117, London, UK, 2000. Springer-Verlag.

[143] Luca Di Gaspero and Andrea Schaerf. Hybrid local search techniques for the generalized balanced academic curriculum problem. In *Proceedings of the 5th International Workshop on Hybrid Metaheuristics*, HM '08, pages 146–157, Berlin, Heidelberg, 2008. Springer-Verlag.

[144] Corrado Gini. Measurement of inequality of incomes. *The Economic Journal*, 31(121):pp. 124–126, 1921.

[145] Christos Gogos, P. Alefragis, and E. Housos. A multi-staged algorithmic process for the solution of the examination timetabling problem. In *Proceedings of the 7th international conference on the practice and theory of automated timetabling*, 19-22, August 2008 2008.

[146] Christos Gogos, Panayiotis Alefragis, and Efthymios Housos. An improved multi-staged algorithmic process for the solution of the examination timetabling problem. *Annals of Operations Research*, 194(1):203–221, 2012.

[147] Christos Gogos, G. Goulas, P. Alefragis, and E. Housos. Pursuit of better results for the examination timetabling problem using grid resources. In *Computational Intelligence in Scheduling, 2009. CI-Sched '09. IEEE Symposium on*, pages 48–53, April 2009.

[148] D. E. Goldberg. *Genetic algorithms in search, optimization, and machine learning.* Addison-Wesley, New York, 1989.

[149] Juan Carlos Gomez and Hugo Terashima-Marín. Approximating multi-objective hyper-heuristics for solving 2d irregular cutting stock problems. In *Advances in Soft Computing*, pages 349–360. Springer, 2010.

[150] Marie-Anne Guerry, Pieter Smet, Tim De Feyter, Greet Vanden Berghe, et al. A two-phase heuristic and a lexicographic rule for improving fairness in personnel rostering. In *Proceedings of the 10th international conference on the practice and theory of automated timetabling*, pages 292–308. PATAT, 2014.

[151] Limin Han, Graham Kendall, and Peter Cowling. An adaptive length chromosome hyperheuristic genetic algorithm for a trainer scheduling problem. In *Fourth Asia-Pacific Conference on Simulated Evolution And Learning, (SEAL'02)*, pages 267–271, 2002.

[152] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[153] A. Hertz. Tabu search for large scale timetabling problems. *European Journal of Operational Research*, 54(1):39–47, 1991.

[154] Ali Hmer and Malek Mouhoub. A multi-phase hybrid metaheuristics approach for the exam timetabling. In *Proceeding of 10th International Conference of the Practice and Theory of Automated Timetabling, PATAT 2014*, pages 233–251, 2014.

[155] Mohd Hakimi Aiman Ibrahim, Rabiah Ahmad, Nuzulha Khilwani Ibrahim, Suriayati Chuprat, and Habibollah Haron. Nurse scheduling with fairness criteria for public hospital. In *IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE)*, pages 91–95, Dec 2011.

[156] Hisao Ishibuchi, Naoya Akedo, Hiroyuki Ohyanagi, and Yusuke Nojima. Behavior of emo algorithms on many-objective optimization problems with correlated objectives. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 1465–1472. IEEE, 2011.

[157] Hisao Ishibuchi and Tadahiko Murata. A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 28(3):392–403, Aug 1998.

[158] Jaffrey M. Jaffe. Bottleneck Flow Control. *IEEE Transactions on Communications*, 29(7), July 1981.

[159] Rajendra K Jain, Dah-Ming W. Chiu, and William R Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer system, 1984.

[160] Andrzej Jaszkiewicz. Genetic local search for multi-objective combinatorial optimization. *European Journal of Operational Research*, 137(1):50–71, 2002.

[161] Mikkel T. Jensen. Guiding single-objective optimization using multi-objective methods. In Stefano Cagnoni, Colin G. Johnson, Juan J. Romero Cardalda, Elena Marchiori, David W. Corne, Jean-Arcady Meyer, Jens Gottlieb, Martin Middendorf, Agnès Guillot, Günther R. Raidl, and Emma Hart, editors, *Applications of Evolutionary Computing*, volume 2611 of *Lecture Notes in Computer Science*, pages 268–279. Springer Berlin Heidelberg, 2003.

[162] D. Johnson. Timetabling university examinations. *Journal of the Operational Research Society*, 41(1):39–47, 1990.

[163] David E. Joslin and David P. Clements. 'Squeaky Wheel' optimization. *Journal of Articial Intelligence Research*, 10:353–373, 1999.

[164] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *J. Artif. Int. Res.*, 4(1):237–285, May 1996.

[165] M. N. Mohmad Kahar and G. Kendall. Universiti malaysia pahang examination timetabling problem: scheduling invigilators. *Journal of the Operational Research Society*, 65(2):214–226, 2014.

[166] M. N. Mohmad Kahar and G. Kendall. A great deluge algorithm for a real-world examination timetabling problem. *Journal of the Operational Research Society*, 66(1):116–133, 2015.

[167] M. N. Mohmad Kahar and Graham Kendall. The examination timetabling problem at universiti malaysia pahang: Comparison of a constructive heuristic with an existing software solution. *European Journal of Operational Research*, 207(2):557–565, 2010.

[168] D. Karaboga. An idea based on honey bee swarm for numerical optimization. Technical report, Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[169] Dervis Karaboga and Bahriye Akay. A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation*, 214(1):108–132, 2009.

[170] Dervis Karaboga and Bahriye Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of Global Optimization*, 39(3):459–471, 2007.

[171] Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger, editors, *Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Springer US, 1972.

[172] F. P. Kelly, A. Maulloo, and D. Tan. Rate control for communication networks: Shadow prices, proportional fairness and stability. *Journal of the Operation Research Society*, 49:237–252, 1998.

[173] Graham Kendall and Naimah Mohd Hussin. A tabu search hyper-heuristic approach to the examination timetabling problem at the MARA university of technology. In Edmund Burke and Michael Trick, editors, *Practice and Theory of Automated Timetabling V*, volume 3616 of *Lecture Notes in Computer Science*, pages 270–293. Springer Berlin Heidelberg, 2005.

[174] Graham Kendall and NaimahMohd Hussin. An investigation of a tabu-search-based hyper-heuristic for examination timetabling. In Graham Kendall, EdmundK. Burke, Sanja Petrovic, and Michel Gendreau, editors, *Multidisciplinary Scheduling: Theory and Applications*, pages 309–328. Springer US, 2005.

[175] Graham Kendall and Mazlan Mohamad. Channel assignment optimisation using a hyperheuristic. In *Proceeding of 2004 IEEE Conference on Cybernetics and Intelligent Systems (CIS 2004)*, 2004.

[176] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[177] Joshua D. Knowles and David W. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evol. Comput.*, 8(2):149–172, June 2000.

[178] Joshua D. Knowles and D.W. Corne. M-PAES: a memetic algorithm for multiobjective optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation, 2000*, volume 1, pages 325–332 vol.1, 2000.

[179] Joshua D. Knowles, Richard A. Watson, and David W. Corne. Reducing local optima in single-objective problems by multi-objectivization. In Eckart Zitzler, Lothar Thiele, Kalyanmoy Deb, CarlosArtemio Coello Coello, and David Corne, editors, *Evolutionary Multi-Criterion Optimization*, volume 1993 of *Lecture Notes in Computer Science*, pages 269–283. Springer Berlin Heidelberg, 2001.

[180] H. T. Kung, F. Luccio, and F. P. Preparata. On finding the maxima of a set of vectors. *J. ACM*, 22(4):469–476, October 1975.

[181] Ting Kuo and Shu-Yuen Hwang. Using disruptive selection to maintain diversity in genetic algorithms. *Applied Intelligence*, 7(3):257–267, 1997.

[182] G. Laporte and S. Desroches. Examination timetabling by computer. *Computers & Operations Research*, 11(4):351–360, 1984.

[183] Souad Larabi Marie-Sainte. A survey of particle swarm optimization techniques for solving university examination timetabling problem. *Artificial Intelligence Review*, pages 1–10, 2015.

[184] Eugene L. Lawler. *Combinatorial Optimization: Networks and Matroids*. Holt Rinehart and Winston, 1976.

[185] Fabien Le Huédé, Michel Grabisch, Christophe Labreuche, and Pierre Savéant. MCS-A new algorithm for multicriteria optimisation in constraint programming. *Annals of Operations Research*, 147(1):143–174, 2006.

[186] Yu Lei, Maoguo Gong, Licheng Jiao, Wei Li, Yi Zuo, and Qing Cai. A double evolutionary pool memetic algorithm for examination timetabling problems. *Mathematical Problems in Engineering*, 2014(2014), 2014.

[187] Douglas J. Leith, Qizhi Cao, and Vijay G. Subramanian. Max-min fairness in 802.11 mesh networks. *IEEE-ACM Transactions on Networking*, 20(3):756–769, 2012.

[188] Rhydian Lewis. A survey of metaheuristic-based techniques for university timetabling problems. *OR Spectrum*, 30(1):167–190, 2008.

[189] Jingpeng Li, Ruibin Bai, Yindong Shen, and Rong Qu. Search with evolutionary ruin and stochastic rebuild: A theoretic framework and a case study on exam timetabling. *European Journal of Operational Research*, 242(3):798–806, 2015.

[190] Jingpeng Li, Edmund K. Burke, and Rong Qu. Integrating neural networks and logistic regression to underpin hyper-heuristic search. *Knowledge-Based Systems*, 24(2):322–330, 2011.

[191] S. L. M. Lin. A broker algorithm for timetabling problem. In E. K. Burke and P. De Causmaecker, editors, *4th international conference on practice and theory of automated timetabling*, page 372386, 2002.

[192] Panta Lucic. *Modeling Transportation Problems Using Concepts of Swarm Intelligence and Soft Computing*. PhD thesis, Civil Engineering, 2002.

[193] Martin Lukasiewycz, Michael Glaß, Felix Reimann, and Jürgen Teich. Opt4J - a modular framework for meta-heuristic optimization. In *Proceedings of the Genetic and Evolutionary Computing Conference (GECCO 2011)*, pages 1723–1730, Dublin, Ireland, 2011.

[194] Panta Lučić and Dušan Teodorović. Computing with bees: Attacking complex transportation engineering problems. *International Journal on Artificial Intelligence Tools*, 12(03):375–394, 2003.

[195] Mashael Maashi. *An Investigation of Multi-objective Hyperheuristics for Multi-objective Optimisation*. PhD thesis, computer science, 2014.

[196] Mashael Maashi, Graham Kendall, and Ender Özcan. Choice function based hyper-heuristics for multi-objective optimization. *Applied Soft Computing*, 28:312–326, 2015.

[197] Mashael Maashi, Ender Özcan, and Graham Kendall. A multi-objective hyper-heuristic based on choice function. *Expert Systems with Applications*, 41(9):4475–4493, 2014.

[198] A.M.A. Malik, Abdul Kadir Othman, M. Ayob, and A.R. Hamdan. Hybrid integrated two-stage multi-neighbourhood tabu search-emcq technique for examination timetabling problem. In *Data Mining and Optimization (DMO), 2011 3rd Conference on*, pages 232–236, June 2011.

[199] M. R. Malim, A. T. Khader, and A. Mustafa. Artificial immune algorithms for university timetabling. In E. K. Burke and H. Rudova, editors, *6th international conference on practice and theory of automated timetabling*, pages 234–245, 2006.

[200] Javier G. Marín-Blázquez and Sonia Schulenburg. A hyper-heuristic framework with XCS: Learning to create novel problem-solving algorithms constructed from simpler algorithmic ingredients. In Tim Kovacs, Xavier Llorá, Keiki Takadama, PierLuca Lanzi, Wolfgang Stolzmann, and Stewart W. Wilson, editors, *Learning Classifier Systems*, volume 4399 of *Lecture Notes in Computer Science*, pages 193–218. Springer Berlin Heidelberg, 2007.

[201] Simon Martin, Djamila Ouelhadj, Pieter Smet, Greet V. Berghe, and Ender Özcan. Cooperative search for fair nurse rosters. *Expert Systems with Applications*, 40(16):6674–6683, 2013.

[202] A. Mas-Colell, M. Whinston, and J. Green. *Microeconomic Theory*. Oxford University Press, Oxford, U.K., 1995.

[203] Kent McClymont and Edward C Keedwell. Markov chain hyper-heuristic (MCHH): an online selective hyper-heuristic for multi-objective continuous problems. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 2003–2010. ACM, 2011.

[204] Barry McCollum. A perspective on bridging the gap between theory and practice in university timetabling. In *Proceedings of the 6th international conference on Practice and theory of automated timetabling VI*, PATAT'06, pages 3–24, Berlin, Heidelberg, 2007. Springer-Verlag.

[205] Barry McCollum, Paul McMullan, Edmund K. Burke, Andrew J. Parkes, and Rong Qu. The second international timetabling competition: Examination timetabling track. Technical Report QUB/IEEE/Tech/ITC2007/Exam/v4.0/17. September 20, 2007, Queen University of Belfast, 2007.

[206] Barry McCollum, Paul McMullan, Andrew J. Parkes, Edmund K. Burke, and Salwani Abdullah. An extended great deluge approach to the examination timetabling problem. In *The 4th Multidisciplinary International Conference on Scheduling: Theory and Applications*, pages 424–434, Dublin, August 2009.

[207] Barry McCollum, Paul McMullan, Andrew J. Parkes, Edmund K. Burke, and Rong Qu. A new model for automated examination timetabling. *Annals of Operations Research*, 194(1):291–315, 2012.

[208] Paul McMullan. An extended implementation of the great deluge algorithm for course timetabling. In Yong Shi, GeertDick van Albada, Jack Dongarra, and PeterM.A. Sloot, editors, *Computational Science ICCS 2007*, volume 4487 of *Lecture Notes in Computer Science*, pages 538–545. Springer Berlin Heidelberg, 2007.

[209] N. K. Mehta. The application of a graph-coloring method to an examination scheduling problem. *Interfaces*, 11(5):57–65, 1981.

[210] N. K. Mehta. Computer-based examination management system. *Journal of Educational Technology Systems*, 11(2):185–198, 1982.

[211] Liam T.G. Merlot, Natashia Boland, Barry D. Hughes, and Peter J. Stuckey. A hybrid algorithm for the examination timetabling problem. In Edmund Burke and Patrick De Causmaecker, editors, *Practice and Theory of Automated Timetabling IV*, volume 2740 of *Lecture Notes in Computer Science*, pages 207–231. Springer Berlin Heidelberg, 2003.

[212] Kaisa Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer, Boston, MA, 1999, 1999.

[213] N. Mladenović and P. Hansen. Variable neighborhood search. *Comput. Oper. Res.*, 24(11):1097–1100, November 1997.

[214] Jeonghoon Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5):556–567, Oct 2000.

[215] MOEA. MOEA framework, version 2.1. `http://www.moeaframework.org`. Accessed: 2014-09-23.

[216] Pablo Moscato and Michael G. Norman. A "memetic" approach for the traveling salesman problem implementation of a computational ecology for combinatorial optimization on message-passing systems. In *Proceedings of The International Conference on Parallel Computing And Transputter Applications*, pages 177–186. IOS Press, 1992.

[217] Moritz Mühlenthaler and Rolf Wanka. Fairness in academic timetabling. In *Proc. 9^{th} Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT)*, pages 114–130, 2012.

[218] Moritz Mühlenthaler and Rolf Wanka. Fairness in academic course timetabling. *Annals of Operations Research*, 239(1):171–188, 2016.

[219] Ahmad Muklason. Preliminary experiments with classical scalarisation techniques based hyper-heuristics. Technical report, School of Computer Science, The University of Nottingham, 2017.

[220] Ahmad Muklason. Survey on examination timetabling preferences over university students. Technical report, School of Computer Science, The University of Nottingham, 2017.

[221] Tomáš Müller. ITC2007 solver description: A hybrid approach. In *Proceedings of the 7th international conference on the practice and theory of automated timetabling*, 19-22, August 2008 2008.

[222] Tomáš Müller. ITC2007 solver description: a hybrid approach. *Annals of Operations Research*, 172(1):429–446, 2009.

[223] Christine L. Mumford. Simple population replacement strategies for a steady-state multi-objective evolutionary algorithm. In Kalyanmoy Deb, editor, *Genetic and Evolutionary Computation GECCO 2004*, volume 3102 of *Lecture Notes in Computer Science*, pages 1389–1400. Springer Berlin Heidelberg, 2004.

[224] Christine L. Mumford. A multiobjective framework for heavily constrained examination timetabling problems. *Annals of Operations Research*, 180(1):3–31, 2010.

[225] Antonio J. Nebro, Juan J. Durillo, Francisco Luna, Bernabé Dorronsoro, and Enrique Alba. Design issues in a multiobjective cellular genetic algorithm. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization*, volume 4403 of *Lecture Notes in Computer Science*, pages 126–140. Springer Berlin Heidelberg, 2007.

[226] Antonio J. Nebro, Juan José Durillo, Carlos A. Coello Coello, Francisco Luna, and Enrique Alba. A study of convergence speed in multi-objective metaheuristics. In Günter Rudolph, Thomas Jansen, Simon Lucas, Carlo Poloni, and Nicola Beume, editors, *Parallel Problem Solving from Nature PPSN X*, volume 5199 of *Lecture Notes in Computer Science*, pages 763–772. Springer Berlin Heidelberg, 2008.

[227] Antonio J. Nebro, F. Luna, E. Alba, B. Dorronsoro, J.J. Durillo, and A Beham. AbYSS: Adapting scatter search to multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 12(4):439–457, Aug 2008.

[228] Gabriela Ochoa, Rong Qu, and Edmund K. Burke. Analyzing the landscape of a graph based hyper-heuristic for timetabling problems. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, GECCO '09, pages 341–348, New York, NY, USA, 2009. ACM.

[229] Ibrahim H. Osman and James P. Kelly. Meta-heuristics: An overview. In Ibrahim H. Osman and James P. Kelly, editors, *Meta-Heuristics*, pages 1–21. Springer US, 1996.

[230] Norwahida Syazwani Othman and Faizuniza Mashhod. Minimizing examination duration using clustering and graph colouring heuristic approach: A case study. In *16th International-Business-Information-Management-Association Conference*, Innovation and Knowledge Management: A Global Competitive Advantage, Vols 1-4, pages 702–711, 2011.

[231] Yong Ouyang and Hongfang Luo. Design of automated examination timetabling system based on hybrid genetic algorithm. In *3rd International Conference on Information Technology and Computer Science*, pages 402–405, 2011.

[232] Ender Özcan, Yuri Bykov, Murat Birben, and Edmund K. Burke. Examination timetabling using late acceptance hyper-heuristics. In *IEEE Congress on Evolutionary Computation CEC '09*, pages 997–1004, May 2009.

[233] Ender Özcan and Ersan Ersoy. Final exam scheduler - FES. In *The 2005 IEEE Congress on Evolutionary Computation*, volume 2, pages 1356–1363, Sept 2005.

[234] Ender Özcan, Mustafa Misir, Gabriela Ochoa, and Edmund K. Burke. A reinforcement learning - great-deluge hyper-heuristic for examination timetabling. *International Journal of Applied Metaheuristic Computing (IJAMC)*, 1(1):39–59, 2010.

[235] Ender Özcan, Andrew J. Parkes, and Alpay Alkan. The interleaved constructive memetic algorithm and its application to timetabling. *Computers & Operations Research*, 39(10), 2012.

[236] B. Paechter, A. Cumming, H. Luchian, and M. Petriuc. Two solutions to the general timetable problem using evolutionary methods. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, pages 300–305 vol.1, Jun 1994.

[237] Tiago Cardal Pais and Paula Amaral. Managing the tabu list length using a fuzzy inference system: an application to examination timetabling. *Annals of Operations Research*, 194(1):341–363, 2012.

[238] Quan-Ke Pan, M. Fatih Tasgetiren, P. N. Suganthan, and T. J. Chua. A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. *Information Sciences*, 181(12):2455–2468, 2011.

[239] L. Paquete and C. Fonseca. A study of examination timetabling with multiobjective evolutionary algorithms. In *4th Metaheuristics International Conference (MIC 2001)*, page 149154, 2001.

[240] Luis Paquete and Thomas Sttzle. An experimental investigation of iterated local search for coloring graphs. In Stefano Cagnoni, Jens Gottlieb, Emma Hart, Martin Middendorf, and GntherR. Raidl, editors, *Applications of Evolutionary Computing*, volume 2279 of *Lecture Notes in Computer Science*, pages 122–131. Springer Berlin Heidelberg, 2002.

[241] Andrew J. Parkes and Ender Özcan. Properties of yeditepe examination timetabling benchmark instances. In 8^{th} *International Conference on Practice and Theory of Automated Timetabling (PATAT)*, 2010.

[242] Elisha A. Pazner. Pitfalls in the theory of fairness. *Journal of Economic Theory*, 14:458–466, 1977.

[243] Sanja Petrovic and Yuri Bykov. A multiobjective optimisation technique for exam timetabling based on trajectories. In Edmund Burke and Patrick De Causmaecker, editors, *Practice and Theory of Automated Timetabling IV*, volume 2740 of *Lecture Notes in Computer Science*, pages 181–194. Springer Berlin Heidelberg, 2003.

[244] Nelishia Pillay. An analysis of representations for hyper-heuristics for the uncapacitated examination timetabling problem in a genetic programming system. In *Proceedings of the 2008 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries: Riding the Wave of Technology*, SAICSIT '08, pages 188–192, New York, NY, USA, 2008. ACM.

[245] Nelishia Pillay. Evolving hyper-heuristics for the uncapacitated examination timetabling problem. In *Proceeding Multidisciplinary International Conference on Scheduling 2009 (MISTA 2009)*, page 409422, 2009.

[246] Nelishia Pillay. Evolving hyper-heuristics for a highly constrained examination. In *PATAT 2010*, page 336346, 2010.

[247] Nelishia Pillay. Evolving hyper-heuristics for the uncapacitated examination timetabling problem. *Journal of the Operational Research Society*, 63(1):47–58, 2011.

[248] Nelishia Pillay and W. Banzhaf. A study of heuristic combinations for hyper-heuristic systems for the uncapacitated examination timetabling problem. *European Journal of Operational Research*, 197(2):482 – 491, 2009.

[249] Nelishia Pillay and W. Banzhaf. An informed genetic algorithm for the examination timetabling problem. *Applied Soft Computing*, 10(2):457–467, 2010.

[250] Nelishia Pillay and Wolfgang Banzhaf. A genetic programming approach to the generation of hyper-heuristics for the uncapacitated examination timetabling problem. In Josó Neves, Manuel Filipe Santos, and Josó Manuel Machado, editors, *Progress in Artificial Intelligence*, volume 4874 of *Lecture Notes in Computer Science*, pages 223–234. Springer Berlin Heidelberg, 2007.

[251] Nelishia Pillay and Wolfgang Banzhaf. A developmental approach to the uncapacitated examination timetabling problem. In Gnter Rudolph, Thomas Jansen, Nicola Beume, Simon Lucas, and Carlo Poloni, editors, *Parallel Problem Solving from Nature PPSN X*, volume 5199 of *Lecture Notes in Computer Science*, pages 276–285. Springer Berlin Heidelberg, 2008.

[252] Rodrigo L Pinheiro, Dario Landa-Silva, and Jason Atkin. Analysis of objectives relationships in multiobjective problems using trade-off region maps. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 735–742. ACM, 2015.

[253] Robin C Purshouse and Peter J Fleming. Conflict, harmony, and independence: Relationships in evolutionary multi-criterion optimisation. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 16–30. Springer, 2003.

[254] Rong Qu. Benchmark data sets in exam timetabling. `http://www.cs.nott.ac.uk/~pszrq/benchmarks.htm`, 2008.

[255] Rong Qu and Edmund K Burke. Hybrid neighbourhood hyper-heuristics for exam timetabling problems. In *The Sixth Metaheuristics International Conference 2005*, 2005.

[256] Rong Qu and Edmund K. Burke. Hybrid variable neighbourhood hyperheuristics for exam timetabling problems. In *Proceedings of the MIC2005: the sixth metaheuristics international conference*, 2005.

[257] Rong Qu and Edmund K. Burke. Adaptive decomposition and construction for examination timetabling problems. In *Multidisciplinary International scheduling: theory and applications 2007 (MISTA 2007)*, pages 418–425, 2007.

[258] Rong Qu and Edmund K. Burke. Hybridizations within a graph-based hyper-heuristic framework for university timetabling problems. *Journal of the Operational Research Society*, 60(9):1273–1285, 2009.

[259] Rong Qu, Edmund K. Burke, and Barry McCollum. Adaptive automated construction of hybrid heuristics for exam timetabling and graph colouring problems. *European Journal of Operational Research*, 198(2):392 – 404, 2009.

[260] Rong Qu, Edmund K. Burke, Barry McCollum, Liam T.G. Merlot, and Sau Y. Lee. A survey of search methodologies and automated system development for examination timetabling. *Journal of Scheduling*, 12(1):55–89, 2009.

[261] Nicholas J. Radcliffe and Patrick D. Surry. Formal memetic algorithms. In Terence C. Fogarty, editor, *Evolutionary Computing*, volume 865 of *Lecture Notes in Computer Science*, pages 1–16. Springer Berlin Heidelberg, 1994.

[262] Siti Khatijah Nor Abdul Rahim, Andrzej Bargiela, and Rong Qu. Granular modelling of exam to slot allocation. In *23rd European Conference on Modelling and Simulation*, pages 861–866, 2009.

[263] J. Rawls. *A Theory of Justice*. Belknap Press, 1971.

[264] Timothy A. Redl. On using graph coloring to create university timetables with essential and preferential conditions. In *Proceedings of the 3rd International Conference on Management, Marketing and Finances*, pages 162–167, 2009.

[265] L. P. Reis and E. Oliveira. Constraint logic programming using set variables for solving timetabling problems. In *Proceedings of 12th international conference on applications of Prolog*, September, 1999 1999.

[266] Keivan Ronasi, Vincent W. S. Wong, and Sathish Gopalakrishnan. Distributed scheduling in multihop wireless networks with maxmin fairness provisioning. *IEEE Transactions on Wireless Communications*, 11(5):1753–1763, 2012.

[267] Peter Ross and Dave Corne. Comparing genetic algorithms, simulated annealing, and stochastic hillclimbing on timetabling problems. In Terence C. Fogarty, editor, *Evolutionary Computing*, volume 993 of *Lecture Notes in Computer Science*, pages 94–102. Springer Berlin Heidelberg, 1995.

[268] Peter Ross, Dave Corne, and Hsiao-Lan Fang. Improving evolutionary timetabling with delta evaluation and directed mutation. In Yuval Davidor, Hans-Paul Schwefel, and Reinhard Männer, editors, *Parallel Problem Solving from Nature PPSN III*, volume 866 of *Lecture Notes in Computer Science*, pages 556–565. Springer Berlin Heidelberg, 1994.

[269] Peter Ross, David Corne, and Hugo Terashima-Marn. The phase-transition niche for evolutionary algorithms in timetabling. In Edmund Burke and Peter Ross,

editors, *Practice and Theory of Automated Timetabling*, volume 1153 of *Lecture Notes in Computer Science*, pages 309–324. Springer Berlin Heidelberg, 1996.

[270] Peter Ross, Emma Hart, and Dave Corne. Some observations about GA-based exam timetabling. In Edmund Burke and Michael Carter, editors, *Practice and Theory of Automated Timetabling II*, volume 1408 of *Lecture Notes in Computer Science*, pages 115–129. Springer Berlin Heidelberg, 1998.

[271] Peter Ross, Emma Hart, and Dave Corne. Genetic algorithms and timetabling. In Ashish Ghosh and Shigeyoshi Tsutsui, editors, *Advances in Evolutionary Computing*, Natural Computing Series, pages 755–771. Springer Berlin Heidelberg, 2003.

[272] Peter Ross and J.G. Marfn-Blazquez. Constructive hyper-heuristics in class timetabling. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 2, pages 1493–1500 Vol. 2, Sept 2005.

[273] Peter Ross, J.G. Marin-Blazquez, and E. Hart. Hyper-heuristics applied to class and exam timetabling problems. In *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 2, pages 1691–1698, June 2004.

[274] Nasser R. Sabar and M. Ayob. Examination timetabling using scatter search hyper-heuristic. In *Data Mining and Optimization, 2009. DMO '09. 2nd Conference on*, pages 127–131, Oct 2009.

[275] Nasser R. Sabar, M. Ayob, and G. Kendall. Tabu exponential monte-carlo with counter heuristic for examination timetabling. In *IEEESymposium on Computational Intelligence in Scheduling, 2009. CI-Sched '09*, pages 90–94, April 2009.

[276] Nasser R. Sabar, M. Ayob, G. Kendall, and Rong Qu. Grammatical evolution hyper-heuristic for combinatorial optimization problems. *IEEE Transactions on Evolutionary Computation*, 17(6):840–861, Dec 2013.

[277] Nasser R. Sabar, Masri Ayob, Graham Kendall, and Rong Qu. Roulette wheel graph colouring for solving examination timetabling problems. In Ding-Zhu Du, Xiaodong Hu, and PanosM. Pardalos, editors, *Combinatorial Optimization and Applications*, volume 5573 of *Lecture Notes in Computer Science*, pages 463–470. Springer Berlin Heidelberg, 2009.

[278] Nasser R. Sabar, Masri Ayob, Graham Kendall, and Rong Qu. A honey-bee mating optimization algorithm for educational timetabling problems. *European Journal of Operational Research*, 216(3):533 – 543, 2012.

[279] Nasser R. Sabar, Masri Ayob, Graham Kendall, and Rong Qu. A dynamic multi-armed bandit-gene expression programming Hyper-Heuristic for combinatorial optimization problems. *IEEE Transactions on Cybernetics*, PP(99):1–1, 2014.

[280] Nasser R. Sabar, Masri Ayob, Rong Qu, and Graham Kendall. A graph coloring constructive hyper-heuristic for examination timetabling problems. *Applied Intelligence*, 37(1):1–11, 2012.

[281] Andrea Schaerf. A survey of automated timetabling. *Artificial Intelligence Review*, 13(2):87–127, April 1999.

[282] Peter Schroder. *Multivariable control of active magnetic bearings*. PhD thesis, 1998.

[283] Paolo Serafini. Simulated Annealing for Multiple Objective Optimization Problems. In G.H. Tzeng, H.F. Wang, U.P. Wen, and P.L. Yu, editors, *Proceedings of the Tenth International Conference on Multiple Criteria Decision Making: Expand and Enrich the Domains of Thinking and Application*, volume 1, pages 283–294, Berlin, 1994. Springer-Verlag.

[284] Shalin Shah. Randomized heuristic for the maximum clique problem. `http://shah.freeshell.org/graphcoloring/`, 2016.

[285] K. Sheibani. An evolutionary approach for the examination timetabling problems. In *Proceedings of the 4th international conference on practice and theory of automated timetabling*, page 387396, 2123 August 2002 2002.

[286] Margarita R. Sierra and Carlos A. Coello Coello. Improving PSO-Based multi-objective optimization using crowding, mutation and ε-dominance. In CarlosA. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization*, volume 3410 of *Lecture Notes in Computer Science*, pages 505–519. Springer Berlin Heidelberg, 2005.

[287] J. Dario Landa Silva, Edmund K. Burke, and Sanja Petrovic. An introduction to multiobjective metaheuristics for scheduling and timetabling. In Xavier Gandibleux, Marc Sevaux, Kenneth Srensen, and Vincent Tkindt, editors, *Metaheuristics for Multiobjective Optimisation*, volume 535 of *Lecture Notes in Economics and Mathematical Systems*, pages 91–129. Springer Berlin Heidelberg, 2004.

[288] Pieter Smet, Burak Bilgin, Patrick De Causmaecker, and Greet Vanden Berghe. Modelling and evaluation issues in nurse rostering. *Annals of Operations Research*, 218(1):303–326, 2014.

[289] Pieter Smet, Simon Martin, Ouelhadj Djamila, Ender Özcan, and Greet Vanden Berghe. Investigation of fairness measures for nurse rostering. In 9^{th} *International Conference on the Practice and Theory of Timetabling (PATAT)*, 2012.

[290] Amr Soghier and Rong Qu. Adaptive selection of heuristics for assigning time slots and rooms in exam timetables. *Applied Intelligence*, 39(2):438–450, 2013.

[291] Aiguo Song and Jiren Lu. Ranking based adaptive evolutionary operator genetic algorithm. *Tien Tzu Hsueh Pao/Acta Electronica Sinica*, 27(1):85–88, 1999. cited By 8.

[292] M. J. Soomer and G. M. Koole. Fairness in the aircraft landing problem. In *Anna Valicek Competition 2008*, 2008.

[293] N. Srinivas and Kalyanmoy Deb. Muilti-objective optimization using nondominated sorting in genetic algorithms. *Evol. Comput.*, 2(3):221–248, September 1994.

[294] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1^{st} edition, 1998.

[295] Kay C. Tan, C.Y. Cheong, and C.K. Goh. Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation. *European Journal of Operational Research*, 177(2):813 – 839, 2007.

[296] Kay C. Tan, Eik Fun Khor, Tong Heng Lee, and Ramasubramanian Sathikannan. An evolutionary algorithm with advanced goal and priority specification for multiobjective optimization. *J. Artif. Int. Res.*, 18(1):183–215, February 2003.

[297] D. Teodorovic and P. Lucic. Bee colony optimization principles and applications. In *Neural Network Applications in Electrical Engineering 2006. NEUREL 2006. 8th Seminar on*, pages 151–156, 2006.

[298] H. Terashima-Marín, E. J. Flores-Álvarez, and P. Ross. Hyper-heuristics and classifier systems for solving 2D-regular cutting stock problems. In *Proceedings of the 2005 conference on Genetic and evolutionary computation*, GECCO '05, pages 637–643, New York, NY, USA, 2005. ACM.

[299] H. Terashima-Marin, P. Ross, and M. Valenzuela-Rendon. Application of the hardness theory when solving the timetabling problem with genetic algorithms. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 1, pages –611 Vol. 1, 1999.

[300] H. Terashima-Marin, P. Ross, and M. Valenzuela-Rendon. Evolution of constraint satisfaction strategies in examination timetabling. In *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 635–642, 1999.

[301] J. Joshua Thomas, Ahamad Tajudin Khader, and Bahari Belaton. Information visualization approach on the university examination timetabling problem. In Mao Lin Huang, Quang Vinh Nguyen, and Kang Zhang, editors, *Visual Information Communication*, pages 255–264. Springer US, 2010.

[302] J. Joshua Thomas, Ahamad Tajudin Khader, and Bahari Belaton. A practical rough sets analysis in real-world examination timetabling problem instances. In K.R. Venugopal and L.M. Patnaik, editors, *Computer Networks and Intelligent Computing*, volume 157 of *Communications in Computer and Information Science*, pages 21–30. Springer Berlin Heidelberg, 2011.

[303] J. Joshua Thomas, Ahamad Tajudin Khader, Bahari Belaton, and Amy Leow. Exploration of rough sets analysis in real-world examination timetabling problem instances. In Ying Tan, Yuhui Shi, Yi Chai, and Guoyin Wang, editors, *Advances in Swarm Intelligence*, volume 6729 of *Lecture Notes in Computer Science*, pages 173–182. Springer Berlin Heidelberg, 2011.

[304] J. Joshua Thomas, AhamadTajudin Khader, and Bahari Belaton. A parallel coordinates visualization for the uncapaciated examination timetabling problem. In Halimah Badioze Zaman, Peter Robinson, Maria Petrou, Patrick Olivier, TimothyK. Shih, Sergio Velastin, and Ingela Nystrm, editors, *Visual Informatics: Sustaining Research and Innovations*, volume 7066 of *Lecture Notes in Computer Science*, pages 87–98. Springer Berlin Heidelberg, 2011.

[305] J. Joshua Thomas, AhamadTajudin Khader, Bahari Belaton, and ChoyChee Ken. Integrated problem solving steering framework on clash reconciliation strategies for university examination timetabling problem. In Tingwen Huang, Zhigang Zeng, Chuandong Li, and ChiSing Leung, editors, *Neural Information Processing*, volume 7666 of *Lecture Notes in Computer Science*, pages 297–304. Springer Berlin Heidelberg, 2012.

[306] J. Joshua Thomas, A.T. Khader, and B. Belaton. Visualization techniques on the examination timetabling pre-processing data. In *Computer Graphics, Imaging and Visualization, 2009. CGIV '09. Sixth International Conference on*, pages 454–458, Aug 2009.

[307] Jonathan M. Thompson and K. A. Dowsland. Variants of simulated annealing for the examination timetabling problem. *Annals of Operations Research*, 63:105–128,

1996.

[308] Jonathan M. Thompson and Kathryn A. Dowsland. General cooling schedules for a simulated annealing based timetabling system. In Edmund Burke and Peter Ross, editors, *Practice and Theory of Automated Timetabling*, volume 1153 of *Lecture Notes in Computer Science*, pages 345–363. Springer Berlin Heidelberg, 1996.

[309] Jonathan M. Thompson and Kathryn A. Dowsland. A robust simulated annealing based examination timetabling system. *Computers & Operations Research*, 25(7-8):637–648, 1998.

[310] Hamza Turabieh and Salwani Abdullah. A hybrid fish swarm optimisation algorithm for solving examination timetabling problems. In CarlosA.Coello Coello, editor, *Learning and Intelligent Optimization*, volume 6683 of *Lecture Notes in Computer Science*, pages 539–551. Springer Berlin Heidelberg, 2011.

[311] Hamza Turabieh and Salwani Abdullah. An integrated hybrid approach to the examination timetabling problem. *Omega-International Journal of Management Science*, 39(6):598–607, 2011.

[312] Özgür Ülker, Ender Özcan, and Emin Erkan Korkmaz. Linear linkage encoding in grouping problems: applications on graph coloring and timetabling. In *Proceedings of the* 6^{th} *international conference on Practice and theory of automated timetabling VI*, PATAT'06, pages 347–363, Berlin, Heidelberg, 2007. Springer-Verlag.

[313] E. L. Ulungu, J. Teghem, P. Fortemps, and D. Tuyttens. MOSA method: a tool for solving multiobjective combinatorial optimization problems. *Journal of Multi-Criteria Decision Analysis*, 8(4):221–236, 1999.

[314] Christine L. Valenzuela. A simple evolutionary algorithm for multi-objective optimization (SEAMO). In *Congress on Evolutionary Computation (CEC)*, pages 717–722, 2002.

[315] David A. Van Veldhuizen and Gary B. Lamont. Multiobjective evolutionary algorithm research: A history and analysis. Technical report, Citeseer, 1998.

[316] Nadarajen Veerapen, Dario Landa-Silva, and Xavier Gandibleux. Hyperheuristic as component of a multi-objective metaheuristic. In *Doctoral Symposium on Engineering Stocahstic Local Search Algorithm*, page 51, 2009.

[317] Hanwu Wang and Weijia Jia. Design a novel fairness model in WiMAX mesh networks. *Computer Communications*, 35(12):1447–1456, 2012.

[318] D. J. A. Welsh and M. B. Powell. An upper bound for the chromatic number of a graph and its application to timetabling problems. *The Computer Journal*, 10:85–86, 1967.

[319] George M. White and P. W. Chan. Towards the construction of optimal examination timetables. *INFOR*, 17:219–229, 1979.

[320] George M. White and Bill S. Xie. Examination timetables and tabu search with longer-term memory. In Edmund Burke and Wilhelm Erben, editors, *Practice and Theory of Automated Timetabling III*, volume 2079 of *Lecture Notes in Computer Science*, pages 85–103. Springer Berlin Heidelberg, 2001.

[321] George M. White, Bill S Xie, and Stevan Zonjic. Using tabu search with longer-term memory and relaxation to create examination timetables. *European Journal of Operational Research*, 153(1):80 – 91, 2004. Timetabling and Rostering.

[322] T. Wong, P. Cote, and P. Gely. Final exam timetabling: a practical approach. In *Electrical and Computer Engineering, 2002. IEEE CCECE 2002. Canadian Conference on*, volume 2, pages 726–731, 2002.

[323] T. Wong, P. Cote, and R. Sabourin. A hybrid MOEA for the capacitated exam proximity problem. In *Congress on Evolutionary Computation, 2004. CEC2004.*, volume 2, pages 1495–1501 Vol.2, June 2004.

[324] D. C. Wood. A system for computing university examination timetables. *Computer Journal*, 11(1):41–47, 1968.

[325] Anthony Wren. Scheduling, timetabling and rostering a special relationship? In Edmund Burke and Peter Ross, editors, *Practice and Theory of Automated Timetabling*, volume 1153 of *Lecture Notes in Computer Science*, pages 46–75. Springer Berlin Heidelberg, 1996.

[326] Meirong Xu, Yuzhen Wang, and Airong Wei. Robust graph coloring based on the matrix semi-tensor product with application to examination timetabling. *Control Theory and Technology*, 12(2):187–197, 2014.

[327] Yong Yang and Sanja Petrovic. A novel similarity measure for heuristic selection in examination timetabling. In Edmund Burke and Michael Trick, editors, *Practice and Theory of Automated Timetabling V*, volume 3616 of *Lecture Notes in Computer Science*, pages 247–269. Springer Berlin Heidelberg, 2005.

[328] Gao Yuntian and Gao Yixing. Nested partitions method to solving examination timetabling problem. In *World Automation Congress (WAC), 2012*, pages 1–4, June 2012.

[329] Liang Zhang, Wen Luo, Shigang Chen, and Ying Jian. End-to-end maxmin fairness in multihop wireless networks: Theory and protocol. *Journal of Parallel and Distributed Computing*, 72(3):462–474, March 2012.

[330] Aimin Zhou, Yaochu Jin, Qingfu Zhang, B. Sendhoff, and E. Tsang. Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion. In *Proceeding IEEE Congress on Evolutionary Computation, 2006. CEC 2006.*, pages 892–899, 2006.

[331] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, and V.G. da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, April 2003.

[332] Eckart Zitzler. *Evolutionary algorithms for multiobjective optimization: Methods and applications*. PhD thesis, ETH Zurich, Switzerland, 1999.

[333] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In *Evolutionary Methods for Design, Optimisation, and Control*, pages 95–100. CIMNE, Barcelona, Spain, 2002.

[334] Eckart Zitzler and Lothar Thiele. An evolutionary algorithm for multiobjective optimization: The strength pareto approach, 1998.

[335] Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

Appendices

Appendix A

Tables of Literature Survey

This appendix presents tables from literature survey discussed in Chapter 2. The tables include the summary of methods as well as the experimental results over the benchmark problems from prior works.

Table A.1: The comparison of examination timetabling problems benchmark datasets constraints

Constraints	CAR1 [89]	ITC [205]	YED [233]	CAR2 [233]	NOT1 [74]	NOT2 [71]	MEL [211]	MAR [173]	UKM [28]	UMP [167]	KAH [122]
Hard Constraints											
Each exam must be assigned to a timeslot	√	√	√	√	√	√	√	√	√	√	√
Each exam must be assigned to a room	x	√	x	x	x	x	x	x	x	√	√
Clash-free (no student is required to sit for more than one exam in the same time)	√	√	√	√	√	√	√	√	√	√	√
Total Room capacity per timeslot is not exceeded	x	√	√	√	√	√	x	x	√	√	√
Timeslot/period duration	x	√	x	x	x	x	x	x	x	x	x
Period related hard(exam ordering)	x	√	x	x	x	x	x	x	x	x	x
Room related (exclusive room requirement)	x	√	x	x	x	x	x	x	√	x	x
Three exams in a day	x	x	x	x	x	x	√	x	√	x	x
examination availability	x	x	x	x	x	x	√	x	x	x	x
large examinations	x	x	x	x	x	x	√	x	x	x	x
Co-schedule exams	x	√	x	x	x	x	x	√	x	x	x
No Sharing Room	x	x	x	x	x	x	x	x	x	√	x
Room Must be in the same building	x	x	x	x	x	x	x	x	x	√	x
Student which has consecutive exams on the same day should be assigned to the same room	x	x	x	x	x	x	x	x	√	x	x
Soft Constraints											
Two examinations in a row	x	√	√	√	√	√	√	x	x	x	x
Two examinations in a day	x	√	x	x	x	x	x	x	x	x	x
Specified spread of examinations	x	√	x	x	x	x	√	x	x	x	√
Mixed duration of examinations within individual period	x	√	x	x	x	x	x	x	x	x	x
Larger examination appearing later in timetable	x	√	x	x	x	x	x	x	x	x	x
Period related soft constraints	x	√	x	x	x	x	x	x	x	x	x
Room related soft constraints	x	√	x	x	x	x	x	x	x	x	x
Consecutive Overnight	x	x	x	x	x	√	√	x	x	x	x
Proximity cost	√	x	x	x	x	x	x	√	√	√	x
Weekend Cost	x	x	x	x	x	x	x	√	x	x	x
Room Distance (if an exam is splitted into several room)	x	x	x	x	x	x	x	x	x	√	x
Splitting Exam Cost	x	x	x	x	x	x	x	x	x	√	x
All written exams of the same subject should be scheduled in the same timeslot	x	x	x	x	x	x	x	x	x	x	√
All oral exams should be scheduled such that the maximum number of students per time slot is at most 20	x	x	x	x	x	x	x	x	x	x	√
Lecturers who are responsible for oral exams cannot examine more than one student group at the same time	x	x	x	x	x	x	x	x	x	x	√

Table A.2: The best result reported in the literature for Carter dataset

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Instance	[277]	[23]	[106]	[133]	[70]	[89]	[73]	[26]	[84]	[327]	[54]	[259]	[259]	[6]	[50]	[72]	[125]	[211]	[257]	[224]	[108]	[249]	[142]	[90]	[29]	[47]	[189]	[55]	[248]	[53]
CAR-F-92-I	6.21	4.56	6.24	4.7	4.53	6.2	4.32	4.54	6	3.93	4	4.32	4.26	4.36	4.2	4.1	5.2	4.3	4.5	4.5	4.2	4.22	5.2	4.4	4.51	3.74	4.91	4.1	4.28	_
CAR-S-91-I	7.01	5.29	7.21	5.7	5.36	7.1	4.97	5.29	6.6	4.5	4.6	5.11	5.09	5.21	4.8	4.65	6.2	5.1	5.45	5.3	5.4	4.92	6.2	5.4	4.9	4.42	5.81	4.9	4.97	_
EAR-F-83-I	42.81	37.02	49.44	36.8	37.92	36.4	36.16	37.02	29.3	33.7	32.8	35.56	35.48	34.87	35.4	37.05	45.7	35.1	36.15	34.2	34.2	35.87	45.7	34.8	36.28	32.76	35.08	33.2	36.86	45.6
HEC-S-92-I	12.9	11.87	13.57	11.2	12.25	10.8	11.61	11.78	9.2	10.83	10	11.62	11.46	10.28	10.8	11.54	12.4	10.6	11.38	10.3	10.4	11.5	12.4	10.8	11.06	10.15	10.44	10.3	11.85	_
KFU-S-93	18.47	15.81	19.9	15	15.2	14	15.02	15.8	13.8	13.82	13	15.18	14.68	13.46	13.7	13.9	18	13.5	14.74	13.9	14.3	14.37	18	14.1	14.74	12.96	15.08	13.2	14.62	_
LSE-F-91	15.62	12.09	14.99	12.2	11.33	10.5	10.96	12.09	9.6	10.35	10	11.32	11.2	10.24	10.4	10.82	15.5	10.5	10.85	11.3	11.3	10.89	15.5	14.7	12.08	9.83	11.28	10.4	11.14	_
PUR-S-93-I	7.92	_	_	5.4	_	_	_	_	_	_	_	_	_	_	4.8	_	_	_	_	_	_	4.65	_	_	4.66	_	_	_	4.73	_
RYE-F-92	10.5	10.35	_	10.1	_	7.3	_	10.38	6.8	8.53	_	_	_	_	8.9	_	_	8.4	_	9.1	8.8	9.3	_	_	10.67	_	_	_	9.65	_
STA-F-83-I	161	160.42	159.28	157.4	158.19	161.5	161.91	160.42	158.2	158.35	159.9	158.88	158.28	159.2	159.1	168.73	160.8	157.3	157.2	157	157	157.81	160.8	134.9	157.32	157.03	157.06	156.9	158.33	158.2
TRE-S-92	10.98	8.67	10.77	8.9	8.92	9.6	8.38	8.67	9.4	7.92	7.9	8.52	8.51	8.4	8.3	8.35	10	8.4	8.79	8.5	8.6	8.38	10	8.7	8.92	7.75	8.39	8.3	8.48	_
UTA-S-92-I	4.76	3.57	4.48	3.8	2.88	3.5	3.36	3.57	3.5	3.14	3.2	3.21	3.15	3.63	3.4	3.2	4.2	3.5	3.55	3.7	3.5	3.35	4.2	_	3.58	3.06	_	_	3.4	4.52
UTE-S-92	29.69	27.78	31.25	27.7	28.01	25.8	27.41	28.07	24.4	25.39	24.8	28	27.9	26	25.7	25.83	29	25.1	26.68	25.2	25.3	27.24	29	25.4	26.36	24.82	25.17	24.9	28.88	35.4
YOR-F-83-I	39.83	40.66	_	39.3	41.37	41.7	40.77	39.8	36.2	36.35	37.3	40.71	40.49	36.2	36.7	37.28	/41	37.4	42.2	37.2	36.4	39.33	41	37.5	38.97	34.84	36.97	36.3	40.74	_
	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
Instance	[23]	[255]	[70]	[247]	[247]	[247]	[24]	[245]	[258]	[60]	[245]	[259]	[122]:SA-4	[173]	[133]	[48]	[280]	[256]	[53]	[260]	[80]	[276]	[4]	[4]	[2]	[80]	[125]	[85]	[240]	[3]
CAR-F-92-I	4.56	4.7	4.84	4.22	4.19	4.18	4.51	4.28	4.16	4.45	_	4.32	3.78	4.67	4.3	3.81	4.7	4.7	_	3.93	4.22	4	4.74	4.76	4.38	4.22	5.2	6	_	4.41
CAR-S-91-I	5.29	5.4	5.41	4.95	4.95	4.93	5.19	4.97	5.16	5.37	_	5.11	4.52	5.37	5.2	4.58	5.14	5.4	_	4.5	5.03	4.62	5.17	5.31	5.08	5.03	6.2	6.6	_	5.12
EAR-F-83-I	37.02	37.29	38.19	35.95	35.94	36.64	36.64	35.86	35.86	37.89	35.56	35.56	32.49	40.18	36.8	32.65	37.86	37.29	45.6	29.3	36.06	34.71	40.91	40.91	38.44	36.06	45.7	29.3	38.9	36.91
HEC-S-92-I	11.78	12.23	12.72	11.27	11.21	11.26	11.6	11.85	11.94	11.78	11.43	11.62	10.03	11.86	11.1	10.06	11.9	12.23	_	9.2	11.71	10.68	12.26	12.36	11.61	11.71	12.4	9.2	11.2	11.31
KFU-S-93	15.81	15.11	15.76	14.12	14.13	14.21	15.34	14.62	14.79	15.45	_	15.18	12.9	15.84	14.5	12.81	15.3	15.11	_	13	16.02	13	15.85	16.31	14.67	16.02	18	13.8	16.5	14.75
LSE-F-91	12.09	12.71	13.15	10.76	10.83	10.81	11.35	11.14	11.15	12.12	_	11.32	10.04	_	11.3	9.86	12.33	12.71	_	9.6	11.15	10.11	12.58	12.84	11.69	11.15	15.5	9.6	13.2	11.41
PUR-S-93-I	_	_	_	_	_	_	_	_	_	_	_	_	5.67	_	_	4.32	5.37	_	_	_	_	4.8	5.87	6.07	_	_	_	3.7	_	5.87
RYE-F-92	10.35	_	_	9.23	9.23	9.25	10.05	9.65	_	_	_	_	8.05	_	9.8	7.93	10.71	_	_	6.8	9.42	10.79	10.11	10.4	9.49	9.42	_	6.8	_	9.61
STA-F-83-I	160.42	158.8	158.19	157.69	157.82	157.39	160.79	158.33	159	158.94	158.38	158.88	157.03	157.38	157.3	157.03	160.12	158.8	158.2	157.2	158.86	158.02	158.12	159.2	157.72	158.86	160.8	158.2	158.1	157.52
TRE-S-92	8.67	8.67	8.85	8.43	8.37	8.48	8.47	8.48	8.6	8.99	_	8.52	7.69	8.39	8.6	7.72	8.32	8.67	_	7.9	8.37	7.9	9.3	9.3	8.78	8.37	10	9.4	9.3	8.76
UTA-S-92-I	3.57	3.54	3.88	3.33	3.37	3.32	3.52	3.4	3.59	3.5	_	3.21	3.13	_	3.5	3.16	3.88	3.54	4.52	3.14	3.37	3.12	3.65	3.74	3.55	3.37	4.2	3.5	_	3.54
UTE-S-92	27.78	29.68	31.65	26.95	27.13	27.16	27.55	28.88	28.3	26.62	27.31	28	24.77	27.6	26.4	27.79	32.67	29.68	35.4	24.4	27.99	26	27.71	27.8	26.63	27.99	29	24.4	27.8	26.25
YOR-F-83-I	40.66	43	40.13	39.63	40	39.84	39.79	40.74	41.81	42.19	39.96	40.71	34.64	_	39.4	34.78	40.53	43	_	36.2	39.53	36.2	43.98	43.98	40.45	39.53	41	36.2	38.9	39.67
	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85					
Instance	[82]	[311]	[9]	[90]	[274]	[10]	[198]	[20]	[302]	[7]	[186]	[18]	[13]	[124]	[139]	[278]	[138]	[19]	[77]	[303]	[310]	[311]	[55]	[237]	[82]	**BEST**		**BEST-REF**		
CAR-F-92-I	4.31	4.1	3.76	4.2	4.41	3.82	4.83	4.36	4	3.94	5.2	4	4.29	_	3.77	3.9	3.89	3.88	4.31	4	4.11	4.1	4.1	4.57	4.31	**3.74**	[47]			
CAR-S-91-I	5.19	4.8	4.46	5.4	5.36	4.35	5.22	5.19	4.42	4.76	6.4	4.62	4.99	5.7	4.71	4.79	4.79	4.38	5.01	4.42	4.81	4.8	4.9	5.46	5.19	**4.35**	[10]			
EAR-F-83-I	35.79	34.92	32.12	34.2	37.74	33.76	39.34	32.26	35.9	33.61	39.8	33.14	34.42	39.4	33.15	34.69	33.43	33.34	36.85	35.9	36.1	34.92	33.2	33.5	35.79	**29.3**	[84]			
HEC-S-92-I	11.19	10.73	9.72	10.2	11.87	10.29	11.18	10.89	9.1	10.56	11.8	10.43	10.4	10.9	10.38	10.66	10.49	10.39	12.24	9.1	10.95	10.73	10.3	10.52	11.19	**9.1**	[302]			
KFU-S-93	14.51	13	12.62	14.2	15.7	12.86	15.24	13.73	12.44	13.44	16.2	13.59	13.5	_	13.69	13	13.72	13.23	14.38	12.44	13.21	13	13.2	14.05	14.51	**12.44**	[302]			
LSE-F-91	10.92	10.01	10.03	14.2	12.53	10.23	12.64	11.15	11.83	10.87	14.5	10.75	10.48	12.6	10.25	10	10.29	10.52	10.99	11.83	10.2	10.01	10.4	_	10.92	**9.6**	[84]			
PUR-S-93-I	_	4.73	_	_	6.27		_	_	10.81	_	_	_	_	_	_	_	_	_	_	10.81	_	4.73	_	_	_	**3.7**	[85]			
RYE-F-92	_	9.65	_	_	9.88		9.74	_	10.38	8.81	12.3	9.17	8.79	_	_	_	_	8.92	9.81	10.38	_	9.65	_	9.11	_	**6.8**		[84],[85],[260]		
STA-F-83-I	157.18	158.26	156.94	134.9	160.77	156.9	157.12	157.23	155.01	157.09	157.2	157.06	157.04	157.4	157.03	157.04	157.07	157.06	160.46	155.01	159.74	158.26	156.9	157.29	157.18	**134.9**	[90]			
TRE-S-92	8.49	7.88	7.86	8.2	8.9	8.21	9.07	9.22	_	7.94	9.5	8	8.16	_	7.84	7.87	7.86	7.89	8.55	5.89	8	7.88	8.3	8.71	8.49	**5.89**	[303]			
UTA-S-92-I	3.44	3.2	2.99	_	3.87	3.22	4.48	3.83	5.89	3.27	4.3	3.27	3.43	4.1	3.1	3.1	3.1	3.13	3.38	25.5	3.32	3.2	3.3	3.71	3.44	**2.88**	[70]			
UTE-S-92	26.7	26.11	24.9	25.2	32.67	25.41	24.89	26.73	25.5	25.36	28.6	25.16	25.09	_	25.32	25.94	25.33	25.12	27.97		26.17	26.11	24.9	25.18	26.7	**24.4**	[84]			
YOR-F-83-I	39.47	36.22	34.95	37.2	39.11	36.35	39.31	40.63	38.3	35.74	40.5	35.58	35.86	39.7	36.06	36.15	36.12	35.49	39.69	38.3	36.23	36.22	36.3	39.08	39.47	**34.64**	[122]			

Table A.3: The best result reported in the literature for ITC 2007 dataset

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
INSTANCE	[221]	[145]	[27]	[119]	[251]	[8]	[8]	[122]	[206]	[146]	[280]	[222]	[290]	[249]	[21]	[17]	[311]	[276]	[81]
EXAM1	4370	5905	8006	6670	12035	4000	4350	6060	4633	4775	6234	4370	5752	8559	11823	6582	4368	4362	6235
EXAM2	400	1008	3470	623	3074	385	385	515	405	385	395	385	1693	830	976	1517	390	380	2974
EXAM3	10049	13862	18622	_	15917	10,190	9951	23580	9064	8996	13002	9378	14586	11576	26770	11912	9830	8991	15832
EXAM4	18141	18674	22559	_	23582	17,494	18,000	_	15663	16204	17940	15368	21491	21901	-	19657	17251	15094	35106
EXAM5	2988	4139	4714	3847	6860	3030	3040	4855	3042	2929	3900	2988	3844	3969	6772	17659	3022	2912	4873
EXAM6	26950	27640	29155	27815	32250	25,995	26,010	27605	25880	25740	27000	26365	28480	28340	30980	26905	25995	25735	31756
EXAM7	4213	6683	10473	5420	17666	4117	4250	6065	4037	4087	6214	4138	5182	8167	11762	6840	4067	4025	11562
EXAM8	7861	10521	14317	_	16184	7360	7450	9038	7461	7777	8552	7516	13711	12658	16286	11464	7519	7452	20994
EXAM9	1047	1159	1737	1288	2055	_	_	1184	1071		_	_	_	_	_	_	_	_	_
EXAM10	16682	-	15085	14778	17724	_	_	15561	14374	_	_	_	_	_	_	_	_	_	_
EXAM11	34129	43888	_	_	40535	_	_	_	29180	_	_	_	_	_	_	_	_	_	_
EXAM12	5535	_	5264	_	6310	_	_	5483	5693	_	_	_	_	_	_	_	_	_	_

	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34		
INSTANCE	[246]	[22]	[279]	[82]	[146]	[311]	[3]	[20]	[7]	[18]	[146]	[311]	[147]	[147]	[82]	**BEST**	REF
EXAM1	8559	9885	4371	6235	4775	4368	5231	6354	5375	5328	4,775	4368	4,699	4,699	6235	**4000**	[8]
EXAM2	830	393	380	2974	385	390	433	1352	445	512	385	390	385	385	2974	**380**	[276],[279]
EXAM3	11576	19931	8965	15,832	8996	9830	9265	10146	10,246	10,178	8,996	9830	8,546	8,500	15832	**8500**	[147]
EXAM4	21901	INF	15381	35,106	16,204	24,822	17,787	18214	16,019	16,465	16,204	17,251	15,884	14,879	35106	**14879**	[147]
EXAM5	3969	4065	2909	4873	2929	3022	3083	16124	3511	3,624	2,929	3022	2,823	2,795	4873	**2795**	[147]
EXAM6	28340	29935	25750	31,756	25,740	25,995	26,060	22309	26,130	26,240	25,74	25,995	25,565	25,410	31756	**22309**	[20]
EXAM7	8167	8801	4037	11,562	4087	4067	10,712	6317	4418	4,562	4,087	4067	3,975	3,884	11562	**3884**	[147]
EXAM8	12658	12145	7468	20,994	7777	7519	12,713	10293	8410	8,043	7,777	7519	7,493	7,440	20994	**7360**	[8]
EXAM9	_	_	_	_	964	_	1111	_	_	_	_	_	_	_	_	**964**	[146]
EXAM10	_	_	_	_	13,203	_	14,825	_	_	_	_	_	_	_	_	**13203**	[146]
EXAM11	_	_	_	_	28,704	_	28,891	_	_	_	_	_	_	_	_	**28704**	[146]
EXAM12	_	_	_	_	5197	_	6181	_	_	_	_	_	_	_	_	**5197**	[146]

Table A.4: Survey of approaches for examination timetabling problems by 2009 reported in [260]

Approaches	References
Graph based sequential techniques	Reduction to Graph Colouring: [318, 209, 210] Graph based ordering: saturation degree [45], largest weighted degree [89], largest enrolment [324], [64], extended clique [87], [76], [73], fuzzy system [23, 25], neural network [106], backtracking [89], look ahead [71]
Constraint based techniques	See studies in [44, 117, 265, 211, 129, 185].
Meta-heuristics	
Local search based techniques	Tabu search : [125, 124, 320, 321, 240] Simulated annealing: [309, 307, 46, 211, 129] Great deluge algorithm: [50, 72, 73] Large neighbourhood search: [6, 5] Variable neighbourhood search (VNS): [54, 55, 258] Iterated local search: [240, 84, 85] Greedy randomised adaptive search procedure (GRASP): [90]
Population based algorithms	Evolutionary algorithm: Genetic algorithm: [105, 104, 271, 269, 270, 300, 299, 134, 285, 322, 107, 312] Memetic algorithm: [74, 75, 76, 71] Ant algorithm: [31, 126, 133] Artificial immune algorithms: [199]
Multi-criteria Technique	See studies in [287, 100, 51, 243, 185].
Hyper-heuristics	VNS to find good combinations of parametrised heuristics for different exam timetabling problems [12] GA-based hyper-heuristics [273] Tabu search hyper-heuristics [66, 173, 174, 70] Case base reasoning hyper-heuristics [53, 79, 327] Others: [258, 41, 135]
Decomposition/clustering techniques	See studies in [71, 191, 257].

Table A.5: Survey of approaches for examination timetabling problems reported between 2009-2015

Approaches	Methods and References
Graph based sequential construction techniques	Standard constructive graph colouring base heuristics [167, 264], adaptive hybrid heuristics [259], fuzzy multiple heuristic ordering [26], roulette wheel graph colouring [277], construction with grid resources [147], clustering and graph colouring heuristics [230], rough set approach [303], adaptive selection of heuristics [82], matrix semi product approach [326].
Improvement with meta-heuristics	
Local search	Integrated problem solving steering framework [305], roughsets [302], hill climbing combined with neighbourhood structure [11], hybrid multi neighbourhood tabu search [198], scatter search [274], tabu search [275, 237], great deluge algorithm [9, 166], local search [85], integrated hybrid approach [311], iterated two-stage multi-neighbourhood tabu search [1], Iterated local search [52], GRASP with hybrid meta-heuristic local search [146], intelligent water drop algorithm [15].
Population base	Genetic algorithm [249], memetic Algorithm [8, 235], evolutionary ruin and stochastic rebuild [189], Hybrid genetic algorithm [231], MOEA with micro-genetic algorithm [127] and hill climber local search [97], hybrid fish swarm [310], honey-bee mating [278], hybrid artificial bee colony [20], hybrid bee colony [19], tabu-based memetic approach [10], hybrid imperialist swarm-based optimisation [139], double evolutionary pool memetic algorithm [186], memetic algorithm with harmony search algorithm [13], adaptive artificial bee colony and late acceptance hill climbing [18], Hybrid self-adaptive bees algorithm [7], particle swarm optimisation algorithms [183].
Multi-criteria technique	Multi-criteria [224], Multi-objective evolutionary algorithm [97].
Hyper-heuristics	Graph-colouring constructive hyper-heuristics [259, 247, 63], Monte-carlo hyper-heuristics [63], linear combination heuristics hyper-heuristic [80], Harmony search hyper-heuristics [21, 22], grammatical evolution hyper-heuristics [276], adaptive selection of heuristics [82, 290], tournament based hyper-heuristic [122], hierarchical hybridizations graph colouring based construction heuristics [280, 248], Adaptive linier combination heuristics [3], Dynamic multiarmed bandit-gene expression programming [279], Adaptive decomposition and ordering strategy [4].
Decomposition/ clustering technique	Granular modelling [262], nested partition [328], clustering and graph colouring heuristics [230].
Visualisation techniques	Information visualisation [301, 306], parallel coordinates visualisations [304].
Others	Invigilator exam timetabling [165].

Appendix B

Survey on Examination Timetable Preferences of University Students

B.1 Introduction

This chapter reports the result of survey on examination timetable preferences of University Students. On April 2014, we conducted a survey among undergraduate and taught postgraduate students at The University of Nottingham. The purpose of survey was to seek in-depth understanding on the preferences of university students over their examination timetable and to be more specific to investigate whether fairness issues does matter in real-world University examination timetabling. The survey was carried out by mean of questionnaire distributed directly to the students to complete as well as via on-line version.

Although surveys on exam timetabling in practice have been conducted by prior works such as in [56], and [110] there are other important issues that are neglected. In [56] the survey was conducted over registrars rather than directly over students. While in [110], the survey conducted over students and invigilators but the students preference were captured in not detail fashion. In the survey students were only asked general questions on how their preference over their

exam timetable and the survey found that the students expect that their exams are distributed as evenly as possible over the timetable period, at least one day gap, and no exam in the weekend. However, to what extend they will penalise different pattern of timetable was not captured. Additionally, other issues such as different difficulty level of exam, and fairness issues among students were not captured too. Though, the author experience found that there is some students complaining their terrible exam timetable whereas in the same time they know their fellows have very good exam timetable. Therefore, the survey conducted in this study aims to extend the previous survey in order to get better understanding on students preferences over their exam timetable as well as capture other issues related with exam timetabling problem, e.g. fairness.

The following sections will give brief description about the questionnaire followed by the survey feedback as well as the finding from the survey.

B.2 The Questionnaire

The method that is used in the survey is a questionnaire. Generally, in addition to general questions regarding students experience on their latest examination, the questionnaire consisted of two main parts. The first part was concerned with students perspective on fairness issue in examination while the second part dealt with students detailed personal preferences over their exam timetable.

In the first part, the students were asked their opinion regarding fairness as well as how they defined fairness in their examination timetable. The questions include questions to know whether fairness should be enforced among the entire students body in the university or among students within the same course only.

In the second part, the students were asked their detail preference how their examinations are timetabled. The preferences include preference regarding the time of the exams and the gap between exams. To be more specific, the questionnaire also asked students to consider between hard and easy exams. To the best of our knowledge, the hardness of exam was neglected in the state-of-the art examination timetabling formulation in which all exams were assumed have the same level of difficulty. On the other hand, in order to oppose to Carter problem formulation, in which the gap between two exams are penalised by a fixed artificial formula, in this survey, how

students penalised the gap between two exams were investigated. Given the first exam scheduled in the first time slot of the first day in different timetable pattern, the students are asked to fill the penalty for any possibilities in which the second exam will be timetabled as shown by Figure B.1, students were asked to penalised the gap between two exams by giving a score 0-9. Table B.1 summarises the questionnaire questions and the detailed questionnaire questions can be observed in [220].

GAP								
Day#1			Day#2			Day#3		
1	2	3	4	5	6	7	8	9
First Exam								

(a) Pattern 1 (Both exams have equal difficulty level)

GAP								
Day#1			Day#2			Day#3		
1	2	3	4	5	6	7	8	9
Easy Exam								

(b) Pattern 2 (An easy exam followed by a hard exam)

GAP								
Day#1			Day#2			Day#3		
1	2	3	4	5	6	7	8	9
Hard Exam								

(c) Pattern 3 (A hard exam followed by an easy exam)

Figure B.1: Some different timetable patterns

B.3 Questionnaire Feedback

In our survey, the feedback data had been collected from 50 undergraduate and taught post-graduate students at The University of Nottingham. From the questionnaire feedback, the most significant findings are as follows.

From the general questions response, we got information that the average number of examinations that students had during their last examination session was four examination lasted within 10 days. Regarding with their examination timetable, as shown by Table B.2 it was found that 4%

Table B.1: The Summary of Questionnaire Questions

Partitions	Descriptions
Part I : Fairness	Students general opinion about their latest examination timetable. Students opinion about fairness issue in their their latest examination timetable. Students opinion about the notions of fairness in examination timetabling.
Part II : Student Personal Preferences	Students preference regarding timeslot (i.e. morning, afternoon, evening) and day of exams. Students preference on the exams should be timetabled over the exam period. Students opinion about exams different level of difficulty and how the should be timetabled
Part III : General Questions	The number of exams in a single exam period and the length of exam period. Other issues supposed to be addressed in examination timetabling.

of the students were very happy and 36% were happy whereas 14% of them were unhappy, 0% were very unhappy, and 46% were moderate. Even, 10% of the respondents strongly believed and 18% of the respondents believed that their examination timetable negatively affected their academic achievement. These finding suggests that although the current examination timetable in the University of Nottingham is generally good, it could be further improved in order to increase students satisfaction.

Table B.2: The Questionnaire Feedback

Questions	Feedback (%)				
	Very Happy	Happy	Moderate	Unhappy	Very Unhappy
Students opinion about their latest exam timetable.	4	36	46	14	0
	Disagree Strongly	Disagree	Neutral	Agree	Agree Strongly
Whether the exam timetable affect negatively the students' academic achievement.	6	24	42	18	10
Whether students' latest exam timetable fair among students.	0	10	30	56	4

The common reasons that made them unhappy were that the timetable of two examinations are

too close to each other (less than 24 hours gap in between). It is even worse, when one or both of the exams are considered as hard exam. The location of exam that is different from the student base campus is also another reason to not happy as well as having an exam in the last day of examination period.

Table B.3: Students response regarding fairness: whether fairness should be enforced in different scenarios.

Fairness among students:	Students Response(%)				
	Disagree Strongly	Disagree	Neutral	Agree	Agree Strongly
Taking the same exam	2	2	14	46	36
Taking the same course	2	4	10	42	42
Overall, although different course	2	8	24	42	24

In response to fairness issue, our survey revealed that 10% of students think that the examination timetable is unfair amongst students, whereas 60% of them think it is fair and the rest are neutral. However, as expected, almost all students agreed that the examination timetable should be fair. Regarding the scope of fairness, as summarised in Table B.3, it is shown that 36% of respondents strongly agreed and 46% agreed that examination timetable should be fair amongst students taking the same exams. Furthermore, when the respondents were asked to detail their perception with respect the scope of fairness, 42% strongly agreed and 42% agreed that examination timetable should be fair amongst students enrolled on the same course. Interestingly, the statistic changed to 24% strongly agreed and 42% agreed, if they were asked whether the examination timetable should be fair amongst the entire students body of the university (though enrolled on different courses). This finding indicates that fairness within a course is more important than fairness amongst the entire student body of the university.

A further finding in our survey is related with what students think about the quality of timetable, in which students personal preference over their examination timetable are investigated. We found some factors that affected students preferences. Overall, it is not surprising that almost three quarters (74%) preferred to have exams that are spread out evenly throughout the time period in oppose to only 12% who preferred the 'bunched' one. When students were asked to trade-off between total exam period length and the gap between exams in which a shorter total exam period would mean a reduced gap between exams, 40% preferred longer total exam period while 12% preferred the opposite and the rest preferred remain the same.

Furthermore, if the students were asked to make a trade-off between fairness and the enough gap between exams, 82% of students were not willing to accept more than one exam a day although

it was fair between students. It confirms that enough gap between exams is very critical for students and more important than fairness.

In relation with the timeslot of exams, assuming that there are three timeslots a day, the afternoon session was the most preferred one while morning and evening session ranked second and third respectively. In addition, some students (31%) preferred to have no exam at particular days. The least preferred days were Saturday or Sunday, Friday, Monday, and any day after the day student had an exam. More than a half (54%) preferred to have no exam on the weekend.

In the current state-of-the-art exam timetabling problem formulation, the exams are assumed having the equal difficulty level, in contrast, our finding showed that 53% of the students strongly agreed and 37% of them agreed that some examinations are harder than the others. Thus, harder exams should be given longer gap for preparation. Furthermore, 50% of students preferred difficult exams to be scheduled earlier while only 20% preferred the opposite. Note that the difficulty level of exam in this case is based on the students personal opinion.

In order to know how long the gap between exams that the students prefer, the students were asked to provide a penalty (0-9) to the three different timetable scenarios as shown by Figure B.1. Given two examinations, three days period with three timeslot per day (morning, afternoon, and evening), if the first exam was scheduled in the first timeslot, i.e. morning of the first day, students were asked to indicate penalty expressing their unwillingness if the second exam was scheduled in the second timeslot of the first day, the third timeslot of the first day, and so on until the third timeslot of the third day.

In the first scenario both exams are assumed have the same difficulty level, i.e. hard exam followed by hard exam or easy exam followed by easy exam (`H-H/EE`)while in the second scenario, the first exam is assumed as easy exam and the second exam was hard (`E-H`). Contrast with the second scenario, in the third scenario, the first exam is assumed as hard exam and the second was easy (`H-E`). The average penalty given by respondents is summarised by Figure B.2. The full detail of questionnaire feedback can be observed in [220].

The x-axis in Figure B.2 indicates the scenario when the second exam is planned to be timetabled, given that the first exam is scheduled in the first timeslot of the first day, while the y-axis indicates the penalty. In `x-axis`, `D1.T2` stands for the first day, the second timeslot, `D2.T1` stands for the second day, the first timeslot and so forth. The score `0` in `y-axis` means that students have no problem with the timetable while `9` means students really don't expect that timetable.

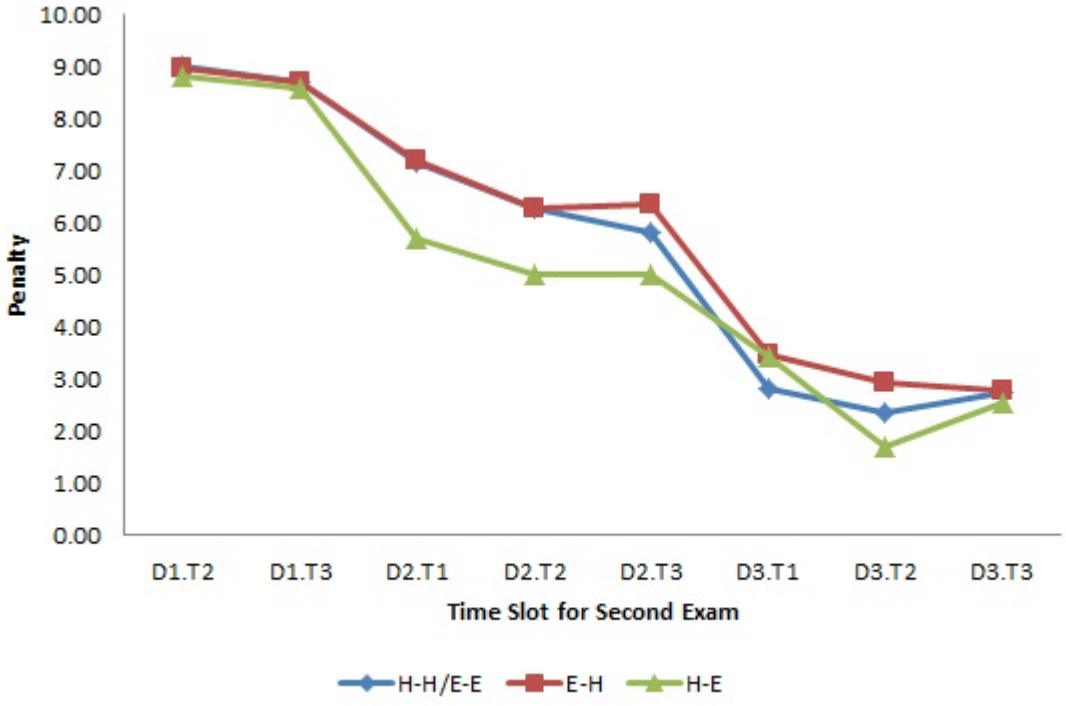


Figure B.2: Penalty given by the students if the first exam is scheduled in Day 1 Timeslot 1 (`D1.T1`) and the second exam is scheduled after the first exam with three different scenarios.

From Figure B.2 we know that no student expects to have two exams in the same day. From the figure we can also observe that before the hard exam, the students expect more gaps than that before the easy one. It is very reasonable, because students need more time for preparing the hard exam.

When the respondents were asked to raise other issues that they would like to be addressed, we got only few other feedbacks out from above mentioned issue. The majority of students highlighted the urgency of at least one day gap between exams, especially for difficult exams. Two consecutive hard exams supposed to be avoided in the timetable.

B.4 Conclusion

This chapter reports our investigation on students preference over their examination timetabling problems in order to know the complexity of real-world examination timetabling problems. Our finding reveals that generally, only 40% of students were happy or very happy with their examination timetable. Whereas there were 14% of them that were unhappy or very happy. Even, we also found that there 28% of students believed that their examination timetable negatively affected their academic achievement. Also, there were 10% of students think that the examination timetable were unfair among students. Since examination contributed significantly on students academic achievement, universities should pay attention on this issues.

In order to provide more representative to the real-world examination timetable, our findings suggest that there are some key issues that should be addressed. First, fairness issue among students especially students enrolled on the same course. Second, the fact that some exams are more difficult than other exams. For the majority of students, difficult exams deserve more time for preparation. These issues should be taken into account in addition to the state-of-the-art examination timetabling problem formulation with the objective to spread out the exams as evenly as possible.

Appendix C

Detailed Experimental Result Data with 'Sum of Powers' Approach

This appendix presents the detailed experimental results data of the experiments with 'Sum of Powers' approach within SA-GD Hyper-heuristic algorithm as discussed in Section 4.3 of Chapter 4 over 13 problem instances of Carter Dataset. Figures C.1 - C.2 visualise the correlation between the number of iteration, standard objective function values, and Anti Jain Fairness Index (AJFI). Figures C.3 - C.7 visualise the comparison of the objective functions values during iterations of SA-GD hyper-heuristic with standard objective function versus 'Sum of Powers' objective function.

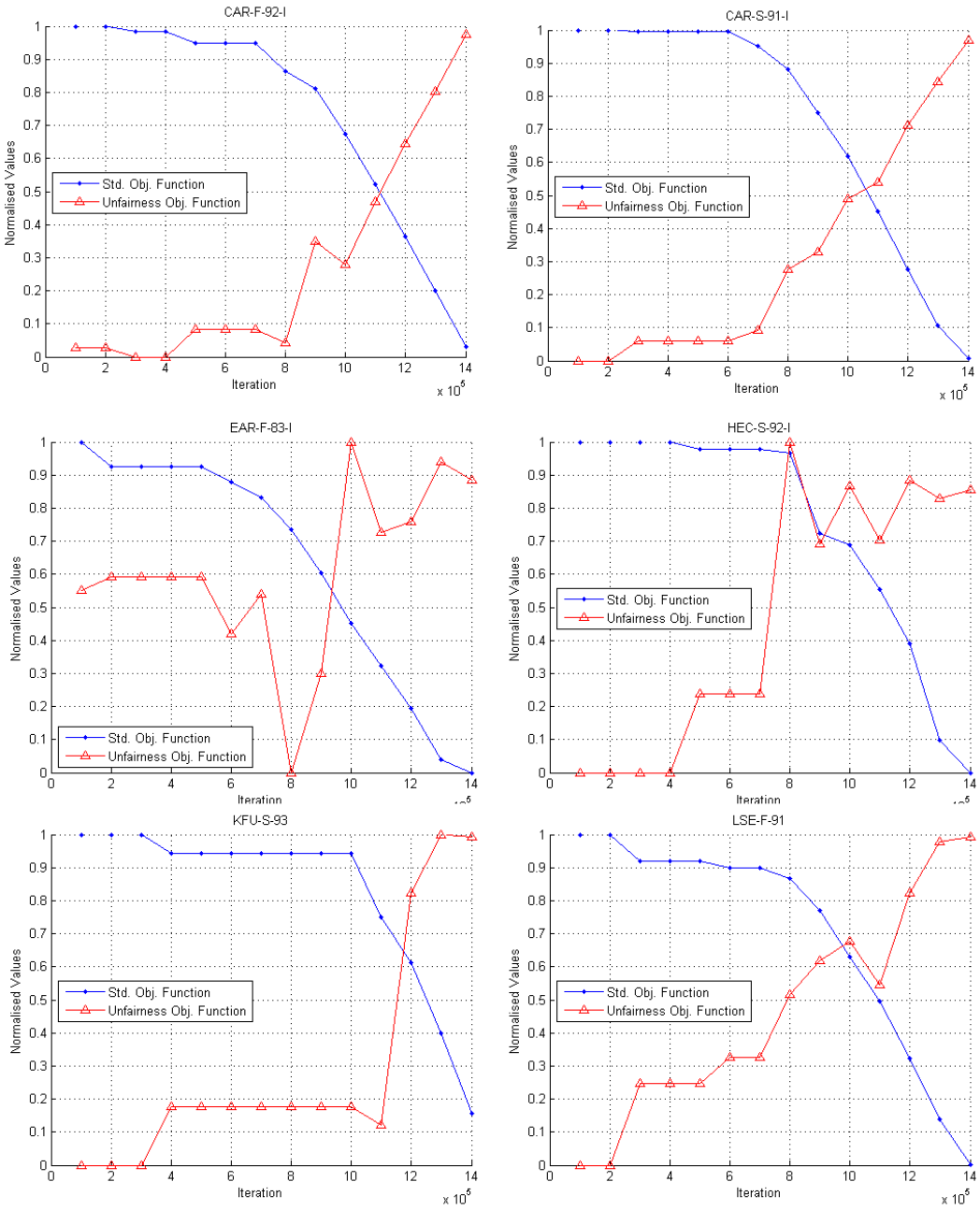


Figure C.1: Correlation between the number of iteration, standard objective function (OF) values, and Anti Jain fairness index (AJFI) values (Normalised Value): Part I

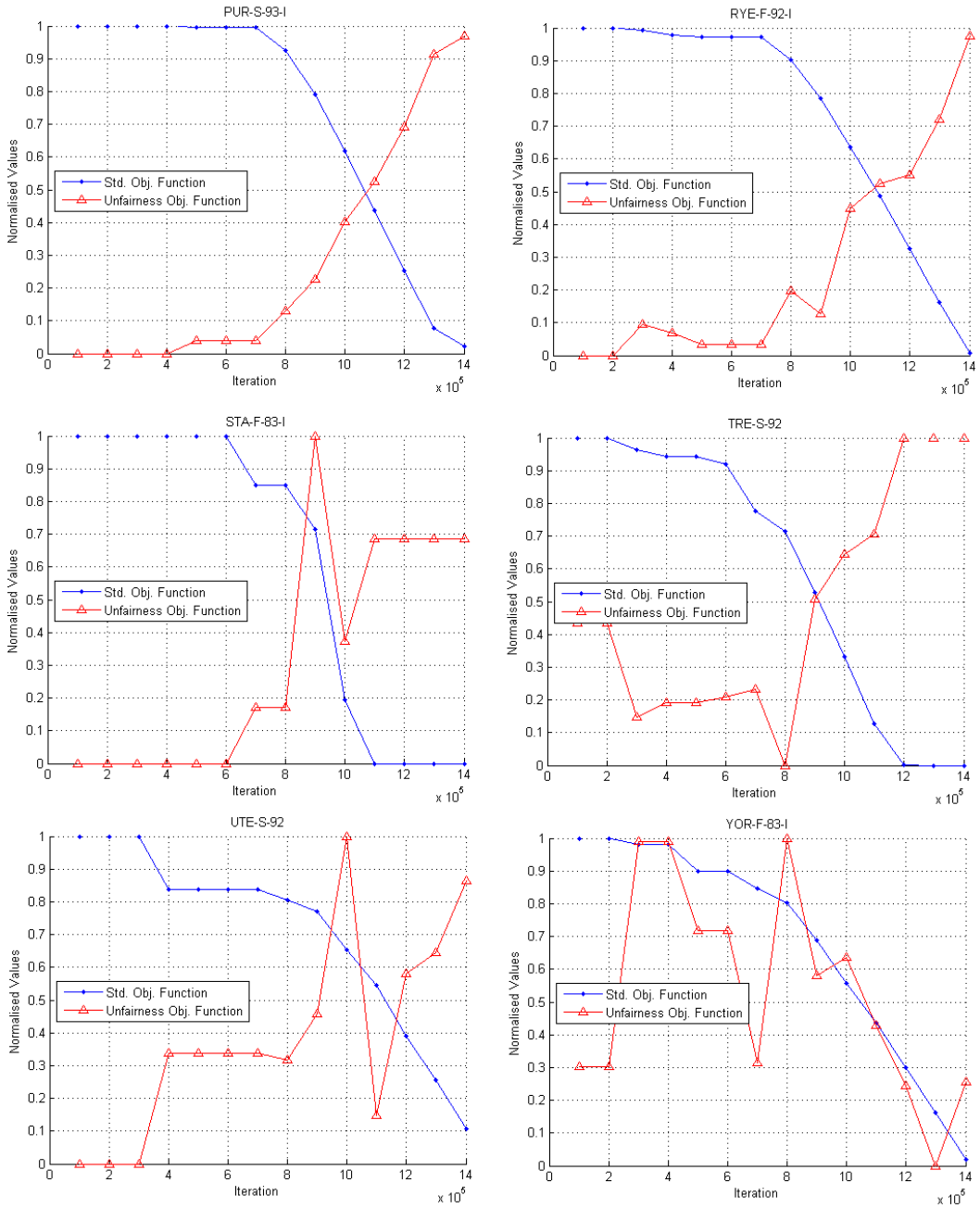


Figure C.2: Correlation between the number of iteration, standard objective function (OF) values, and Anti Jain fairness index (AJFI) values (Normalised Value): Part II

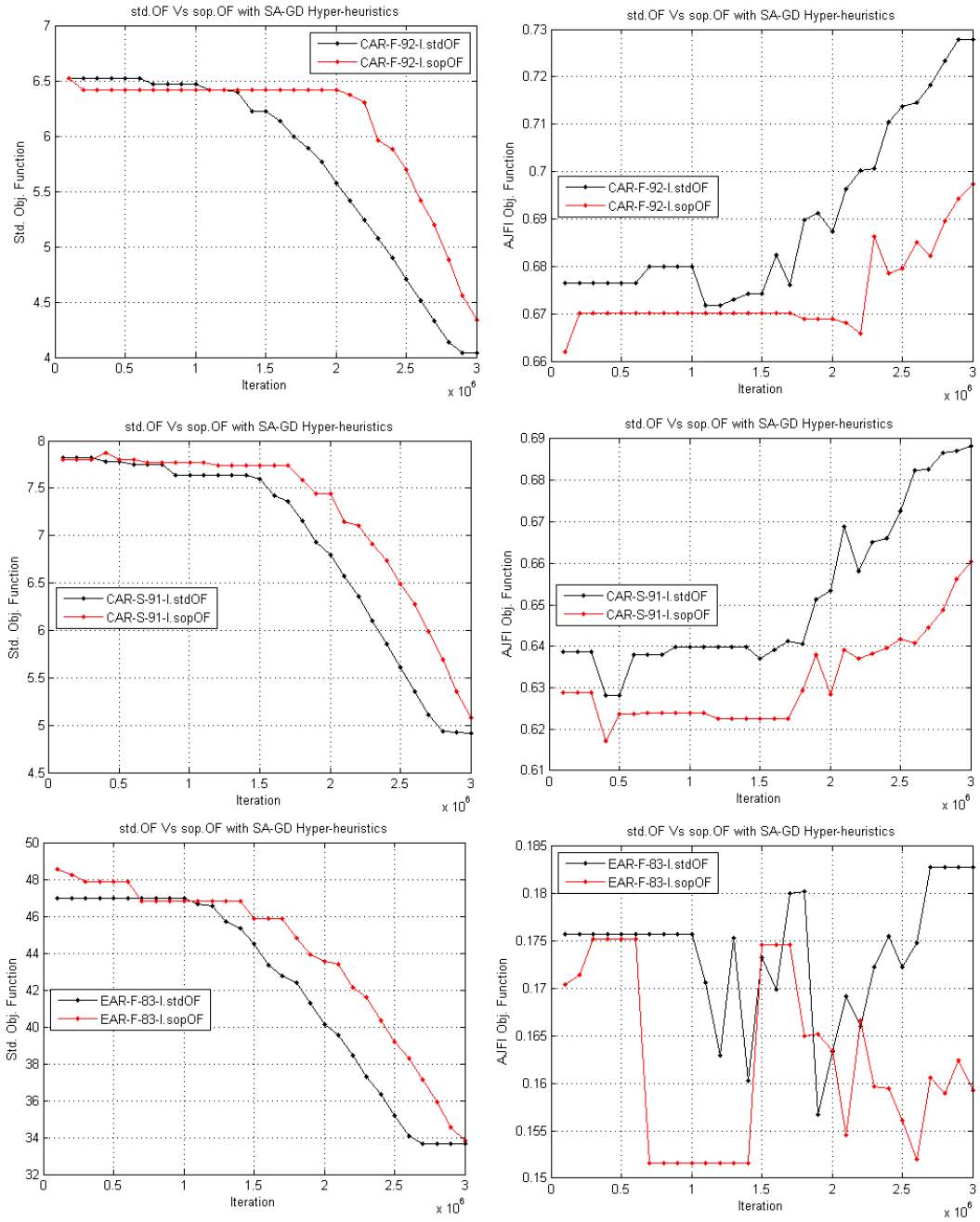


Figure C.3: Comparing the objective functions values during iterations of SA-GD hyper-heuristic with different objective function: standard objective function (stdOF) vs 'Sum of Powers' with q=2 or 'Sum of Square' objective function (sopOF): Part 1

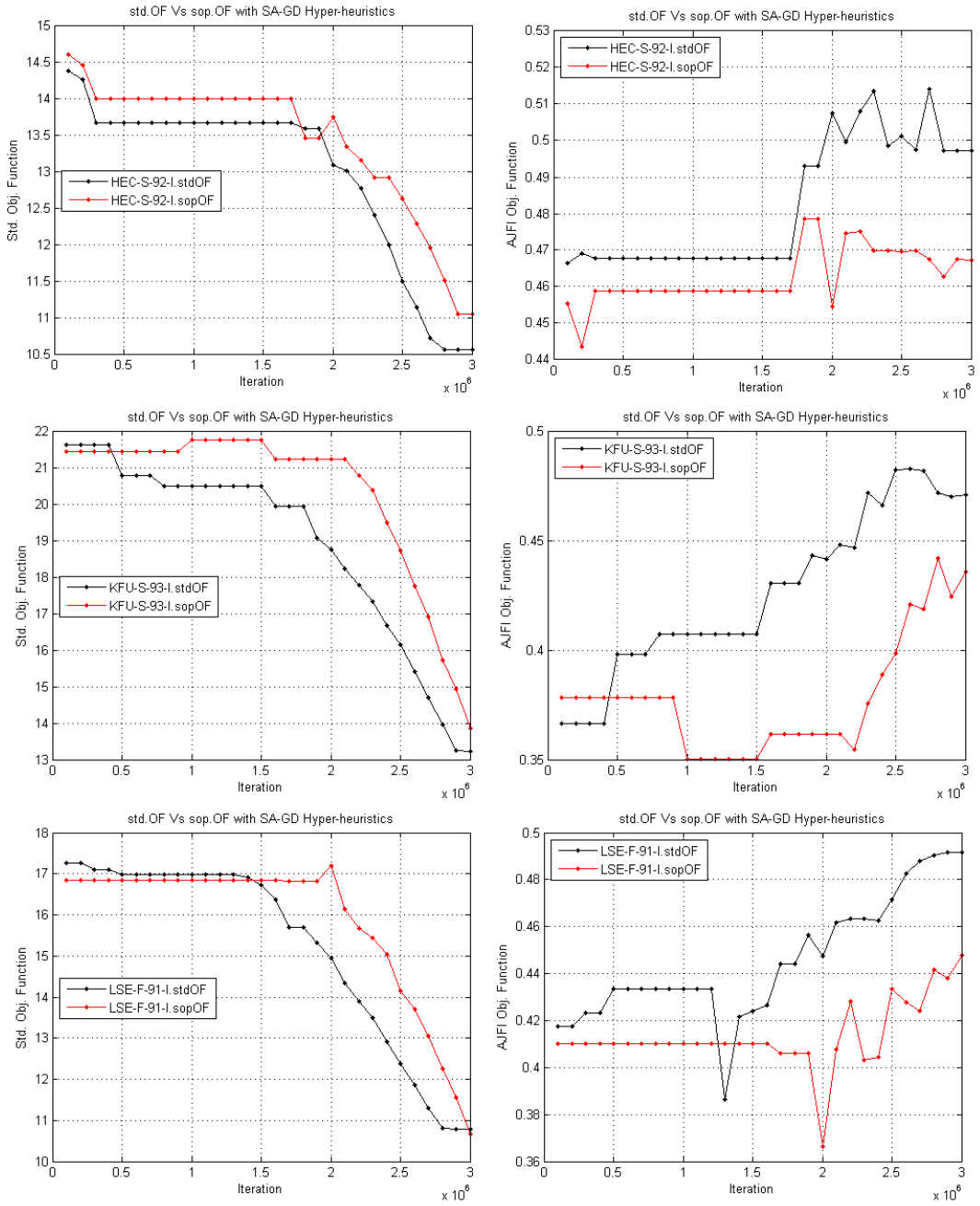


Figure C.4: Comparing the objective functions values during iterations of SA-GD hyper-heuristic with different objective function: standard objective function (stdOF) vs 'Sum of Powers' with q=2 or 'Sum of Square' objective function (sopOF): Part 2

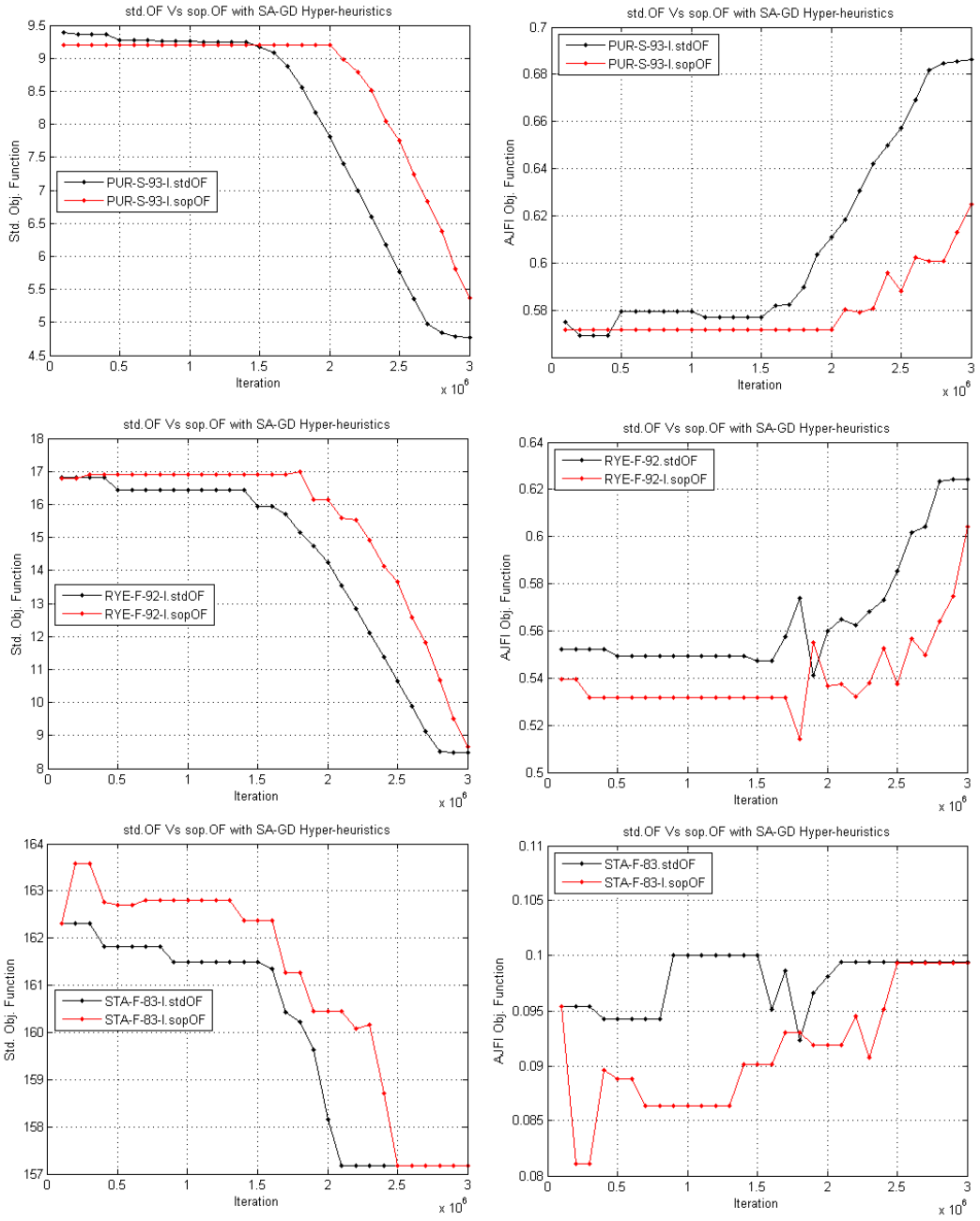


Figure C.5: Comparing the objective functions values during iterations of SA-GD hyper-heuristic with different objective function: standard objective function (stdOF) vs 'Sum of Powers' with `q=2` or 'Sum of Square' objective function (sopOF): Part 3

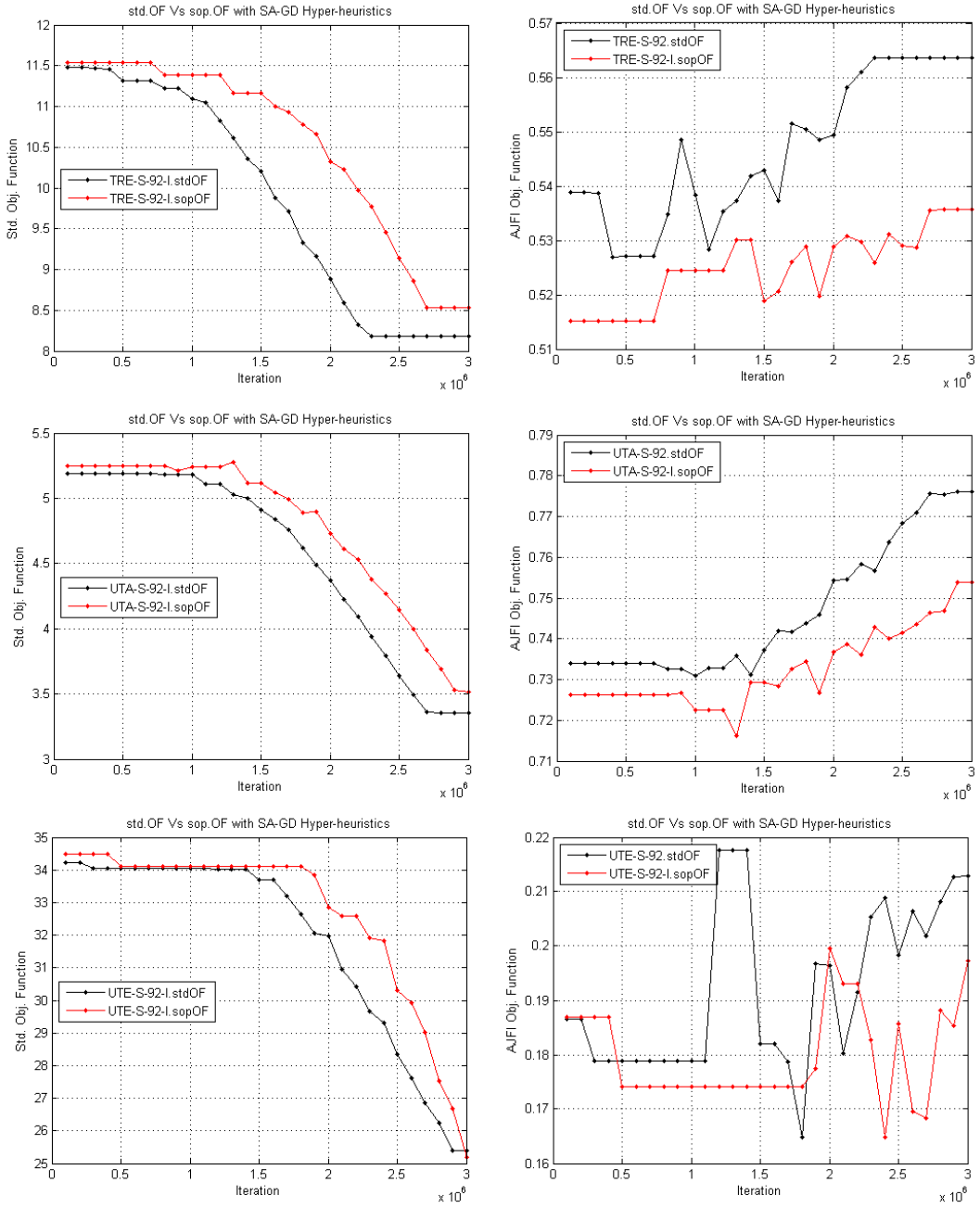


Figure C.6: Comparing the objective functions values during iterations of SA-GD hyper-heuristic with different objective function: standard objective function (stdOF) vs 'Sum of Powers' with `q=2` or 'Sum of Square' objective function (sopOF): Part 4

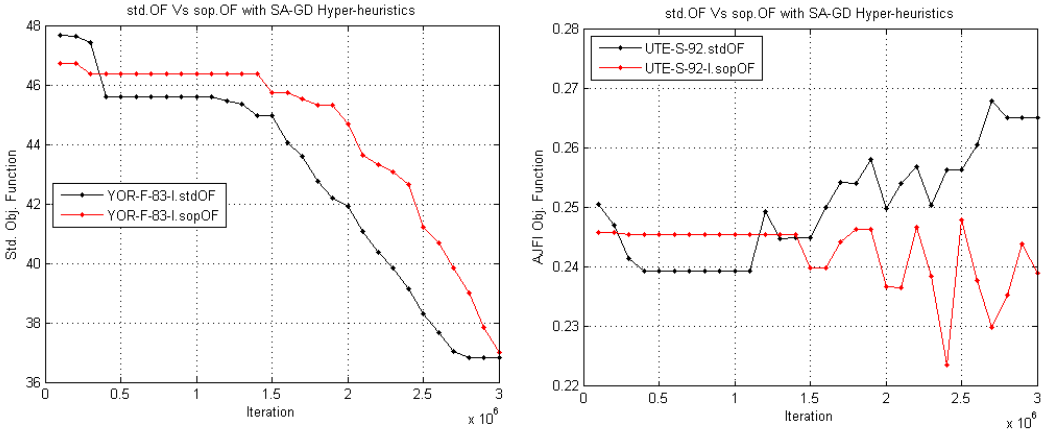


Figure C.7: Comparing the objective functions values during iterations of SA-GD hyper-heuristic with different objective function: standard objective function (stdOF) vs 'Sum of Powers' with q=2 or 'Sum of Square' objective function (sopOF): Part 5

Appendix D

Complete Experimental Results with Classical Scalarisation Techniques-based Hyper-heuristics

This appendix presents the complete experimental results of classical scalarisation techniques-based hyper-heuristics to solve multi-objective examination timetabling problems with fairness over all 39 problem instances of four different datasets, i.e. Nottingham, Carter, ITC 2007, and Yeditepe; instead over only four selected problem instances as given in Section 6.4 of Chapter 6.

Table D.1: The objective function values range of the Pareto solution set from the experiments of a hyper-heuristic given in Algorithm 11 using 'Weighted Summation' as the scalarisation technique under the parameter settings given in Table 6.18 tested over multi-objective examination timetabling problem with bi-objective functions: standard objective function and global unfairness objective function.

P. Instance	Init Solution		STD.OF			Global Unfairness		
	STD.OF	AJFI	min	max	range	min	max	range
NOTT-I	11923	0.79	11600	12515	915	0.73	0.77	0.04
NOTT-II-32	0	0.00	0	0	0	0.00	0.00	0.00
NOTT-II-31	0	0.00	0	0	0	0.00	0.00	0.00
NOTT-II-30	0	0.00	0	0	0	0.00	0.00	0.00
NOTT-II-26	8	0.00	8	8	0	1.00	1.00	0.00
NOTT-II-23	68	0.99	68	71	3	0.99	0.99	0.00
CAR91	5.30	0.68	5.28	5.56	0.28	0.64	0.66	0.02
CAR92	4.51	0.71	4.48	4.73	0.25	0.67	0.70	0.03
EAR83	36.73	0.17	36.61	38.56	1.95	0.13	0.17	0.03
HEC92	10.91	0.51	10.88	11.42	0.54	0.46	0.50	0.04
KFU93	14.36	0.45	14.36	15.08	0.72	0.41	0.45	0.04
LSE91	11.02	0.49	11.00	11.56	0.57	0.43	0.48	0.05
PUR93	5.03	0.67	5.03	5.28	0.25	0.65	0.67	0.02
RYE92	9.01	0.63	9.01	9.46	0.45	0.59	0.63	0.04
STA83	157.12	0.10	157.12	164.97	7.85	0.07	0.10	0.03
TRE92	8.75	0.56	8.73	9.18	0.45	0.51	0.55	0.04
UTA92	3.60	0.77	3.57	3.78	0.20	0.75	0.77	0.02
UTE92	25.20	0.21	25.18	26.43	1.25	0.16	0.20	0.04
YOR83	38.03	0.25	38.03	39.93	1.89	0.21	0.25	0.04
EXAM1	6856	0.82	6743	7195	452	0.77	0.82	0.05
EXAM2	632	1.00	569	663	94	0.99	1.00	0.00
EXAM3	11659	0.93	11405	12212	807	0.92	0.93	0.01
EXAM4	16325	0.73	16287	17141	854	0.70	0.73	0.03
EXAM5	3837	0.86	3837	4028	191	0.84	0.86	0.02
EXAM6	27370	0.79	27265	28725	1460	0.78	0.79	0.01
EXAM7	5528	0.84	5467	5789	322	0.82	0.84	0.02
EXAM8	9798	0.65	9515	10287	772	0.57	0.63	0.06
EXAM9	1246	0.75	1221	1308	87	0.67	0.74	0.07
EXAM10	14556	0.64	14519	15258	739	0.60	0.63	0.03
EXAM11	36810	0.91	36536	38563	2027	0.90	0.91	0.01
EXAM12	5300	0.88	5286	5564	278	0.88	0.88	0.01
YUE20011	56	0.92	51	58	7	0.91	0.92	0.01
YUE20012	122	0.86	121	128	7	0.84	0.85	0.01
YUE20013	29	0.88	29	30	1	0.88	0.88	0.00
YUE20021	76	0.92	76	79	3	0.92	0.92	0.00
YUE20022	162	0.87	150	170	20	0.85	0.87	0.02
YUE20023	56	0.89	56	58	2	0.89	0.89	0.00
YUE20031	143	0.90	138	150	12	0.89	0.90	0.01
YUE20032	434	0.77	417	455	38	0.74	0.77	0.03

Table D.2: The objective function values of the solution with the best standard objective function and the solution with the best global fairness in the Pareto solution set from the experiments of a hyper-heuristic given in Algorithm 11 using 'Weighted Summation' as the scalarisation technique under the parameter settings given in Table 6.18 tested over multi-objective examination timetabling problem with bi-objective functions: standard objective function and global unfairness objective function. The d(%) values are the percentage of the respective objective function values' change from the initial solutions values.

P. Instance	Init. Solution		Sol. With Best STD.OF				Sol. With Best Global Fairness			
			STD.OF		AJFI		STD.OF		AJFI	
	STD.OF	AJFI	Val.	d (%)	Val.	d(%)	Val.	d (%)	Val.	d(%)
NOTT-I	11923	0.79	11600	-2.71	0.77	-3.16	12515	4.97	0.73	-7.97
NOTT-II-32	0	0.00	0	0.00	0.00	0.00	0	0.00	0.00	0.00
NOTT-II-31	0	0.00	0	0.00	0.00	0.00	0	0.00	0.00	0.00
NOTT-II-30	0	0.00	0	0.00	0.00	0.00	0	0.00	0.00	0.00
NOTT-II-26	8	0.00	8	0.00	1.00	0.00	8	0.00	1.00	0.00
NOTT-II-23	68	0.99	68	0.00	0.99	0.00	71	4.41	0.99	-0.06
CAR-S-91	5.30	0.68	5.28	-0.40	0.66	-1.84	5.56	4.91	0.64	-5.45
CAR-F-92	4.51	0.71	4.48	-0.59	0.70	-1.10	4.73	4.95	0.67	-4.82
EAR-F-83	36.73	0.17	36.61	-0.31	0.17	-4.35	38.56	4.99	0.13	-22.79
HEC-S-92	10.91	0.51	10.88	-0.22	0.50	-0.95	11.42	4.71	0.46	-8.89
KFU-S-93	14.36	0.45	14.36	0.00	0.45	0.00	15.08	4.98	0.41	-9.37
LSE-F-91	11.02	0.49	11.00	-0.21	0.48	-1.30	11.56	4.93	0.43	-11.91
PUR-S-93	5.03	0.67	5.03	0.00	0.67	0.00	5.28	4.99	0.65	-2.94
RYE-S-92	9.01	0.63	9.01	0.00	0.63	0.00	9.46	4.98	0.59	-6.48
STA-F-83	157.12	0.10	157.12	0.00	0.10	0.00	164.97	4.99	0.07	-31.49
TRE-S-92	8.75	0.56	8.73	-0.14	0.55	-2.39	9.18	4.99	0.51	-8.85
UTA-S-92	3.60	0.77	3.57	-0.70	0.77	-0.53	3.78	4.95	0.75	-3.25
UTE-S-92	25.20	0.21	25.18	-0.08	0.20	-3.93	26.43	4.87	0.16	-22.98
YOR-F-83	38.03	0.25	38.03	0.00	0.25	0.00	39.93	4.98	0.21	-15.59
EXAM1	6856	0.82	6743	-1.65	0.82	-0.27	7195	4.94	0.77	-6.26
EXAM2	632	1.00	569	-9.97	1.00	-0.03	663	4.91	0.99	-0.47
EXAM3	11659	0.93	11405	-2.18	0.93	0.10	12212	4.74	0.92	-1.30
EXAM4	16325	0.73	16287	-0.23	0.73	-0.11	17141	5.00	0.70	-4.01
EXAM5	3837	0.86	3837	0.00	0.86	-0.03	4028	4.98	0.84	-2.45
EXAM6	27370	0.79	27265	-0.38	0.79	0.05	28725	4.95	0.78	-0.73
EXAM7	5528	0.84	5467	-1.10	0.84	-0.10	5789	4.72	0.82	-1.91
EXAM8	9798	0.65	9515	-2.89	0.63	-3.17	10287	4.99	0.57	-11.83
EXAM9	1246	0.75	1221	-2.01	0.74	-1.30	1308	4.98	0.67	-10.32
EXAM10	14556	0.64	14519	-0.25	0.63	-2.12	15258	4.82	0.60	-7.32
EXAM11	36810	0.91	36536	-0.74	0.91	0.21	38563	4.76	0.90	-1.05
EXAM12	5300	0.88	5286	-0.26	0.88	0.02	5564	4.98	0.88	-1.01
YUE20011	56	0.92	51	-8.93	0.92	0.40	58	3.57	0.91	-0.95
YUE20012	122	0.86	121	-0.82	0.85	-1.01	128	4.92	0.84	-2.51
YUE20013	29	0.88	29	0.00	0.88	0.00	30	3.45	0.88	-0.48
YUE20021	76	0.92	76	0.00	0.92	-0.19	79	3.95	0.92	-0.58
YUE20022	162	0.87	150	-7.41	0.87	-0.32	170	4.94	0.85	-2.82
YUE20023	56	0.89	56	0.00	0.89	0.00	58	3.57	0.89	-0.52
YUE20031	143	0.90	138	-3.50	0.90	0.12	150	4.90	0.89	-1.52
YUE20032	434	0.77	417	-3.92	0.77	0.45	455	4.84	0.74	-3.73

Table D.3: The objective function values range of the Pareto solution set from the experiments of a hyper-heuristic given in Algorithm 11 using 'Weighted Summation' as the scalarisation technique under the parameter settings given in Table 6.18 tested over multi-objective examination timetabling problem with bi-objective functions: standard objective function and 'specified cohort unfairness' objective function

P. Instance	Init Solution		STD.OF			Specified Cohort Unfairness		
	STD.OF	AJFIC1	min	max	range	min	max	range
NOTT-I	11923	0.81	11920	12510	590	0.65	0.79	0.14
NOTT-II-32	0	0.00	0	0	0	0.00	0.00	0.00
NOTT-II-31	0	0.00	0	0	0	0.00	0.00	0.00
NOTT-II-30	0	0.00	0	0	0	0.00	0.00	0.00
NOTT-II-26	8	0.22	8	8	0	0.22	0.22	0.00
NOTT-II-23	68	0.77	68	71	3	0.77	0.77	0.00
CAR-S-91	5.30	0.66	5.28	5.57	0.28	0.62	0.66	0.04
CAR-F-92	4.51	0.68	4.45	4.73	0.28	0.65	0.68	0.03
EAR-F-83	36.73	0.16	36.73	38.56	1.83	0.12	0.16	0.04
HEC-S-92	10.91	0.24	10.90	11.45	0.55	0.20	0.24	0.04
KFU-S-93	14.36	0.32	14.36	15.08	0.72	0.25	0.32	0.07
LSE-F-91	11.02	0.48	11.01	11.57	0.56	0.40	0.46	0.06
PUR-S-93	5.03	0.70	5.02	5.27	0.24	0.65	0.69	0.04
RYE-S-92	9.01	0.41	9.01	9.45	0.45	0.32	0.41	0.09
STA-F-83	157.12	0.00	157.12	161.54	4.41	0.00	0.00	0.00
TRE-S-92	8.75	0.56	8.74	9.18	0.45	0.49	0.53	0.04
UTA-S-92	3.60	0.74	3.57	3.76	0.19	0.71	0.73	0.03
UTE-S-92	25.20	0.14	25.11	26.45	1.35	0.11	0.14	0.02
YOR-F-83	38.03	0.24	38.03	39.93	1.90	0.20	0.24	0.04
EXAM1	6856	0.78	6856	7184	328	0.46	0.78	0.32
EXAM2	632	0.67	607	651	44	0.66	0.67	0.01
EXAM3	11659	0.90	11422	12135	713	0.88	0.90	0.02
EXAM4	16325	0.78	16056	17136	1080	0.56	0.79	0.22
EXAM5	3837	0.46	3837	4018	181	0.32	0.46	0.13
EXAM6	27370	0.61	27080	28560	1480	0.59	0.60	0.02
EXAM7	5528	0.62	5528	5788	260	0.57	0.62	0.05
EXAM8	9798	0.44	9790	10273	483	0.34	0.44	0.10
EXAM9	1246	0.68	1241	1304	63	0.58	0.66	0.08
EXAM10	14556	0.50	14556	15282	726	0.45	0.50	0.05
EXAM11	36810	0.86	36000	38136	2136	0.84	0.86	0.02
EXAM12	5300	0.75	5300	5565	265	0.49	0.75	0.26
YUE20011	56	0.90	54	58	4	0.89	0.90	0.01
YUE20012	122	0.85	120	128	8	0.83	0.85	0.02
YUE20013	29	0.90	29	30	1	0.89	0.89	0.01
YUE20021	76	0.73	69	79	10	0.53	0.54	0.01
YUE20022	162	0.73	155	170	15	0.68	0.73	0.06
YUE20023	56	0.59	56	58	2	0.59	0.59	0.01
YUE20031	143	0.92	142	150	8	0.74	0.92	0.18
YUE20032	434	0.64	433	455	22	0.60	0.63	0.03

Table D.4: The objective function values of the solution with the best standard objective function and the solution with the best global fairness in the Pareto solution set from the experiments of a hyper-heuristic given in Algorithm 11 using 'Weighted Summation' as the scalarisation technique under the parameter settings given in Table 6.18 tested over multi-objective examination timetabling problem with bi-objective functions: standard objective function and 'specified cohort unfairness' objective function. The d(%) values are the percentage of the respective objective function values' change from the initial solutions values.

P. Instance	Init Solution		Sol. With Best STD.OF				Sol. With Best Spec. Cohort Unfairness			
			STD.OF		AJFIC1		STD.OF		AJFIC1	
	STD.OF	AJFIC1	Val.	d (%)	Val.	d(%)	Val.	d (%)	Val.	d(%)
NOTT-I	11923	0.81	11920	-0.03	0.79	-2.16	12510	4.92	0.65	-19.56
NOTT-II-32	0	0.00	0	0.00	0.00	0.00	0	0.00	0.00	0.00
NOTT-II-31	0	0.00	0	0.00	0.00	0.00	0	0.00	0.00	0.00
NOTT-II-30	0	0.00	0	0.00	0.00	0.00	0	0.00	0.00	0.00
NOTT-II-26	8	0.22	8	0.00	0.22	0.00	8	0.00	0.22	0.00
NOTT-II-23	68	0.77	68	0.00	0.77	0.00	71	4.41	0.77	-0.15
CAR-S-91	5.30	0.66	5.28	-0.33	0.66	-0.28	5.57	4.97	0.62	-6.34
CAR-F-92	4.51	0.68	4.45	-1.23	0.68	-1.30	4.73	4.94	0.65	-5.70
EAR-F-83	36.73	0.16	36.73	0.00	0.16	0.00	38.56	5.00	0.12	-27.26
HEC-S-92	10.91	0.24	10.90	-0.07	0.24	-2.76	11.45	4.93	0.20	-19.65
KFU-S-93	14.36	0.32	14.36	0.00	0.32	0.00	15.08	5.00	0.25	-20.96
LSE-F-91	11.02	0.48	11.01	-0.09	0.46	-3.44	11.57	4.97	0.40	-15.33
PUR-S-93	5.03	0.70	5.02	-0.08	0.69	-0.84	5.27	4.78	0.65	-6.79
RYE-S-92	9.01	0.41	9.01	0.00	0.41	0.00	9.45	4.97	0.32	-21.82
STA-F-83	157.12	0.00	157.12	0.00	0.00	0.00	161.54	2.81	0.00	-88.47
TRE-S-92	8.75	0.56	8.74	-0.10	0.53	-4.52	9.18	4.99	0.49	-11.82
UTA-S-92	3.60	0.74	3.57	-0.74	0.73	-1.13	3.76	4.59	0.71	-4.74
UTE-S-92	25.20	0.14	25.11	-0.37	0.14	-3.80	26.45	4.97	0.11	-19.78
YOR-F-83	38.03	0.24	38.03	0.00	0.24	0.00	39.93	5.00	0.20	-17.77
EXAM1	6856	0.78	6856	0.00	0.78	0.00	7184	4.78	0.46	-40.86
EXAM2	632	0.67	607	-3.96	0.67	0.00	651	3.01	0.66	-0.81
EXAM3	11659	0.90	11422	-2.03	0.90	0.10	12135	4.08	0.88	-1.73
EXAM4	16325	0.78	16056	-1.65	0.79	0.47	17136	4.97	0.56	-28.26
EXAM5	3837	0.46	3837	0.00	0.46	0.00	4018	4.72	0.32	-29.65
EXAM6	27370	0.61	27080	-1.06	0.60	-0.81	28560	4.35	0.59	-3.33
EXAM7	5528	0.62	5528	0.00	0.62	0.00	5788	4.70	0.57	-7.67
EXAM8	9798	0.44	9790	-0.08	0.44	-0.52	10273	4.85	0.34	-23.18
EXAM9	1246	0.68	1241	-0.40	0.66	-3.28	1304	4.65	0.58	-14.49
EXAM10	14556	0.50	14556	0.00	0.50	0.00	15282	4.99	0.45	-10.30
EXAM11	36810	0.86	36000	-2.20	0.86	-0.43	38136	3.60	0.84	-2.92
EXAM12	5300	0.75	5300	0.00	0.75	0.00	5565	5.00	0.49	-34.94
YUE20011	56	0.90	54	-3.57	0.90	0.07	58	3.57	0.89	-1.39
YUE20012	122	0.85	120	-1.64	0.85	0.49	128	4.92	0.83	-2.13
YUE20013	29	0.90	29	0.00	0.89	-0.13	30	3.45	0.89	-0.95
YUE20021	76	0.73	69	-9.21	0.54	-25.73	79	3.95	0.53	-27.69
YUE20022	162	0.73	155	-4.32	0.73	0.30	170	4.94	0.68	-7.27
YUE20023	56	0.59	56	0.00	0.59	0.00	58	3.57	0.59	-1.01
YUE20031	143	0.92	142	-0.70	0.92	0.08	150	4.90	0.74	-19.14
YUE20032	434	0.64	433	-0.23	0.63	-0.82	455	4.84	0.60	-5.36

Table D.5: The objective function values range of the Pareto solution set from the experiments of a hyper-heuristic given in Algorithm 11 using 'Weighted Summation' as the scalarisation technique under the parameter settings given in Table 6.18 tested over multi-objective examination timetabling problem with bi-objective functions: standard objective function and 'exam-count-based cohort' (ECB Cohort) unfairness objective function

P. Instance	Init Solution		STD.OF			ECB Cohort Unfairness		
	STD.OF	AJFI.C2	min	max	range	min	max	range
NOTT-I	11923	0.54	11918	12510	592	0.47	0.54	0.07
NOTT-II-32	0	0.00	0	0	0	0.00	0.00	0.00
NOTT-II-31	0	0.00	0	0	0	0.00	0.00	0.00
NOTT-II-30	0	0.00	0	0	0	0.00	0.00	0.00
NOTT-II-26	8	0.22	8	8	0	0.22	0.22	0.00
NOTT-II-23	68	0.65	68	71	3	0.64	0.65	0.00
CAR-S-91	5.30	0.37	5.30	5.57	0.26	0.30	0.37	0.07
CAR-F-92	4.51	0.43	4.51	4.72	0.21	0.35	0.43	0.07
EAR-F-83	36.73	0.10	36.59	38.56	1.97	0.06	0.09	0.04
HEC-S-92	10.91	0.29	10.91	11.41	0.50	0.25	0.29	0.04
KFU-S-93	14.36	0.33	14.36	15.07	0.71	0.19	0.33	0.15
LSE-F-91	11.02	0.26	11.02	11.57	0.55	0.20	0.26	0.07
PUR-S-93	5.03	0.42	5.03	5.28	0.25	0.38	0.42	0.04
RYE-S-92	9.01	0.30	9.01	9.45	0.44	0.25	0.30	0.05
STA-F-83	157.12	0.00	157.12	161.96	4.83	0.00	0.00	0.00
TRE-S-92	8.75	0.37	8.75	9.18	0.43	0.31	0.37	0.06
UTA-S-92	3.60	0.44	3.59	3.78	0.19	0.39	0.41	0.02
UTE-S-92	25.20	0.21	25.20	26.41	1.21	0.10	0.21	0.11
YOR-F-83	38.03	0.23	38.03	39.93	1.90	0.17	0.23	0.06
EXAM1	6856	0.47	6784	7180	396	0.41	0.46	0.06
EXAM2	632	0.49	627	661	34	0.43	0.49	0.06
EXAM3	11659	0.58	11615	12117	502	0.54	0.58	0.03
EXAM4	16325	0.50	16325	17134	809	0.43	0.50	0.07
EXAM5	3837	0.61	3837	4027	190	0.51	0.61	0.10
EXAM6	27370	0.29	27370	27800	430	0.28	0.29	0.01
EXAM7	5528	0.36	5439	5804	365	0.33	0.36	0.03
EXAM8	9798	0.29	9798	10278	480	0.25	0.29	0.05
EXAM9	1246	0.24	1241	1302	61	0.18	0.22	0.04
EXAM10	14556	0.08	14556	15279	723	0.02	0.08	0.06
EXAM11	36810	0.55	36759	38586	1827	0.48	0.55	0.06
EXAM12	5300	0.57	5287	5565	278	0.51	0.53	0.02
YUE20011	56	0.45	54	58	4	0.41	0.45	0.04
YUE20012	122	0.57	117	128	11	0.41	0.43	0.02
YUE20013	29	0.37	29	30	1	0.35	0.37	0.01
YUE20021	76	0.56	76	79	3	0.52	0.54	0.02
YUE20022	162	0.54	157	170	13	0.46	0.50	0.04
YUE20023	56	0.47	56	58	2	0.32	0.33	0.01
YUE20031	143	0.60	138	150	12	0.58	0.60	0.02
YUE20032	434	0.55	434	455	21	0.37	0.55	0.18

Table D.6: The objective function values of the solution with the best standard objective function and the solution with the best global fairness in the Pareto solution set from the experiments of a hyper-heuristic given in Algorithm 11 using 'Weighted Summation' as the scalarisation technique under the parameter settings given in Table 6.18 tested over multi-objective examination timetabling problem with bi-objective functions: standard objective function and 'exam-count-based cohort' (ECB Cohort) unfairness objective function

P. Instance	Init Solution		Sol. With Best STD.OF				Sol. With Best ECB Cohort Unfairness			
			STD.OF		AJFIC2		STD.OF		AJFIC2	
	STD.OF	AJFIC2	Val.	d (%)	Val.	d(%)	Val.	d (%)	Val.	d(%)
NOTT-I	11923	0.54	11918	-0.04	0.54	-0.56	12510	4.92	0.47	-13.48
NOTT-II-32	0	0.00	0	0.00	0.00	0.00	0	0.00	0.00	0.00
NOTT-II-31	0	0.00	0	0.00	0.00	0.00	0	0.00	0.00	0.00
NOTT-II-30	0	0.00	0	0.00	0.00	0.00	0	0.00	0.00	0.00
NOTT-II-26	8	0.22	8	0.00	0.22	0.00	8	0.00	0.22	0.00
NOTT-II-23	68	0.65	68	0.00	0.65	0.00	71	4.41	0.64	-0.76
CAR-S-91	5.30	0.37	5.30	0.00	0.37	0.00	5.57	4.98	0.30	-19.06
CAR-F-92	4.51	0.43	4.51	0.00	0.43	0.00	4.72	4.65	0.35	-17.34
EAR-F-83	36.73	0.10	36.59	-0.37	0.09	-5.50	38.56	4.99	0.06	-43.16
HEC-S-92	10.91	0.29	10.91	0.00	0.29	0.00	11.41	4.56	0.25	-13.98
KFU-S-93	14.36	0.33	14.36	0.00	0.33	0.00	15.07	4.95	0.19	-44.30
LSE-F-91	11.02	0.26	11.02	0.00	0.26	0.00	11.57	4.98	0.20	-25.06
PUR-S-93	5.03	0.42	5.03	0.00	0.42	0.00	5.28	5.00	0.38	-10.64
RYE-S-92	9.01	0.30	9.01	0.00	0.30	-0.38	9.45	4.87	0.25	-16.06
STA-F-83	157.12	0.00	157.12	0.00	0.00	0.00	161.96	3.07	0.00	-84.57
TRE-S-92	8.75	0.37	8.75	0.00	0.37	0.00	9.18	4.94	0.31	-15.86
UTA-S-92	3.60	0.44	3.59	-0.39	0.41	-6.81	3.78	4.94	0.39	-12.13
UTE-S-92	25.20	0.21	25.20	0.00	0.21	0.00	26.41	4.79	0.10	-52.63
YOR-F-83	38.03	0.23	38.03	0.00	0.23	0.00	39.93	4.99	0.17	-27.47
EXAM1	6856	0.47	6784	-1.05	0.46	-2.48	7180	4.73	0.41	-14.44
EXAM2	632	0.49	627	-0.79	0.49	0.00	661	4.59	0.43	-12.31
EXAM3	11659	0.58	11615	-0.38	0.58	0.02	12117	3.93	0.54	-5.78
EXAM4	16325	0.50	16325	0.00	0.50	0.00	17134	4.96	0.43	-14.63
EXAM5	3837	0.61	3837	0.00	0.61	0.00	4027	4.95	0.51	-16.38
EXAM6	27370	0.29	27370	0.00	0.29	0.00	27800	1.57	0.28	-3.98
EXAM7	5528	0.36	5439	-1.61	0.36	0.39	5804	4.99	0.33	-7.20
EXAM8	9798	0.29	9798	0.00	0.29	0.00	10278	4.90	0.25	-16.14
EXAM9	1246	0.24	1241	-0.40	0.22	-8.12	1302	4.49	0.18	-26.31
EXAM10	14556	0.08	14556	0.00	0.08	0.00	15279	4.97	0.02	-77.92
EXAM11	36810	0.55	36759	-0.14	0.55	-0.49	38586	4.82	0.48	-11.99
EXAM12	5300	0.57	5287	-0.25	0.53	-6.54	5565	5.00	0.51	-10.49
YUE20011	56	0.45	54	-3.57	0.45	0.27	58	3.57	0.41	-9.54
YUE20012	122	0.57	117	-4.10	0.43	-23.66	128	4.92	0.41	-27.02
YUE20013	29	0.37	29	0.00	0.37	0.00	30	3.45	0.35	-2.98
YUE20021	76	0.56	76	0.00	0.54	-1.87	79	3.95	0.52	-5.42
YUE20022	162	0.54	157	-3.09	0.50	-7.75	170	4.94	0.46	-15.88
YUE20023	56	0.47	56	0.00	0.33	-29.70	58	3.57	0.32	-31.19
YUE20031	143	0.60	138	-3.50	0.60	0.25	150	4.90	0.58	-3.70
YUE20032	434	0.55	434	0.00	0.55	-0.03	455	4.84	0.37	-32.86

Table D.7: The objective function values range of the Pareto solution set from the experiments of a hyper-heuristic given in Algorithm 11 using 'Weighted Summation' as the scalarisation technique under the parameter settings given in Table 6.18 tested over multi-objective examination timetabling problem with three-objective functions: standard objective function, global unfairness objective function and 'specified cohort unfairness' objective function

P. Instance	Init Solution			STD.OF			Global Fairness			Spec. Cohort Unfairness		
	STD.OF	AJFI	AJFIC1	min	max	range	min	max	range	min	max	range
NOTT-I	11923	0.79	0.81	11920	12519	599	0.77	0.79	0.02	0.75	0.81	0.06
NOTT-II-32	0	0.00	0.00	0	0	0	0.00	0.00	0.00	0.00	0.00	0.00
NOTT-II-31	0	0.00	0.00	0	0	0	0.00	0.00	0.00	0.00	0.00	0.00
NOTT-II-30	0	0.00	0.00	0	0	0	0.00	0.00	0.00	0.00	0.00	0.00
NOTT-II-26	8	1.00	0.22	8	8	0	1.00	1.00	0.00	0.22	0.22	0.00
NOTT-II-23	68	0.99	0.77	68	71	3	0.99	0.99	0.00	0.77	0.77	0.00
CAR-S-91	5.30	0.68	0.66	5.30	5.57	0.26	0.66	0.68	0.02	0.64	0.66	0.03
CAR-F-92	4.51	0.71	0.68	4.51	4.73	0.22	0.69	0.71	0.02	0.66	0.68	0.02
EAR-F-83	36.73	0.17	0.16	36.73	38.56	1.84	0.14	0.17	0.03	0.12	0.16	0.03
HEC-S-92	10.91	0.51	0.24	10.91	11.45	0.54	0.47	0.51	0.05	0.20	0.24	0.04
KFU-S-93	14.36	0.45	0.32	14.36	15.08	0.72	0.41	0.45	0.04	0.27	0.32	0.05
LSE-F-91	11.02	0.49	0.48	11.02	11.57	0.55	0.44	0.49	0.04	0.43	0.48	0.04
PUR-S-93	5.03	0.67	0.70	5.02	5.28	0.25	0.65	0.67	0.01	0.64	0.70	0.05
RYE-S-92	9.01	0.63	0.41	9.00	9.46	0.46	0.60	0.63	0.03	0.33	0.41	0.08
STA-F-83	157.12	0.10	0.00	157.12	164.97	7.84	0.07	0.10	0.03	0.00	0.00	0.00
TRE-S-92	8.75	0.56	0.56	8.75	9.18	0.44	0.53	0.56	0.03	0.51	0.56	0.04
UTA-S-92	3.60	0.77	0.74	3.60	3.78	0.18	0.76	0.77	0.02	0.72	0.74	0.02
UTE-S-92	25.20	0.21	0.14	25.20	26.45	1.25	0.17	0.21	0.04	0.12	0.14	0.02
YOR-F-83	38.03	0.25	0.24	38.03	39.93	1.90	0.22	0.25	0.03	0.21	0.24	0.03
EXAM1	6856	0.82	0.78	6846	7195	349	0.79	0.82	0.03	0.51	0.78	0.27
EXAM2	632	1.00	0.67	627	663	36	1.00	1.00	0.00	0.66	0.67	0.00
EXAM3	11659	0.93	0.90	11636	12241	605	0.92	0.93	0.01	0.88	0.90	0.02
EXAM4	16325	0.73	0.78	16325	17130	805	0.71	0.73	0.03	0.68	0.78	0.10
EXAM5	3837	0.86	0.46	3837	4027	190	0.84	0.86	0.02	0.32	0.47	0.15
EXAM6	27370	0.79	0.61	27370	28735	1365	0.78	0.79	0.01	0.59	0.61	0.02
EXAM7	5528	0.84	0.62	5518	5803	285	0.82	0.84	0.01	0.60	0.62	0.02
EXAM8	9798	0.65	0.44	9784	10287	503	0.62	0.65	0.03	0.38	0.44	0.06
EXAM9	1246	0.75	0.68	1246	1308	62	0.71	0.75	0.04	0.63	0.68	0.06
EXAM10	14556	0.64	0.50	14556	15141	585	0.61	0.64	0.04	0.46	0.50	0.04
EXAM11	36810	0.91	0.86	36777	38620	1843	0.90	0.91	0.01	0.85	0.86	0.01
EXAM12	5300	0.88	0.75	5300	5565	265	0.88	0.88	0.01	0.50	0.77	0.27
YUE20011	56	0.92	0.90	54	58	4	0.91	0.92	0.01	0.88	0.89	0.01
YUE20012	122	0.86	0.85	122	128	6	0.84	0.86	0.02	0.83	0.85	0.02
YUE20013	29	0.88	0.90	29	30	1	0.88	0.88	0.00	0.89	0.89	0.01
YUE20021	76	0.92	0.73	76	79	3	0.92	0.92	0.00	0.70	0.73	0.03
YUE20022	162	0.87	0.73	162	170	8	0.86	0.87	0.01	0.70	0.73	0.03
YUE20023	56	0.89	0.59	56	58	2	0.89	0.89	0.00	0.59	0.59	0.01
YUE20031	143	0.90	0.92	143	150	7	0.89	0.90	0.01	0.90	0.92	0.02
YUE20032	434	0.77	0.64	434	455	21	0.75	0.77	0.02	0.61	0.78	0.17

Table D.8: The objective function values of the solution with the best specified cohort fairness in the Pareto solution set from the experiments of a hyper-heuristic given in Algorithm 11 using 'Weighted Summation' as the scalarisation technique under the parameter settings given in Table 6.18 tested over multi-objective examination timetabling problem with three-objective functions: standard objective function, global unfairness objective function and 'specified cohort unfairness' objective function. The d(%) values are the percentage of the respective objective function values' change from the initial solutions values.

P. Instance	Init Solution			Sol. With Best STD.OF						Sol. With Best Global Unfairness						Sol. With Best Specified Cohort Unfairness					
				STD.OF		AJFI		AJFIC1		STD.OF		AJFI		AJFIC1		STD.OF		AJFI		AJFIC1	
	STD.OF	AJFI	AJFIC1	Val.	d (%)	Val.	d(%)	Val.	d(%)	Val.	d (%)	Val.	d(%)	Val.	d(%)	Val.	d (%)	Val.	d(%)	Val.	d(%)
NOTT-I	11923	0.79	0.81	11920	-0.03	0.79	-0.11	0.81	-0.41	12480	4.67	0.77	-2.73	0.77	-4.56	12377	3.81	0.78	-1.55	0.75	-7.91
NOTT-II-32	0	0.00	0.00	0	0.00	0.00	0.00	0.00	0.00	0	0.00	0.00	0.00	0.00	0.00	0	0.00	0.00	0.00	0.00	0.00
NOTT-II-31	0	0.00	0.00	0	0.00	0.00	0.00	0.00	0.00	0	0.00	0.00	0.00	0.00	0.00	0	0.00	0.00	0.00	0.00	0.00
NOTT-II-30	0	0.00	0.00	0	0.00	0.00	0.00	0.00	0.00	0	0.00	0.00	0.00	0.00	0.00	0	0.00	0.00	0.00	0.00	0.00
NOTT-II-26	8	1.00	0.22	8	0.00	1.00	0.00	0.22	0.00	8	0.00	1.00	0.00	0.22	0.00	8	0.00	1.00	0.00	0.22	0.00
NOTT-II-23	68	0.99	0.77	68	0.00	0.99	0.00	0.77	0.00	71	4.41	0.99	-0.06	0.77	-0.14	71	4.41	0.99	-0.04	0.77	-0.15
CAR-S-91	5.30	0.68	0.66	5.30	-0.01	0.68	0.00	0.66	0.00	5.57	4.97	0.66	-3.22	0.64	-3.85	5.56	4.81	0.66	-3.18	0.64	-3.96
CAR-F-92	4.51	0.71	0.68	4.51	0.00	0.71	-0.01	0.68	-0.01	4.73	4.97	0.69	-2.45	0.67	-2.92	4.73	4.98	0.69	-2.41	0.66	-3.02
EAR-F-83	36.73	0.17	0.16	36.73	0.00	0.17	-0.83	0.16	-0.94	38.56	5.00	0.14	-19.98	0.12	-21.41	38.55	4.97	0.14	-18.91	0.12	-22.00
HEC-S-92	10.91	0.51	0.24	10.91	0.00	0.51	0.00	0.24	0.00	11.44	4.88	0.47	-7.83	0.21	-13.01	11.37	4.21	0.48	-5.69	0.20	-15.83
KFU-S-93	14.36	0.45	0.32	14.36	-0.01	0.45	-0.05	0.32	-0.04	15.08	4.99	0.41	-9.29	0.28	-11.89	15.06	4.90	0.42	-8.46	0.27	-15.17
LSE-F-91	11.02	0.49	0.48	11.02	0.00	0.49	-0.13	0.48	-0.12	11.57	4.95	0.44	-9.05	0.44	-8.09	11.57	4.99	0.44	-8.99	0.43	-8.93
PUR-S-93	5.03	0.67	0.70	5.02	-0.04	0.67	-0.01	0.70	0.00	5.28	4.98	0.65	-2.10	0.65	-6.76	5.27	4.95	0.65	-2.03	0.64	-7.49
RYE-S-92	9.01	0.63	0.41	9.00	-0.13	0.63	0.11	0.41	-0.07	9.46	4.98	0.60	-4.48	0.38	-6.84	9.45	4.95	0.61	-2.13	0.33	-18.44
STA-F-83	157.12	0.10	0.00	157.12	0.00	0.10	0.00	0.00	0.00	164.97	4.99	0.07	-31.69	0.00	-31.61	159.62	1.59	0.09	-6.51	0.00	-72.62
TRE-S-92	8.75	0.56	0.56	8.75	0.00	0.56	0.00	0.56	0.00	9.18	4.99	0.53	-5.76	0.51	-7.41	9.18	4.95	0.53	-5.67	0.51	-7.49
UTA-S-92	3.60	0.77	0.74	3.60	0.00	0.77	0.00	0.74	0.00	3.77	4.85	0.76	-1.95	0.72	-2.34	3.78	4.92	0.76	-1.82	0.72	-2.56
UTE-S-92	25.20	0.21	0.14	25.20	0.00	0.21	-0.02	0.14	-0.02	26.45	4.96	0.17	-19.23	0.12	-15.09	26.39	4.71	0.17	-19.23	0.12	-15.12
YOR-F-83	38.03	0.25	0.24	38.03	0.00	0.25	0.00	0.24	0.00	39.89	4.90	0.22	-12.09	0.21	-11.64	39.90	4.92	0.22	-11.19	0.21	-13.25
EXAM1	6856	0.82	0.78	6846	-0.15	0.82	-0.07	0.78	-0.06	7192	4.90	0.79	-3.39	0.67	-13.56	7168	4.55	0.80	-2.60	0.51	-34.95
EXAM2	632	1.00	0.67	627	-0.79	1.00	0.00	0.67	0.00	653	3.32	1.00	-0.09	0.67	-0.10	663	4.91	1.00	-0.07	0.66	-0.53
EXAM3	11659	0.93	0.90	11636	-0.20	0.93	0.01	0.90	0.01	12229	4.89	0.92	-0.97	0.89	-1.12	12235	4.94	0.92	-0.65	0.88	-1.69
EXAM4	16325	0.73	0.78	16325	0.00	0.73	0.00	0.78	0.00	17130	4.93	0.71	-3.57	0.68	-12.59	17120	4.87	0.71	-3.56	0.68	-12.77
EXAM5	3837	0.86	0.46	3837	0.00	0.86	-0.01	0.46	-0.01	4027	4.95	0.84	-2.20	0.45	-1.41	3974	3.57	0.85	-1.47	0.32	-29.24
EXAM6	27370	0.79	0.61	27370	0.00	0.79	0.00	0.61	0.00	28735	4.99	0.78	-0.71	0.60	-0.34	28210	3.07	0.79	0.11	0.59	-2.23
EXAM7	5528	0.84	0.62	5518	-0.18	0.84	-0.09	0.62	-0.05	5800	4.92	0.82	-1.69	0.61	-1.36	5803	4.97	0.83	-0.90	0.60	-3.93
EXAM8	9798	0.65	0.44	9784	-0.14	0.65	0.02	0.43	-0.95	10287	4.99	0.62	-4.94	0.39	-10.23	10241	4.52	0.63	-2.67	0.38	-13.91
EXAM9	1246	0.75	0.68	1246	0.00	0.75	0.00	0.68	0.00	1308	4.98	0.71	-5.42	0.63	-7.09	1308	4.98	0.71	-5.37	0.63	-8.23
EXAM10	14556	0.64	0.50	14556	0.00	0.64	0.00	0.50	0.00	15141	4.02	0.61	-5.45	0.46	-8.24	15141	4.02	0.61	-5.45	0.46	-8.24
EXAM11	36810	0.91	0.86	36777	-0.09	0.91	0.02	0.86	0.02	38601	4.87	0.90	-1.03	0.85	-1.19	38562	4.76	0.90	-0.93	0.85	-1.36
EXAM12	5300	0.88	0.75	5300	0.00	0.88	0.00	0.75	0.00	5562	4.94	0.88	-0.89	0.53	-29.92	5563	4.96	0.88	-0.43	0.50	-33.15
YUE20011	56	0.92	0.90	54	-3.57	0.92	-0.18	0.89	-0.50	58	3.57	0.91	-0.95	0.88	-2.06	58	3.57	0.91	-0.95	0.88	-2.06
YUE20012	122	0.86	0.85	122	0.00	0.86	0.00	0.85	0.00	128	4.92	0.84	-2.08	0.83	-2.13	128	4.92	0.84	-2.08	0.83	-2.13
YUE20013	29	0.88	0.90	29	0.00	0.88	0.00	0.89	-0.13	30	3.45	0.88	-0.48	0.89	-0.95	30	3.45	0.88	-0.48	0.89	-0.95
YUE20021	76	0.92	0.73	76	0.00	0.92	-0.19	0.73	-0.19	79	3.95	0.92	-0.58	0.70	-4.11	79	3.95	0.92	-0.58	0.70	-4.11
YUE20022	162	0.87	0.73	162	0.00	0.87	-0.38	0.73	-0.47	170	4.94	0.86	-1.62	0.71	-3.50	170	4.94	0.86	-0.96	0.70	-4.05
YUE20023	56	0.89	0.59	56	0.00	0.89	0.00	0.59	0.00	58	3.57	0.89	-0.52	0.59	-1.01	58	3.57	0.89	-0.52	0.59	-1.01
YUE20031	143	0.90	0.92	143	0.00	0.90	0.00	0.92	0.00	150	4.90	0.89	-1.13	0.91	-0.78	150	4.90	0.90	-0.65	0.90	-1.96
YUE20032	434	0.77	0.64	434	0.00	0.77	0.00	0.64	0.00	455	4.84	0.75	-2.75	0.61	-3.69	455	4.84	0.75	-2.10	0.61	-4.42

Table D.9: The objective function values range of the Pareto solution set from the experiments of a hyper-heuristic given in Algorithm 11 using 'Weighted Summation' as the scalarisation technique under the parameter settings given in Table 6.18 tested over multi-objective examination timetabling problem with three-objective functions: standard objective function, global unfairness objective function and 'exam-count-based cohort unfairness' objective function

P. Instance	Init Solution			STD.OF			Global Fairness			ECB Cohort Unfairness		
	STD.OF	AJFI	AJFIC1	min	max	range	min	max	range	min	max	range
NOTT-I	11923	0.79	0.54	11923	12517	594	0.77	0.79	0.03	0.50	0.54	0.04
NOTT-II-32	0	0.00	0.00	0	0	0	0.00	0.00	0.00	0.00	0.00	0.00
NOTT-II-31	0	0.00	0.00	0	0	0	0.00	0.00	0.00	0.00	0.00	0.00
NOTT-II-30	0	0.00	0.00	0	0	0	0.00	0.00	0.00	0.00	0.00	0.00
NOTT-II-26	8	1.00	0.22	8	8	0	1.00	1.00	0.00	0.22	0.22	0.00
NOTT-II-23	68	0.99	0.65	68	71	3	0.99	0.99	0.00	0.64	0.65	0.00
CAR-S-91	5.30	0.68	0.37	5.30	5.57	0.27	0.66	0.68	0.02	0.32	0.37	0.05
CAR-F-92	4.51	0.71	0.43	4.51	4.73	0.22	0.69	0.71	0.02	0.38	0.43	0.05
EAR-F-83	36.73	0.17	0.10	36.70	38.55	1.86	0.15	0.18	0.03	0.07	0.11	0.05
HEC-S-92	10.91	0.51	0.29	10.91	11.45	0.55	0.47	0.51	0.04	0.26	0.29	0.03
KFU-S-93	14.36	0.45	0.33	14.36	15.07	0.71	0.41	0.45	0.04	0.28	0.33	0.05
LSE-F-91	11.02	0.49	0.26	11.02	11.57	0.55	0.45	0.49	0.04	0.23	0.26	0.03
PUR-S-93	5.03	0.67	0.42	5.02	5.27	0.25	0.65	0.67	0.01	0.41	0.42	0.01
RYE-S-92	9.01	0.63	0.30	9.01	9.46	0.45	0.60	0.63	0.03	0.27	0.30	0.03
STA-F-83	157.12	0.10	0.00	157.12	164.96	7.84	0.07	0.10	0.03	0.00	0.00	0.00
TRE-S-92	8.75	0.56	0.37	8.74	9.18	0.44	0.53	0.56	0.03	0.34	0.37	0.04
UTA-S-92	3.60	0.77	0.44	3.60	3.78	0.18	0.76	0.77	0.01	0.41	0.44	0.03
UTE-S-92	25.20	0.21	0.21	25.20	26.44	1.24	0.17	0.21	0.04	0.16	0.21	0.05
YOR-F-83	38.03	0.25	0.23	38.03	39.83	1.80	0.22	0.25	0.03	0.20	0.23	0.04
EXAM1	6856	0.82	0.47	6825	7196	371	0.79	0.82	0.02	0.43	0.47	0.04
EXAM2	632	1.00	0.49	627	663	36	1.00	1.00	0.00	0.47	0.59	0.12
EXAM3	11659	0.93	0.58	11639	12234	595	0.92	0.93	0.01	0.55	0.58	0.02
EXAM4	16325	0.73	0.50	16325	17137	812	0.71	0.73	0.03	0.44	0.51	0.06
EXAM5	3837	0.86	0.61	3836	4026	190	0.84	0.86	0.02	0.51	0.61	0.10
EXAM6	27370	0.79	0.29	27370	28735	1365	0.78	0.79	0.01	0.28	0.30	0.01
EXAM7	5528	0.84	0.36	5526	5804	278	0.82	0.84	0.01	0.34	0.36	0.02
EXAM8	9798	0.65	0.29	9792	10285	493	0.61	0.65	0.04	0.26	0.29	0.03
EXAM9	1246	0.75	0.24	1246	1308	62	0.70	0.75	0.04	0.19	0.24	0.05
EXAM10	14556	0.64	0.08	14556	15283	727	0.61	0.64	0.03	0.03	0.08	0.05
EXAM11	36810	0.91	0.55	36750	38639	1889	0.90	0.91	0.01	0.53	0.55	0.02
EXAM12	5300	0.88	0.57	5300	5564	264	0.88	0.88	0.01	0.51	0.57	0.06
YUE20011	56	0.92	0.45	56	58	2	0.91	0.92	0.01	0.43	0.45	0.02
YUE20012	122	0.86	0.57	122	128	6	0.84	0.86	0.02	0.54	0.57	0.02
YUE20013	29	0.88	0.37	29	30	1	0.88	0.88	0.00	0.35	0.37	0.01
YUE20021	76	0.92	0.56	72	79	7	0.91	0.92	0.01	0.51	0.54	0.03
YUE20022	162	0.87	0.54	162	170	8	0.86	0.87	0.01	0.49	0.54	0.06
YUE20023	56	0.89	0.47	56	58	2	0.89	0.89	0.00	0.46	0.47	0.01
YUE20031	143	0.90	0.60	143	150	7	0.89	0.90	0.01	0.58	0.60	0.02
YUE20032	434	0.77	0.55	434	455	21	0.75	0.77	0.02	0.38	0.55	0.17

Table D.10: The objective function values of the solution with the best standard objective function, the solution with the best global fairness, and the solution with exam-count-based cohort (ECB) cohort fairness in the Pareto solution set from the experiments of a hyper-heuristic given in Algorithm 11 using 'Weighted Summation' as the scalarisation technique under the parameter settings given in Table 6.18 tested over multi-objective examination timetabling problem with three-objective functions: standard objective function, global unfairness objective function and 'exam-count-based cohort unfairness' objective function. The d(%) values are the percentage of the respective objective function values' change from the initial solutions values.

P. Instance	Init Solution			Sol. With Best STD.OF						Sol. With Best Global Unfairness						Sol. With Best Specified Cohort Unfairness					
				STD.OF		AJFI		AJFIC2		STD.OF		AJFI		AJFIC2		STD.OF		AJFI		AJFIC2	
	STD.OF	AJFI	AJFIC2	Val.	d (%)	Val.	d(%)	Val.	d(%)	Val.	d (%)	Val.	d(%)	Val.	d(%)	Val.	d (%)	Val.	d(%)	Val.	d(%)
NOTT-I	11923	0.79	0.54	11923	0.00	0.79	0.00	0.54	0.00	12506	4.89	0.77	-3.53	0.52	-4.07	12494	4.79	0.77	-2.68	0.50	-7.21
NOTT-II-32	0	0.00	0.00	0	0.00	0.00	0.00	0.00	0.00	0	0.00	0.00	0.00	0.00	0.00	0	0.00	0.00	0.00	0.00	0.00
NOTT-II-31	0	0.00	0.00	0	0.00	0.00	0.00	0.00	0.00	0	0.00	0.00	0.00	0.00	0.00	0	0.00	0.00	0.00	0.00	0.00
NOTT-II-30	0	0.00	0.00	0	0.00	0.00	0.00	0.00	0.00	0	0.00	0.00	0.00	0.00	0.00	0	0.00	0.00	0.00	0.00	0.00
NOTT-II-26	8	1.00	0.22	8	0.00	1.00	0.00	0.22	0.00	8	0.00	1.00	0.00	0.22	0.00	8	0.00	1.00	0.00	0.22	0.00
NOTT-II-23	68	0.99	0.65	68	0.00	0.99	0.00	0.65	0.00	71	4.41	0.99	-0.06	0.64	-0.68	71	4.41	0.99	-0.04	0.64	-0.76
CAR-S-91	5.30	0.68	0.37	5.30	0.00	0.68	0.00	0.37	0.00	5.57	4.97	0.66	-2.54	0.34	-8.45	5.55	4.65	0.67	-1.28	0.32	-13.67
CAR-F-92	4.51	0.71	0.43	4.51	0.00	0.71	0.00	0.43	0.00	4.73	4.84	0.69	-2.22	0.40	-6.59	4.73	4.98	0.70	-1.00	0.38	-12.16
EAR-F-83	36.73	0.17	0.10	36.70	-0.08	0.17	-0.02	0.10	-0.31	38.38	4.49	0.15	-14.25	0.08	-16.49	38.55	4.97	0.16	-5.68	0.07	-29.97
HEC-S-92	10.91	0.51	0.29	10.91	0.00	0.51	0.00	0.29	0.00	11.45	5.00	0.47	-7.09	0.26	-9.65	11.39	4.43	0.48	-6.26	0.26	-11.23
KFU-S-93	14.36	0.45	0.33	14.36	0.00	0.45	0.00	0.33	0.00	15.07	4.96	0.41	-8.86	0.31	-8.42	15.02	4.58	0.42	-6.86	0.28	-15.62
LSE-F-91	11.02	0.49	0.26	11.02	0.00	0.49	-0.05	0.26	0.03	11.55	4.85	0.45	-7.75	0.25	-4.33	11.56	4.94	0.46	-5.12	0.23	-10.41
PUR-S-93	5.03	0.67	0.42	5.02	-0.03	0.67	-0.02	0.42	-0.04	5.25	4.52	0.65	-1.89	0.41	-1.77	5.27	4.96	0.66	-0.86	0.41	-3.42
RYE-S-92	9.01	0.63	0.30	9.01	-0.01	0.63	0.02	0.30	0.05	9.46	5.00	0.60	-4.71	0.27	-10.57	9.45	4.96	0.60	-4.61	0.27	-10.60
STA-F-83	157.12	0.10	0.00	157.12	0.00	0.10	0.04	0.00	-7.54	164.96	4.99	0.07	-31.74	0.00	-23.69	160.89	2.40	0.09	-12.84	0.00	-70.42
TRE-S-92	8.75	0.56	0.37	8.74	0.00	0.56	0.04	0.37	-0.34	9.18	4.98	0.53	-5.71	0.34	-8.98	9.18	4.94	0.54	-5.02	0.34	-9.83
UTA-S-92	3.60	0.77	0.44	3.60	-0.03	0.77	-0.20	0.44	-0.01	3.77	4.85	0.76	-1.65	0.43	-3.78	3.77	4.83	0.76	-1.12	0.41	-7.51
UTE-S-92	25.20	0.21	0.21	25.20	-0.01	0.21	-0.57	0.20	-2.35	26.40	4.76	0.17	-17.12	0.17	-18.33	26.18	3.90	0.18	-13.67	0.16	-24.54
YOR-F-83	38.03	0.25	0.23	38.03	0.00	0.25	0.00	0.23	0.00	39.43	3.69	0.22	-10.37	0.21	-9.06	39.42	3.65	0.24	-2.41	0.20	-15.18
EXAM1	6856	0.82	0.47	6825	-0.45	0.82	0.00	0.47	-1.76	7196	4.96	0.79	-3.01	0.46	-4.11	7178	4.70	0.79	-2.74	0.43	-9.44
EXAM2	632	1.00	0.49	627	-0.79	1.00	0.00	0.49	0.00	663	4.91	1.00	-0.08	0.59	20.25	648	2.53	1.00	-0.03	0.47	-4.97
EXAM3	11659	0.93	0.58	11639	-0.17	0.93	0.01	0.58	0.01	12213	4.75	0.92	-0.89	0.57	-1.15	12152	4.23	0.93	-0.35	0.55	-4.05
EXAM4	16325	0.73	0.50	16325	0.00	0.73	0.00	0.50	0.00	17128	4.92	0.71	-3.63	0.48	-3.64	17112	4.82	0.72	-2.30	0.44	-11.67
EXAM5	3837	0.86	0.61	3836	-0.03	0.86	0.01	0.61	0.01	4026	4.93	0.84	-2.26	0.60	-2.21	4024	4.87	0.84	-1.68	0.51	-15.84
EXAM6	27370	0.79	0.29	27370	0.00	0.79	0.00	0.29	0.00	28730	4.97	0.78	-0.69	0.30	3.44	27820	1.64	0.79	0.39	0.28	-0.99
EXAM7	5528	0.84	0.36	5526	-0.04	0.84	-0.02	0.36	-0.04	5798	4.88	0.82	-1.72	0.35	-2.58	5779	4.54	0.83	-0.85	0.34	-5.08
EXAM8	9798	0.65	0.29	9792	-0.06	0.65	0.04	0.29	0.05	10284	4.96	0.61	-5.41	0.27	-8.09	10259	4.71	0.62	-4.67	0.26	-10.32
EXAM9	1246	0.75	0.24	1246	0.00	0.75	0.00	0.24	0.00	1307	4.90	0.70	-5.55	0.19	-19.38	1308	4.98	0.71	-5.45	0.19	-19.48
EXAM10	14556	0.64	0.08	14556	0.00	0.64	0.00	0.08	0.00	15243	4.72	0.61	-5.17	0.04	-47.71	15240	4.70	0.62	-4.32	0.03	-57.95
EXAM11	36810	0.91	0.55	36750	-0.16	0.91	0.02	0.55	0.31	38639	4.97	0.90	-1.03	0.54	-1.49	38052	3.37	0.90	-0.48	0.53	-3.12
EXAM12	5300	0.88	0.57	5300	0.00	0.88	0.00	0.57	0.00	5547	4.66	0.88	-0.97	0.55	-2.38	5508	3.92	0.88	-0.60	0.51	-9.84
YUE20011	56	0.92	0.45	56	0.00	0.92	0.00	0.45	-0.23	58	3.57	0.91	-0.65	0.44	-3.13	58	3.57	0.91	-0.38	0.43	-5.13
YUE20012	122	0.86	0.57	122	0.00	0.86	0.00	0.57	-0.08	128	4.92	0.84	-2.08	0.54	-3.77	128	4.92	0.84	-2.08	0.54	-3.77
YUE20013	29	0.88	0.37	29	0.00	0.88	0.00	0.37	0.00	30	3.45	0.88	-0.48	0.35	-2.98	30	3.45	0.88	-0.48	0.35	-2.98
YUE20021	76	0.92	0.56	72	-5.26	0.92	-0.08	0.54	-2.75	79	3.95	0.91	-0.99	0.51	-7.39	79	3.95	0.91	-0.99	0.51	-7.39
YUE20022	162	0.87	0.54	162	0.00	0.87	0.00	0.54	0.00	170	4.94	0.86	-1.62	0.52	-3.99	170	4.94	0.87	-0.58	0.49	-10.43
YUE20023	56	0.89	0.47	56	0.00	0.89	0.00	0.47	0.00	58	3.57	0.89	-0.52	0.46	-2.46	58	3.57	0.89	-0.52	0.46	-2.46
YUE20031	143	0.90	0.60	143	0.00	0.90	0.00	0.60	0.00	150	4.90	0.89	-1.25	0.59	-2.31	150	4.90	0.89	-1.00	0.58	-2.80
YUE20032	434	0.77	0.55	434	0.00	0.77	0.00	0.55	0.00	455	4.84	0.75	-2.84	0.49	-11.31	455	4.84	0.76	-1.56	0.38	-31.44