



Faculty of Computer Science and Information Technology

Rule Prioritisation Heuristic in Shift Design of Airline Ground Crew

Lee Kong Weng

**Doctor of Philosophy
2024**

Rule Prioritisation Heuristic in Shift Design of Airline Ground Crew

Lee Kong Weng

A thesis submitted

In fulfillment of the requirements for the degree of Doctor of Philosophy

(Computational Science)

Faculty of Computer Science and Information Technology

UNIVERSITI MALAYSIA SARAWAK

2024

DECLARATION

I declare that the work in this thesis was carried out in accordance with the regulations of Universiti Malaysia Sarawak. Except where due acknowledgements have been made, the work is that of the author alone. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.



.....

Signature

Name: Lee Kong Weng

Matric No.: 13010084

Faculty of Computer Science and Information Technology

Universiti Malaysia Sarawak

Date: 10 March 2024

ACKNOWLEDGEMENT

I want to express my sincere gratitude to Associate Professor Dr Sze San Nah and Associate Professor Dr Chiew Kang Leng, my research supervisors, for their patient guidance, enthusiastic encouragement, and useful critiques of this research work. My sincere gratitude to the Centre for Graduate Studies and the university for the support given during my period of study in Universiti Malaysia Sarawak.

I want to thank my parents and my family for support. I wish to thank my loving and supportive wife, Jessica, and our cats and Benji the dog, who provide unending stress-free therapy throughout the ordeal.

Finally, I would like to thank all that directly or indirectly has assisted me and making it possible for me to complete my study here in UNIMAS. Thank you all.

ABSTRACT

In various industries, the exigency of workforce shift scheduling and rostering demands immediate, innovative solutions. Many off-the-shelf scheduling products lack the flexibility to adapt to varied organizational requirements, resulting in widespread reliance on manual interventions. This research, rooted in a case study of a local airline at Kuala Lumpur International Airport (KLIA), a major regional hub, endeavours to formulate an algorithm addressing workforce shift scheduling and rostering complexities. The proposed algorithm aims to ensure judicious allocation of operational tasks, emphasizing minimal workforce utilization, reduced idleness, and heightened adaptability in employee schedules. Utilizing an existing workforce prediction method and shift design criteria, the study determines the minimum workforce required for task completion. Shifts are assigned based on this minimum, employing heuristic and genetic algorithms. Subsequent optimization of potential schedules culminates in the identification of the most efficacious ones. Empirical tests reveal the proposed heuristic method's superiority over manual practices and genetic algorithms. It not only minimizes shift idle time and aligns with optimal shift starting times but also significantly reduces workforce requirements, streamlining the shift scheduling process, irrespective of data size. This research contributes valuable insights to the advancement of workforce management strategies, offering practical and innovative solutions within the specified context.

Keywords: Heuristic, genetic algorithm, minimum workforce, shift design, shift scheduling

Heuristik Penentuan Keutamaan Aturan dalam Rekabentuk Shif untuk Krew Syarikat Penerbangan

ABSTRAK

Dalam pelbagai industri, kemendasakan mengenai penjadualan shif tenaga kerja dan penjadualan tugas menuntut penyelesaian segera dan inovatif. Kebanyakan produk penjadualan siap-pakai kurang fleksibel untuk menyesuaikan keperluan organisasi yang berbeza, menyebabkan bergantung kepada penyelesaian manual secara meluas. Kajian ini, berakar dalam kajian kes sebuah syarikat penerbangan di Lapangan Terbang Antarabangsa Kuala Lumpur (KLIA), sebuah pusat serantau utama, bertujuan merumuskan algoritma menangani kekompleksan jadual pergeseran tenaga kerja dan penjadualan tugas. Algoritma yang dicadangkan bertujuan memastikan peruntukan tugas operasi yang bijak, menitikberatkan penggunaan tenaga kerja minimum, mengurangkan pembaziran masa, dan meningkatkan kebolehtelapan jadual pekerja. Dengan menggunakan kaedah ramalan tenaga kerja sedia ada dan kriteria reka bentuk shif, kajian menentukan jumlah minimum pekerjaan yang diperlukan untuk penyelesaian tugas. Giliran diberikan berdasarkan jumlah minimum ini, dengan menggunakan kaedah heuristik dan algoritma genetik. Penyesuaian kemudian dilakukan untuk mengoptimumkan jadual berpotensi hingga pemilihan yang paling berkesan. Ujian empirik menunjukkan keunggulan kaedah heuristik dicadangkan berbanding amalan manual dan algoritma genetik. Ia tidak hanya mengurangkan masa menganggur giliran dan sejajar dengan waktu permulaan giliran yang optimum, malah mengurangkan secara ketara keperluan tenaga kerja, menyelaraskan proses penjadualan giliran, tanpa mengira saiz data. Kajian ini memberikan pandangan berharga untuk kemajuan strategi pengurusan tenaga kerja, menawarkan penyelesaian praktikal dan inovatif dalam konteks yang ditentukan.

Kata kunci: *Heuristik, algoritma genetik, tenaga kerja minimum, reka bentuk shif, penjadualan shif*

TABLE OF CONTENTS

	Page
DECLARATION	i
ACKNOWLEDGEMENT	ii
ABSTRACT	iii
<i>ABSTRAK</i>	iv
TABLE OF CONTENTS	vi
LIST OF TABLES	xi
LIST OF FIGURES	xii
LIST OF ABBREVIATIONS	xiv
CHAPTER 1 INTRODUCTION	1
1.1 Introduction	1
1.2 Problem Statement	3
1.3 Objectives	5
1.4 Thesis Structure	6
CHAPTER 2 LITERATURE REVIEW	8
2.1 Introduction	8
2.2 Pioneering Studies	9
2.3 Heuristic	11
2.4 Genetic Algorithm (GA)	15

2.5	Shift Design for Airline Ground Crew	19
2.6	Conclusion	21
CHAPTER 3	RESEARCH METHODOLOGY	26
3.1	Introduction	26
3.2	Research Framework	31
3.3	Phase 1: Planning and Data Collection	32
3.4	Phase 2: Modelling and Prototyping	34
3.4.1	General Model	34
3.4.2	Proposed Model	35
3.4.3	Rostering and Scheduling Formulation	36
3.4.3.1	Sets	37
3.4.3.2	Parameters	38
3.4.3.3	Decision Variables	40
3.4.3.4	Constraints	41
3.4.4	Rule Prioritisation Heuristic Algorithm	43
3.4.4.1	Randomise Pick	41
3.4.4.2	Criteria-Based Selection	43
3.4.4.3	Optimisation	46
3.4.5	The Genetic Algorithm	51
3.4.5.1	Fundamentals	52
3.4.5.2	Fitness Function	54

3.4.5.3 Initial Pool Generation	55
3.4.5.4 Selection	54
3.4.5.5 Crossover	55
3.4.5.6 Mutation	60
3.4.5.7 Evaluation	61
3.4.5.8 Optimisation	62
3.5 Analysis Methodology	62
3.5.1 Sources of Data	64
3.5.2 Performance Analysis	63
3.5.3 Sensitivity Analysis	64
3.6 Conclusion	65
CHAPTER 4 RULE PRIORITISATION HEURISTIC APPROACH	66
4.1 Introduction	66
4.1.1 Criteria Preparation	69
4.1.2 Shift Scheduling	70
4.2 Computational Results	71
4.3 Results Analysis	72
4.3.1 Performance Analysis	72
4.3.2 Sensitivity Analysis	76
4.4 Discussion	80

4.5	Conclusion	82
CHAPTER 5	GENETIC ALGORITHM APPROACH	82
5.1	Introduction	82
5.2	Computational Results	83
5.3	Results Analysis	87
5.3.1	Performance Analysis	87
5.3.1.1	Analysis of Shift Pool Size and GA Iteration	87
5.3.1.2	Analysis of GA vs Manual Assignment Shift (MAS) Schedule	91
5.3.2	Sensitivity Analysis	92
5.3.2.1	Analysis of GA Initialisation Pool Size Distribution and Shift Pool Size	93
5.3.2.2	Analysis of GA Iteration Distribution	95
5.3.2.3	Analysis of GA Operators Distribution	99
5.4	Conclusion	101
CHAPTER 6	ANALYSIS	103
6.1	Introduction	103
6.2	Computational Analysis	106
6.3	Performance Analysis	104
6.4	Sensitivity Analysis	107
6.4.1	Workforce	107
6.4.2	Overlapped	108
6.4.3	Gap Hour	109

6.5	Conclusion	110
CHAPTER 7	CONCLUSION AND RECOMMENDATIONS	112
7.1	Conclusion	112
7.2	Contributions	114
7.3	Future recommendations	117
REFERENCES		120
APPENDICES		131

LIST OF TABLES

	Page
Table 2.1 Research Gap Analysis	23
Table 3.1 Manually Assigned Shifts	33
Table 3.2 Jobs Demand Data	33
Table 3.3 Typical Shift Type	37
Table 3.4 Suggested Shift-Type Configuration	37
Table 3.5 Shift Schedule Parameters	51
Table 3.6 Selection Method	55
Table 4.1 Shift Start Time Range Used for Testing	69
Table 4.2 Randomise Pick Shift Schedule	70
Table 4.3 Criteria Based Selection	70
Table 4.4 Computational Time with Different Datasets Size	73
Table 4.5 Overlapped Shift Durations in Hours for RP and CBS	80
Table 4.6 Gap Hour Duration Values for RP and CBS	80
Table 5.1 GA shift schedule	84
Table 5.2 GA Shift Schedule Costs	85
Table 6.1 Computational Time with Different Datasets Size	102

LIST OF FIGURES

	Page
Figure 3.1 Work Force Demand Curve Chart	29
Figure 3.2 Research Framework	31
Figure 3.3 Model Design and Development	36
Figure 3.4 Proposed Design of RPH Algorithm	40
Figure 3.5 Shift Scheduling Process	41
Figure 3.6 Randomise Pick Function in Picking Schedule	42
Figure 3.7 Criteria-Based Selection Function in Selecting Schedule	45
Figure 3.8 Illustration of the Optimisation	47
Figure 3.9 Shifts Candidates	52
Figure 3.10 Sorting of the Candidate Based on Fitness Values	53
Figure 3.11 Crossover Operation Between Days	55
Figure 3.12 Crossover Operation Between Shifts	56
Figure 3.13 Swapping Operation	57
Figure 3.14 Swapping Operation Between Days	58
Figure 3.15 Sorting Operations after GA Operations	58
Figure 3.16 Real Dataset of 4 days	61
Figure 3.17 Simulated Data with 1 Peak Hour	62
Figure 3.18 Simulated Data with 2 Peak Hour	62
Figure 3.19 Simulated Data with 3 Peak Hour	63
Figure 4.1 Defining A Day Shift Time	68
Figure 4.2 Comparison Between Manual, Lower Bound and Proposed Heuristic Algorithms Workforce Requirements	71
Figure 4.3 Comparison of the Workforce Assigned for Shifts Between Proposed Algorithm Vs. Manual Assignment	73

Figure 4.4	The Relation Between Resource Values Vs. A Different Range of Shift Start Time and Duration	75
Figure 4.5	The Relation Between Uncovered Jobs and Overlapped of Shifts Vs. Different Range of Shift Start Time and Duration	77
Figure 4.6	The Relation Between Workforce Vs. Different Range of Shift Start Time and Duration	78
Figure 4.7	Comparison of Workforce Utilisation	79
Figure 5.1	Prototype User Interface for Parameters Input	83
Figure 5.2	Workforce Requirements	86
Figure 5.3	Tests on Various Shift Duration and Shift Start Time Range	88
Figure 5.4	Tests on Various Iteration Pool Size	90
Figure 5.5	Tests on Various Numbers of Iteration	91
Figure 5.6	Workforce Utilization Comparison	92
Figure 5.7	Tests Running on Various Initial Pool Number, Shift Duration and Shift Start Time Range	94
Figure 5.8	Tests on Various Numbers of Iteration Pool Population	95
Figure 5.9	Tests on Various Numbers of Iteration	96
Figure 5.10	Tests of Various Selection Methods Impact on the Workforce	97
Figure 5.11	Tests of Various Selection Methods Impact on the Overlap	98
Figure 5.12	Tests of Various Selection Methods Impact on the Gap Hour	99
Figure 6.1	Workforce Requirement by Assignment Methods	103
Figure 6.2	Heuristic vs. Manual Workforce Distribution	105
Figure 6.3	GA vs. Manual Workforce Distribution	106
Figure 6.4	Heuristic vs. GA Workforce Distribution	106
Figure 6.5	Different Shift Duration Test	108
Figure 6.6	Different Shift Duration Test	109
Figure 6.7	Different Shift Duration Test	110

LIST OF ABBREVIATIONS

CBS	Criteria based selection
CGS	Centre for Graduate Studies
FP	Flat peak
RP	Randomise pick
RPH	Rule prioritisation heuristic
RPH-CBS	Rule prioritisation heuristic – criteria-based selection
RPH-RP	Rule prioritisation heuristic – randomise pick
SP	Significant peak
TSH	Two-stage heuristic
UNIMAS	Universiti Malaysia Sarawak

CHAPTER 1

INTRODUCTION

1.1 Introduction

In various industries, workforce rostering and scheduling poses problems that require immediate solutions. The demand for fast and reliable operation in any organisation and businesses has never been higher, especially in manufacturing, logistics, transportation, and various other industries. With the rising cost face by operations and managing, all businesses are struggling to keep their expenditures low. The most significant potential area for cost saving is workforce administration through which most organisations pursue approaches that would lessen wastage and optimise workforce utilisation. This solution is hardly new; American Airlines adopted an optimisation software solution developed by (Anbil et al., 1991) for crew rostering which resulted in annual savings of millions. (Afifi & Aly, 2018) and (Odoni & Barnhart, 2003) discuss optimization models and techniques for crew scheduling and schedule planning, respectively, which can lead to increased efficiency and cost savings for airlines. While (Mallick & Mishra, 2018) focus on revenue management practices, their insights can indirectly impact scheduling by helping airlines optimize pricing and revenue streams. In most organisations, having the ideal number of workforces is critical, as it reduces not only cost but also enhances effectiveness and decreases bureaucracy. Therefore, the enormous spending by organisations to secure a viable system for scheduling and rostering is justified and even indispensable, as it provides a forecast of the optimal number of the workforce needed by the organisations.

In solving the rostering and scheduling, there have been various solutions proposed by researchers all over the world. The widely used approach is the heuristic-based solution,

where computational performance gain has higher priority than accuracy or precision. The heuristic method will generate the most probable solutions and pick the best solution instead of generating all the solutions. This approach can find an approximate solution in a reasonable timeframe or when in a situation where finding an optimal solution is not possible. There is also a possibility that an optimal solution would not be found. Researchers all over the world have discovered various heuristic approaches for rostering and scheduling solutions.

Another widely used approach in solving rostering and scheduling is by using an integer programming method which is based on mathematical optimisation. An integer programming method is a mathematical optimisation or feasibility program where it is restricted in which some or all of the variables are integers. Integer programming is a powerful solution technique for obtaining an optimum or near-optimal solution. Regarding solution quality, integer programming has proven to produce a better solution when compared to the heuristic approach but is inferior with regards to computational speed. The literature review will elaborate and discuss the details on the approaches taken over the years and also current development to solve rostering and scheduling.

Rostering and scheduling algorithm aims to address the challenge of ensuring that it assigns all jobs using minimum resources while preventing wastage. Its secondary goal is to avoid unutilized resource such as idling workforce during the working hours. Aside from that, the algorithm will schedule a shift with a consistent start time pattern throughout the week. This consistency in the start time pattern will ensure workers' adaptability to the schedule and exercise fairness. The fundamental need to solve this problem is to ensure that it assigns all shifts to cover all or as many jobs as possible. It should additionally plan the

shifts with minimum workforce utilisation, least workforce idleness and enhanced adaptability of employees' time.

Our approach is to utilise the existing workforce prediction method to decide the minimum workforce required to complete all jobs. Based on the minimum workforce number and shift criteria, the shifts were assigned to form schedules using a random pick and criteria-based selection methods. The potential schedules were then optimised, and the best ones were selected. Based on several realistic test instances, the proposed heuristic approach appears to offer promising solutions for rostering and scheduling as it improves shift idle time, complies with better shift start time and significantly reduces the workforce needed and the time spent on creating schedules, regardless of the data.

1.2 Problem Statement

The scheduling or rostering problem is not new, and there will be a continual change to the requirements because of the dynamic and competitive nature of the airline industry. Schedule or roster is an organised structure of keeping track of staff working timetable to complete the tasks or services for a period defined. Shifts are created in order to split the day's work in schedule or roster so that the workforce or personnel can be assigned accordingly. The main problem involves determining the best possible schedule with shifts for at least 100 jobs per day within 12 hours or fewer. The schedule and shifts generated should cover at least one-week period or more and should be flexible in accepting parameters changes.

A set of rules which dictate the structure of the schedule or roster typically governs a schedule or roster. In this research, these rules are known as parameters from the programming perspective or operations constraints in the airline industry. The lists of

parameters or operational constraints that will be considered are discussed in the following section.

i. Shift start time

Shift start time for different tasks is generally interdependent since staff cannot cover more than one task at a time. This is true, especially at peak times, when tasks will overlap each other. In shift assignment, the shift schedule or roster will need to have minimum rest times between shifts and maximum offsets between start times on consecutive days. Each shift-type defines a valid starting and ending time for a shift duty and prescribes one or several meals or relief breaks. Usually, break time rules are based on state and federal laws, union agreements, company policies, or practical considerations. Shift start time is a parameter definable based on the requirement of the operations. Shift start time comprises two attributes that form the shift start time range, which allow flexibility of shifts to cover jobs as much as possible; the attributes are:

- Earliest start time (EST) refers to the earliest time the shift can start.
- Latest start time (LST) refers to the latest time the shift can start.

ii. Shift duration

Personnel then covered the base workload with typical shift lengths of between six and eleven hours and up to three meals and relief breaks. Planners often try to use full-time shifts, which cover at least two peak periods within the day. Similar to shift start time, the shift duration is definable parameters and comprises two attributes that form the shift duration range. The attributes are:

- Min duration refers to the shortest period of duration for a shift.
- Max duration refers to the longest period of duration for a shift.

1.3 Objectives

The goal of this research is to develop a computationally bounded heuristic algorithm that can solve rostering and scheduling problems in the case study within a short time frame.

The research will satisfy the following objectives by achieving the goal:

- i. To design and develop a computational bounded heuristic algorithm that ensures efficiency within defined constraints on time, space, and practicality.
- ii. To formulate a simulated model of airline scheduling/rostering model by digitally replicating operations.
- iii. To design and develop a hybrid meta-heuristic rostering algorithm.

The research will also subject to the following limitations:

- a. Simulated models may involve simplifications.
- b. Involve trade-offs between algorithmic efficiency and solution optimality.
- c. Limited or inaccurate data may impact the realism.

Apart from the research objectives and limitation mentioned, it is also vital to evaluate the quality of current industries practices and the proposed algorithms. We take into consideration three types of cost savings under different operational constraints. The cost savings are given below:

- All jobs must be assigned/covered. All jobs must be assigned and completed by the personnel assigned to the job, no compromise on this constraint.
- To reduce the use of resource/workforce. In shift scheduling, to lower the operational cost, the shift schedule should utilise the most minimum resource.
- Avoid overlaps of shift period. Overlapping of shifts happened when two teams of workforce is on duty in the same period where one of the team will be assigned to jobs while the other will most likely be idling which is a waste of resource and cost.

In these studies, two algorithms are being proposed and developed. The heuristic approach will be the focus, as the genetic algorithm approach will be used as a control algorithm. An optimisation function would be developed and implemented on both algorithms. The optimisation function will try to tackle three types of cost savings under different operational constraints mentioned earlier.

1.4 Thesis Structure

An overview of current rostering and scheduling literature is given in Chapter 2. It gave a detailed review of various methods to solve the rostering and scheduling problem in the first subsection. The second subsection emphasis of the chapter looks at the previous implementation of the two heuristic approaches. Last, the chapter concludes with the pass studies on genetic algorithm approaches.

Chapter 3 looks at rostering and scheduling modelling; its constraints and objectives are explained. All operational constraints, decision variables and parameters are explained in detail with the goal is to minimise the number of resources and wastage.

Two rule-prioritisation heuristic algorithms are introduced in Chapter 4. The first heuristic-based algorithm is the randomise pick approach where all the possibilities of the shifts schedule are generated randomly and pre-processed and optimised before the proceeding with the process of the best schedule selection. An optimisation algorithm is developed and implemented so that the resource utilisation is more efficient throughout the schedule. The second heuristic-based algorithm is a criteria-based selection approach where the main goal is to pick schedules that fulfil specific required criteria, such as minimum workforce, minimum duration, or the earliest/latest start time of a shift. This chapter illustrates the computational effort of generating an optimal solution using the heuristic algorithm.

Chapter 5 presents the Genetic Algorithm method for rostering and scheduling. The algorithm is implemented with the various improvements, which include adaptive crossover, which tries to encourage a better shift schedule to be generated. Optimisation from Chapter 4 was integrated into the algorithm to improve the shift schedule. The chapter concludes with results from various parameter and strategy settings experimentation.

In Chapter 6, to validate the results obtained so far, all previous algorithms implementation applies to the same real-life data obtained from an airline in Malaysia. A numerical study of the heuristic and genetic algorithm solutions is performed with a range of data. The comparison of the different approaches and extensive sensitivity analysis are performed in order to show the effectiveness and robustness of the proposed algorithms. The numerical results and discussion are also reported at the end of the chapter.

Chapter 7 summarises the findings in this thesis and contains concluding remarks, contribution and ideas for future investigation are identified.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Research on scheduling and rostering is extensive and identified several problems involving real-life shift scheduling. The problems mostly involve assigning workers to shifts, determining working days and rest days, or constructing flexible shifts and their starting times for each shift. Furthermore, the problem includes a wide range of operational constraints. These constraints pose a challenge in addressing optimal scheduling because of integer's large size and pure nature; thus, these problems are typically labelled as NP-hard and NP-complete, as stated by both (Lau, 1996) and (Heimerl & Kolisch, 2010). Methods and models commonly used for solving these problems have been surveyed by (Ernst et al., 2004; Van den Bergh et al., 2013). In this paper by (Ernst et al., 2004), the authors categorise the solution methods for scheduling problems based on method and approaches such as artificial intelligence, fuzzy set theory, constraint programming, meta-heuristics, and mathematical model. Most literature reviews are intensely skewed towards mathematical programming and meta-heuristic for scheduling as opposed to constraint programming and other techniques.

Since the linear ordering problem is NP-hard, we cannot expect to solve practical problem instances of arbitrary size to optimality. Depending on the size of an instance or depending on the available CPU time, we will often have to be satisfied with approximate computing solutions. Also, under such circumstances, it might be impossible to assess the real quality of approximate solutions. In this chapter, we will attend to the question of how to find good enough solutions in a short or reasonable time by using heuristic algorithms.

The term heuristic stems from the Greek word *heuriskein*, which means to find or discover. It is used in the field of optimisation to characterise a certain kind of problem-solving methods. There are a high number and variety of challenging problems, which come up in practice and need to be solved efficiently, and this has promoted the development of efficient procedures in an attempt to find reasonable solutions, even if they are not optimal. These methods, in which the processing speed is as essential as the quality of the solution obtained, are called heuristics or approximative algorithms.

2.2 Pioneering Studies

One of the earlier shift scheduling surveys was done by (Baker, 1976) in attempt to classified scheduling problems. In Baker's investigation, he classified the approaches into two categories: shift scheduling and day-off scheduling. Baker also found a common weakness in both the theory and the practice of solving staffing problems lies in a failure to fully comprehend the broader context in which the staffing problem arises.

One of the earliest articles on shift scheduling was introduced in Laporte et al. (1980) on rotating schedule construction based on integer linear programming (ILP) algorithm in which different constraints such as length of work and off days is in itself a complex combinatorial problem. The disadvantage of ILP method is the solution generated may not be optimal or even workable as some constraint might be relaxed in order to get an acceptable solution. A simple heuristic approach is undertaken by Morris and Showalter (1983), the authors employed an expedient optimising procedure and a heuristic procedure to solve the tour scheduling procedure. The solutions generated consistently have low error bounds with a workforce of 30 people; the downside is the approach did not fully simulate attributes in real life and associate budget considerations.

Anbil et al. (1991) presented an optimisation software solution for crew rostering to American Airlines, which resulted in annual savings of more than USD20 million. American Airlines's crew pairing optimisation system is known as the trip re-evaluation and improvement program (TRIP). TRIP is based on an approach in which the pairings are iteratively improved by generating and solving a series of subproblems; the higher the number and size of subproblem trials, the better the solution quality. The success of TRIP is imminent, as the solution was sold to 10 major airlines and one railroad.

One of the earliest works on shift scheduling with simulated annealing is done by Brusco and Jacobs, (1993); their design is to minimise the number of workforces to satisfy forecast demand. Simulated annealing approach produces better quality solution when compared to branch-and-bound algorithms and exhibited rapid convergence to a low-cost solution.

Tabu search with strategic oscillation was employed by Katryn (1998) to tackled nurse scheduling in a large hospital. The tabu search approach repeatedly oscillates to find a feasible solution that makes sure that nurses are on duty all the time while taking accounts of individual preference of off days with fairness to all nurses. The result obtains from the tabu search approach seem to be comparable to those done by a human expert.

The most extensive survey of recent time was done by Van den Bergh et al. (2013) which have classified approaches from multiple perspectives. Besides details and through reviews, the author also makes recommendations for future works. Among the recommendations are to include multiple decisions making, avoid deterministic approach and to focus on the quality of the solutions by comparing with other approaches done by earlier researchers.

2.3 Heuristic

Musliu et al. (2002) introduced a new framework to solve rotating workforce schedule using the backtracking algorithm. The method can generate a high-quality result with human input in a reasonable time duration but not able to guarantee optimal results for large instances due to search space. Johannes and Slany (2004) presented a paper on the automatic generation of shifts with breaks. The problem of generating shifts with breaks appears in many areas of workforce scheduling, like in airlines, airports, call centres and others. Their solution utilised the greedy assignment of breaks in shifts and repair steps to find the best position of breaks inside of shifts. Hybrid heuristic was introduced by Musliu (2006) by combining tabu search with minimum conflict algorithm to tackle automatic rotating workforce scheduling. The hybrid heuristic approach is fast with acceptable results but sacrifice accuracy and quality of solutions when compared to the existing method tested.

Montoya and Mejía (2006) presented a general heuristic approach for solving workforce scheduling problems with the primary goal is to minimise the number of required workers given a pre-established shift demand over a planning horizon. The proposed algorithm starts with an initial solution and iteratively searches the state space, moving towards better solutions via a local search procedure which local optima. A hybrid approach was introduced by Di Gaspero et al. (2007) in their paper entitled "The minimum design problem" to find a minimum number of shifts and the number of employees to be assigned to these shifts to minimise the deviation from workforce requirements. The authors proposed a greedy heuristic hybrid with a local search using a tabu search method, which outperforms a commercial software in terms of quality and running time.

To solve the shift generation Di Gaspero et al. (2010) proposed automating the whole process of shift design and break assignment using a new hybrid Local Search-Constraint Programming method. Constraint programming was employed to solve the break assignment for the local search to avoid over/under staffing. In Clausen (2010), the author considers the heterogeneous shift design problem for a workforce with varied skills, where work shifts are created to cover a given demand as well as possible while minimising cost and satisfying a flexible set of constraints. Different labour regulations or organisational rules can apply to various ground handling operations, so the rules and restrictions can be numerous and vary significantly.

Dahmen and Rekik (2012) addresses a multi-activity shift scheduling problem in a continuous and flexible environment, including a heterogeneous workforce, using a hybrid heuristic that combines tabu search and a branch-and-bound procedure is proposed to solve the problem. Given the days-off schedules associated with each employee, the aim is to construct and assign admissible multi-activity shifts to employees on their workdays in a way that minimises under-staffing and over-staffing with a restricted budget on workforce cost. The workforce scheduling process is split into two fundamental procedures, namely the workload prediction and shift generation, to solve unlimited shift generation problem (Kynge et al., 2012). In the same year Sze et al. (2012) implemented the workload prediction algorithm on multi-trip vehicle routing and scheduling problems with time window with meal break considerations problem. In Rodič and Baggia (2013), the authors describe the development of a heuristic scheduling algorithm for workforce scheduling and shift construction at the Ljubljana Jože Pučnik airport which perform task schedule optimisation, personnel and equipment requirements optimisation and shift planning for airport ground crews.

The paper by Constantino et al. (2014) tackles a Nurse Scheduling Problem, which comprises generating work schedules for a set of nurses while considering their shift preferences and other requirements. This paper presents a new deterministic heuristic algorithm, called MAPA (multi-assignment problem-based algorithm), which is based on two phases: a constructive phase and an improvement phase. The constructive phase builds a full schedule by solving successive assignment problems, one for each day in the planning period, and the improvement phase uses a couple of procedures that re-solve assignment problems to produce a better schedule. Variable Neighborhood Search (VNS) is used to tackle workforce scheduling and routing problem (WSRP) (Pineiro et al., 2015). The authors implemented two greedy constructive heuristics to give VNS a good starting point, followed by multiple heuristic search capability, which would yield a substantial improvement to the final solution found.

The paper from Castillo-Salazar et al. (2015) present a greedy heuristic (GHI) designed to tackle five time-dependent activities constraints (synchronisation, overlap, minimum difference, maximum difference, and minimum-maximum difference) on workforce scheduling and routing problems. The study in Mustafa et al. (2015) investigates the performance of heuristics while solving problems with routing and rostering characteristics. The results from their implementation reveal that different low-level heuristics perform better on different problems and that their performance varies during a search process. Paper by Dahmen and Rekik (2015) addresses a multi-activity multi-day shift scheduling problem in a continuous and flexible environment, including a heterogeneous workforce. Given days off schedules associated with each employee, the authors objective is to construct and assign admissible multi-activity shifts to employees on their workdays in a way that minimises understaffing and over-staffing with a restricted budget on workforce

cost using a hybrid heuristic which combines tabu search with a branch-and-bound procedure.

Work by Pan et al. (2018) investigates the tour scheduling problem with a multi-activity context, a challenging problem that often arises in personnel scheduling. The authors propose a hybrid heuristic, which combines tabu search and large neighbourhood search techniques. A Mixed-Integer Linear Programming (MILP) based heuristic algorithm Éles et al. (2018) is proposed to obtain economical schedules for the mobile workforce, considering travelling, execution and resource utilisation costs and other requirements. Their solution method involves a MILP model, as well as a greedy heuristic algorithm, which assigns tasks one by one to teams' schedules. This two-level approach ensures the practically acceptable computational time. Barbosa-Póvoa et al. (2018) lay the groundwork by providing a comprehensive review of optimization models and algorithms, offering insights into heuristic methodologies employed in crew scheduling within the airline industry. Elharrouss (2022) advances this by presenting an efficient algorithm explicitly tailored for crew scheduling, incorporating legality constraints, showcasing the practical application of heuristics to address complex regulatory requirements. Fang and Wang's (2021) work contribute to the literature by introducing an algorithm designed for large-scale crew scheduling, featuring heuristics to efficiently manage constraints related to multiple days off. Gao (2021) brings a fresh perspective with a heuristic-driven machine learning approach, particularly focusing on preference-based constraints in crew scheduling. Finally, Wang and Zhao's (2022) fuzzy-based algorithm introduce a heuristic perspective to handle uncertainties in crew scheduling scenarios, illustrating the adaptability of heuristics in addressing dynamic and unpredictable factors. Together, these works demonstrate the versatility and

effectiveness of heuristic methods in tackling various challenges within the domain of airline crew scheduling.

2.4 Genetic Algorithm (GA)

The genetic algorithm first appears in the 70s proposed by Holland (1975); it is based on Darwin's theory of biological evolution. In computer science, GA is a meta-heuristic procedure that may provide a sufficiently right solution for a complex optimisation problem with limited computation capacity. Tanomaru in his paper Tanomaru (1995) uses a GA based heuristic for a staff scheduling problem of similar complexity to real-life. He encounters problem involves a significant number of variables, rather complex data structures for program implementation, and several probabilistic-heuristic operators. Simple GA can be applied with rotating workforce with multiple instances and constraints, as shown in Musliu (2004), the implementation gives promising results with further improvement still possible.

Experimental results from Dean (2008) show that GA with two-dimensional array staff-scheduling implementation outperforms GA with bit-string staff-scheduling implementation. Both approaches generate a comparable high percentage of optimal schedules, but the two-dimensional array chromosome approach converges much more rapidly to an optimal solution. GA also can be applied to solve the problem of multi-period staff allocation, considering the flexibility of working time, versatility, learning and unlearning phenomena as presented by Attia et al. (2014) in their research paper.

One of the most well-known problems which always relate to GA or scheduling is the nurse scheduling problem, which has NP-hard (non-deterministic polynomial-time hardness) complexity. There are various researches in solving nurse scheduling (NSP) problem using the GA approach to find an optimal way to assign nurses to shift while

fulfilling sets of hard and soft constraints. The indirect GA suggested by Aickelin and Dowsland (2004) extends the classical fitness calculation by the use of separate heuristic decoder function and changing the problem into a permutation problem. The quality of the final results, together with the fact that the indirect GA approach proved to be more flexible and robust than Tabu Search, makes the indirect GA an excellent choice to solve the NSP and also can be applied to all constrained scheduling problems.

Simulated annealing (SA) is a heuristic optimization technique that is commonly used to solve combinatorial optimization problems. It is based on the physical process of annealing, which involves heating and cooling a material to modify its properties. SA was proven to be able to produce better quality solutions for NSP than basic GA (Kundu et al., 2008), the authors findings indicate that aside from the quality of solutions, the time needed is also more, this is because the authors implement simple GA and did not include constraint checks. An interesting approach to solve NSP was to hybridise SA and GA, and this was the approach taken by Bai et al. (2010) to combine SA with local search and GA framework. Computational results show that the hybrid of SA, local search and GA algorithm outperforms both the GA with stochastic ranking and the SA hyper-heuristic alone. Some other GA approaches to solve NSP include using cost bit matrix (Kim et al., 2013) and two-point crossover Leksakul and Phetsawat (2014), which provide some improvements and localised solutions. There is also hybridisation of fuzzy set theory with GA where fuzzy set theory is used for fitness evaluation Mutingi and Mbohwa (2014) in order to solve NSP. Another GA with cost bit matrix Alharbi and Alqahtani (2016) is present to solve doctor scheduling problem (DSP) which the basic is similar to NSP.

Variants of hybrid GA began to surface in studies such as Souai and Teghem (2009) where the authors implement multi-points crossover operator, local search heuristic and heuristic-based initialisation on GA to improve the feasibility of the obtained solutions for airline crew-pairing and scheduling problem. Paper by Zeren and Özkol (2012) also implemented airline crew-pairing scheduling with improvised GA by introducing a perturbation operator. This new perturbation operator deliberately made the chromosomes infeasible, which provides better results, increase in convergence rate, and with consideration of multiple crew bases. A GA based scheduling for aircraft ground service was introduced by Ip et al. (2013) to solve scheduling optimisation problem of ground service with vehicle routing problem with time windows of multiply non-identical vehicles. The authors developed a hybrid encoding scheme that had the adaptive search-ability and also embedded with greedy heuristics for initialisation and re-initialisation of the GA population.

It is possible to adapt GA in a wide range of fields where GA is used to resolve constrained scheduling of employee resourcing as proven (Brezulianu et al., 2009; Brezulianu et al., 2012). In the manufacturing environment with multi-shift staff allocation, GA implementation with multi-skilled and flexible working hours is proven possible (Attia et al., 2014). Aydemir and Koruca (2015) in their paper presented priority-rule based GA in the manufacturing environment for job shop scheduling problem.

Some research focus on GA improvement by adding the adaptive approach to increase the right chromosomes or feasible schedules hoping to get the best optimal solutions. One research (Shen et al., 2013) employed an adaptive evolutionary approach by incorporating a hybrid GA for public transport crew scheduling problems. The authors prove that, by varying the chromosome lengths, adaptively better solutions can be found; the

process is achieved by GA operations of crossover and mutation with removal and replenishment strategies aided by a simple greedy algorithm. Adaptive GA demonstrate better results (Algethami & Landa-Silva, 2017) compared to traditional GA in workforce scheduling and routing problem (WSRP), the authors present a GA that uses a diversity-based adaptive parameter control method for three elements: adaptive mutation/crossover rates, dynamic population size and random crossover exchange. The authors in their next paper (Algethami et al., 2018) focus on multiple crossover operator of GA, where a learning process is employed.

Investigation on GA operators helps to understand better how GA can be effectively employed, Algethami and Landa-silva (2015) developed Steady State Genetic Algorithm (SSGA) using two cost-based genetic operators tailored to the WSRP to identify genetic operators that could help to minimise the violation of customer/workers preferences. In the subsequent WSRP paper Algethami et al. (2016), the authors investigate customised chromosome representation and with various settings of crossover rate, mutation rate and population size, for each combination of one crossover operator and one mutation operator which yield improved results. An extension of research on WSRP using SSGA and generational GA (GenGA) by the same primary author (Algethami et al., 2017) explore twelve operators where the results showed that random swap mutation and uniform crossover perform the best. Liu and Gao's (2021a) hybrid algorithm combines genetic algorithm and simulated annealing for efficient airline crew scheduling, synergizing evolutionary principles with probabilistic search for enhanced optimization.

2.5 Shift Design for Airline Ground Crew

Airline ground crew scheduling is a complex optimization problem that involves assigning duties to crew members while satisfying a set of constraints, such as flight time limits, rest or break requirements, and crew pairing requirements. Many researchers have proposed various methods and algorithms to solve this problem, ranging from mathematical programming to heuristic-based approaches.

In recent years, there has been an increasing interest in developing adaptive optimization algorithms for airline crew scheduling. Guo and Lv (2017) also proposed an effective algorithm for airline crew scheduling based on ALNS. The authors showed that their algorithm can achieve better results than other algorithms in terms of solution quality and efficiency. Liu and Gao (2018) proposed an integrated algorithm that considers flight connections in airline crew scheduling. The proposed algorithm was tested in real-world instances and showed significant improvements in solution quality. Huang (2021) proposed an adaptive optimization algorithm that combines the adaptive large neighborhood search (ALNS) with the greedy randomized adaptive search procedure (GRASP). The algorithm was shown to outperform other state-of-the-art algorithms in terms of solution quality and computational time.

Another popular approach is the use of hybrid algorithms that combine different optimization methods to improve the quality of solutions. Zhang (2021) proposed a hybrid algorithm for airline crew scheduling that combines the simulated annealing (SA) algorithm with a genetic algorithm (GA). The proposed algorithm was compared with other state-of-the-art algorithms and showed significant improvements in solution quality. Liu and Gao (2021b) proposed a hybrid algorithm that combines GA with SA to solve the airline crew

scheduling problem. The proposed algorithm was tested in real-world instances and showed significant improvements in solution quality.

Fuzzy logic is also a popular approach used in airline crew scheduling. Bai (2021) proposed a novel heuristic algorithm that uses fuzzy logic to solve the airline crew scheduling problem with fuzzy constraints. The proposed algorithm was compared with other state-of-the-art algorithms and showed significant improvements in solution quality.

Machine learning approaches have also been applied to solve the airline crew scheduling problem. Gao (2021) proposed a machine learning approach that uses preference-based constraints to solve the problem. The proposed approach was shown to improve solution quality and computational time compared to other state-of-the-art algorithms.

Some researchers have also proposed algorithms that consider specific constraints. Elharrouss (2022) proposed an algorithm that considers legality constraints in airline crew scheduling. The proposed algorithm was shown to be efficient in finding feasible solutions for large-scale problems.

Wei and Yuan (2017) proposed a branch-and-price algorithm specifically addressing duty constraints in the airline crew pairing problem. Building on this, Li and Yin (2018) introduced a two-phase algorithm in 2018, offering a robust solution to the complexities of crew scheduling under intricate constraints. Liu's integrated algorithm (2018) focused on crew scheduling with flight connections, emphasizing the challenges associated with connecting flights. Advancing to 2019, Liu and Gao presented a distributed algorithm tailored for crew pairing, specifically addressing scenarios with a large number of constraints. In 2020, Kim et al. contributed with an optimization model, incorporating a preferential bidding system for crew scheduling efficiency. Feng and Wang (2020) added to

the literature with an efficient algorithm designed for various crew scheduling scenarios. Other approaches that have been used in airline crew scheduling include ant colony optimization (Zhang & Zhao, 2021), variable neighborhood search (Wang & Wang, 2019), and tabu search (Zhang & Jin, 2020).

In summary, there are many approaches and algorithms that have been proposed to solve the airline ground crew scheduling problem, each with its unique advantages and disadvantages. Adaptive optimization algorithms, hybrid algorithms, fuzzy logic, and machine learning approaches have shown promising results in improving solution quality and computational time. Researchers continue to develop and refine these methods to provide better solutions to the airline industry.

2.6 Conclusion

This chapter reviewed the problems of rostering and scheduling from various perspectives to methodologies used by previous researchers. There are various approaches for shift scheduling problems based on the analysis of literature; each of them demonstrates its relevance and utility in numerous domains, and their purpose is often to contribute to better results than current ones. To acquire the best solution, most of these techniques either suffer from heavy computational baggage or trade-off efficiency to achieve a shorter processing time. More recent studies have shown promising results in producing a high-quality solution within a shorter amount of time by hybridisation of techniques.

Several impressive results were identified while reviewing the literature on how to construct the roster and schedule computationally. There is a fragile balance between computational complexity, convergence characteristics, and algorithm portability for the traditional methods.

Many approaches are employing classical methods that are too slow and complicated, with massive calculation involved. For example, mathematical-based strategies such as linear, integer, mixed integer programming, column generation, Lagrange relaxation, etc. are still popular among researchers (e.g., Laporte et al., 1980; Brusco & Jacobs, 2001; Auke et al., 2008; Fang & Rong, 2012; Cemal & Banu, 2013; İbrahim et al., 2013; Dohn & Mason, 2013; Telhada, 2014; Jacquillat & Odoni, 2015; Éles et al., 2018). The reason for the popularity is that the mathematical method can get high-quality outcomes that give a guarantee of optimality. The methods are comparably complex and run the risk that it will return no viable solution, so it is typically used to find the answer where small instances are involved.

Compared to the optimization approach, heuristic-based algorithms are thus remarkably efficient in terms of processing time with the trade-off of optimality, which is mentioned in various studies before (Gärtner et al., 2005; Montoya & Mejía, 2006; Musliu, 2006, 2010; Di Gaspero et al., 2010; Clausen, 2010; Rodič & Baggia, 2013; Constantino et al., 2014; Misir et al., 2015; Castillo-Salazar et al., 2015; Pan et al., 2016; Sze et al., 2017; Pan et al., 2018; Éles et al., 2018). The efficiency lies in the production of reasonably good feasible solutions within a limited amount of running time. The use of heuristics, however, often results in some disadvantages, as they can neither demonstrably produce optimal, accurate and precise solutions nor demonstrably provide completeness for the purported solution. Gap analysis table as shown in as shown in Table 2.1.

Table 2.1: Research Gap Analysis

Requirement	Current State	Ideal Literature	Gap in Literature	Proposed Method
Efficiency in Shift Scheduling	Methods with heavy computational baggage prioritize efficiency over optimality.	Ideal literature balances computational efficiency and solution quality.	Gap in achieving a balance between complexity and speed.	Proposed hybrid heuristic leverages workload prediction and local search.
Overreliance on Classical Methods	Classical methods (e.g., linear, integer programming) are popular but slow and complex.	Literature lacks emphasis on faster, adaptable methods for real-world scenarios.	Gap in addressing the need for efficient, adaptable methods.	Hybrid heuristic offers an efficient alternative to traditional methods.
Trade-off between Heuristic and Optimization	Heuristic algorithms sacrifice optimality, limiting solution quality.	Ideal literature achieves a better trade-off between efficiency and optimality.	Gap in balancing heuristic efficiency and solution quality.	Proposed hybridization method aims for a more effective trade-off.
Flexibility and Robustness	Literature recognizes hybrid benefits but lacks emphasis on flexibility and robustness.	Ideal literature provides comprehensive insights into hybrid flexibility and robustness.	Gap in addressing the importance of flexibility and robustness.	Proposed hybrid heuristic integrates workload prediction and rule prioritization.

The challenge is to develop a system that demonstrates the most appropriate compromise between both the complexity of computation that offers accuracy and fast execution time. The use of hybrid methods in solving shift scheduling problems has several strengths, including:

- Improved solution quality: Hybrid methods combine different approaches to improve the quality of the solutions obtained. By combining the strengths of different methods, hybrid algorithms can find better solutions than individual methods alone.
- Reduced computational time: Hybrid methods can reduce computational time by using different techniques in different stages of the algorithm. This allows the algorithm to focus on areas of the problem where a particular technique is most effective and avoid unnecessary computations in other areas.
- Flexibility: Hybrid methods are flexible and can be customized to suit the specific requirements of different airline crew shift scheduling problems. By combining different approaches, a hybrid algorithm can be tailored to address the specific constraints and objectives of the problem at hand.
- Robustness: Hybrid methods are often more robust than individual methods alone. By using multiple techniques, hybrid algorithms can better handle uncertainties and variations in the problem, resulting in more reliable solutions.
- Adaptability: Hybrid methods can adapt to changes in the problem and adjust their approach accordingly. This makes them particularly useful in dynamic scheduling environments where conditions can change rapidly.

With the above considerations, this study proposes to leverage methods of hybridisation to take advantage of its ability to solve a complex shift design problem with a reasonably decent solution rapidly. For workload prediction, a two-stage heuristic is employed, and a simple local search heuristic method is implemented to obtain the result. To ensure the quality of the solution produced, the result is then further optimised with rule prioritisation optimisation. Based on the literature review undertaken in this study, the proposed hybridisation methods using workload demand prediction while taking into consideration operational constraints for a Malaysian airline are considered novel in solving the rostering and scheduling problem.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

In various industries, including aviation, workforce rostering and scheduling, is an important, complicated and requires a vast amount of time to plan. Hence, careful and meticulous planning can lead to significant savings and efficiency. Airline ground operations have become crucial functions in the planning and scheduling of airline operations and services. Airlines are expected to meet stipulated timing that has been committed to passengers even with a high volume of scheduled flights and tight time windows. Any delay will result in a significant loss for airlines as delays lead to additional costs, mainly for the crew, fuel, aircraft, and maintenance.

A comprehensive study on flight delay estimated USD\$8.3 billion was lost by airlines in 2007 in the United States alone, as highlighted by (Peterson et al., 2013). In the same study, passengers were estimated to have lost USD\$16.7 billion due to schedule buffer, delayed flights, flight cancellations, and missed connections. With forecasts of substantial increases in flight operations in the future by the Federal Aviation Administration (FAA), the prospect of even higher levels of flight delay is likely if no actions are taken. A study published in the Journal of Air Transport Management in 2018 found that an automated crew scheduling system could reduce crew costs by up to 3% for a major European airline. Another study published in the Journal of Scheduling in 2020 found that an automated scheduling system could reduce crew costs by up to 6% for a major North American airline. The emergence of low-cost carriers has created additional demands for short aircraft turnaround times, while the hub-and-spoke network structures of larger carriers are creating

a demand for reduced connection times between airports. Therefore, capable rostering and scheduling has become paramount for airlines operating in this competitive environment.

In this research, workforce rostering and scheduling is studied, particularly on airline ground staff based at Kuala Lumpur International Airport (KLIA) which in 2018 is 23rd busiest airport in the world with close to 60 million passenger traffic that year. Rostering and scheduling data was obtained from the airline directly besides observing the behaviour and duties of people located within teams and management. Rostering and scheduling design problems were considered with the specific emphasis on the airport where the airline ground staff was operating. When an aircraft lands at an airport, and before it is ready for departure, there are numerous tasks that must be carried out. Many airports are already jam-packed and are operating at full or near-full capacity while expecting further growth in passenger numbers in the future. The flight arrival or departure times are scheduled at peak periods to provide short transit times for transferring passengers. The flight scheduling nature causes the airport to be hectic during periods of high frequent arrivals and departures, while other periods of the day are relatively quiet.

The rostering and scheduling problem research will be based on the airline's security teams, which are tasked to provide security monitoring and assistance to all arrival and departure flights. Airline security teams are a good case study for airline crew scheduling for several reasons:

- 24/7 Operations: Airline security teams often work around the clock to ensure the safety of passengers and crew. This means that their scheduling requirements are similar to those of the airline crew, who also work in shifts to keep flights operational at all times.

- **Complex Scheduling Requirements:** Airline security teams have complex scheduling requirements that need to be managed efficiently. For example, security personnel may need to be deployed to different airports or flights at short notice, and their schedules need to be adjusted accordingly. Similarly, airline crew scheduling also involves complex requirements, such as managing rest periods, crew availability, and flight schedules.
- **Importance of Safety and Security:** Both airline security teams and airline crew are responsible for the safety and security of passengers and crew. This makes their scheduling requirements critical to ensure that they are alert and focused during their shifts.
- **Regulatory Compliance:** Both airline security and crew scheduling are subject to strict regulatory compliance requirements. For example, airline crew scheduling needs to comply with regulations around rest periods and maximum flight hours, while airline security teams need to comply with regulations around airport security and screening procedures.

By studying airline security teams' scheduling requirements, researchers can gain insights into the challenges and best practices of managing complex scheduling requirements for 24/7 operations. This information can then be applied to optimize airline crew scheduling to improve operational efficiency, ensure regulatory compliance, and enhance the safety and security of passengers and crew.

All aircraft must be served within its time window considering the Standard Time of Arrival (STA) and Standard Time of Departure (STD) which represent the following types of flight patterns:

Inbound (Flights arrive on the day but will depart the following day.)

Outbound (Flights arrived on an earlier day and departed on the day.)

The workforce demand at airports is represented as a demand curve, in which it divided the planning period into time slots or windows with predefined periods. The time slots or windows are planned for fifteen-minute intervals, with each having several required workforces. High frequency of flight schedules during peak periods causes a high variance in inactivity for the day, meaning that the demand curve may contain a high level of irregularity, where the difference between workforce requirements of two adjacent time slots can be enormous. Figure 3.1 shows an example of a demand curve chart.

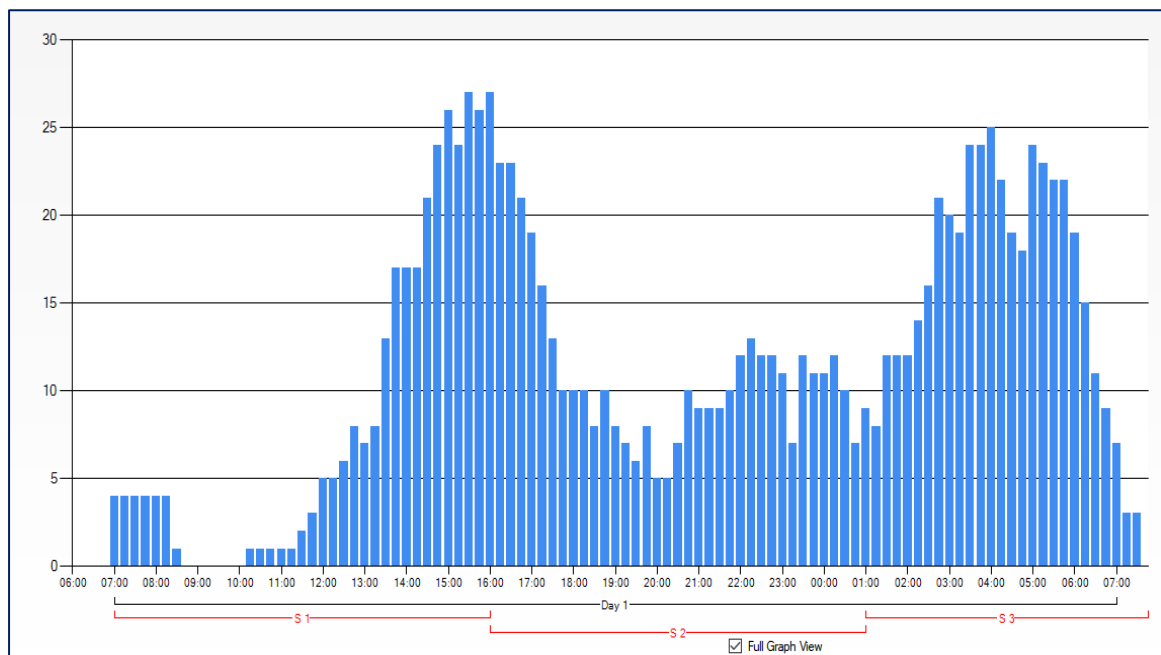


Figure 3.1: Work Force Demand Curve Chart

When the workforce demand is irregular, as shown in Figure 3.1, it is unlikely that one shift (S1 = Shift 1, S2 = Shift 2, and S3 = Shift 3) can cover the demand entirely. Covering a single peak may require a considerable workforce. If a massive number of workforce is assigned to cover a single peak, the workforce for that shift will be idle before and/or after the peak. The workforce idling scenarios emphasise the need to consider both workforce shortages (understaffing) and surplus (over-staffing) as part of the desired solution quality.

The roster and schedule should be designed such that it allows the organisation to satisfy constraints of local labour regulations. Such constraints include such as shift lengths, the placement and number of relief breaks or the allowed placement of shifts with regards to office hours or similar requirements.

This research is focused on finding rostering and scheduling solution for airline ground staff, i.e., the estimation of workloads and the generation of staff before operations. This means that workloads for the planning period are calculated from a fixed flight schedule, resources, and forecasted jobs. Real-time issues aspects i.e., like short-term schedule changes or interruptions due to flight delays or illnesses, will not be considered.

The optimisation models and algorithms developed in this work are generic and apply to airline or airport ground handling activities. However, some specific services may not conform to the general framework and may therefore not be covered such as baggage or check-in services. This study aims to conceive an algorithm which is robust on a wide range of relevant planning scenarios for rostering and scheduling in general.

3.2 Research Framework

The theoretical framework of this study is shown in Figure 3.2. The data collected during the planning and data collection phase is mainly derived from real-world scenarios provided by an industrial partner. To ensure that the study can cover various scenarios, the researchers added randomly simulated data which is explained in Section 3.5.1.

The goal of this study is to ensure that the objectives can be achieved in an organized manner. Phase 1 of the research is focused on planning and data collection. This process involves conducting a literature review to identify the strengths and weaknesses of the proposed model.

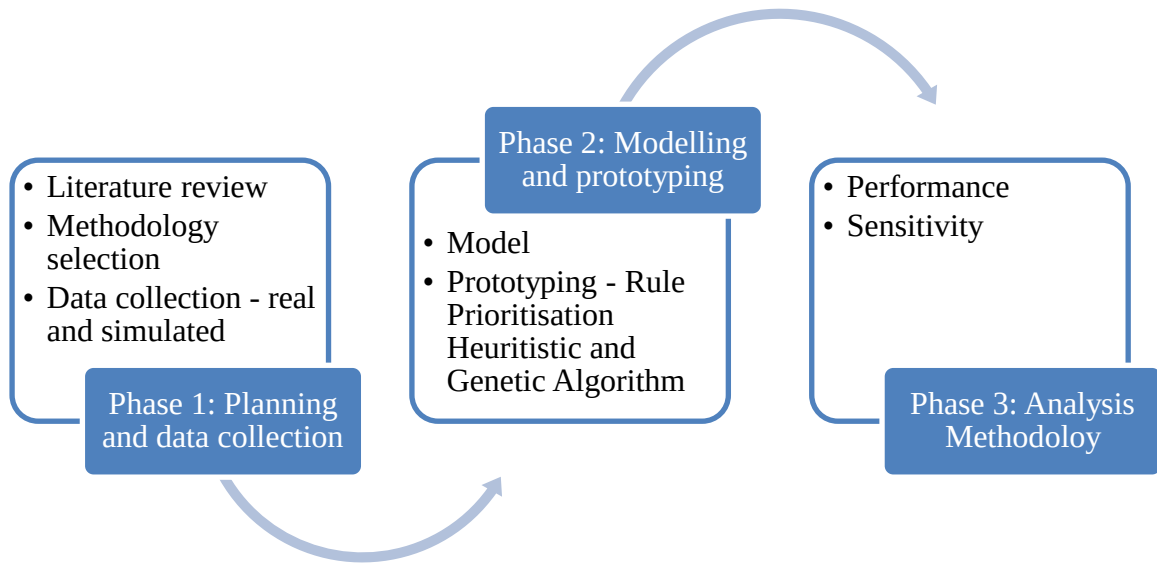


Figure 3.2: Research Framework

Phase 2 of the study is focused on the design and prototyping of the models. This process involves creating and prototyping the different RPH and GA models. Phase 3 of this study is focused on analyzing the performance and sensitivity of the RPH and GA methods. It also aims to ensure that the results of the proposed approach are consistent and robust.

3.3 Phase 1: Planning and Data Collection

During the initial phase of the research, data collection and problem investigation were conducted with the help of a major airline based in KLIA, Malaysia. Several weeks of fieldwork were organised during the initial phase to identify and analyse the processes that adhere to the daily operation of the airline. Some of the key activities include:

Fieldwork: The research team organized several weeks of fieldwork to observe and analyze the daily operations of the airline. This involved closely examining various processes and procedures, such as crew scheduling, maintenance, and ground handling, to identify areas where improvements could be made.

Interviews: The research team conducted interviews with key personnel at the airline, including schedulers, dispatchers, crew members, and ground handling staff. These interviews were designed to gather insights into the challenges faced by the airline in terms of scheduling and resource allocation.

Data Collection: Real data on the airline's actual roster and shift schedules for the month of September and October 2013 were also collected during this phase of the research. This data together with the data retrieve from (Sze et al., 2010) was used to acquire insights into the airline's scheduling practices and to identify patterns or trends that could be used to inform the development of an optimization algorithm.

Documentation Review: The research team reviewed documentation related to the airline's scheduling practices, such as scheduling policies, procedures, and regulations. This helped the team to gain a deeper understanding of the constraints and requirements that must be taken into account when scheduling crew members, maintenance, and other resources.

Through these activities, the research team was able to gain a comprehensive understanding of the scheduling challenges faced by the airline and to identify potential solutions. This information was critical in developing an effective optimization algorithm that could help the airline to better manage its resources and reduce operational costs. Real data of the actual roster and shift schedule were also gathered to help with the investigation of potential solutions. Table 3.1 shows one of the examples of the shift schedule.

Table 3.1: Manually Assigned Shifts

Day 1	Day 2	Day 3	Day 4
0700-1700	0700-1700	0700-1600	0700-1600
1500-2400	1500-2400	1600-2400	1600-2400
2200-0700	2200-0700	2200-0700	2200-0700

Real-life datasets were used to generate shift scheduling with different start time ranges to analyse the sensitivity of the proposed algorithm. Aside from that, jobs demand data were also collected, as shown in Table 3.2.

Table 3.2: Jobs Demand Data

Job ID	Release Time	Deadline	Processing Time	Number of Task
1	0	4	3	1
2	0	4	3	1
3	90	98	5	1
4	90	98	5	1
5	96	100	1	1

Job's demand data are vital information needed by the proposed algorithm to ensure that the shifts created took into consideration all jobs available and reduce wastage. The job's demand data consist of id, release time which is the earliest time the job can be processed, deadline when the job must be completed, processing time which is how long is needed to complete the job and number of task or job.

3.4 Phase 2: Modelling and Prototyping

In this section, the design and construction of roster and shift scheduling are described to show the overall structure of the rostering and scheduling problems. A generic model is described and improvement to the model is proposed with the formulation of the problems. The proposed algorithm and benchmark algorithm are described in full detail.

3.4.1 General Model

General rostering and scheduling model formulation for the suggested configuration can be formulated as described by (Baker, 2009) shown here:

$$\min \sum_{j \in J} c_j X_j$$

$$\text{where} \quad \sum_{j \in J} a_{ij} X_j \geq d_i \quad \forall i \in I$$

$$X_j \geq 0 \text{ and integer } \forall j \in J \quad \text{Equation 3.1}$$

The coefficients and variables used have the following meanings:

X_j : number of workforces assigned to shift j

c_j : cost of shift j

d_i : number of workforces required to work in the i 'th time period

I : set of time periods in a day

J : set of daily shift types

$a_{ij} = 1$ if time period i is a work period in daily shift-type j (0 otherwise)

Equation 3.1 formulate a shift set to reduce the number of workforces and reduce idle time. As a result, the aim is to try and obtain the best fit of available resources to the fluctuating pattern of demand over the forecasting horizon.

3.4.2 Proposed Model

In this section, Phase 2, the heuristic model of rostering and scheduling problems are proposed and implemented. Sets, parameters, decision variables and constraints are identified and defined to provide examples of the work rules and to show the overall structure of the optimisation problem. The model will guide the implementation of the system and maintain a strict separation between the optimisation algorithms and the constraint checking procedures. The proposed model generally can be written as follows.

$$\text{Minimise } \sum_{t=0}^M (r_t - B_t)^2$$

Equation 3.2

where time interval, $t = 1, 2, \dots, M$ and r_t = the actual workforce requires at t .

Equation 3.2 shows the primary objective function (1) was chosen to minimise the number of workforces required to complete all the jobs or tasks in set A to meet the rostering

and scheduling requirements. The proposed model Rule Prioritisation Heuristic Algorithm aims to improve the current implementation in obtaining the best fit of available resources to the changing pattern of demand over the planning horizon. The Genetic Algorithm is adapted to the current problem to serve as a benchmark for Step 3. Figure 3.3 illustrates the steps planned that guided the development of the RPHA model and GA.

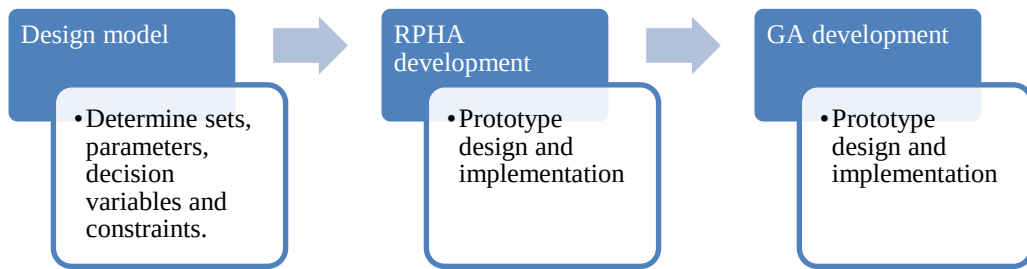


Figure 3.3: Model Design and Development

Therefore, this Phase 2 comprises a model design step to identify the formulation, the description of RPHA and the design of the GA for benchmarking and analysis for Phase 3.

3.4.3 Rostering and Scheduling Formulation

In this section, a mathematical model of roster and scheduling problems is presented and explained. Formulation of the mathematical model is to provide an example of the work rules and to show the overall structure of the optimisation problem.

3.4.3.1 Sets

Rostering and scheduling consist of a few phases to identify the number of workforces required for each day. For each day, shifts will be created to determine the workforce required for a specific phase of duration in the day, as shown in Table 3.3.

Table 3.3: Typical Shift Type

Shift	Period	Workforce Requirement
Morning	0700 – 1600	27
Evening	1600 – 0100	27
Night	0100 – 0900	25

The typical shift-type shown in Table 3.3 is easy to implement, as the configuration of the shift is non-overlapping. However, the configuration is too rigid to be implemented in airline ground staff operations, where the demand fluctuates. For this research, the suggested configuration approach would be four shifts a day where over-lapping is allowed as shown in Table 3.4, which is also applied in (Di Gaspero et al., 2007).

Table 3.4: Suggested Shift-Type Configuration

Shift	Period	Workforce Requirement
Morning	0500 – 1300	4
Afternoon	1000 – 1800	27
Evening	1600 – 0000	13
Night	2300 – 0700	25

The following sets are identified and derived for the mathematical model formulation:

T : set of days in the planning period = $\{1, 2, 3, \dots, n\}$

A : set of jobs/tasks = $\{a_1, a_2, a_3, \dots, a_n\}$

B : set of the workforce = $\{1, 2, 3, \dots, m\}$

F : set of shift type = $\{1, 2, 3, 4\}$

where 1 = “morning”, 2 = “afternoon”, 3 = “evening” and 4 = “night”

D : set of days in week $\{1, 2, 3, \dots, 7\}$

S : set of shift slots = $\{s_{ij} \mid i \in D \text{ \& } j \in F\}$

Set A represents the number of tasks or jobs that are required to be completed in a particular period. Each job or task, $a \in A$ requires a number of workforces B to complete the tasks. Set F represents the shift type of the day while Set D represent the days in the week.

3.4.3.2 Parameters

The parameters for shift scheduling design have following parameters.

n = End time of planning period.

a_i = Number of tasks in period $i \in F$.

m = Number of workforces required on duty in time interval $t = 1, 2, \dots, M$

$[Start_i, End_i]$ = Time window for shift slot $i \in F$.

3.4.3.3 Decision Variables

The problem in designing the shifts for rostering and scheduling is defined by determining the values of the following decision variables for each shift, s_{ij} .

1. A number of workforces where $\sum_{i=1}^{k=m} y_{ij}^k \geq \max[a_n]$ for $n = p, \dots, q \in T$. p = earliest start time and q = latest end time.

$$y_{ij}^k = \begin{cases} 1, & \text{if workforce } k \in B \text{ is assigned to shift slot } i, j \in s_{ij} \\ 0, & \text{otherwise} \end{cases}$$

$$w_{ij}^d = \text{feasible shift start time of } s_{ij} \in S, \text{ where } d \in T$$

2. A duration $[d_{\min}, d_{\max}]$, where $d_{\min}$ is the minimum duration and $d_{\max}$ is the maximum duration of the shift.

$$s_{ij}^{\text{duration}} = \text{feasible duration for } s_{ij} \in S$$

3.4.3.4 Constraints

In this section, to model restrictions of the workforce, constraints that are related to rostering and shift scheduling are defined.

1. All jobs or tasks A must be assigned workforce B .
2. Shift duration to be between x hours and y hours as defined.

$$x \leq (s_{ij}^{\text{start time}} + s_{ij}^{\text{duration}}) \leq y$$

3. Shift start time difference to be between x hours and y hours as defined.

$$x \leq (s_{ij}^{\text{earliest start time}} - s_{ij}^{\text{latest start time}}) \leq y$$

3.4.4 Rule Prioritisation Heuristic Algorithm

To solve the problem, shift design based on the proposed model, the Two-Stage Scheduling Heuristic (TSH) strategy is adopted with a local search heuristic. A two-stage scheduling heuristic is a type of optimization algorithm commonly used in scheduling

problems. In this approach, the scheduling problem is broken down into two stages. In the first stage, a priority list of jobs or tasks is created based on a set of predetermined criteria, such as processing time or due date. The priority list is used to determine the sequence in which the jobs are processed. In the second stage, the jobs are assigned to specific machines or resources. This assignment is based on another set of predetermined criteria, such as availability or capacity of the machines. The two-stage scheduling heuristic is a popular approach because it is relatively simple to implement and can provide good results for many scheduling problems. However, it may not always produce optimal solutions and may require fine-tuning for specific applications. Overall, the two-stage scheduling heuristic can be an effective tool for reducing the complexity of scheduling problems and improving the efficiency of scheduling algorithms. TSH produces a sound valid integer solution for workload prediction, and subsequently, the local search heuristic procedure is implemented to generate optimum shift scheduling. The flow of the proposed RPH architecture is shown in Figure 3.4.

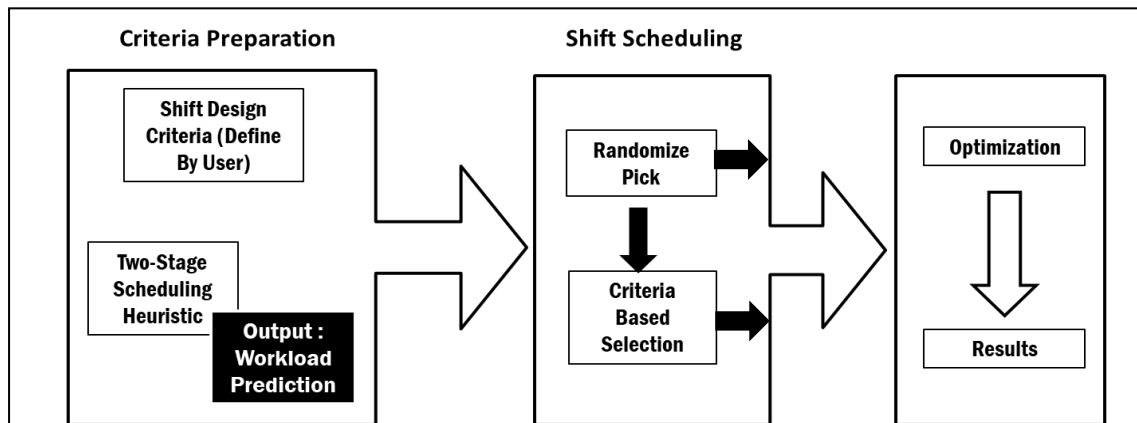


Figure 3.4: Proposed Design of RPH Algorithm

A set of parameters defined by the user is used to generate all the possibilities of the shifts. The number of possibilities depends on the start time range of the shift and the duration range of the shift defined. As the parameters are definable, the only bottleneck will be on the hardware where the solution is running. The shifts are assigned accordingly to form schedules. The list of schedules is then processed using two different methods to identify efficient schedules to increase the chances of obtaining the best schedule candidates within a short time frame. All the selected schedules are then optimised as best as possible to cover all jobs with minimum workforce and the least wastage. Figure 3.5 illustrates the shift scheduling process.

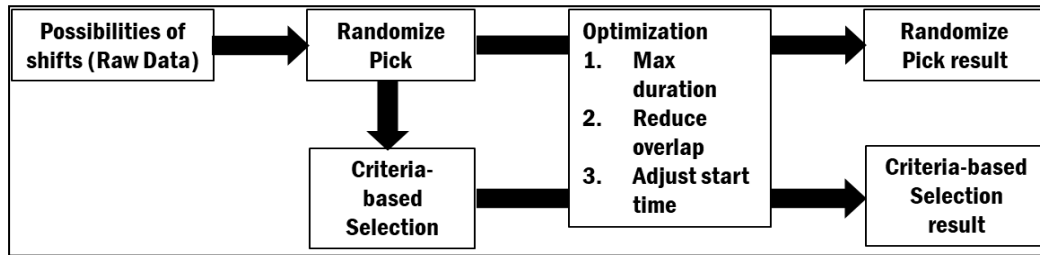


Figure 3.5: Shift Scheduling Process

Both approaches shared the same initial data with pre-processing, optimisation, and post-processing functions. In the next section, further details of the local search based on the heuristic approach will be discussed and explain.

3.4.4.1 Randomise Pick

This process applies a simple randomise pick (RP) method to select good schedules from all the shifts available. The schedules are picked based on the smallest number of the following parameters: workforce follow by uncover job, overlapped and gap hour. Recursive randomisation is also implemented to reduce the number of potential candidate schedules to

maintain the good schedules among the potential good candidate schedules identified. The implementation of the randomise pick function is documented in the algorithm shown:

Algorithm: Randomise pick function

```

for i = 0, .....ShiftMax do

    MorningShift.Add(Random(MorningCandidate));

    DayShift.Add(Random(DayCandidate));

    AfternoonShift.Add(Random(AfternoonCandidate));

    NightShift.Add(Random(NightCandidate));

end for

Schedule = CreateSchedule(MorningShift, DayShift, AfternoonShift,
    NightShift);

CalculatePenalty(Schedule);

Optimize(Schedule);

Present(Schedule);

```

The algorithm will generate all the possibilities of the shifts and pair it randomly to form a schedule as shown in Figure 3.6.

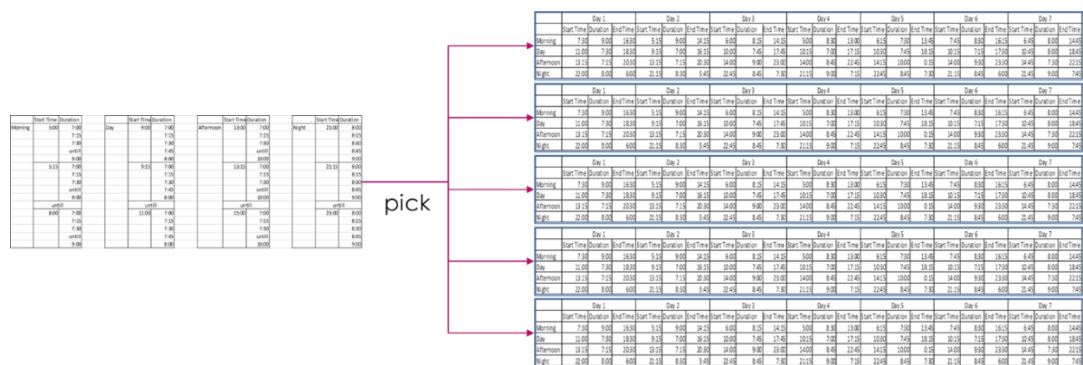


Figure 3.6: Randomise Pick Function in Picking Schedule

The process will be repeatedly recursively to so that the best schedule can be identified and pick. The candidates of schedules will be pre-processed, optimised, and presented before the process to select the best schedule.

3.4.4.2 Criteria-Based Selection

The criteria-based selection (CBS) is aimed at picking schedules that fulfil specific required criteria such as minimum workforce/resource ($f(x)$) required to complete all jobs/tasks, duration required ($f(y)$) and start time of a shift ($f(z)$).

$$\alpha^{min/max} = min/max(\{f(x): x = 1, \dots, n\}), \alpha = workforce/resource \quad (1)$$

$$\beta^{min/max} = min/max(\{f(y): y = 1, \dots, n\}), \beta = duration \quad (2)$$

$$\gamma^{earliest/latest} = min/max(\{f(z): z = 1, \dots, n\}), \gamma = starttime \quad (3)$$

Forty-eight combinations of criteria are identified by listing out all the possible combinations. All the combinatory criteria are used to determine the candidates for the “good” schedule for the next selection process. The list of criteria combinations is as follows:

$$\alpha^{min/max} > \beta^{min/max} > \gamma^{earlies/latest}, \quad (4)$$

resource is prioritized over duration, and duration is prioritized over start time.

$$\beta^{min/max} > \gamma^{earlies/latest} > \alpha^{min/max}, \quad (5)$$

duration takes precedence over start time, and start time takes precedence over resource.

$$\beta^{min/max} > \alpha^{min/max} > \gamma^{earlies/latest}, \quad (6)$$

duration is favored over resource, and resource is favored over start time.

$$\gamma^{earlies/latest} > \beta^{min/max} > \alpha^{min/max}, \quad (7)$$

start time is prioritized over duration, and duration is prioritized over resource.

$$\gamma^{earlies/latest} > \alpha^{min/max} > \beta^{min/max}, \quad (8)$$

start time takes precedence over resource, and resource takes precedence over duration.

$$\alpha^{min/max} > \gamma^{earlies/latest} > \beta^{min/max}, \quad (9)$$

resource is favored over start time, and start time is favored over duration.

From these selections of criteria, the number of candidates of good schedule picked is 48 derive from 2^3 (3 parameters with 2 possibilities) x 6 ((4) – (9)) combination of arrangement. The intentions of applying the criteria-based selection are to evaluate and to discover the criteria that most likely influence the best solution. The implementation of the algorithm based on the methods explained is shown below.

Algorithm: Criteria-based selection function (minimum resource $\alpha^{\min}$)

```

minResources = -1;
for m = 0, ...resources.Count
    if uncover[m] == minUncovered then
        if minResources == -1 then
            minResources = resources[m];
            index = m;
        else
            if minResources > resources[m] then
                minResources = resources[m];
                index = m;
            end if
        end if
    end if
end for

```

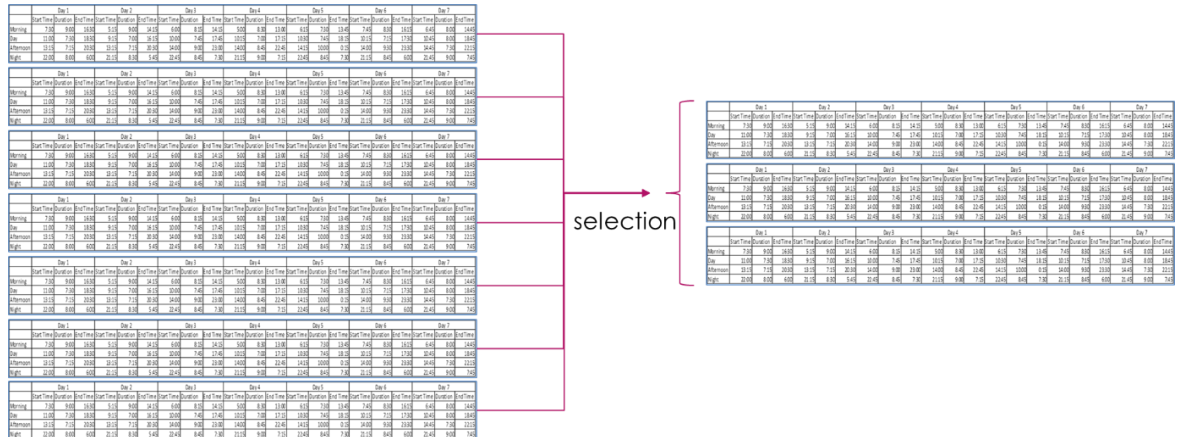


Figure 3.7: Criteria-Based Selection Function in Selecting Schedule

The implementation of the algorithm shown in Figure 3.7 is extended to cover the rest of the possibilities in order to produce the best solutions for each criterion determined earlier.

3.4.4.3 Optimisation

Good schedule candidates picked or selected from both randomised and criteria-based selections were optimised to improve the quality of the schedules. The objectives of the optimisation are to reduce the use of the resource/workforce, avoid overlaps of shift period and uncover jobs. The algorithm for optimisation is as shown below.

Algorithm: Optimisation function

<pre> for i = 0, ...ScheduleMax do for j = 0, ...ShiftMax do e1 = GetStartTime(Schedule[i][j]); e2 = GetFirstJobStartTime(jobs); if e1 >= e2 then newStartTime = e2 else newStartTime = e1 end if Schedule[i][j] = VerifyTruncateStartTime(newStartTime); d1 = Schedule[i][j].Duration; d2 = MaxShiftLength(Schedule[i][j+1]); if d1 >= d2 then newDuration = d1 else newDuration = d2 end if Schedule[i][j] = VerifyTruncateEndTime(newDuration); </pre>

```

if j == ShiftMax then
    Schedule[i][j] = VerifyTruncateEndTime(maxDuration);
end if
end for
end for

```

Initially, the algorithm tries to optimise the duration within the permitted time window restriction to cover more jobs while avoiding an overlap of shifts. Next, it tries to adjust the start time to reduce overlap of shifts while respecting the time window restriction. Last, it tries to cover the jobs by pushing the previous shift start time, even though the benefit is for the subsequent shifts. All the optimisations avoid overlapping with other shifts and stay within the range of parameters defined by the user. The optimisation illustration is shown in Figure 3.8.

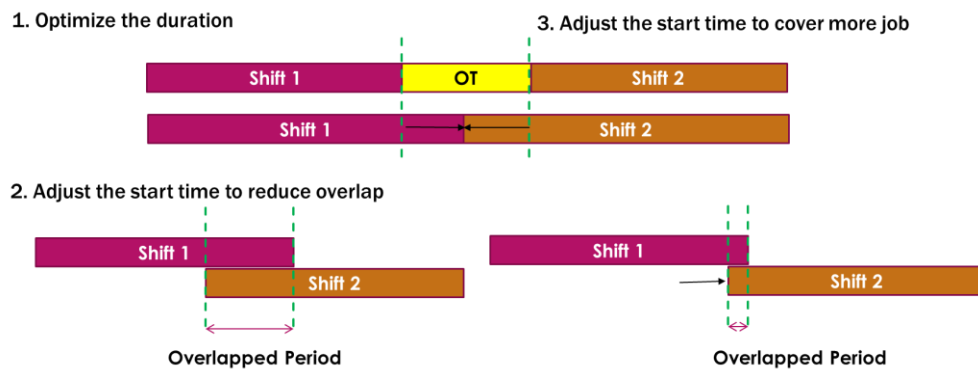


Figure 3.8: Illustration of the Optimisation

The optimisation steps are repeated until there is no improvement. Finally, the candidate for the good schedule is considered optimised and ready to be placed in the evaluation pool.

3.4.5 The Genetic Algorithm

In order to benchmark the proposed approach described in the previous section GA approach is developed and implemented. GA based shift scheduling is often used as a control to compare with heuristic-based shift scheduling methods because of several reasons:

- GA-based methods are a popular optimization technique that has been widely used in various fields, including airline crew scheduling. They have been shown to be effective in solving complex optimization problems by mimicking the process of natural selection and evolution.
- GA-based methods are generally more computationally intensive than heuristic-based methods. They typically require more time to generate solutions, but they are often able to find better solutions than heuristic-based methods.
- Heuristic-based methods, on the other hand, are often faster and more efficient in generating solutions, but may not always find the optimal solution. They rely on rules of thumb, intuition, and experience to guide the search for solutions.
- Comparing GA-based methods with heuristic-based methods provides a benchmark for evaluating the quality of the solutions generated by different methods. It allows researchers and practitioners to determine whether the additional computational effort required by GA-based methods is justified by the improved solution quality.
- Finally, comparing GA-based methods with heuristic-based methods provides insights into the strengths and weaknesses of different optimization techniques. It can help identify areas where each method performs well and areas where they may need improvement.

Overall, using GA-based shift scheduling as a control to compare with heuristic-based shift scheduling methods provides a useful benchmark for evaluating the quality and efficiency of different optimization techniques.

The implementation starts with the same two-stage scheduling heuristic (TSH) method used in heuristic in order to predict the workload in order to get an indicative number of the resource required. The number of resources requires which is derived from the assumption of unlimited workforce available, is subsequently used as initial input for GA implementation in finding the most optimum shift schedule. The flow of the proposed algorithm is shown below.

Algorithm: Main GA function
User input RandomGenerate(Schedules); CalculateFitness(Schedules); for i = 0, ...limit do SelectScheduleCandidates(Schedules); AdaptiveCrossOver (ScheduleCandidates)); Mutation(ScheduleCandidates)); Schedules.Add(ScheduleCandidates)); Schedules.Sort(); Schedules.RemoveBadSchedules(); end for Optimize(Schedule); Present(Schedule);

The prototype was developed in C# using the proposed algorithm shown as a guide, further explanation on each of the function will be further elaborated in the following sections.

3.4.5.1 Fundamentals

The GA methodology starts with a randomly generated an initial pool of a predefined number of possible solutions or candidates. Subsequently, the initial pool of solutions is subjected to pre-define a repetitive process of selection, crossover, and mutation operators for improvement. Generically, the pseudocode GA is as follows:

- 1) randomly initialize population(t)
- 2) determine the fitness of the population(t)
- 3) repeat
 - i) select parents from population(t)
 - ii) perform crossover on parents creating population($t+1$)
 - iii) perform mutation of population($t+1$)
 - iv) determine fitness of population($t+1$)
- 4) until the best individual is good enough or predefined iteration

The GA approach and implementation are explained in detail in subsequent sections.

3.4.5.2 Fitness Function

Selecting shift schedule candidates to form a pool for the next generation shift schedule candidates requires many constraints evaluation. Hence, fitness function is formulated to simplify the determination of the fittest shift schedule candidate. For each constraint violation, penalty will be given based on the parameters which will be discussed following. There are four parameters in assigning shift schedule, as shown in Table 3.5 below (in the order of from the most to the least important).

Table 3.5: Shift Schedule Parameters

Parameters	Description
Uncovered jobs (<i>uc</i>)	The number of jobs that are not assigned to any workforce.
Workforce (<i>wf</i>)	The number of workforces needed.
Overlapped shift (<i>os</i>)	The number of idling person hour when shift overlapped.
Shift start time difference (<i>sstd</i>)	The number of hour difference between the shifts start time for the shift duration.

All the penalties are then summed up and assigned as the fitness function value. The fitness function is defined as:

$$fitness(z) = \sum (uc \times C1) + (wf \times C2) + (os \times C3) + (sstd \times C4)$$

The specific factors and costs represented in the equation are as follows:

- *uc* represents the number of understaffed shifts, and *C1* represents the cost associated with each understaffed shift.
- *wf* represents the number of workers with overtime or insufficient work hours, and *C2* represents the cost associated with each worker with overtime or insufficient work hours.
- *os* represents the number of overlapping shifts, and *C3* represents the cost associated with each overlapping shift.
- *sstd* represents the standard deviation of the start times of the shifts, and *C4* represents the cost associated with deviation from the desired start times.

GA will select based on fitness function value to determine if the candidates are suitable to be in the pool for the next generation evolution.

3.4.5.3 Initial Pool Generation

Based on the parameters input, all the possibilities of the shifts and duration of shifts for a day is generated. For example, based on the start time and duration illustrated in Figure 3.9, the possibilities for each shift are listed out.

	Start Time	Duration			Start Time	Duration			Start Time	Duration			Start Time	Duration
Morning	5:00	7:00		Day	9:00	7:00		Afternoon	13:00	7:00		Night	21:00	8:00
		7:15				7:15				7:15				8:15
		7:30				7:30				7:30				8:30
		untill				7:45				untill				8:45
		9:00				8:00				10:00				9:00
	5:15	7:00			9:15	7:00			13:15	7:00			21:15	8:00
		7:15				7:15				7:15				8:15
		7:30				7:30				7:30				8:30
		untill				7:45				untill				8:45
		9:00				8:00				10:00				9:00
	untill				untill				untill				untill	
	8:00	7:00			11:00	7:00			15:00	7:00			23:00	8:00
		7:15				7:15				7:15				8:15
		7:30				7:30				7:30				8:30
		untill				7:45				untill				8:45
		9:00				8:00				10:00				9:00

Figure 3.9: Shifts Candidates

Between 7am and 9am, there are nine possible times - 7:00, 7:15, 7:30, 7:45, 8:00, 8:15, 8:30, 8:45 and 9:00 - with an interval of 15 minutes. There are 13 options for the morning shift start time, ranging from 5:00 to 8:00. The total morning shift start time possibilities considering of range of start time and range of duration can be calculated; $9 \times 13 = 117$ possibilities. With 117 potentials for the day, afternoon, and night shift, the total possibilities in a single day amount to 187,388,721. What if it gets extended to 7 days? Even the most powerful machines today would find it impossible to crunch the number.

Hence, predefined numbers of candidates of schedule are generated by randomly picked shifts. If a seven day's schedule is to be generated; 7 times for the morning shift, day shift, afternoon shift and night shift is picked randomly to form the schedule, as shown in Figure 5.2.

The shift start time and duration can be repeated, but the schedule generated is guaranteed to be unique in the initial pool. Schedule candidates like in Figure 3.10 will be generated to fill in the initial pool based on the predefined number of candidates.

Shift Schedule	Fitness Value	Sort based on fitness value descending	Shift Schedule	Fitness Value	Best
SS1	264		SS1	264	
SS2	771		SS4	304	
SS3	646		SS7	356	Good
SS4	304		SS8	406	
SS5	686		SS9	536	
SS6	665		SS10	592	
SS7	356		SS3	646	Bad
SS8	406		SS6	665	
SS9	536		SS5	686	
SS10	592		SS2	771	

Figure 3.10: Sorting of the Candidate Based on Fitness Values

Once the initial pool is filled up with schedule candidates that are generated randomly, the fitness value calculation for each shift schedule is determined. Based on this fitness value, the schedule candidates in the initial pool are categorised as best, good, and bad based on predefined values which can be obtained from multiple runs by observing the performance.

Once the categorisation process is done, the initial pool is ready for the breeding process. From the initial pool, all categorisation of shift schedule candidates either from best, good, and bad are picked to be used as a population or "breeding" pool. The process is designed in such to ensure that all shift schedule candidates have a chance to be selected.

3.4.5.4 Selection

During "breeding" or generation of a new population, a portion of a predefined population of shift schedule candidates is selected to "breed". The chosen shift schedule candidates are the parents or chromosomes that will be used to "breed" so that their genes can be passed down to the next generation. There are three methods implemented;

- 1) Random - parent(s) are randomly selected from the population without taking into consideration fitness function value.
- 2) Best - the shift schedule candidate with the best fitness function will be selected as a parent(s).
- 3) Top - parent(s) are randomly selected from the best population portion, which is defined with a range of best fitness function values; this method is also known as truncation selection.

Parent or shift schedule candidate selection is crucial to produce good quality next-generation shift schedule candidate resources could ultimately lead to a good shift schedule being selected. The best and top selection methods are used to make sure that the best genes are being selected for "breeding" as this will give a better chance of producing a better shift schedule than the previous population. The random selection method is implemented to introduce diversity into the parent selected for "breeding". A less desirable parent or shift schedule candidate with low fitness function value can be the catalyst to produce a good shift schedule even though itself is not a desirable candidate. The pool of parents with good fitness function value might suffer premature convergence or stagnation of evolution. The premature convergence is due to pooling of parents with superior fitness which causes the situation where parent or shift schedule candidate use for "breeding" unable to produce

offspring superior to the parent itself, hence maintaining functional diversity is crucial in ensuring the success of GA.

3.4.5.5 Crossover

During the "breeding" process or generation of a new population, two parents or shift schedule candidates are selected to participate in crossover or swapping of genes to produce new offspring and hopefully fitter than the parents. The selection of parents is implemented with three different types of selection methods, as explained in the Selection section. The three implementations are shown in Table 3.6.

Table 3.6: Selection Method

Selection Method	Parent 1	Parent 2
Selection 1	Random	Random
Selection 2	Best	Random
Selection 3	Top	Random

After the parents (shift schedule candidates) have been selected based on the table shown above, the days to swap between both parents are randomly generated. Referring to Figure 3.11 example, Parent 1's Day 6 will be swap/crossover with Parent's Day 2.

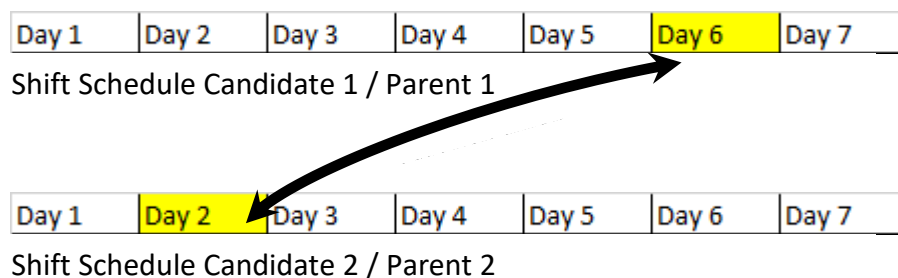


Figure 3.11: Crossover Operation Between Days

Once the days to swapped are determined, the shift to be swapped between both days is also randomly generated, as shown with an example in Figure 3.12.

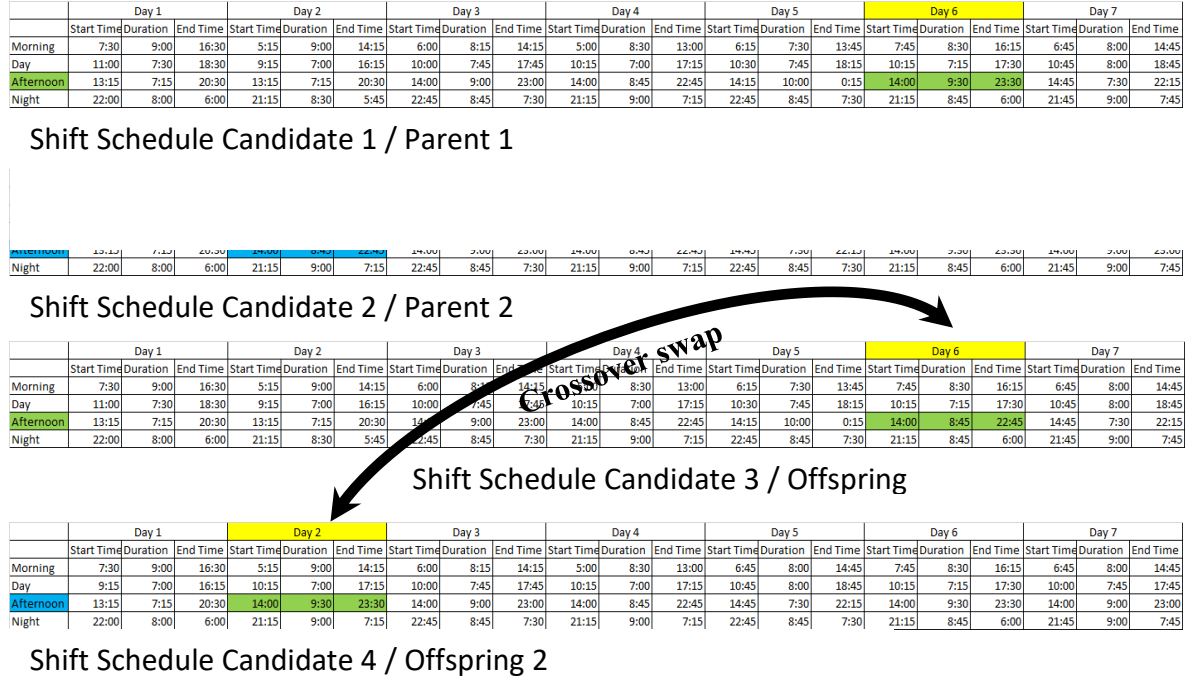


Figure 3.12: Crossover Operation Between Shifts

The crossover is fine-tuned by adding an adaptive function to avoid premature convergence or stagnation of evolution. The function will check the best fitness function value of the new generation population. If the cost remains the same for the next few cycles (number of cycles is predefined) of the generation, then the number of days to swap will be increased by one up to a maximum of three days as the schedule only consists of seven days. This adaptive increment of several days to crossover, increase the genes exchange between parents when stagnation of evolution is detected. So instead of Day 6 crossover with Day 2, it could be Day 6 and seven crossovers with Day 2 and Day 4. If the best function value of the new generation population improved the number of days to swap will be restored to the

original value of one. The purpose of the adaptive crossover is to maintain the diversity to encourage the improvement of shift schedule candidate or offspring fitness.

3.4.5.6 Mutation

After crossover, a chromosome or shift schedule candidate is selected to mutate. The mutation is the genetic process of swapping gene within the chromosome itself. The purpose of mutation is to maintain the genetic diversity of the population from one generation to the next generation.

The mutation process starts with a selection of the chromosome or shifts schedule candidate either by random, best, or top selection methods explained in the earlier section. After the chromosome or shift schedule candidates have been selected, the days to swap in the same shift schedule candidate are randomly generated. Referring to Figure 3.13 example, Day 3 will be swapped with Day 5.

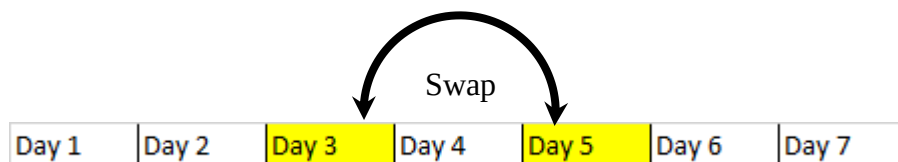


Figure 3.13: Swapping Operation

The shift to swap for both days is also determined by random function, as shown in Figure 3.14.

	Day 1			Day 2			Day 3			Day 4			Day 5			Day 6			Day 7		
	Start Time	Duration	End Time	Start Time	Duration	End Time	Start Time	Duration	End Time	Start Time	Duration	End Time	Start Time	Duration	End Time	Start Time	Duration	End Time	Start Time	Duration	End Time
Morning	7:30	9:00	16:30	5:15	9:00	14:15	6:00	8:15	14:15	5:00	8:30	13:00	6:15	7:30	13:45	7:45	8:30	16:15	6:45	8:00	14:45
Day	11:00	7:30	18:30	9:15	7:00	16:15	10:00	7:45	17:45	10:15	7:00	17:15	10:30	7:45	18:15	10:15	7:15	17:30	10:45	8:00	18:45
Afternoon	13:15	7:15	20:30	13:15	7:15	20:30	14:00	9:00	23:00	14:00	8:45	22:45	14:15	10:00	0:15	14:00	8:45	22:45	14:45	7:30	22:15
Night	22:00	8:00	6:00	21:15	8:30	5:45	22:45	8:45		21:15	9:00	7:15	22:45	8:45	7:30	21:15	8:45	6:00	21:45	9:00	7:45

Original Shift Schedule Candidate

Swap

	Day 1			Day 2			Day 3			Day 4			Day 5			Day 6			Day 7		
	Start Time	Duration	End Time	Start Time	Duration	End Time	Start Time	Duration	End Time	Start Time	Duration	End Time	Start Time	Duration	End Time	Start Time	Duration	End Time	Start Time	Duration	End Time
Morning	7:30	9:00	16:30	5:15	9:00	14:15	6:00	8:15	14:15	5:00	8:30	13:00	6:15	7:30	13:45	7:45	8:30	16:15	6:45	8:00	14:45
Day	11:00	7:30	18:30	9:15	7:00	16:15	10:00	7:45	17:45	10:15	7:00	17:15	10:30	7:45	18:15	10:15	7:15	17:30	10:45	8:00	18:45
Afternoon	13:15	7:15	20:30	13:15	7:15	20:30	14:15	10:00	0:15	14:00	8:45	22:45	14:00	9:00	23:00	14:00	8:45	22:45	14:45	7:30	22:15
Night	22:00	8:00	6:00	21:15	8:30	5:45	22:45	8:45	7:30	21:15	9:00	7:15	22:45	8:45	7:30	21:15	8:45	6:00	21:45	9:00	7:45

Mutated Shift Schedule Candidate

Figure 3.14: Swapping Operation Between Days

The randomness in swapping of both days and shifts are done to encourage diversity.

Diversity of the potential solutions is essential in order to increase the possibility of finding the right solution.

3.4.5.7 Evaluation

Crossover and mutation will produce new chromosomes or shift schedule candidates to be added to the existing pool of a population. Fitness function calculation is applied for new shift schedule candidates that have been added so that ranking and sorting can be done on the pool of population, as shown in Figure 3.15.

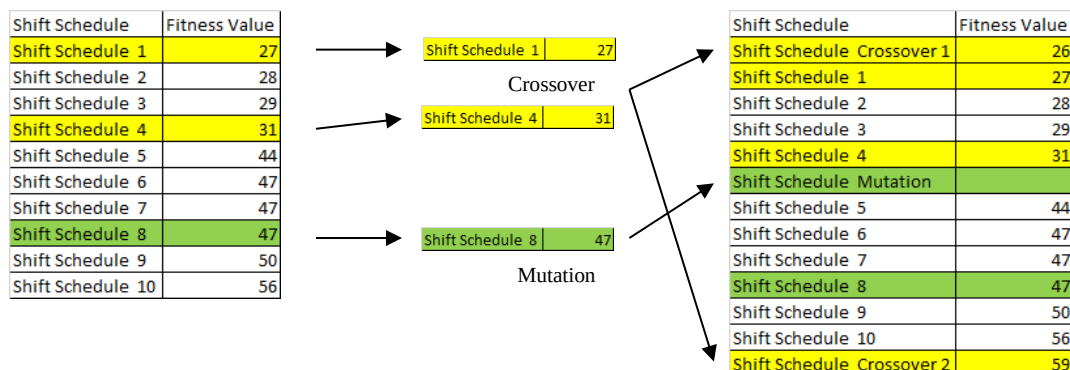


Figure 3.15: Sorting Operations after GA Operations

The weaker chromosomes, or in this case shift schedule candidates with low fitness value, are removed from the pool population, which will be used for next generation "breeding". Referring to Figure 3.15, three new shift schedule candidates produce from crossover and mutation. The new offspring are then added to the current pool of shift schedule candidates or population. The pool is then sorted according to the latest fitness value, and the least fit chromosome (shift schedule candidate with bad fitness value) are removed based on the concept of survival of fittest. The number of shift schedule candidates removed is the same as the number shift schedule candidates generated from crossover and mutation. This new pool of population is then used for the next generation in subsequent GA processing. Elitism is employed to prevent the current fittest from loss due to crossover and mutation. The implementation of this GA prevents the fittest shift schedule candidate from being replaced. During the crossover and mutation, the parents are copied together with the offspring into the current pool population to be evaluated. The GA iteration terminates once the predefined value is reached.

3.4.5.8 Optimisation

Good schedule candidates picked or selected from both randomise and criteria-based selection will be optimised to improve the quality of schedules. The objectives of the optimisation are to reduce the resource/workforce used, overlapped in shift period, and uncover jobs.

- a) First, it tries to optimise the duration within the permitted time window restriction to cover more jobs while avoiding an overlap of shifts.
- b) Next, it tries to adjust the start time to the reduced overlap of shifts while respecting the time window restriction.

- c) The last step is to assign jobs by changing the start time of the previous shifts, even though the reward is for the upcoming shifts.

All the optimisations minimised overlapping with other shifts and stay within the range of parameters defined by the user. The optimisation steps are repeated until there are no changes in the schedule. Once there is no improvement, the candidate for the good schedule is considered optimised and ready to be in the pool for the evaluation of the best schedule.

3.5 Analysis Methodology

The test evaluation is expounded in three distinct sub-sections. This work delves into a comprehensive analysis of the algorithm's complexity and computational time for the test analysis. It also evaluates the algorithm's suitability as an ideal candidate. Moving on to the experimental results, the algorithm's performance is assessed through experiments conducted on both simulated data with varying peak hours and real-world data. In the case of experiments with simulated data, the dataset is generated to mirror real data trends, systematically increasing the number of peak hours throughout the day to test the algorithms' robustness.

3.5.1 Sources of Data

The dataset used for this research consists of real data collected from an airline based in Kuala Lumpur, Malaysia. The general characteristics of the real data only show the workload for 4 days due to missing data at the source are presented in Figure 3.16.

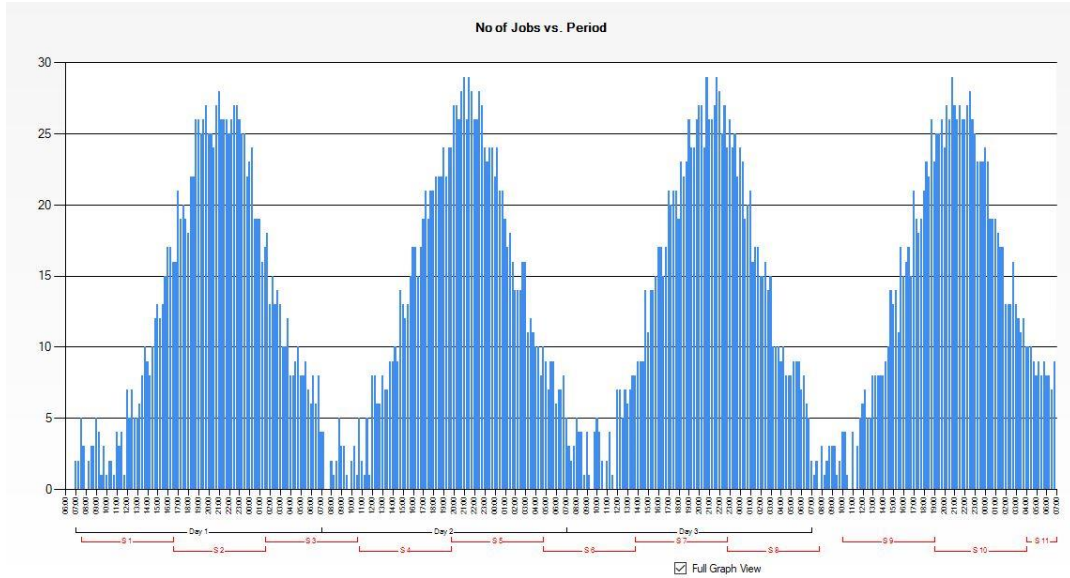


Figure 3.16: Real Dataset of 4 days

For a comprehensive analysis and evaluation, this work considers a wide range of test data which was generated to simulate real data, consisting of multiple peak hours which are randomly generated with the guided range to mimic real-life data. For experimental with simulated data, the duration of the schedule is populated to a full seven days to simulate a one-week shift schedule. Trends Tested:

- **Single Peaks:** The first trend explores scenarios characterized by a single peak in activity, representing typical daily operational patterns.
- **Two Peaks:** The second trend introduces dual peaks, simulating situations with dual high activities period.
- **Three Peaks:** The third trend involves scenarios with three distinct peaks, emulating complex operational cycles.

These trends are visually represented in Figure 3.17 – 3.19, offering a clear insight into the diverse simulated scenarios tested in our analysis."

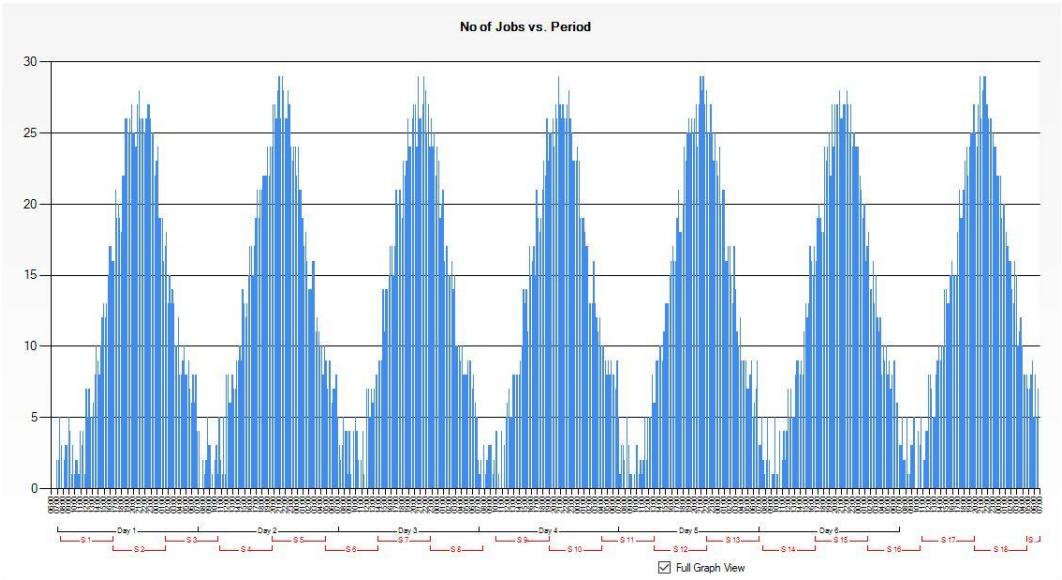


Figure 3.17: Simulated Data with 1 Peak Hour

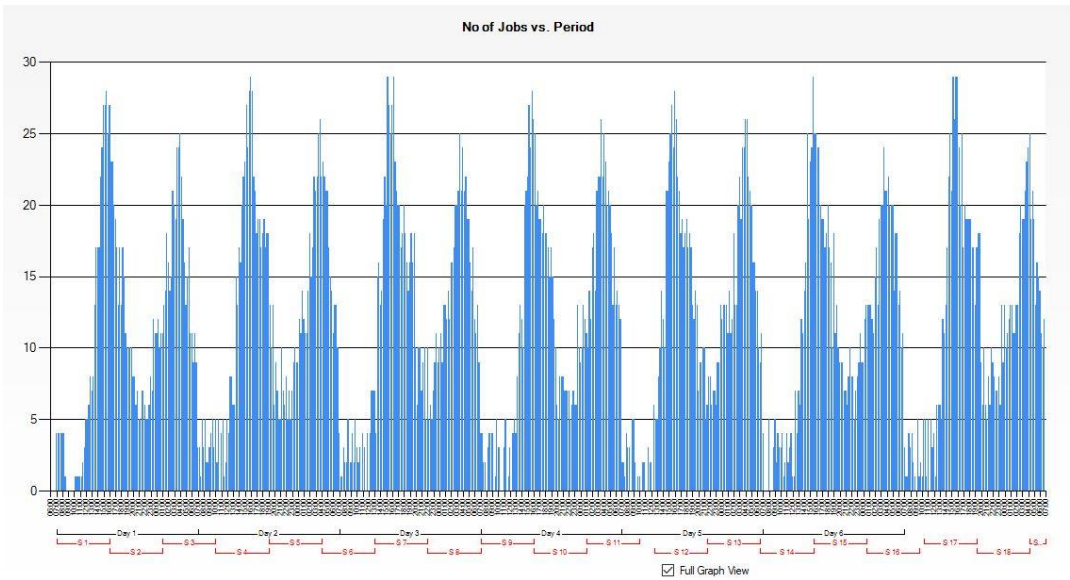


Figure 3.18: Simulated Data with 2 Peak Hour

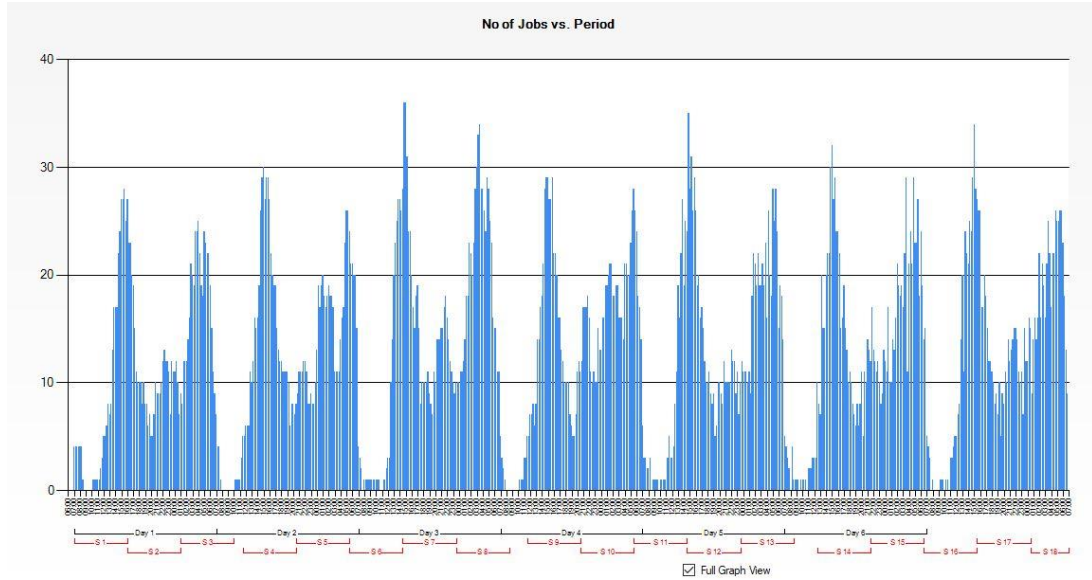


Figure 3.19: Simulated Data with 3 Peak Hour

3.5.2 Performance Analysis

The main complexity of the proposed algorithm comes from the number of possibilities for shift schedule candidates that are available which is huge, that involve RPHA operations based on multiple criteria as described in Section 3.4.4 and multiple operators function with adaptive behaviour of the GA as explained in Section 3.4.5.

In this study, the performance of the proposed algorithm, benchmark algorithm and real-world method will be assess based on the following criteria:

a) Robustness

The ability to withstand or overcome adverse conditions or rigorous testing with real data and simulated data.

b) Feasibility

The ability of the proposed algorithm to solve the problems by reducing the number

of workforces required to complete the shift schedule, in this context the less the better.

The proposed algorithms will be implemented and tested in the same setup and both algorithm results will be compared with the real-world shift schedule.

3.6 Sensitivity Analysis

In order to better understand the impact of input variables on the final result, multiple scenarios testing will be set up for each algorithm. Aside from getting to know the relation between the input variables and output, this will also help with robustness testing. The main aspect of the input variables to be tested for both algorithms will be:

- a) Start time range of each shift.
- b) Duration range of each shift.

The analysis is further extended for each algorithm with:

- i) RPHA implements a simple randomise method to benchmark with the criteria-based method.
- ii) GA implementation of testing with multiple initialisation pool numbers, iteration numbers and different operator selection methods.

The proposed sensitivity analysis testing will be able to assist with identifying of connection between the input variables and output results. It will also lead to better forecasting of computational wastage avoidance, leading to the better algorithm development.

3.7 Conclusion

This chapter presents the proposed methodology for the research. It follows a framework that divides the various phases of a study into three phases. During Phase 1, the models were constructed using criteria such as literature review analysis, methodology selection, and data collection. This study establishes the foundation for the methodology, employing a blend of both fundamental and technical analysis approaches.

In Phase 2, the RPHA model and GA were designed and developed to solve the problem by ensuring a high-quality solution is produced. The models are implemented to accommodate even the most complicated constraints, multiple shifts, inconsistent workforce demand and complex configurable setup. GA methods were used to evaluate the proposed model's robustness and consistency.

CHAPTER 4

RULE PRIORITISATION HEURISTIC APPROACH

4.1 Introduction

Designing schedules with multiple shifts entails numerous challenges; it is governed by a set of rules or operational constraints that induce the complexity in generating an optimum schedule. Shift scheduling algorithm aims to address the challenge of ensuring that all jobs are assigned using minimum resources while preventing wastage. Its secondary goal is to avoid unutilized resource such as idling workforce during the working hours. Aside from that, the algorithm will schedule a shift with a consistent start time pattern throughout the week. This consistency in the start time pattern will ensure worker's adaptability to the schedule and exercise fairness.

In other words, the heuristic method is well-known to be able to deal with the various complex real-world objectives and constraints that cannot be easily solved with a mathematical programming formulation. This behaviour corresponds to heuristic's attribute that entail the capability of performing well with a vast pool of datasets. The heuristic approach demonstrates the potential for producing a reasonable and adequate solution within a short period of computational time when compared to exact methods. The heuristic method can also be adapted to address multiple objectives and highly constrained problems as opposed to meta-heuristic due to its difficulties in operating within a search space where feasible regions are disconnected.

Based on the objectives and problems stated, this research focuses on the heuristic algorithm to produce an efficient shift schedule which covers all assigned jobs with the least

workforce requirement. An optimisation function is added into the proposed algorithm to extend the heuristic capability of finding not only the best schedule with least computing cost, but also to avoid wastage, promote worker-friendly scheduling and exercise fairness.

Based on the literature review, we are inspired to address shift scheduling challenges or what we call shift scheduling design using the heuristic method. The heuristic method was known as a method that has been utilised frequently to tackle scheduling problems which are either too difficult or complicated and impossible to solve using an exact solution approach. A heuristic method is well known to be robust and capable of delivering a great result within a short period, although they may not ultimately produce the best outcome. In our approach, the implementation consists of two heuristic methods with a post-processing constraint-based optimisation method, which is applied to the solutions to improve the quality. Post-processing optimisation ensures that the solution generated is viable because both randomise-pick and criteria-based selection methods are heuristic in nature, which leaves open the possibility that the results might not be satisfactory. In the subsequent subsections, the process of generating the solutions are presented from the user perspective when running the prototype.

4.1.1 Criteria Preparation

The initial step to the solution was to generate an optimum workload prediction using TSH. The shift cannot be implemented based on workload prediction because it will allocate a different workforce at different periods. However, workload prediction can be used as a guide to generate an optimum schedule. User input of shift design criteria was used to list out all possibilities of schedules with all the combinations of start times and durations. The

list of raw schedules, together with the result of workload prediction, formed the input for the shift scheduling algorithm to schedule the shift.

4.1.2 Shift Scheduling

In shift scheduling, a shift is known as time intervals defined by a beginning time, an ending time, and a duration. A one-day shift time comprises four shifts, which are morning, day, afternoon and night. For each of the shift time, the user would be able to choose the earliest start time, latest start time, minimum duration and maximum duration, as illustrated in Figure 4.1.

	Earliest Start Time (EST)		Latest Start Time (LST)		Min Duration		Max Duration	
Morning	05 ▾	00 ▾	08 ▾	00 ▾	07 ▾	00 ▾	09 ▾	00 ▾
Day	09 ▾	00 ▾	11 ▾	00 ▾	07 ▾	00 ▾	08 ▾	00 ▾
Afternoon	13 ▾	00 ▾	15 ▾	00 ▾	07 ▾	00 ▾	10 ▾	00 ▾
Night	21 ▾	00 ▾	23 ▾	00 ▾	08 ▾	00 ▾	09 ▾	00 ▾

Figure 4.1: Defining A Day Shift Time

Two different ranges for the start time, as shown in Table 4.1, namely 2 hours and 4 hours range, were used to analyse the performance of the solution. The start time range refers to the difference between the earliest start time and latest start time. For the 2-hour range of the morning shift, the earliest start time (EST) was 0500, and the latest start time (LST) was 0700.

Table 4.1: Shift Start Time Range Used for Testing

Shift Start Time	2 hours' range (EST – LST)	4 hours' range (EST – LST)
Morning	0500 – 0700	0300 – 0700
Day	1000 – 1200	0800 – 1200
Afternoon	1600 – 1800	1400 – 1800
Night	2100 – 2300	1900 - 2300

The logic of having the 2-hour range is to simulate the loosely restricted start time range so that the shift is more adaptable to employees. The 4-hour range was formed to test whether the relaxation of start time range would yield a schedule that is more cost-effective and with least wastage, which should be the case in theory.

4.2 Computational Results

In this proposed heuristic method of shift scheduling, users are allowed to define shift restrictions parameters such as shift earlier start time, shift latest start time, shift minimum duration, and shift maximum duration. The configurations allowed for four shifts of restrictions parameters per day assignment. The configurations defined will be used as input to generate all the possibilities of the shifts, with the subsequent process explained in the following sections.

The schedule generated by the proposed randomise pick algorithm requires 310 units of the workforce for four days, and the overlapped between shift is 7 hours and 45 minutes, as shown in Table 4.2.

Table 4.2: Randomise Pick Shift Schedule

Day 1	Day 2	Day 3	Day 4
0700-1300	0700-1300	0645-1245	0700-1300
1200-1800	1200-1800	1200-1800	1200-1800
1800-2400	1800-2400	1800-2400	1800-2400
2300-0700	2300-0645	2300-0700	2300-0900

For the criteria-based selection, the workforce number calculated is 316, with an overlapped duration of 8 hours, as shown in Table 4.3. The overlapped shift is between

- 0700-1300 and 1200-1800: 1 hour
- 1800-2400 and 2300-0700: 1 hour

So in total for 4 days will be $((1 + 1) \text{ hour} \times 4 \text{ days}) = 8 \text{ hours}$.

Table 4.3: Criteria Based Selection

Day 1	Day 2	Day 3	Day 4
0700-1300	0700-1300	0700-1300	0700-1300
1200-1800	1200-1800	1200-1800	1200-1800
1800-2400	1800-2400	1800-2400	1800-2400
2300-0700	2300-0700	2300-0700	2300-0900

Our test revealed that the start time range in this scenario did not impact the schedule selected. The randomise pick algorithm performed slightly better with a reduction in the number of workforce required and overlapped shift time. However, the criteria-based

selection did not perform as well as the randomised pick algorithm as the former algorithm needed to adhere to a stricter criteria limit. The highlighted cell indicates the difference between the two schedules shifts from different solutions. The CBS algorithm consistently picked the same start time, while the RP algorithm proved true to its nature by picking without restriction, which led to the outlier in the shift start time. All the algorithms executed the parameters from Table 4.1 to obtain the number of workforces required. Lower bound value is obtained with TSH which is the most lowest value possible without taking into any restriction, this value is use as benchmarking. Results obtained from the proposed algorithms with the TSH and manual shift schedule are illustrated in Figure 4.2.

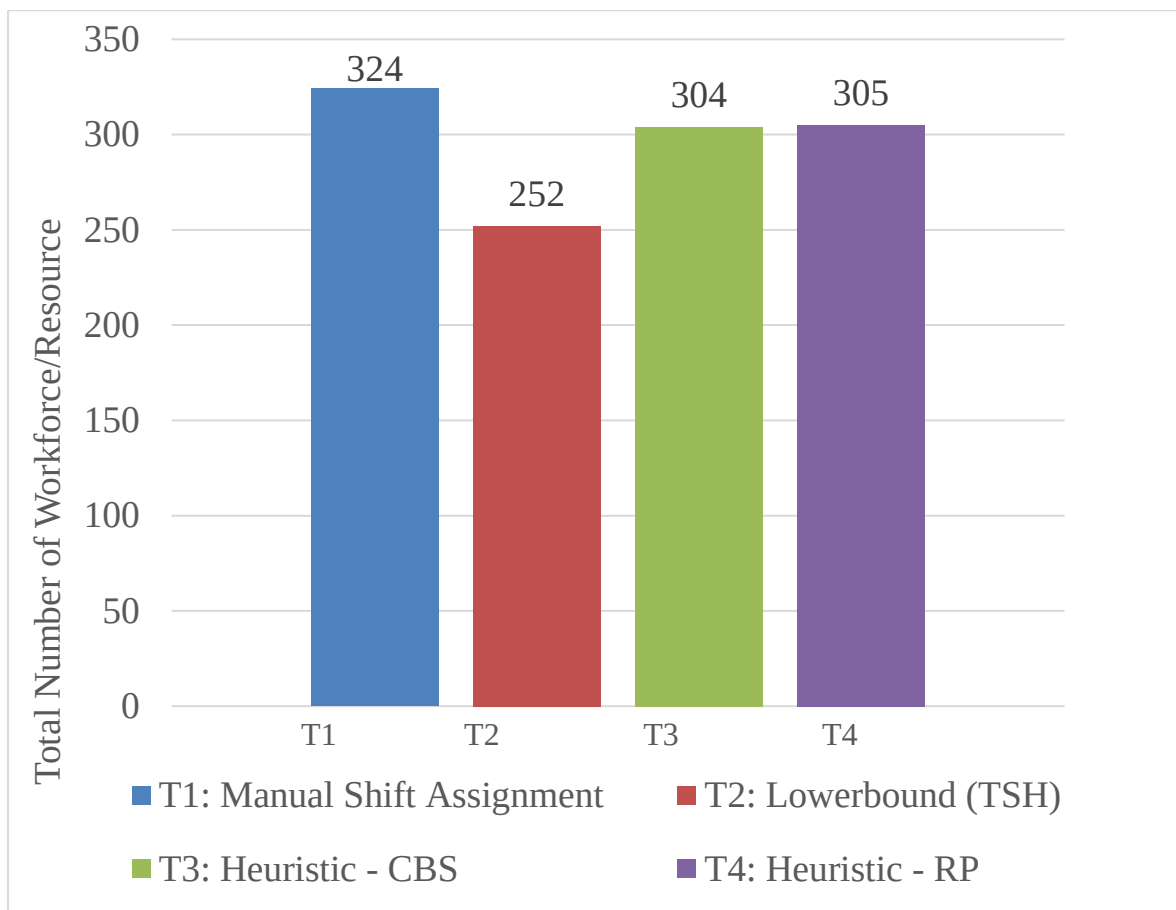


Figure 4.2: Comparison Between Manual, Lower Bound and Proposed Heuristic Algorithms Workforce Requirements

In summary, the workforce required by the proposed shift scheduling algorithms was lower than the manually assigned shifts, namely a reduction in 5.25% to 6.17%. If 5.25% of saving is translated into the current monetary gain, it would be equivalent to USD25K of saving per annum. The saving is based on the calculation of current airline in Malaysia which operation cost of operation staff including overtime is around USD40K a month, the information is based on interview done with airline employee.

4.3 Results Analysis

The results presented in this section were drawn based on the data of the real problem found in an international airport in Malaysia. The real problem datasets contained 1023 jobs; each job has the following attributes: a job number, release time, deadline and processing time. The data sets comprised a collection of the real number of jobs for the first four days of the week. Due to unforeseen circumstances, the datasets were incomplete as the records for the last three days of the week were missing. Nevertheless, the algorithm managed to simulate and produce the desired result, even with only four days of datasets.

4.3.1 Performance Analysis

In this case study, the prototype was developed using C# from scratch. All experiments were carried out on a Toshiba laptop with Intel® CORE™ i5 2.4GHz and 8GB memory. In addition to testing on the 4- day datasets, which comprised 1023 jobs, testing was also done with the 1-day datasets which consisted of 234 jobs. All experimental tests generate results in less than twenty seconds, as shown in Table 4.4.

Table 4.4: Computational Time with Different Datasets Size

Schedule	Average (s)
1 day	4.72
4 days	19.47

The results indicate that RPH can generate outcome rapidly, even running on a vast data set which shows high performance that the heuristic approach brings. The chart from Figure 4.3 shows that the proposed RPH algorithm projected a start time (indicate by ellipsis in chart) pattern that was more or less consistent throughout the week, which indicates that the resources were being utilised efficiently according to the workload distribution.

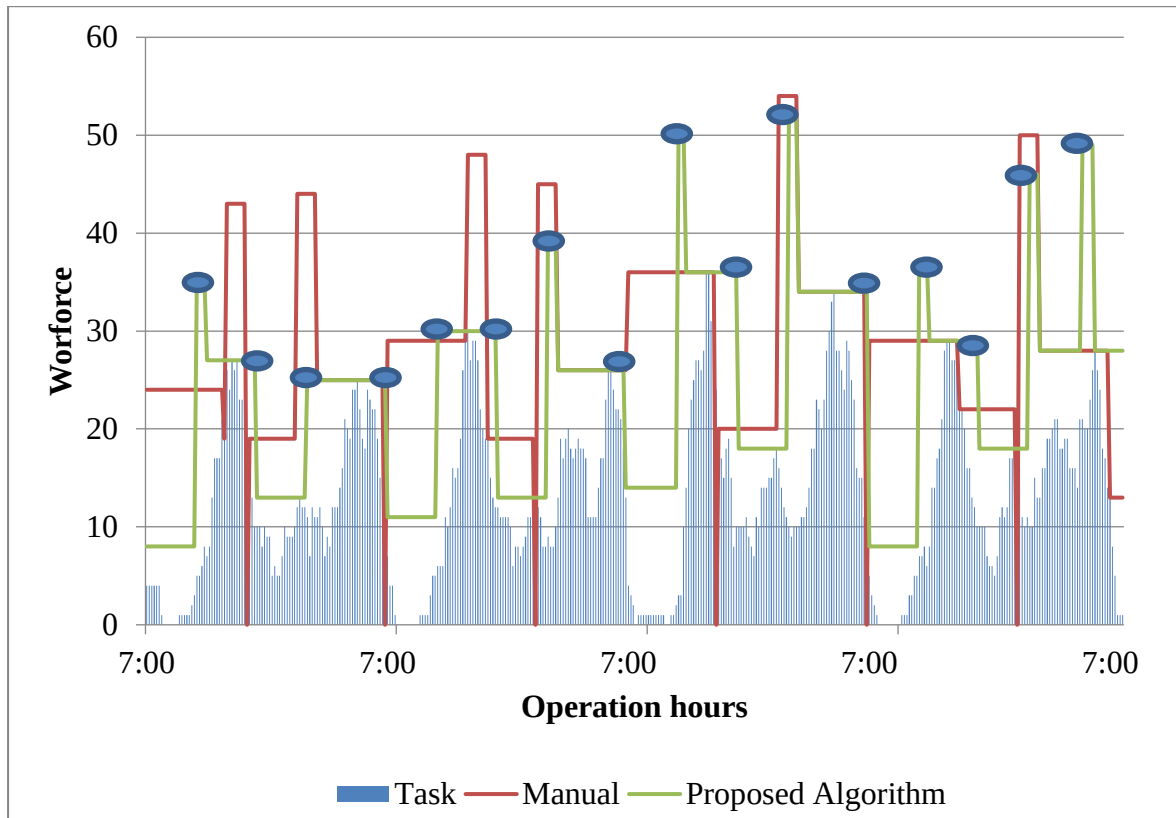


Figure 4.3: Comparison of the Workforce Assigned for Shifts Between Proposed Algorithm Vs. Manual Assignment

The peak exhibited in the chart indicates an overlapping of shift, which implied the wastage of resources. The chart showed that the proposed algorithm produced fewer peaks as compared to manual shift scheduling.

4.3.2 Sensitivity Analysis

The same test on datasets was used to run the algorithm to examine the impact of the shift start time and duration (which was defined as shown in Figure 4.1 and Table 4.1) by disregarding the constraints individually. The sensitivity results are summarised in Figure 4.4 to Figure 4.6.

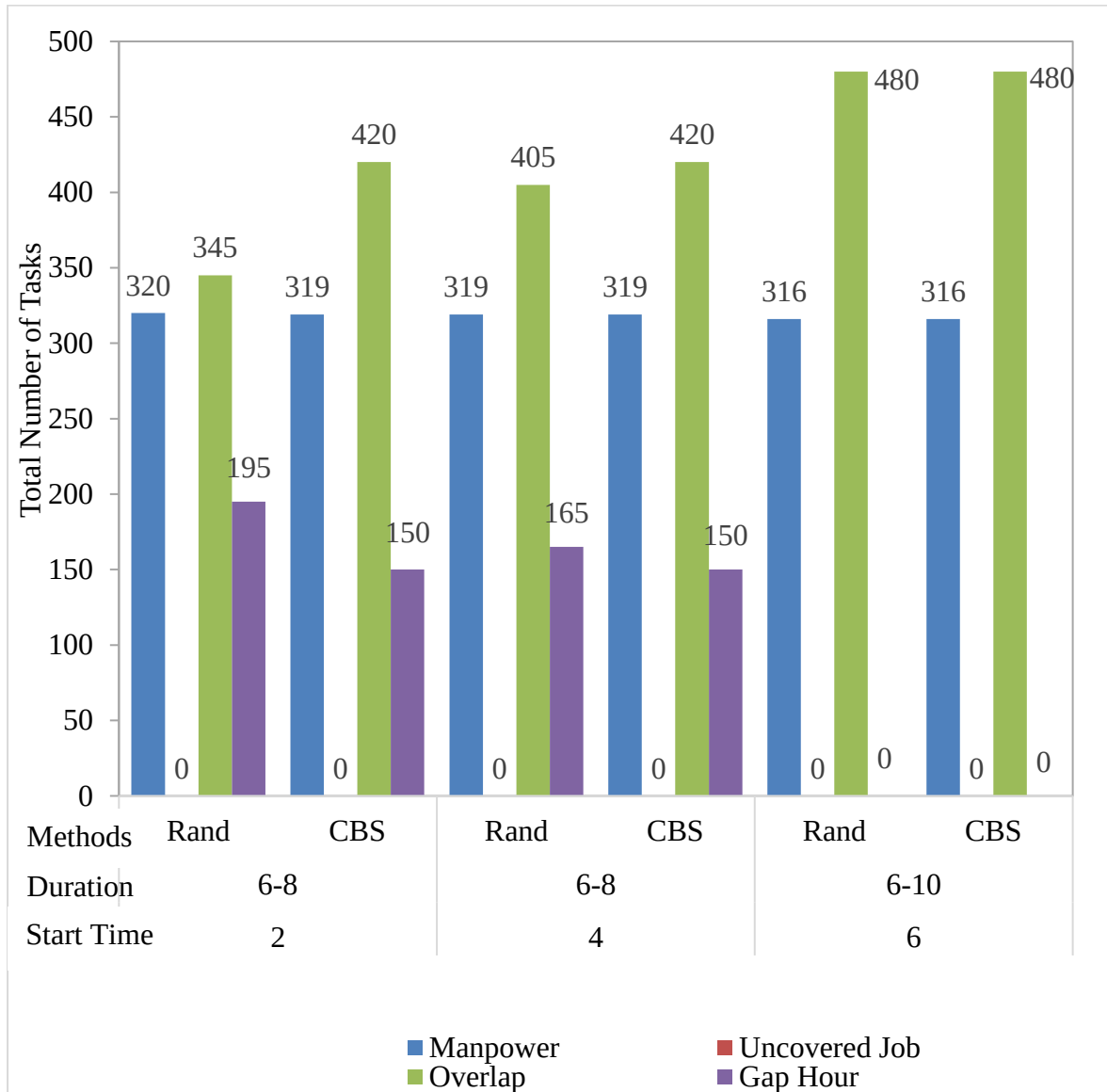


Figure 4.4: The Relation Between Resource Values Vs. A Different Range of Shift Start Time and Duration

The same datasets were run using the same algorithm but with a different range of start time and shift duration. The longer the range and the duration, the less the number of workforce/resources required to complete all the jobs, as shown in Figure 4.4. The workforce reduced from 320 to 316 as the range for start time and shift duration widens. The reduction in workforce is expected as the relaxation of constraints provided the possibility of designing

a better shift. The wider range of the constraints, especially shift duration, also caused overlapping of shifts. The overlapping of shifts was due to the start time, which was a restricted constraint defined at the input stage. The difference between the start time also reduced when the constraints were relaxed, namely the value dropped from 195 to 150, and eventually to zero, which indicates all the shifts have the same start time. CBS is more likely to generate overlapping shifts than a Rand approach. This is because a criteria-based approach relies on minimizing the workforce and covering jobs, it may result in longer shifts that overlap with others. On the other hand, Rand approach generates more varied shift schedules, which could potentially reduce the degree of overlap. Besides showing workforce reduction when compared to the manual process, both algorithms also can reduce the overlapping between shifts if the appropriate duration range was defined. The Gap Hour drops to zero due the Start Time range of 6 hours, allowing a higher possibility of a consistent shift start. The CBS algorithm showed that it has the advantage of assigning the shift start time range in the consistency matter, which is a trait desired by employees.

Next, the same test was repeated, but with a fixed shift range duration of 5, 6 and 8 hours. The difference was the start time range, where three ranges of start time were tested. The trend indicates that the range of the start time did not impact both uncovered jobs and overlap of shift values. Instead, the duration of the shift played an important role here. As shown in Figure 4.4, the longer the shift duration, the higher the value in the overlapping of shifts. However, it was the opposite for uncovered jobs, where the trend shows that the shorter the shift duration, the higher the value of uncovered jobs, and this made perfect sense. Shorter shifts could be designed, but it will not be able to cope with the available jobs, as shown in Figure 4.5. In summary, both Rand/RP and CBS required almost the same number of workforce is due to CBS is also utilising the Rand/RP in the earlier stage which explore a

broader range of possible solutions and the input solutions are almost optimize hence there the improvement that could be achieved is minimum.

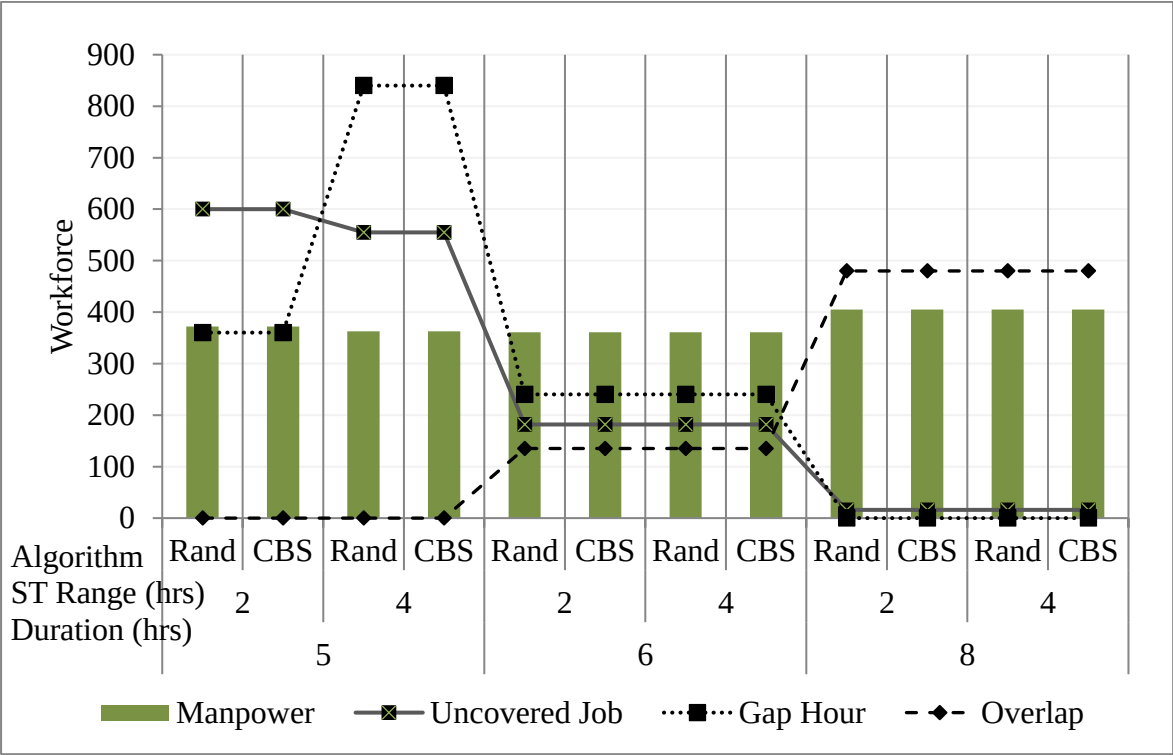


Figure 4.5: The Relation Between Uncovered Jobs and Overlapped of Shifts Vs. Different Range of Shift Start Time and Duration

Figure 4.6 shows that shift duration is essential in determining the required number of workforce. Long duration would cause overlapping of shifts, as shown in this test where 8 hours is defined as the shift duration, with four shifts per day. The overly short duration of shifts would also have the same impact, but in this instance, the value is not as bad as when the shift duration is extended. An optimum number of the workforce was achieved when the shift duration was six hours. This is logical as four shifts with 6-hour shift duration would be able to cover the whole day's jobs.

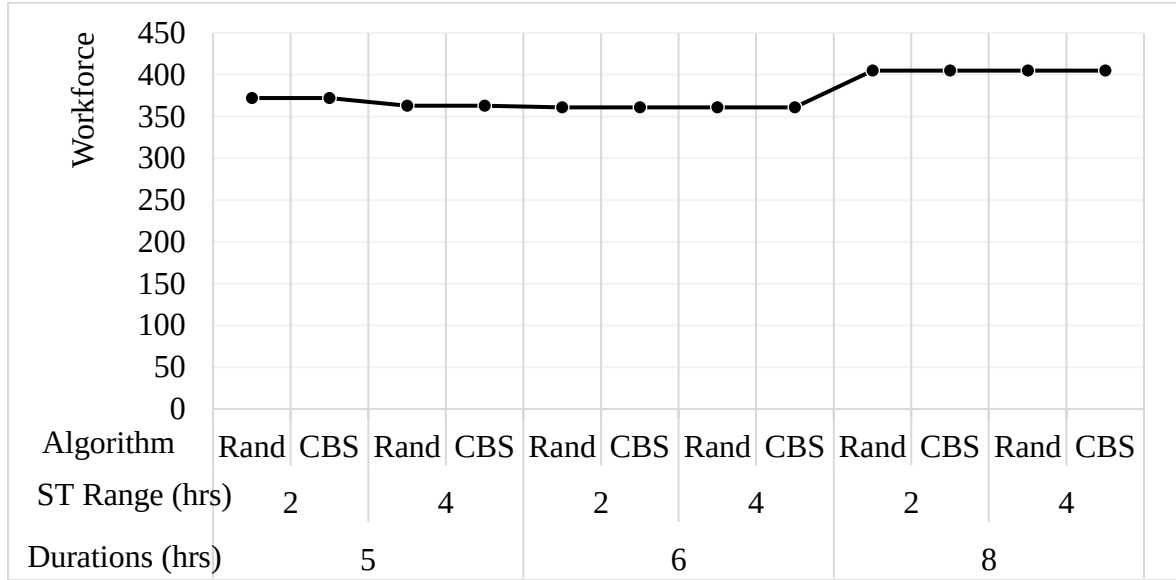


Figure 4.6: The Relation Between Workforce Vs. Different Range of Shift Start Time and Duration

The test results from Figure 4.6 shows that shift start time and shift duration play a critical role in determining how much workforce is required to complete the designated jobs.

4.4 Discussion

We used real-world scenarios provided by an industrial partner, as mentioned in the earlier section. The dataset is composed of 4-day instances for a total of 1023 job instances. We also tested the simulated dataset comprising the original dataset with extended random generated data which was randomly generated with the guided range to mimic real-life data as mentioned in Section 3.5.1. We conducted multiple experiments to evaluate the effectiveness of different components of the proposed algorithm and its performance. In this section, we present the discussion of the results generated by the solution applied using our approach algorithms. Results of the comparisons are shown in Figure 4.7 and Table 4.5 to Table 4.6.

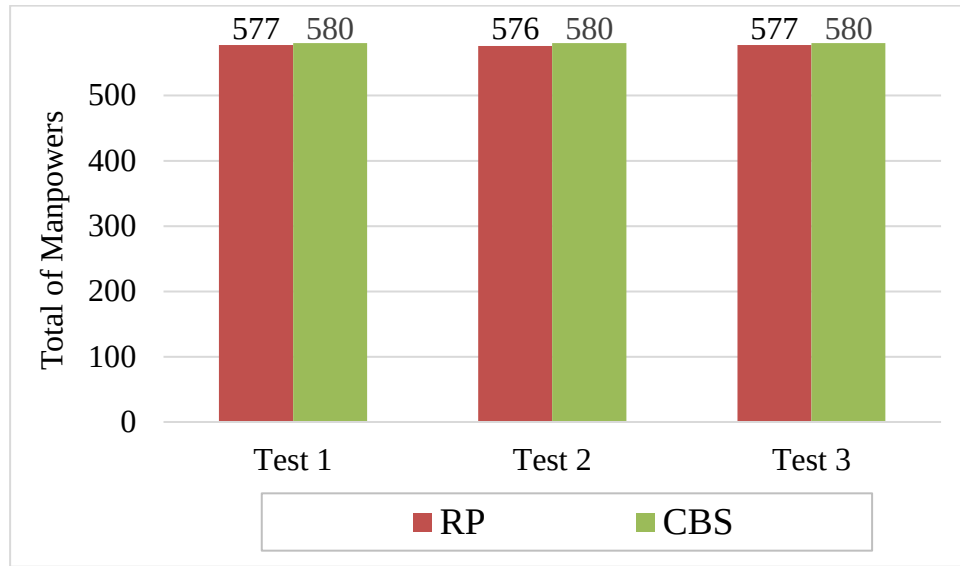


Figure 4.7: Comparison of Workforce Utilisation

The results in Figure 4.7 reveal that reasonable quality solutions can be obtained with both methods. Additionally, the results also show that the RP algorithm produced slightly better results, namely up to 0.69% lesser workforce to complete the same number of jobs. RP algorithm introduce an element of randomness into the optimization process, which can help to avoid getting stuck in local optima and allow for more comprehensive exploration of the search space. This can be particularly useful in complex, high-dimensional problems where it is difficult to determine the optimal solution based on predefined criteria. Aside from the workforce, both methods also managed to cover all jobs given in all three tests.

RP algorithm also generated the schedule with the highest unutilized resource, such as idling workforce during the working hours, and the results were consistently obtained, as clearly indicated in Table 4.5.

Table 4.5: Overlapped Shift Durations in Hours for RP and CBS

Algorithms	Test 1	Test 2	Test 3
RP	124h	125h	122h
CBS	116h	116h	116h

Lastly, the proposed CBS algorithm demonstrated the ability to assign consistent shift start time, as highlighted in Table 4.6, which would ensure a fair schedule as well as higher adaptability of the employees to the schedule.

Table 4.6: Gap Hour Duration Values for RP and CBS

Algorithms	Test 1	Test 2	Test 3
RP	22h	18h	1h
CBS	2h	2h	2h

Hence, the results have shown that the proposed heuristic algorithms can obtain high-quality results with less computational resources. RP algorithm produced the solution with the most minimum workforce require but at the expense of higher overlapped shift duration and gap hour durations. The higher values for overlapped shift and gap hour durations indicates that in order to achieve a minimum workforce, some overlapping of shifts and flexibility in shift start time need to be sacrificed. CBS algorithm result proved that by allowing more workforce overlapping of shifts and flexibility in shift start time can be reduced.

4.5 Conclusion

In this chapter, an algorithm for designing shift schedules was presented and successfully applied to real airline data. The algorithm employed two methods: randomise pick and criteria-based selection to design the shift schedule. The shift schedule designed by both techniques was optimised using the requirements provided to reduce the use of the resource/workforce, cut down overlaps of shift period, and uncover jobs. The proposed algorithm has demonstrated the ability to generate very high-quality solutions within a short period. In short, the algorithm managed to reduce up to 6.17% in resource utilisation and up to 30% in shift duration overlaps as calculated from graph in Figure 4.3. Furthermore, the proposed algorithms exhibited the capability of solving large problem instances in less than a minute. These results suggest that further improvement can be achieved, especially in the reduction of resources. Further studies may consider exploring and taking advantage of the shift change period to optimise workforce utilisation between both shifts to reduce it. As machine learning has gained popularity in solving various types of problems, this area of study can also be applied to a shift scheduling problem.

CHAPTER 5

GENETIC ALGORITHM APPROACH

5.1 Introduction

In this chapter, we present a genetic algorithm (GA) approach to solve the shift scheduling issues. The solution developed based on the GA approach will be used to generate results to be compared with the results from the rule prioritisation heuristic algorithm from chapter 4.

Due to the increasing popularity of genetic algorithms, a vast amount of research has been published in this area. GA has been successfully applied to solve a variety of optimisation problems that are not well suited for standard optimisation algorithms, including problems in which the objective function is discontinuous, nondifferentiable, stochastic, or highly nonlinear.

The genetic algorithm can address problems of mixed-integer programming, where some components are restricted to be integer-valued. GA are chosen for their well-known robustness and ability to solve large and complex discrete optimisation problems. Hence, the central theme of this work is to balance the feasibility and cost of solutions. GA method is particularly appropriate in the domain, such as timetabling and scheduling; for example, the classic problem for nurse scheduling problem or job shop scheduling. Our focus will be solely on shift scheduling or workforce rotation problem.

In this GA method shift scheduling, users are allowed to define shift restrictions parameters such as shift earlier start time, shift latest start time, shift minimum duration, and shift maximum duration similar to Section 4.2 to generate all the possibilities of the shifts.

The list of shifts generated will be picked randomly by GA to form candidates of schedule for further GA processing. GA configuration is also configurable where all the parameters are modifiable. The modifiable parameters are number of initial pool population, number of pool population in iteration, number of GA iterations to run, crossover methods as mentioned in Section 3.4.5.5 earlier and mutation methods as mention in Section 3.4.5.6. Quality composition categorisation for initial and iteration pool population is definable. For both GA operators of crossover and mutation, the user can pick the selection method of random, best, and top, which the details are in Section 3.4.5.4. The prototype developed also allowed for the fitness value to be definable. All the parameters mention discussed in this section can be viewed in screen capture in Figure 5.9.

The screenshot displays a comprehensive parameter input interface for a Genetic Algorithm (GA) scheduler. It is organized into several functional panels:

- Shift Parameters:** A table for defining shift schedules with columns for shift type (Morning, Day, Afternoon, Night), Earliest Start Time (EST), Latest Start Time (LST), Minimum Duration, and Maximum Duration. Each cell contains a time selection interface.
- Jobs Data Set:** Includes a file path input, a 'Browse' button, a 'Generate' button, and options to validate GA with previous best heuristic and save results to a file.
- Greedy Heuristic:** Features a 'Randomized Method' section with sliders for probabilities (A, B, C, D) and a checkbox for 'Optimized results'.
- Criteria Based Methods:** Contains a checkbox for 'Optimized results' and displays performance metrics such as GA Duration, Heuristic Duration, and start/end times.
- Genetic Algorithm:** Includes a 'Pool' section for population size and composition (Best, Good, Bad) and a 'Cross Over' section for parent selection and crossover rate.
- GA Process Parameters:** Includes sliders for 'Population' and 'Iteration' counts, and a checkbox for 'Optimized results'.
- Mutation:** Features a 'Mutation' section with a 'Parent Selection Mode' dropdown and a 'Mutation' checkbox.
- Fitness:** Includes a 'Fitness' section with sliders for 'Weightage', 'Resource', 'Uncovered Jobs', 'Overlapped Periods', and 'ST Gap'.

Figure 5.1: Prototype User Interface for Parameters Input

5.2 Computational Results

Testing was conducted with the parameters shown in Figure 5.9 using the real data from the airline, which comprise 4 day's schedules because of airline limitation. The parameters in Figure 5.9 were selected because of workforce adaptability of the shift timing,

and acceptable processing duration requires. The test parameter of start time was run with four hours of flexibility and flexibility in shift duration was set to any duration between six and ten hours. The initial pool population is set at 5000, which means at the beginning there are already 5000 schedules with the shifts generated randomly. These 5000 shift schedules will be used to produce new shift schedules or, in GA term, chromosomes. Each iteration pool population is set at 1000, wherein every iteration, the 1000 shift schedules will be evaluated and replaced with a better shift schedule if available. The iteration for the GA is set to 5000 times. The pool population composition for both initial and iteration are set to the first 50% as the best chromosome, the next 30% as good chromosome and the rest as a bad chromosome. The values have been obtained from multiple tests and known to give optimum diversify population which provided good solution. Both GA operators utilise random selection with 100 pairs of chromosomes being used in crossover operation. The generated shift schedule for four days using GA is shown in Table 5.1.

Table 5.1: GA shift schedule

Day 1	Day 2	Day 3	Day 4
0700-1300	0700-1300	0645-1245	0545-1145
1200-1800	1200-1800	1200-1800	1200-1800
1800-2400	1800-2400	1800-2400	1800-2400
2300-0700	2300-0645	2300-0645	2300-0900

Table 5.2 shows the values of each essential cost factor element calculated from the shift schedule generated.

Table 5.2: GA Shift Schedule Costs

Workforce (person hour)	307
Uncovered Jobs	0
Overlapped Duration (minutes)	390
Gap Hour (minutes)	1575

Preliminary testing using the same data from Section 4.3 with basic parameters revealed that GA method excels in producing shift schedule which uses far less workforce (person hour) when compared to manual operation, this is shown in Figure 5.2.

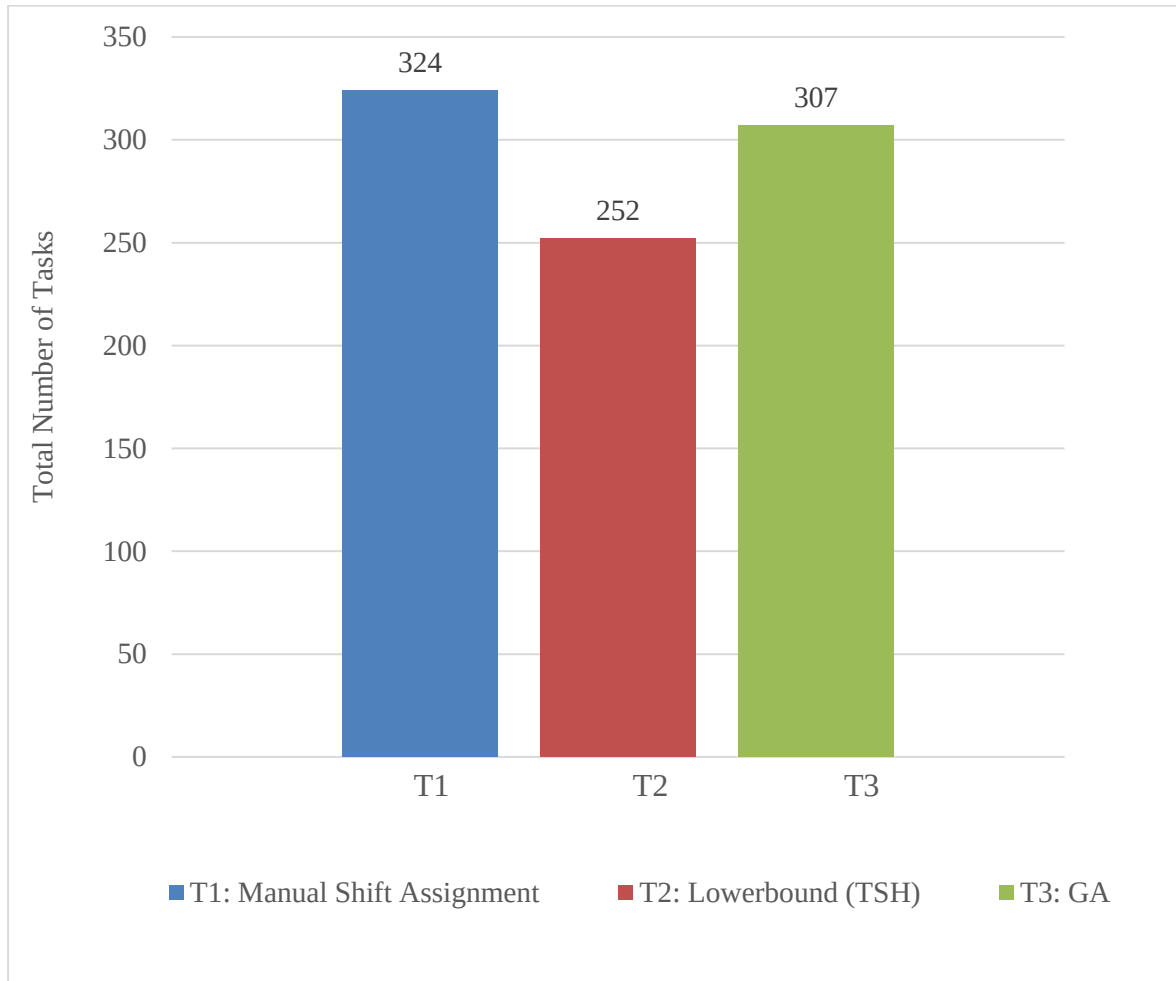


Figure 5.2: Workforce Requirements

The T2 bar graph figure is the determined value where all the jobs in the schedule will have the exact number of staff doing the jobs available. That situation won't work in reality because the human workforce requires to be paid for their downtime, not just when there's work available. For example, there is a period between two jobs/tasks where there is no job/task. It will not be acceptable that during that period the staff will not be paid for the standby time. The T2 graph, which is demonstrated in Figure 5.2, is the perfect workforce number any organization could hope for, and should be utilised as the minimum value for comparison. Manual shift assignment currently in practice uses 324 workforces, which is 72

person-hours more than the lower-bound value for four days, for a month that will require 540 person-hours, which is unutilized. GA method utilised 307, which is 17 person-hours less than manual shift assignment method, in percentage that is a saving of 5.24% or 127.5 person-hours per month. The savings of 127.5 person-hours which is almost equivalent to hiring of one person, which possibly can be turned to overtime instead of paying for one more headcount.

5.3 Results Analysis

In order to evaluate the proposed GA, numerous tests are performed, and results obtained are analysed from a different perspective.

5.3.1 Performance Analysis

A four-day data set based on an in-flight caterer from Malaysia used in Section 4.3 was used to study and for evaluation of the performance of the proposed solution. The GA was coded in C#, and all experiments were carried out on a Toshiba Intel Core™ i5 2.4GHz laptop with 8GB of RAM running on SSD. Computational testing in the subsequent sections with different parameter settings are given to demonstrate the robustness and efficiency of the proposed algorithm performance.

5.3.1.1 Analysis of Shift Pool Size and GA Iteration

We ran a variation of tests to study the relation of various parameters in the GA process with the number workforce and the duration of processing time required in obtaining those results. The test scenarios for Figure 5.3 as follows:

- a) *Shift Start Range* or shift start time range. Two and four hours start time range are for the test because of workforce adaptability. If a six hours start time range is to be

included, there is a potential of significant variation of shift start time, which will not be acceptable and out of norms. The two hours range is from 0500 to 0700 and the four hours range is from 0400 till 0700.

- b) *Shift Duration* or shift duration range. Shift duration of between 6 to 8 hours and 6 to 10 hours are used, this is as per the current practice of the workforce in our studies.
- c) *Init Pool* or initial pool population size. The initial pool population size will be selected from a) and b) based on the method discussed earlier in Section 5.2.4.

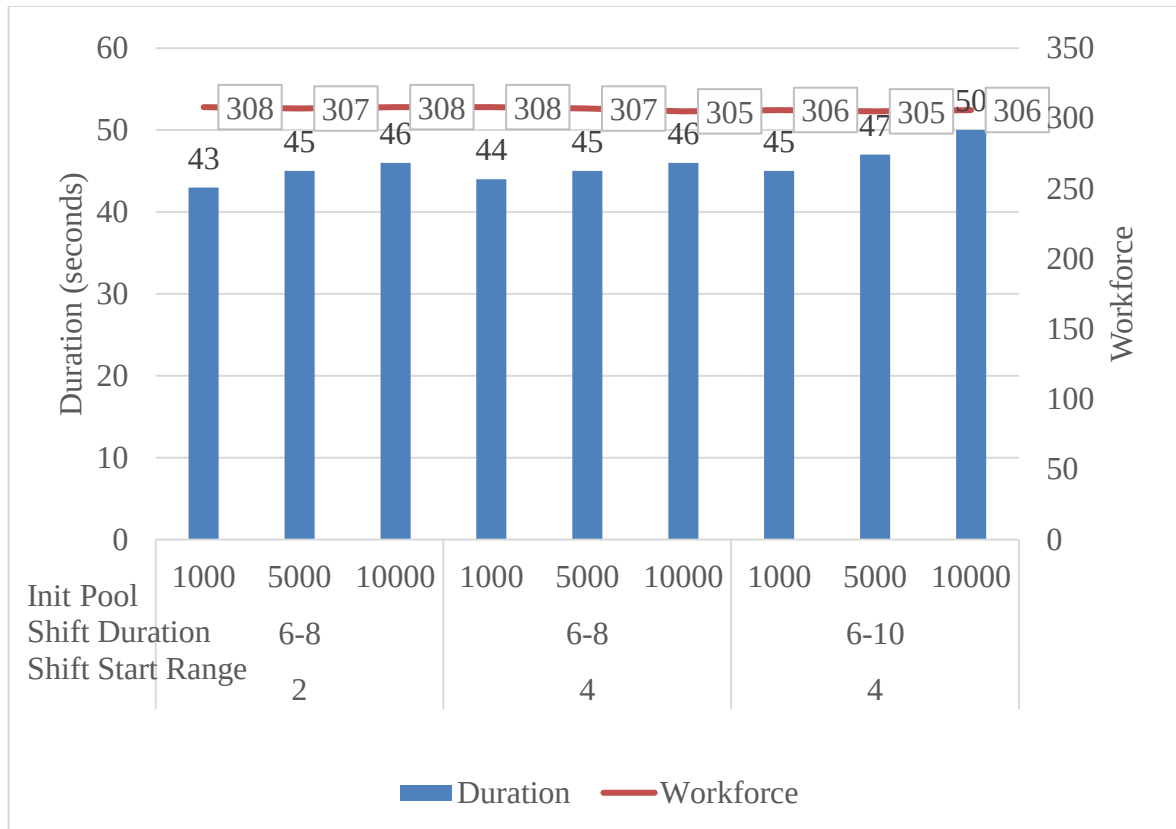


Figure 5.3: Tests on Various Shift Duration and Shift Start Time Range

The test results shown in Figure 5.3 indicate that a bigger range of shift start time and shift duration does provide a better quality shift schedule, which has the lowest number of workforces required. The workforce values generated from the small range are between 307

to 308 but improve to 305 to 306 on the longer range. The processing time also increases with a bigger range and also bigger initial pool population from 43 to 46 seconds to 45 to 50 seconds, an increase for 2-4 seconds, which is acceptable. The following subsequent tests will be conducted with a shift start time of 4 hours (0400 to 0700), shift durations of 6 till 8 hours and initialise pool population size of 5000. These parameters are selected because of the optimum performance in obtaining results. In summary, the improvement that is achievable is about $0.97\% = ((308-305) / 308 \times 100)$ which could be translated USD4656 per annum based on USD40K monthly operation of an airline in Malaysia.

In each GA iteration, a pool of chromosomes or, in this case, shift schedule candidates are selected. On the first iteration, a pool of chromosomes will be selected from the initial pool population, and this pool will be processed by GA operators to become the new pool. The new pool will then be used to select chromosomes for the next iteration; this continues till GA stops once the stop condition is fulfilled. In Figure 5.4, the iteration pool size tested is 500, 1000 and 3000.

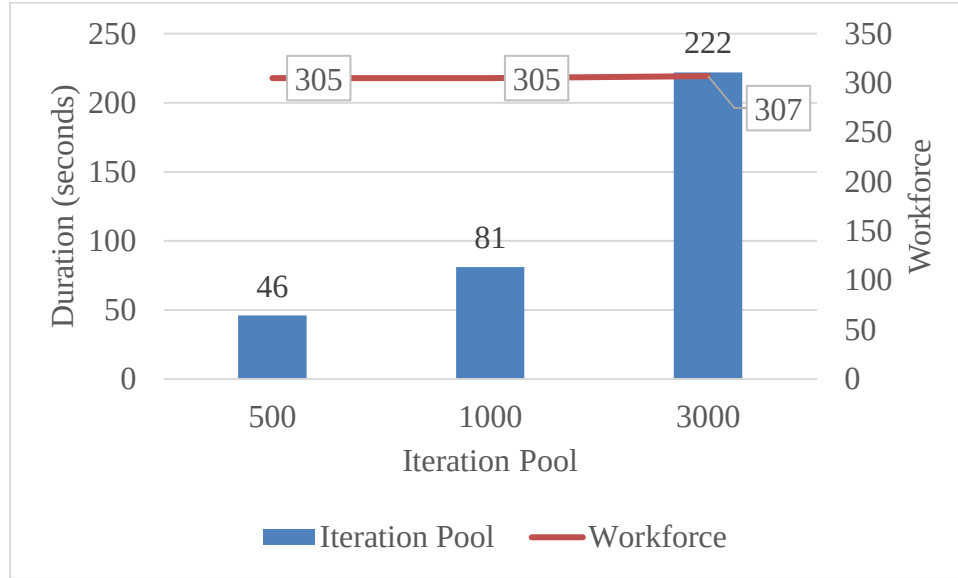


Figure 5.4: Tests on Various Iteration Pool Size

The maximum pool population is 3000 to adhere to the initial pool population size of 5000 from the first test in Figure 5.4. The outcome of the iteration pool population with different size testing indicates that the workforce number does not significantly change by increasing the size of the iteration pool population. This indicate for this problem, the quality and diversity of the initial pool population may be more important in this specific problem. Hence, the selection and crossover operators used in the GA, as well as the fitness function, may also have a greater impact on the performance of the algorithm than the size of the iteration pool population. On the contrary, the processing time does increase almost linearly to the increase in the iteration pool population size.

The next parameter to be tested is the number of iterations in the GA; the tested number of iteration is 1000, 5000 and 10000, as shown in Figure 5.5. Like the previous test, the number of iterations also does not impact the quality of the solution as the initial pool quality and diversity is more significant, but the increase of iteration does significantly

impact the processing time. Identical to the previous result, the processing time is also almost linearly increase with the increase of the number of iterations.

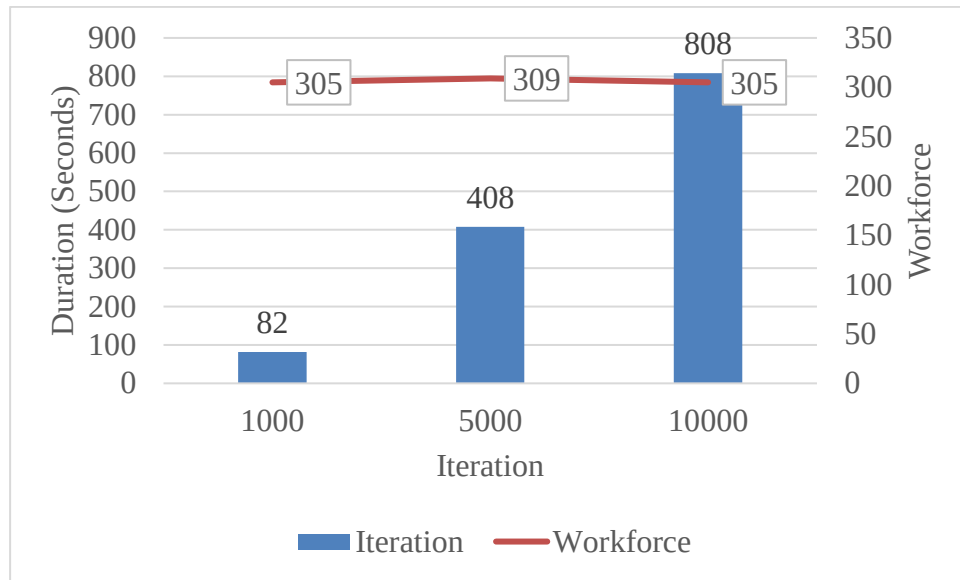


Figure 5.5: Tests on Various Numbers of Iteration

To conclude the three tests, the initial pool size does not play a significant in the quality of the outcome instead the initial pool quality and diversity does. The bigger the range of shift start time and shift duration provided more variety of schedule, hence increasing the possibility of getting better chromosomes or shift schedule. Increasing the size of the iteration pool population and the number of iterations only significantly increase the time linearly.

5.3.1.2 Analysis of GA vs Manual Assignment Shift (MAS) Schedule

To analyze the performance of GA versus manual shift schedule assignment, both the shift schedules selected or generated is illustrated in a chart shown in Figure 5.6.

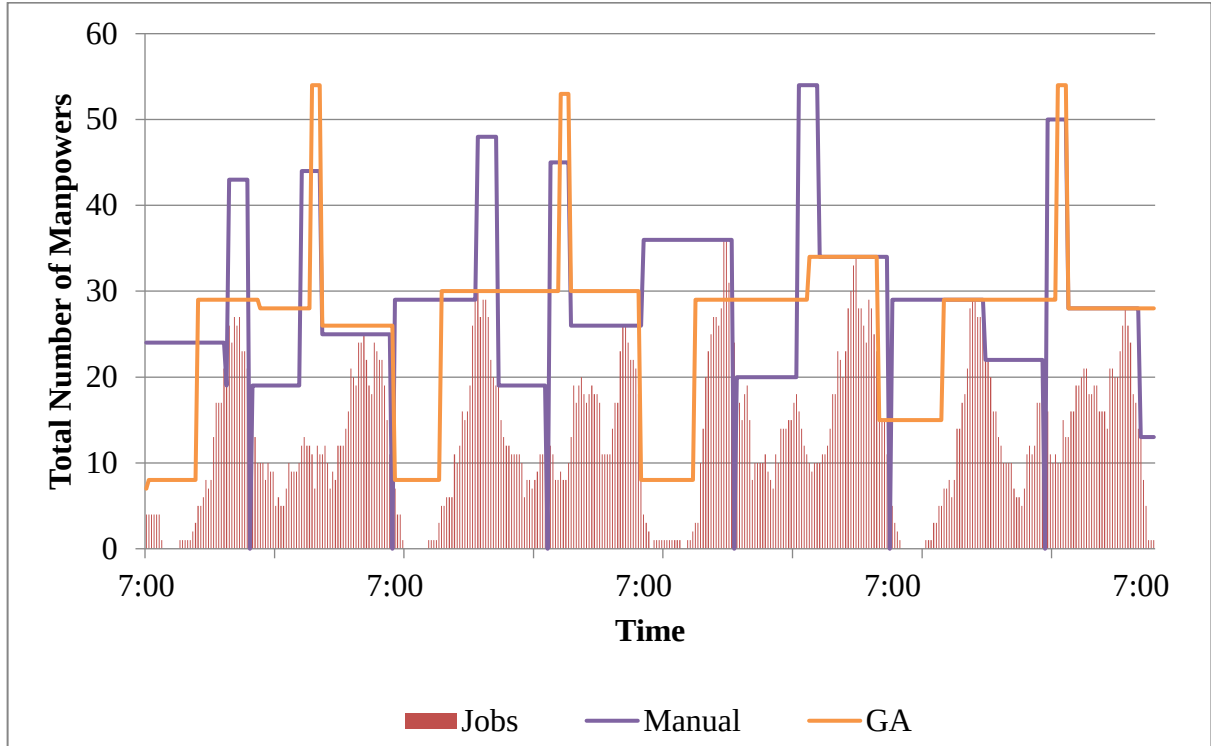


Figure 5.6: Workforce Utilization Comparison

The charts are drawn based on the shift schedule timing and workforce allocation using parameters shown in Figure 5.1, from that it is self-evident that the GA method reign supreme over MAS. GA shift schedule has less overlapped, which is indicated by the peak of each graph line; less overlapped means less wastage or idling workforce. GA shift has one peak less than MAS, and the peak of GA is minimum indicate by slim/thin peak which means wastage of workforce is minimum, hence shift schedule produce by GA will have the minimum workforce and less wastage.

5.3.2 Sensitivity Analysis

In order to examine the impact of particular parameters, tests are run by focusing on the focus parameter and disregarding other parameters.

5.3.2.1 Analysis of GA Initialisation Pool Size Distribution and Shift Pool Size

Analysis from Section 5.3.1.1 shows that initial pool selection size, range of shift start time and shift duration influence the quality of the outcome significantly. The bigger the initial pool selection size, range of shift start time and shift duration, increasing the possibility of getting better chromosomes or shift schedule, which in result reduce the number of workforces required. Figure 5.7 shows the impact on the four parameters on gap hour, overlap, uncovered jobs, and workforce.

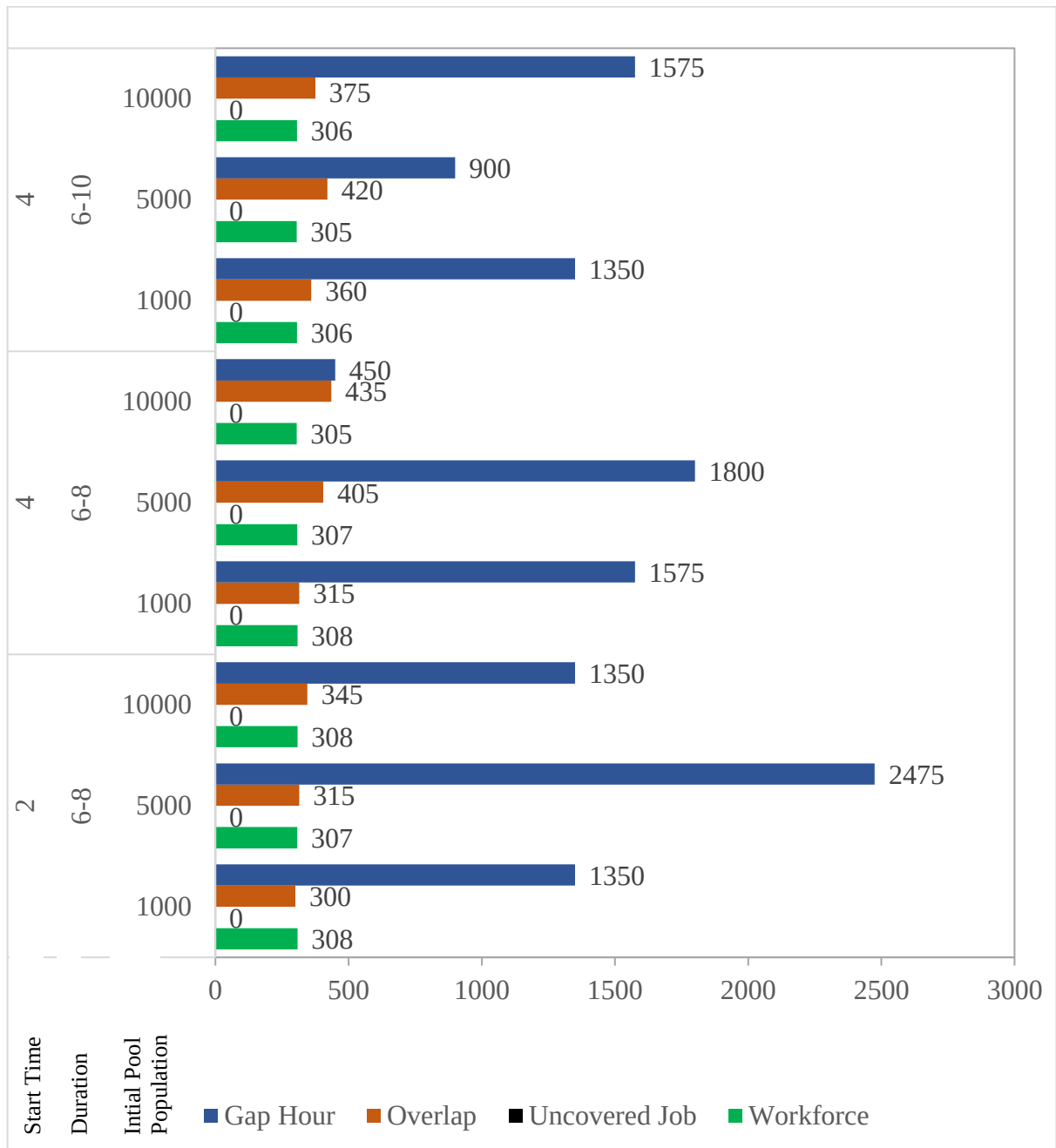


Figure 5.7: Tests Running on Various Initial Pool Number, Shift Duration and Shift Start Time Range

The increase in the three parameter's values not only reduce the workforce but also reduce the gap hour in some cases and, unfortunately, increases the shift overlap duration. Duration of overlap between shifts increased as start time range increases from two hours to

four hours due extra two hours to cover more jobs in a shift, the possibility of overlap has been increased by two-fold. Because of the optimisation function in Section 3.4.5.8, all the test results do not have uncovered job; this means all the jobs are successfully processed and handled by the workforce.

5.3.2.2 Analysis of GA Iteration Distribution

Results from testing of GA iteration pool population with different size using random selection (refer Section 3.4.5.4) in Figure 5.8 reveal that there is no significant reduction in the number of workforces and as usual there are no uncovered jobs.

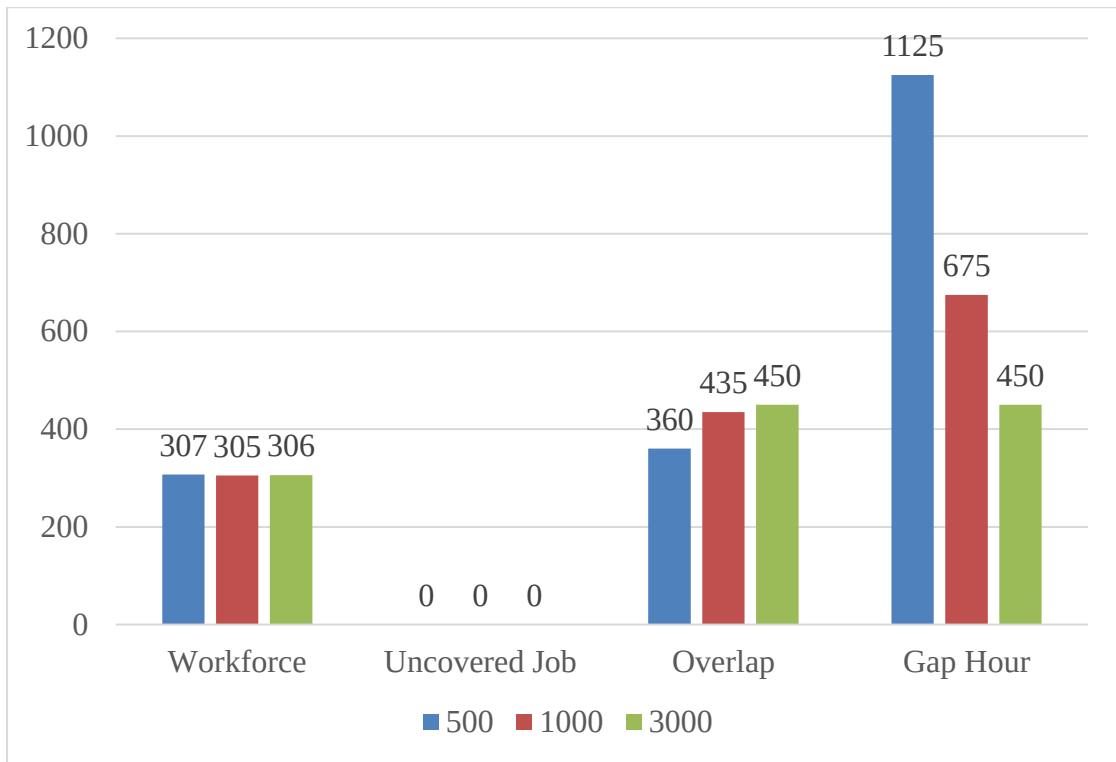


Figure 5.8: Tests on Various Numbers of Iteration Pool Population

From the Figure 5.8, it's observed that the overlap duration of the shift has increased and the gap hour duration has been reduced. This is consistent with findings in the earlier

Section 5.3.2.1. Results from Figure 5.9, where testing of the number of iterations in GA processing, also mirror the reverse trend of Figure 5.8.

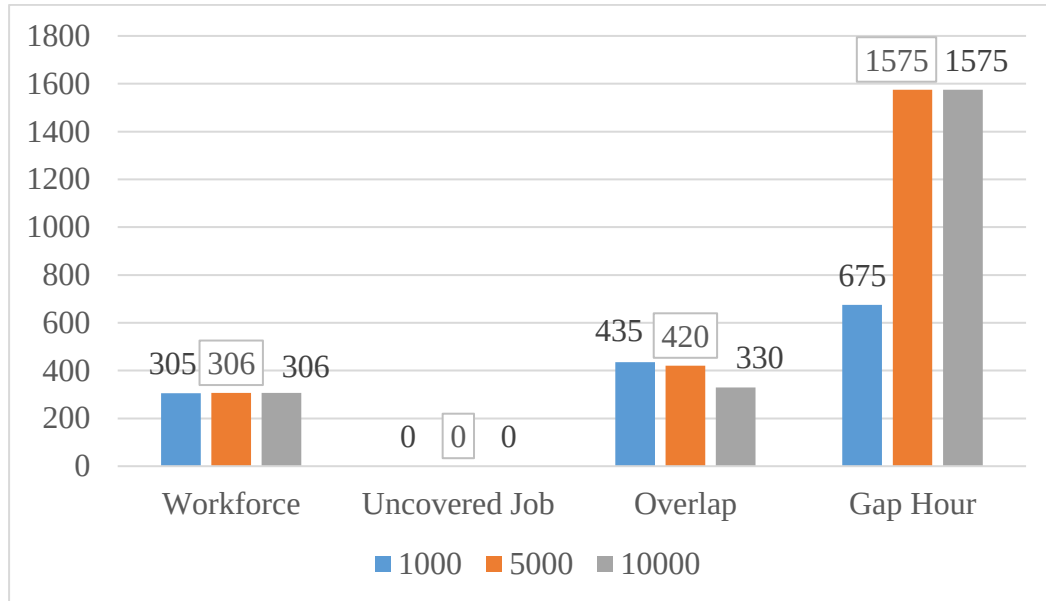


Figure 5.9: Tests on Various Numbers of Iteration

Workforce and uncovered jobs trends are almost identical to the iteration pool population size test. In the earlier testing with various size of iteration pool population, the overlap duration increased, and gap hour decreased with iteration pool population size increment. However, in this test with a different number of iterations, the overlap duration decrease with more iteration run. It proved that with more iteration, GA managed to discover better chromosomes or shift schedules that reduce the overlap duration. This reduction of overlap duration increases the gap hour as less overlap duration means shift tends to start later to avoid overlap of shift.

5.3.2.3 Analysis of GA Operators Distribution

Figure 5.10 shows the sensitivity analysis of GA crossover and mutation operators with the implementation of all the selection methods as described in Section 3.4.5.4.

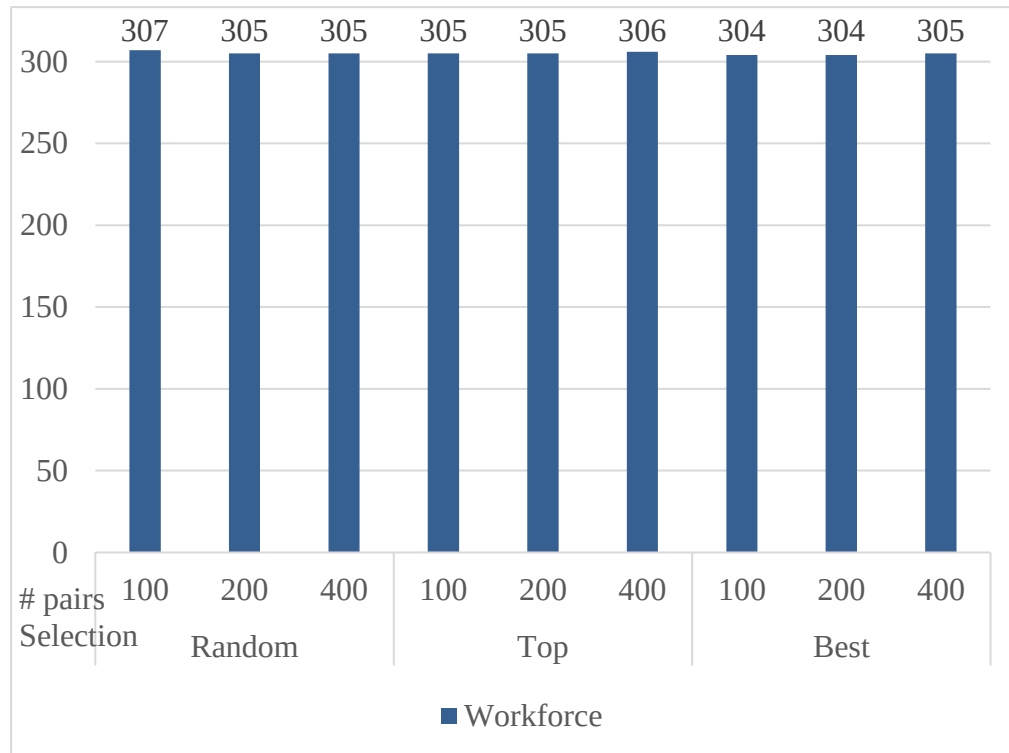


Figure 5.10: Tests of Various Selection Methods Impact on the Workforce

The results demonstrate that the selection of the right chromosomes or shift schedule candidates is essential in GA in order to get the best outcome. The number of pairs selected for crossover and does not impact the results as long as the right chromosomes are selected. Random selection produces the less number of workforces required compared to the top selection, and the best results are with the best chromosomes selection method. Even though the selection method produces only 0.99% ($3/304 \times 100\%$) but in terms of monetary it's about USD4700 per annum based on USD40K monthly operation of an airline in Malaysia.

The results for the overlap duration of shift prove too hard to distinguish, even though the best selection method produces the lowest shift overlap duration, as shown in Figure 5.11.

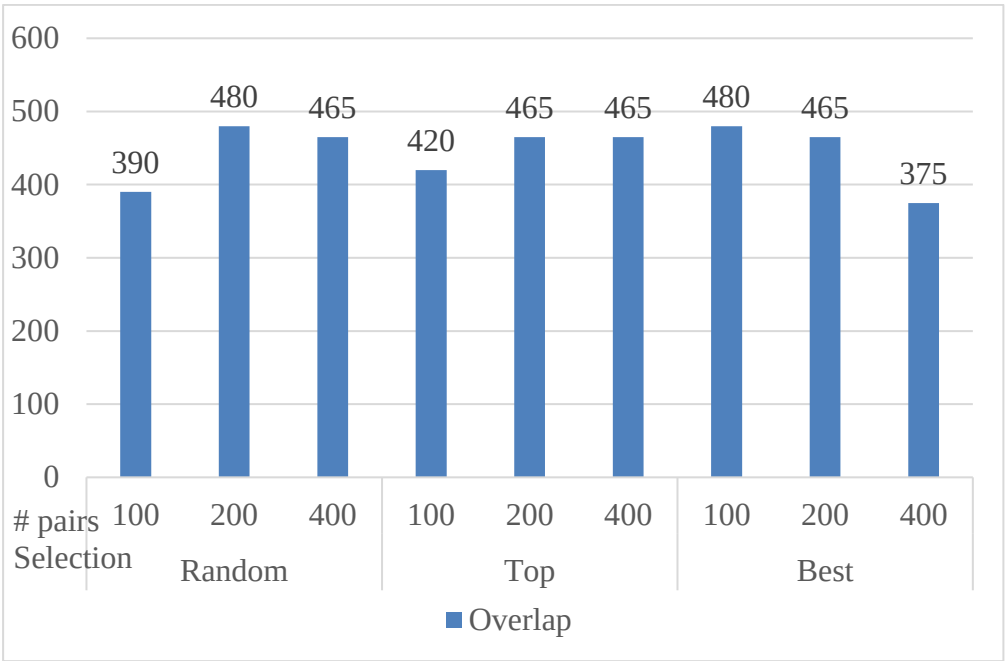


Figure 5.11: Tests of Various Selection Methods Impact on the Overlap

The results for Figure 5.11 also indicate that the number of pairs used for crossover and mutation does not play an essential role in influencing the outcome of the test. The insignificance of the number of pairs used in crossover and mutation is further supported in the gap hour analysis chart in Figure 5.12.

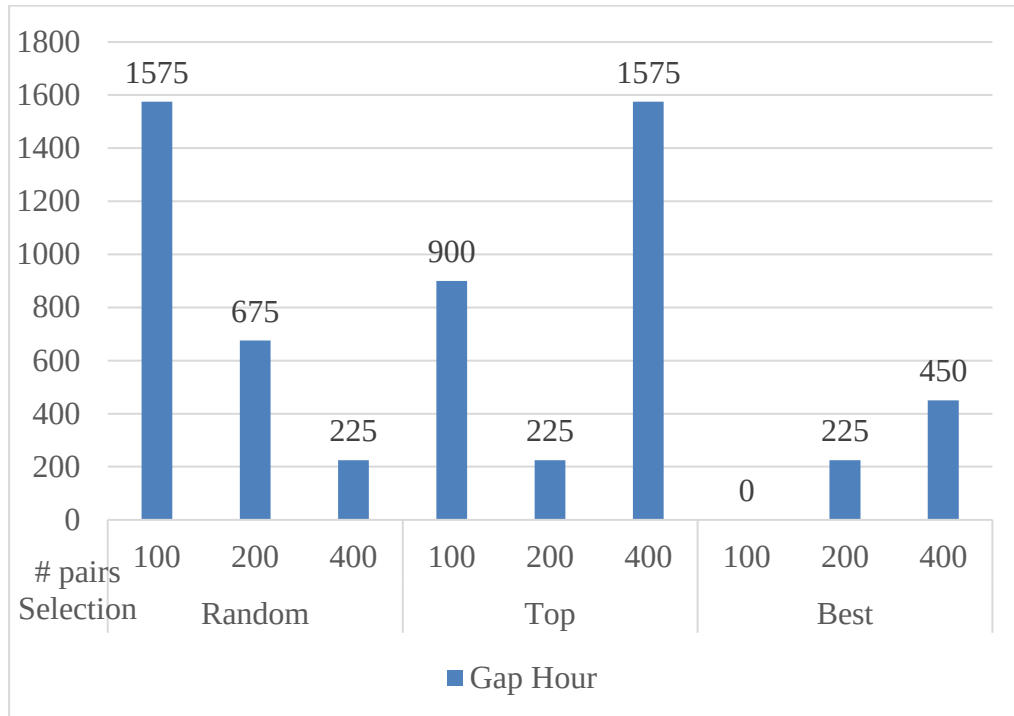


Figure 5.12: Tests of Various Selection Methods Impact on the Gap Hour

The results from the gap hour chart reemphasize that chromosomes selection is vital in getting good results. Top and Best selection methods involve selecting the best performing individuals from the population to produce the next generation which is supposed to be more superior. But both selection methods could lead to a decrease in diversity and an increase in gap hours compared to random selection which could explained why Best method with 400 pairs is getting 450 hours of GAP hour which is higher than Random 400 pairs. The chart in Figure 5.12 shows that the best selection method has the lowest duration in gap hour, followed by the top selection, whereas the random method has the highest gap hour duration.

5.4 Conclusion

GA follows meta-heuristics, which means it is neither fully stochastic nor fully deterministic, but a mix of both. The part of the fitness function is deterministic, but

crossover and mutation parts are random because of crossover and mutation. Performance of GA depends on various parameters like population size, crossover probability, initial mutation probability and error threshold. However, when the size of the dataset increases, the likelihood of getting optimal result increases gradually. The optimum parameter settings are determined by running a lengthy repeated test and the results recorded. However, in almost all cases, the performance of GA is better than the manual assignment by the airline specialist planner. The GA outperformed the human-generated solution by at least $(324-307)/307 \times 100\% = 5.24\%$ or more depending on the parameter's selection as shown in Figure 5.2 in Section 5.2.

CHAPTER 6

ANALYSIS

6.1 Introduction

In the last two chapters, the rule prioritization heuristic algorithms (RP and CBS) and genetic algorithm (GA) has demonstrated to be reliable and efficient in solving the roster and shift scheduling problem through the use of real-life data from an airline company based in KLIA. Both approaches have proven to be able to generate solutions faster than the manual process. The RPH and GA approach developed to solve the problem do not take into consideration the limitation of the workforce as a constraint. The workforce limitation is ignored because of the current manual method in practice, which will resolve the workforce limitation by either enforcing overtime or part-time workforce hiring. Since both approaches can be used to solve the same problem, it will be interesting to know the best algorithm under different settings.

In this chapter, both solutions, the heuristic and the GA approaches are further evaluated and to be compared with a different range of data sets and parameters. All experiments were carried out on a Toshiba laptop with Intel® CORE™ i5 2.4GHz and 8GB memory.

6.2 Computational Analysis

The same set of real data described in previous chapters was used to compare the performance of proposed solutions. As mentioned, the data retrieved was only for four days because of some limitation from the company. For testing, four days and seven days's data (further three more days data were generated) are used to simulate the shift scheduling

process. The quality of the solutions is measured by the number of workforces required to complete the jobs and the computational time required to generate the shift schedules. Table 6.1 shows the computational time for both the heuristic and GA methods.

Table 6.1: Computational Time with Different Datasets Size

Schedule	Heuristic	GA
Four days	19.47 seconds	73.53 seconds
Seven days	30.98 seconds	161.87 seconds

The result demonstrates that both RPH and GA are highly efficient in processing large data sets. However, the RPH approach requires less computational time than the GA, which is logical as GA involves running multiple iterations of GA operations. In order to compare workforce requirement by methods, the four day's data set are used as the result of the manual assignment is available. With the manual assignment shift scheduling result from the studied airline, the other proposed methods can be compared, as shown in Figure 6.1.

From Figure 6.1, both RPH and GA solutions outperform the solution from the manual shift assignment of the airline scheduling specialist. The proposed GA approach reduces the required workforce by 5.25% at the minimum. In total, the proposed RPH approaches manage to reduce the workforce load by as many as 20 jobs (6.17%) over the four days. A reduction of workforce load between 5.25% to 6.17% is a massive saving in monetary for an international airline.

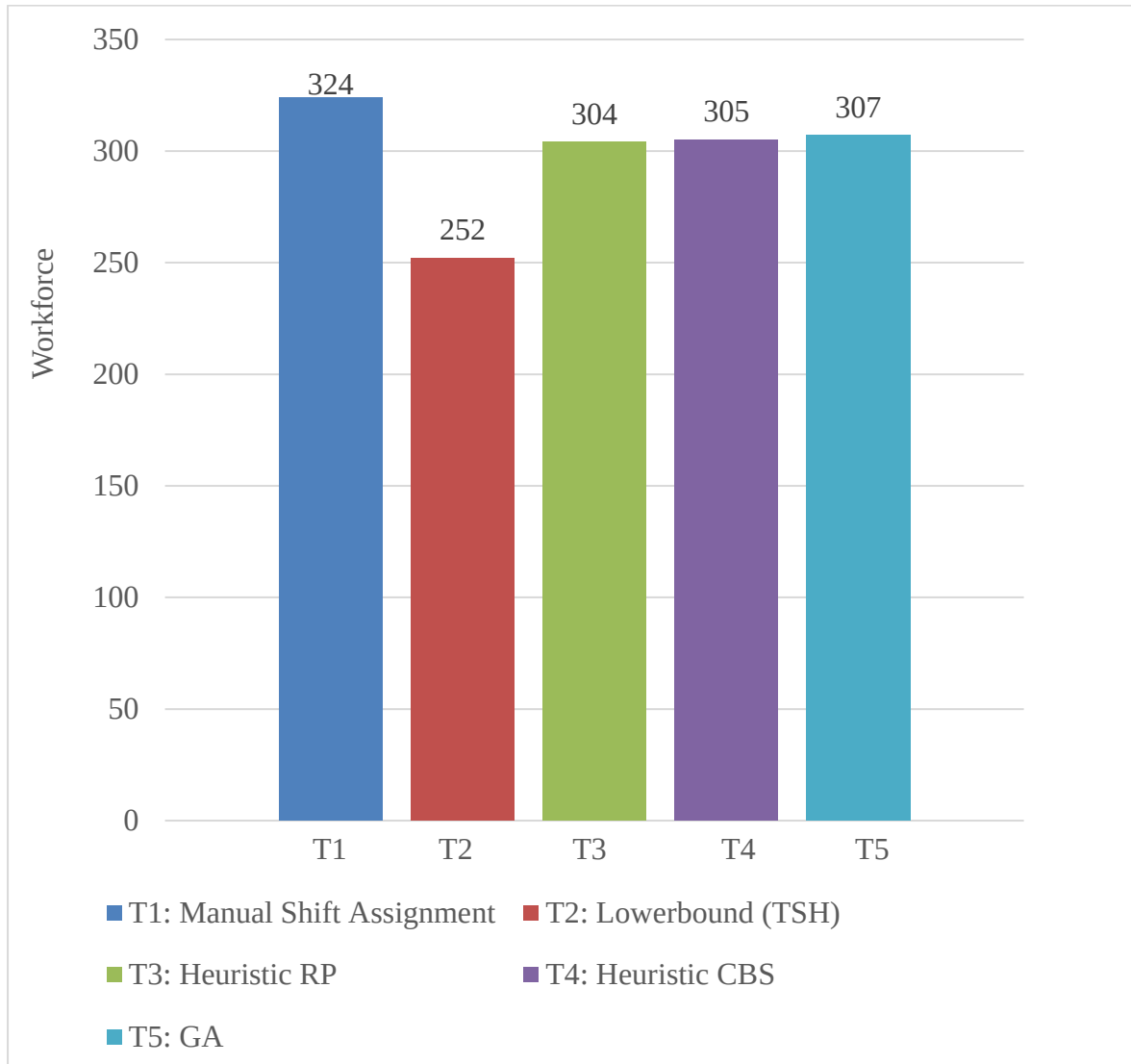


Figure 6.1: Workforce Requirement by Assignment Methods

Besides workforce requirement reduction, productivity can also be increased due to time saved, including the unavoidable idle time from a well-planned shift schedule. This comparative result shows that both RPH and GA solutions can solve the problem effectively and efficiently, even in instances with a massive data set. The RPH approaches have an upper edge when compared to GA solution in terms of the least workforce require and less time to produce the shift schedule.

6.3 Performance Analysis

To analyse the performance of the RPH and GA computational tests are run using a real-life dataset. In the performance analysis, the RPH of randomise pick (RP) variant will be representing heuristic as both RP and CBS (criteria-based selection) yield a less than 1% difference. Hence, the results from the manual assignment by scheduling specialist, RPH-RP variant and GA are compared by workforce distribution on time windows. There are two types of workforce distribution defined, significant peak (SP) and flat peak (FP). The significant peak is when there is a massive surge of the workforce for a short period where, in the chart, it resembles a tower. Meanwhile, a flat peak indicates a substantial workforce for a more extended period, and in the chart, it looks like a box.

From Figure 6.2, the manual assignment solution displayed three SP in the first two days of the schedule. Meanwhile, the RPH solution result is with four SP. Even though with four SP, three of the SP are minor as the peak period is short and significantly less than the manual solution. Approximately 90% of the proposed RPH's FP is equal to or less than the manual solution. In total, the proposed heuristic approaches required 6.17% workforce less over the four days when compared to the manual assignment.

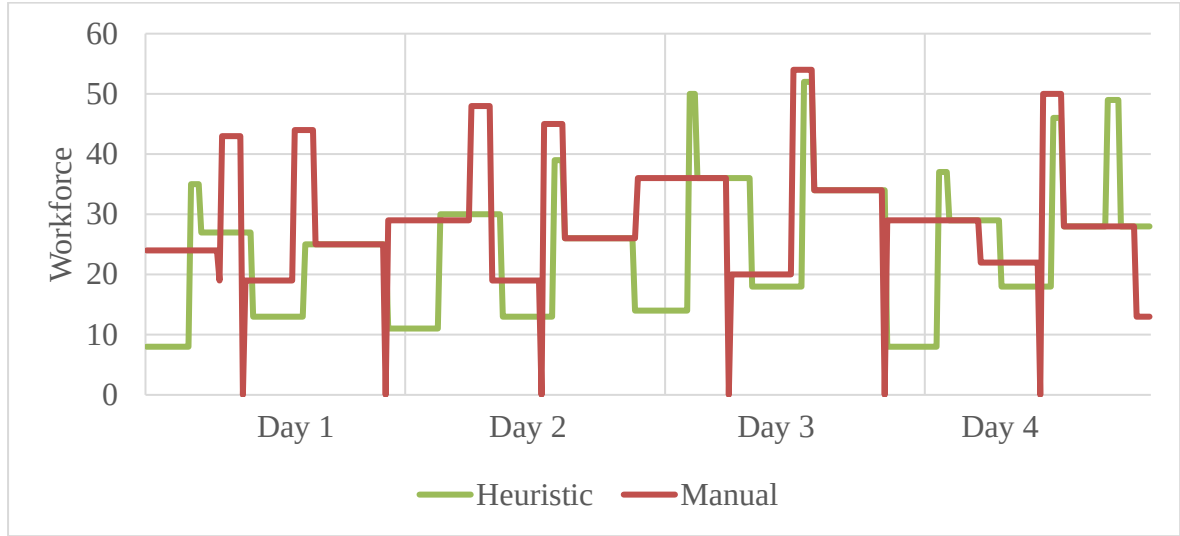


Figure 6.2: Heuristic vs. Manual Workforce Distribution

Similar to Figure 6.2, Figure 6.3 shows that the manual assignment solution has three SP throughout the four days, while the proposed GA approach has less SP and the SP is less significant when compared to the manual approach. It is possible that a dispute could arise regarding the GA algorithm's solution if it indicates that a certain task (SP) is scheduled on multiple days, such as Day 1, Day 2, and Day 4. However, the chart illustrates that the SP for GA solution are overlapped and cancelled out by the SP of the manual assignment solution. The FP of the manual assignment was visually covering a more significant area as compared to GA solution's FP. The GA approaches reduce the workforce load by 5.24% for four days.

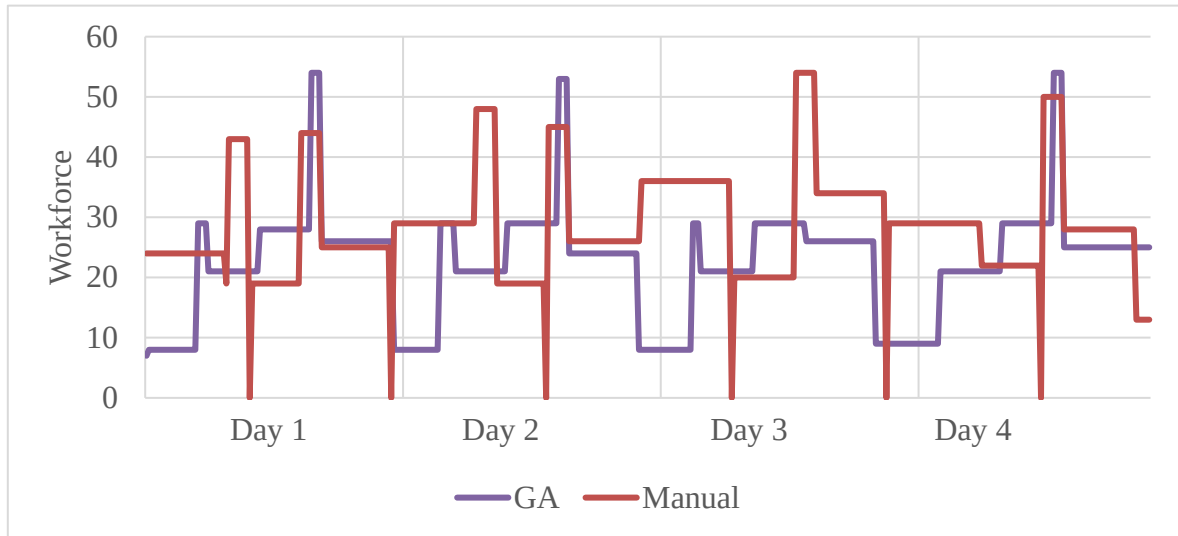


Figure 6.3: GA vs. Manual Workforce Distribution

In Figure 6.4, the performance capabilities of both proposed solutions are shown to be almost equal on par in terms of area coverage of SP and FP. The RPH approach has four SPs while GA has two, but this smaller number of GA's FP was cancelled out by four major FPs of GA solution on each day. The RPH solution's FP is less than half of the FP of the GA solution.

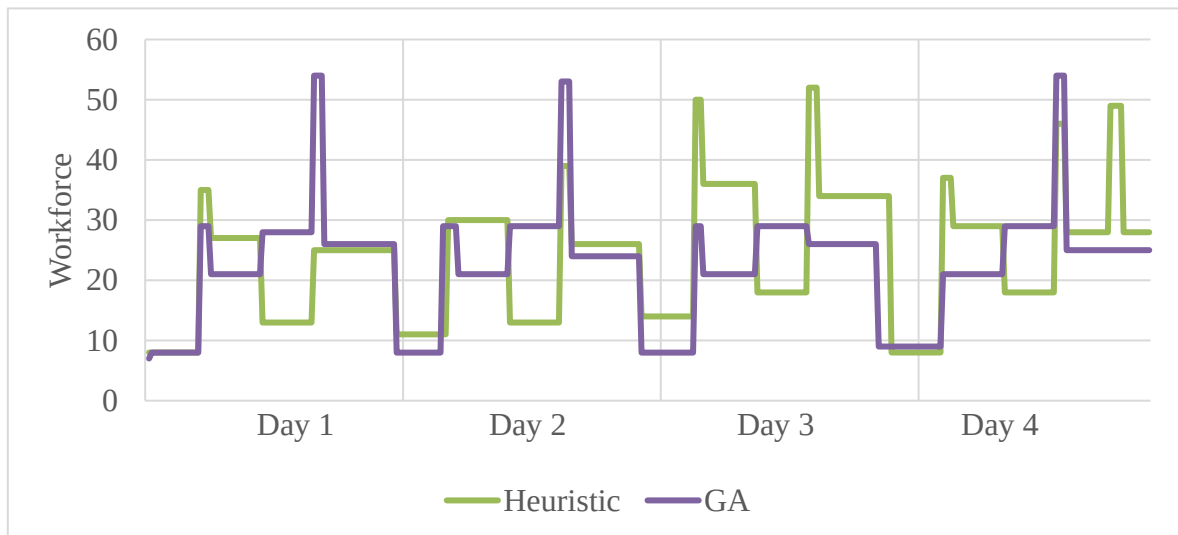


Figure 6.4: Heuristic vs. GA Workforce Distribution

Overall, Figure 6.4 demonstrates that the RPH performs better than the GA in solving the shift scheduling problem. The performance advantage of heuristic over GA is small about 0.97% over four days, but the main advantage is shown in Section 6.2 where computational result reveals that the heuristic solution is almost three to five times faster. This result is expected because the GA is a more complicated algorithm and runs a high number of iterations, compared to the heuristic, for evaluating and choosing the best shift schedule.

6.4 Sensitivity Analysis

Further tests were conducted for sensitivity analysis to show the efficiency and effectiveness of different shift scheduling in both RPH and GA solutions. In this sensitivity study, the focus is given to the scenario with various shift start time range and shift duration. Most of the computational results show that the results tend to improve with the relaxation of parameters. Therefore, for the start time range, three ranges are used in testing; 2 hours, 4 hours, and 6 hours range. For shift duration, the range is 6 to 8 hours and 6 to 10 hours range.

6.4.1 Workforce

Here, workforce demand sensitivity analysis is conducted based on various start time range and various shift duration range, as discussed in Section 6.4 earlier. As presented in Figure 6.5, the number of workforces required to complete all the jobs increase in a similar trend when the shift duration range decrease, for all the solutions. The trend is logical as when a more extended range is allowed on shift duration; the methods have a higher possibility to get a better shift duration that can complete more jobs in one shift assignment. RPH-CBS variant display similar results for both shift duration range is because the best schedule was selected from the same shift schedule that is within both ranges.

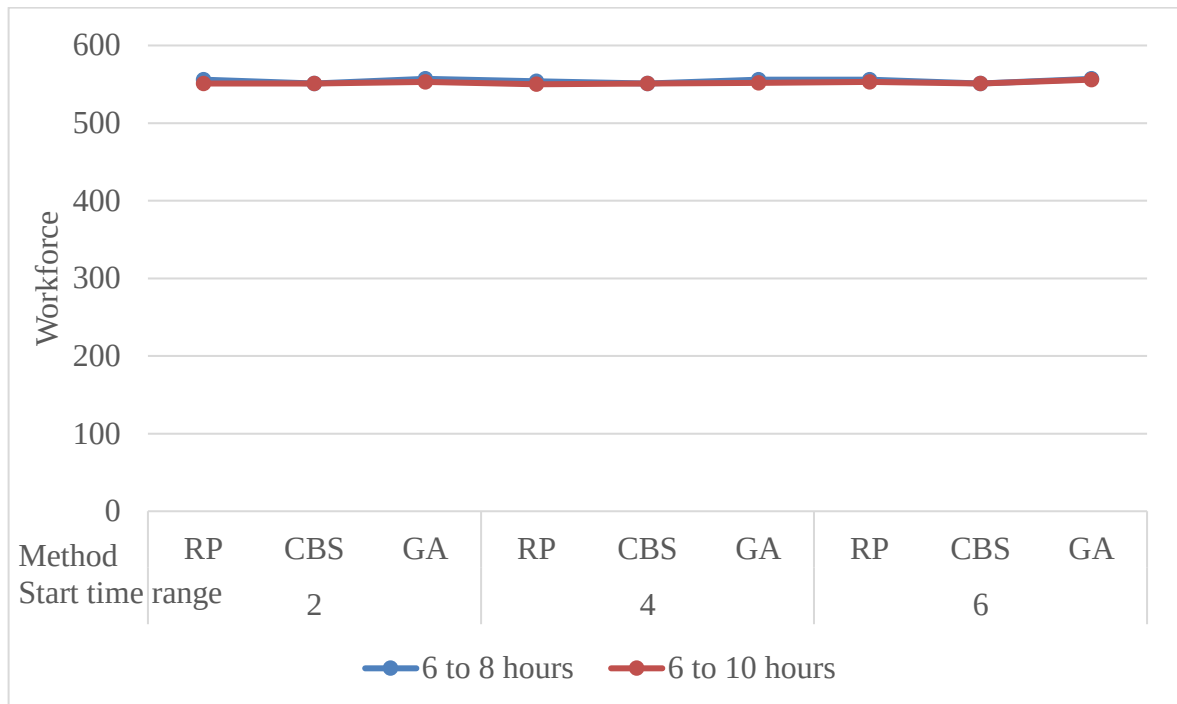


Figure 6.5: Different Shift Duration Test

Start time range does not influence the results visibly on the chart, and this is due there is a tiny time window to play with when the shift duration is extended. Also, GA approaches have the highest workforce demand on either shift duration and performance is worst when on 6 hours start time range.

6.4.2 Overlapped

Similar to the previous section, overlapped duration sensitivity analysis is also tested and analyse based on various start time range and various shift duration range. Referring to Figure 6.6, the duration of overlapped duration between shifts increases in a similar trend when the shift duration range increase. The trend is generally legit for both RPH-RP variant and GA; this is due to the extended range for shift duration, which increased the possibility of longer shift duration, which in turns increase the overlapping of shifts.

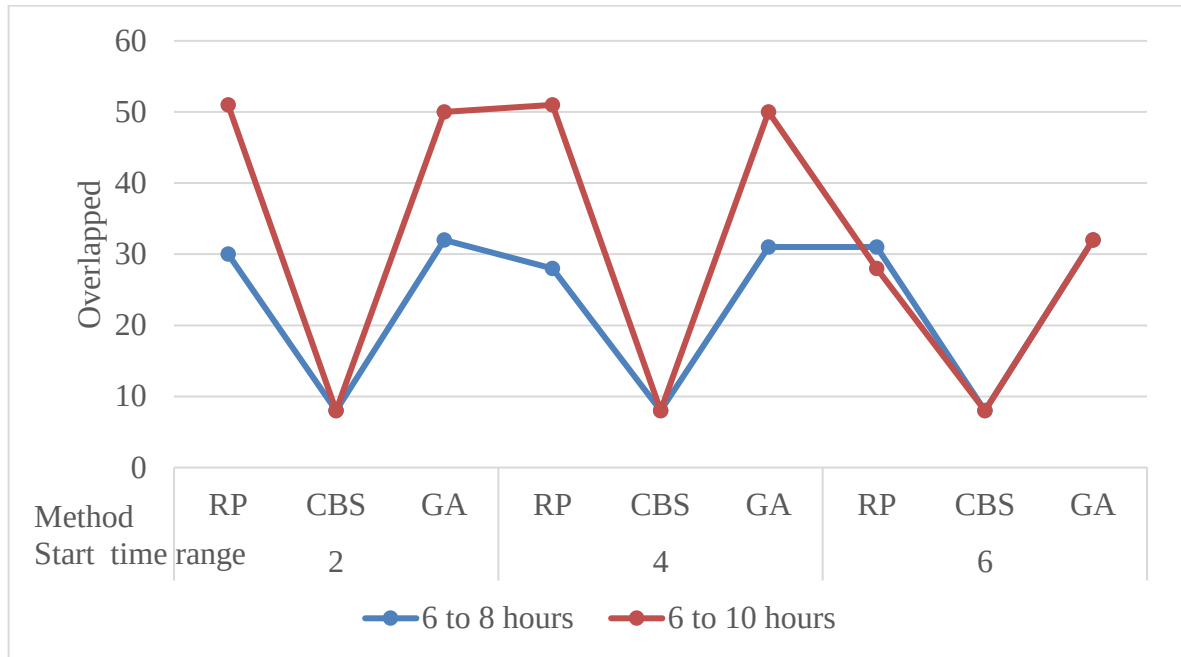


Figure 6.6: Different Shift Duration Test

RPH-CBS variant display similar characteristic, as explained in Section 6.4.1. Similar to workforce demand sensitivity analysis, the start time range also does not impact the outcome of the overlapped durations.

6.4.3 Gap Hour

In this section, start time gap sensitivity analysis is conducted based on various start time range and various shift duration range. As presented in Figure 6.7, RPH-CBS variant approach is not impacted by the parameter's variation. The start time gap for the RPH-RP variant solution increase in a similar trend when the shift duration range increase. Allowing extended range on shift duration increase the possibility of shift with a variety of start time and duration. Shifts with various start times and duration will increase the likelihood of higher differences and reduce the consistency of the shift start time.

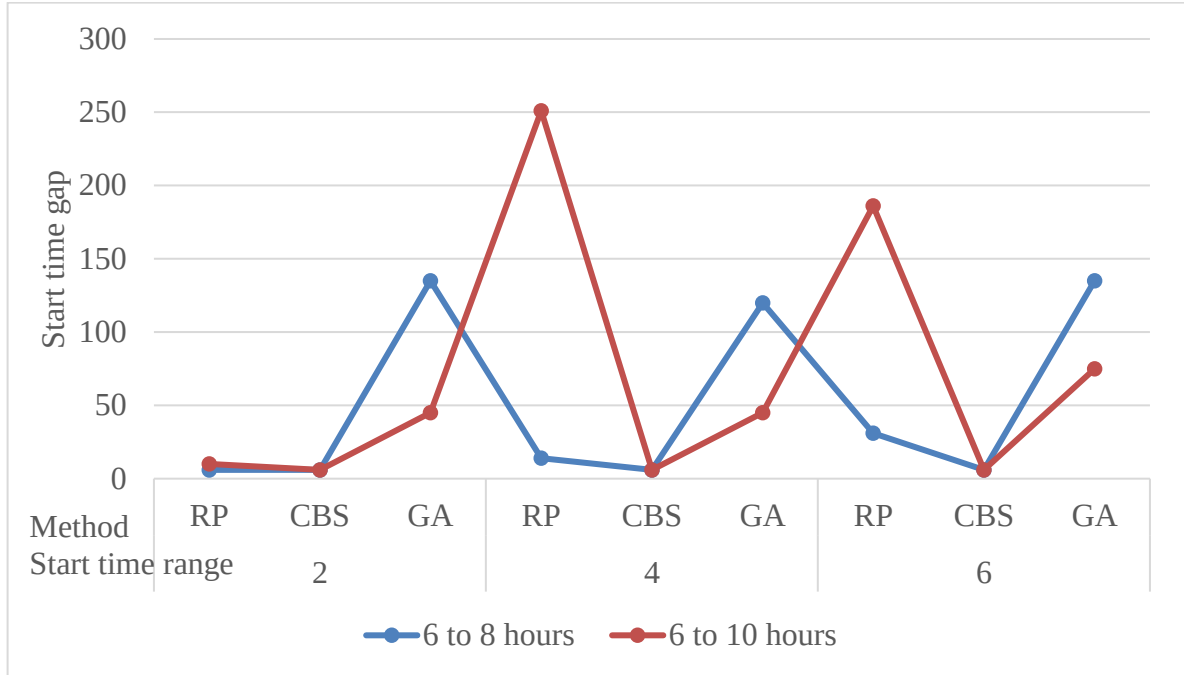


Figure 6.7: Different Shift Duration Test

For GA solution, it is the opposite trend of RPH-RP variant, where the increment of shift duration reduces the start time gap of shift.

6.5 Conclusion

In this chapter, shift scheduling methods of manual assignment, both RPH variance and the GA solution, were analysed in terms of computational and performance. In order to validate the computational superiority, data set using various workforce demand with real-life distributions and various duration lengths. Both RPH and GA solution demonstrated effectiveness and efficiency in solving the shift scheduling problem, regardless of data set size. Furthermore, the presented automated solutions were generated in minutes, or less, this is a tiny fraction of time required when compared to the human expert scheduler in the industry. Ultimately, the RPH approach requires less computational time than the GA as the latter requires a considerable number of iterations.

In order to evaluate the performance, manual assignment and the proposed automated solutions are compared by determining the number of SP and FP coverage. The outcome was evident with both the RPH, and GA demonstrated their strengths and significant advantage over the solution by a human expert scheduler by a reduction of 5-6% in workforce requirement. Both the RPH and GA approaches are also compared to one another to determine the best approach. The RPH approach provided a better solution when compared to GA by less than 1% in term of workforce requirement.

In terms of sensitivity analysis settings, a series of sensitivity test were conducted to analyse the impact of different shift parameters of start time range and duration range. In the test, it is shown that with a longer shift duration, the workforce requirement is reduced for both RPH-RP variant and GA. Meanwhile, the test also shows that with a shorter shift, it will reduce idling time or overlapped of shift. For gap hour sensitivity analysis, RPH-RP variant with short shift duration will have a better gap hour value, but for GA long shift duration will yield a better result. It is also noteworthy to see those tests with different settings do not impact RPH-CBS variant at all.

In the last word, because of the superiority of RPH computational power, the heuristic solution is recommended. The RPH solution also provides a superior solution to all test problems. The sensitivity analysis shows that both RPH methods are capable of assigning and scheduling shift efficiently with less strict constraints if permitted.

CHAPTER 7

CONCLUSION AND RECOMMENDATIONS

7.1 Conclusion

The work described in this thesis has demonstrated that heuristic and genetic algorithms are capable of solving a workforce rostering and scheduling problem in assigning optimum shift. The two proposed algorithms solved the problem by balancing feasibility with high solution quality. The heuristic algorithms present a direct and problem-specific approach to exploiting problem structure, while the genetic algorithm leans toward a generic solution.

From the literature review, the roster and scheduling problem is known to be an NP-hard problem, which is computationally time-consuming even for a small set of data. The literature review, together with the investigation of the problem in the initial phase, inspired the mathematical model for the rostering and scheduling problem.

Because of the problem's complexity, roster and schedule design is typically carried out manually and in an incremental fashion and often small changes. Nevertheless, increasingly sophisticated, and integrated models are being developed to solve the problem fast and optimally for extensive data set as required by various industries, including airline. In this thesis, an original Rule Prioritisation Heuristic (RPH) with two variants was proposed. This approach works well for the airline data collected in the initial phase of the study. A simple rule prioritisation with random and criteria constraint is used in assigning and scheduling the shifts.

Evaluation to gauge the effectiveness of the RPH to solve the roster and scheduling problem is conducted and tested repeatedly with real-life data gathered from airline during the fieldwork. The computational results demonstrate that the RPH can resolve the problem in less than a minute. Furthermore, the proposed RPH algorithm outperforms the airlines' scheduling specialist by reducing up to 6.17% in workforce utilisation and 30% of overlapping in shift duration.

Another approach using GA (genetic algorithm) was developed for comparison and analysis purposes. GA was selected because of its capability to solve multiple constraint problem and flexibility in adapting to complicated constraints. The GA approach is tested using the same real data and analysed similarly to the proposed heuristic solution. The numerical results show that the GA is capable of solving the scheduling problem efficiently and reduced the workforce by 5.24% compared to the real roster. However, it performs slightly behind the RPH (less than 1%) using the same data set.

In Chapter 6, more tests and further assessment are done on both the RPH and GA with recreation data with different sample sizes, distribution types and diverse range of time windows. All the test scenarios are done under a series of sensitivity analysis, such as workforce, overlapping of shift durations and gap hour of shift start time sensitivity.

The outcome from the test results proves both approaches can solve the problem efficiently and effectively, irrespective of the sample sizes. Moreover, both the algorithms are robust and reliable. The results also demonstrate that RPH has a slight advantage when compared to the GA solution. RPH generally tends to perform better when compared to GA in the workforce's reduction, overlapping and start shift time gap hour. However, with RPH-

RP variant, the GA outperforms the heuristic variant in the decrease of overlap shift duration and gap hour in specific scenarios which was discussed in an earlier chapter.

7.2 Contributions

The main intention of this research is fulfilled by the development of computational algorithms that can solve rostering and scheduling problems within a short time frame. Aside from addressing the specific workforce scheduling problem confronted every day by airline companies, the solutions can be tailored to solve the scheduling problem face in other industries as well such as logistic, passenger transportation, cargo operations, manufacturing, etc. From an academic point of view, the main contributions of this study can be categorized as follows:

- Development of a mathematical model based on an airline in Malaysia

Most of the rostering and scheduling problem has been solved by applying various solutions from past studies. The numerically accurate solution has been presented and able to solve with a high degree of accuracy but limited to small instances with a limited number of constraints. The approximation approach utilising heuristic or meta-heuristics typically relaxes the restrictions in order to achieve the desire solution. However, complicated constraints are required pre-requisite in all industries today. Furthermore, there is more than one constraint affecting the rostering and scheduling, which makes the modelling even more complicated which have not been studied in detail to the best of our knowledge.

- Rule Prioritisation Heuristic solutions

- Heuristic solution: RPH-RP

The RPH-RP algorithm was developed by applying randomness as a part of its logic. The algorithm typically uses homogeneously random bits as an assisting input to steer its behaviour in the hope of achieving a good quality outcome. It is proven to be fast and competent in generating desire solutions, even for a vast data set.

- Heuristic solution: RPH-CBS

The RPH-CBS algorithm was developed by taking account of all the criteria during the searching, sorting, and selection processes. The algorithm is easy to implement and flexible enough to cater to all complicated constraints. It is also robust and scalable to satisfy a vast number of criteria. The algorithm is fast in execution, and the solution is high quality. Similar to RPH-RP, the algorithm able to handle an extensive data set.

- Genetic Algorithm solution

GA are chosen for their well-known robustness and ability to solve large and complex discrete optimisation problems. The research starts by applying standard genetic algorithms to the problems and further extended by tweaking each of the operators in extensive computational experiments out on a variety of data. The solutions produced by GA are almost comparable to the heuristic approaches. The disadvantage would be duration require as GA execution time is three times or more when compared to heuristic, but still, it's less than five minutes on our testing.

- Post-processing optimisation algorithm

An optimisation algorithm with the objectives to reduce the workforce required, overlapped in shift period, and uncover jobs had been developed. The optimisation is done on the time window of shift, start time, and shift duration. By applying the algorithm after running the heuristic and GA algorithm, the results were better than running the heuristic and GA algorithm without the optimisation function.

- Excellence initial solutions

Both proposed algorithms, the RPH and the GA are proven to be able to generate high-quality solutions, as shown in computational analysis in chapter 6. An excellent initial solution would be the right candidate as an input to meta-heuristics, hyper-heuristic, machine learning, and so on, in solving the problem further.

- Flexibility, time, and cost-saving

Implementing the proposed heuristics to automate the rostering and shift scheduling will allow flexible changes to the schedule as the schedule can be rapidly generated as opposed to the manual by a human. In addition, the airline would be able to take further actions in enhancing contingency plans as the proposed heuristics would be able to handle abrupt and complicated change in emergency conditions. This is because the time needed to re-generate the solution will be fast. The heuristic solution will also enable the airline to be able to employ the minimum workforce to manage airline operations. Consequently, the resolutions would provide a broad financial saving across by requiring less space, fewer benefits for the employee such as provident fund contribution, healthcare costs, for the employee's family, and so on.

Currently, the airline specialist planner plays a vital role in rostering and scheduling, which could take days to weeks to come out with a roster or schedule. Changes or modification will restart the whole planning process again, which makes this process very

rigid. The proposed heuristic solutions will rapidly accelerate the workforce planning process from days to a few seconds or minutes with a minimal number of planners. Indirectly, the workforce and time could also be reduced while increasing flexibility.

7.3 Future recommendations

This section summarises the various avenues of future work arising from this study and gives an outlook on future research in the area. For the problems already solved in this thesis, the proposed algorithms used could be further extended by the improvement in initial solutions selection or generation and optimisation. Future work on rostering and scheduling problem presented in this thesis are likely to be influenced by the needs of the industries, which is the economic factor.

Most of the research tried to solve the problem with a high degree of accuracy, which requires a high intensity of calculation and a longer duration. The challenge which has proven hard to solve optimally can instead solve by integrating more heuristic components into the algorithm to generate more workable solutions. There are several ways to proceed from this point and to improve the models and methods further; there will be multiple areas of focus to be further suggested for further studies.

Robustness is one of the main criteria in solving rostering and scheduling problem; traditional models tackle the issue with regards to costing and timing by removing uncertainty in the schedule. However, the rosters and schedules are rarely executed without alterations as interferences due to weather, personal delays, unforeseen maintenance, etc. are particularly common. Often, these adjustments require additional modifications, which result in unnecessary additional operational expenditure. As a result, rosters and schedules are rarely executed exactly as planned, and modifications are often required to accommodate

unforeseen events. These modifications can include changes to flight times, crew schedules, and maintenance schedules, among other things. However, making modifications to the schedule can come at a cost. Additional crew members may need to be brought in, flights may need to be delayed or cancelled, and equipment may need to be rescheduled or repositioned. These adjustments can result in additional operational expenses, which can have a significant impact on an organization's bottom line. To minimize these additional costs, it is important to develop schedules and rosters that are as flexible and adaptable as possible. This can include building in additional time between flights, scheduling extra crew members, and having backup equipment available, among other things. By taking a proactive approach and anticipating potential issues, organizations can minimize the impact of unforeseen events and keep operational expenses under control. Airlines and researchers have recognised the need for schedules to perform better under changing conditions. Another factor in improving the robustness of the solution would be to consider to further investigation into a more complicated scheduling constraint, such as workforce skill requirements.

On the GA solution, there is the still unexplored idea of applying more complicated constraints by introducing more intelligent GA operators, which would be worth investing more time and effort. Intelligent GA operators could be further extended by the introduction of adaptive operators' usage based on solution quality. For example, adaptive GA could be implemented to improve the quality of candidate for shifts schedule selection. The implementation can be done by developing an algorithm that can tune the suitable GA operator parameters' value when the current solutions on hand are not of the desired quality.

During the course of experimenting with various shift start time and shift duration, it was discovered that the overlapping of shift is an unproductive workforce period. Future studies may consider exploring and taking advantage of the overlapping of shift period to optimise workforce utilisation between two shifts. In the testing results, it is shown that an overlapped shift period occurs additional workforce that is not utilised throughout the shift. Future work also should look at solving this by taking into consideration of workforce cost, overlapping and gap hour as overall cost. Although intuitive reasons were presented for this, more work is needed to establish the precise behaviour of this operator.

Finally, there is the still unexplored idea for solving rostering and scheduling problem, studies in the area such as meta-heuristics or hyper-heuristics could be conducted using the output from this research. The hybridisation of the algorithms could potentially produce high-quality solutions and hopefully maintaining highly efficient algorithms that could provide results rapidly. At the same time, the proposed solution will need a pre-processing algorithm that will further improve the quality of the candidate of shift generated in order to be used as input for the selection process. Furthermore, with machine learning and artificial intelligence become more popular because of its capability in solving various types of complex problems, these two areas of studies should be considered in future research for rostering and scheduling problems.

REFERENCES

- Afifi, S. S., & Aly, S. A. (2018). Airline Crew Scheduling: An Overview of Research and Optimization Models. *IEEE Access*, 6, 19304-19316. doi: 10.1109/ACCESS.2018.2811393
- Aickelin, U., & Dowsland, K. A. (2004). An indirect genetic algorithm for a nurse-scheduling problem. *Computers & Operations Research*, 31(5), 761-778.
- Algethami, H., & Landa-silva, D. (2015). A Study of Genetic Operators for the Workforce Scheduling and Routing Problem. In *2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, 2016*, 927-934.
- Algethami, H., & Landa-Silva, D. (2017). Diversity-based adaptive genetic algorithm for a Workforce Scheduling and Routing Problem. In *2017 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1771-1778).
- Algethami, H., Landa-Silva, D., & Martínez-Gavara, A. (2017). Selecting genetic operators to maximise preference satisfaction in a workforce scheduling and routing problem. In *Proceedings of the 6th International Conference on Operations Research and Enterprise Systems* (pp. 416-423).
- Algethami, H., Martínez-Gavara, A., & Landa-Silva, D. (2019). Adaptive multiple crossover genetic algorithm to solve workforce scheduling and routing problem. *Journal of Heuristics*, 25(4), 753-792.
- Algethami, H., Pinheiro, R. L., & Landa-Silva, D. (2016). A genetic algorithm for a workforce scheduling and routing problem. In *2016 IEEE Congress on Evolutionary Computation (CEC)* (pp. 927-934).
- Alharbi, A., & AlQahtani, K. (2016). A Genetic Algorithm Solution for the Doctor

- Scheduling Problem. In *Proceeding of the 10th International Conference on Advanced Engineering Computing and Applications in Sciences, ADVCOMP* (pp. 91–97).
- Anbil, R., Gelman, E., Patty, B., & Tanga, R. (1991). Recent advances in crew-pairing optimization at american airlines. *Interfaces*, 21(1), 62-74.
- Attia, E. A., Duquenne, P., & Le-Lann, J. M. (2014). Considering skills evolutions in multi-skilled workforce allocation with flexible working hours. *International Journal of Production Research*, 52(15), 4548-4573.
- Aydemir, E., & Koruca, H. I. (2015). A new production scheduling module using priority-rule based genetic algorithm. *International Journal of Simulation Modelling*, 14(3), 450-462.
- Bai, R., Burke, E. K., Kendall, G., Li, J., & McCollum, B. (2010). A hybrid evolutionary approach to the nurse rostering problem. In *IEEE Transactions on Evolutionary Computation*, 14(4), 580-590.
- Bai, J. (2021). A novel heuristic algorithm for airline crew scheduling problem with fuzzy constraints. *Journal of Ambient Intelligence and Humanized Computing*, 12(12), 15165-15178. doi: 10.1007/s12652-021-03497-9
- Baker, K. R. (1976). Workforce allocation in cyclical scheduling problems: a survey. *Journal of the Operational Research Society*, 27 (1), 155-167.
- Baker, K.R., & Trietsch, D. (2009). Scheduling Groups of Jobs. *Principles of Sequencing and Scheduling*.
- Barbosa-Póvoa, A., Andrade-Campos, J.A.G., & Curralero, S.M.P. (2018). Crew scheduling in airlines: A review of optimization models and algorithms. *Journal of Air Transport Management*, 66, 108-121. doi: 10.1016/j.jairtraman.2017.11.006

- Brezulianu, A., Fira, M., & Fira, L. (2009). A genetic algorithm approach for a constrained employee scheduling problem as applied to employees at mall type shops. In *ACM International Conference Proceeding Series* (pp. 497–501).
- Brezulianu, A., Fira, L., & Fira, M. (2012). A genetic algorithm approach for scheduling of resources in well-services companies. *International Journal of Advanced Research in Artificial Intelligence*, 1(5), 1-6.
- Brusco, M. J., & Jacobs, L. W. (1993). A simulated annealing approach to the cyclic staff-scheduling problem. *Naval Research Logistics*, 40(1), 69-84.
- Brusco, M. J., & Jacobs, L. W. (2001). Starting-time decisions in labor tour scheduling: An experimental analysis and case study. *European Journal of Operational Research*, 131(3), 459-475.
- Castillo-Salazar, J. A., Landa-Silva, D., & Qu, R. (2015). A Greedy Heuristic for Workforce Scheduling and Routing With Time-dependent Activities Constraints. In *Proceedings of the 4th International Conference on Operations Research and Enterprise Systems* (pp. 367-375).
- Clausen, T. (2010). A dynamic programming-based heuristic for the shift design problem in airport ground handling. *DTU Management 2010*, No. 7. Kgs. Lyngby, Denmark: Technical University of Denmark (DTU).
- Constantino, A. A., Landa-Silva, D., de Melo, E. L., de Mendonça, C. F. X., Rizzato, D. B., & Romão, W. (2014). A heuristic algorithm based on multi-assignment procedures for nurse scheduling. *Annals of Operations Research*, 218(1), 165-183.
- Dahmen, S., & Rekik, M. (2012). Solving Multi-Activity Personalized Shift Scheduling Problems with a Hybrid. *CIRRELT-2012-30*. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), Canada.

- Dahmen, S., & Rekik, M. (2015). Solving multi-activity multi-day shift scheduling problems with a hybrid heuristic. *Journal of Scheduling*, 18(2), 207-223.
- Dean, J. (2008). Staff scheduling by a genetic algorithm with a two-dimensional chromosome structure. In *Proceeding of the 7th Conference on the Practice and Theory of Automated Timetabling* (pp. 1-15). Montreal, Canada.
- Di Gaspero, L., Gärtner, J., Kortsarz, G., Musliu, N., Schaerf, A., & Slany, W. (2007). The minimum shift design problem. *Annals of Operations Research*, 155(1), 79-105.
<https://doi.org/10.1007/s10479-007-0221-1>
- Di Gaspero, L., Gärtner, J., Musliu, N., Schaerf, A., Schafhauser, W., & Slany, W. (2010). A hybrid LS-CP solver for the shifts and breaks design problem. In *Proceedings of the 7th international conference on Hybrid metaheuristics (HM'10)* (pp. 46–61). Springer, Berlin, Heidelberg.
- Dohn, A., & Mason, A. (2013). Branch-and-price for staff rostering: An efficient implementation using generic programming and nested column generation. *European Journal of Operational Research*, 230(1), 157-169.
- Dowland, K. A. (1998). Nurse scheduling with tabu search and strategic oscillation. *European Journal of Operational Research*, 106(2-3), 393- 407.
- Éles, A., Cabezas, H., & Heckl, I. (2018). Heuristic algorithm utilizing mixed-integer linear programming to schedule mobile workforce. *Chemical Engineering Transactions*, 70, 895-900.
- Elharrouss, Y. (2022). An efficient algorithm for airline crew scheduling problem with legality constraints. *Journal of Air Transport Management*, 98, 101998. doi: 10.1016/j.jairtraman.2022.101998
- Ernst, A. T., Jiang, H., Krishnamoorthy M., & Sier, D. (2004). Staff scheduling and

- rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153(1), 3-27.
- Fang, X., & Wang, J. (2021). An efficient algorithm for large-scale airline crew scheduling with multiple days-off constraints. *IEEE Transactions on Intelligent Transportation Systems*, 22(9), 5137-5148. doi: 10.1109/TITS.2021.3088302
- Feng, L., & Wang, Y. (2020). An efficient algorithm for airline crew scheduling problem with multiple scenarios. *Journal of Air Transport Management*, 87, 101866.
- Gärtner, J., Musliu, N., & Slany, W. (2004). A heuristic based system for generation of shifts with breaks. In: Macintosh A., Ellis R., Allen T. (Eds.), *Applications and Innovations in Intelligent Systems XII*. SGAI 2004. Springer, London.
- Gao, C. (2021). A machine learning approach to airline crew scheduling with preference-based constraints. *Transportation Research Part C: Emerging Technologies*, 126, 103123. doi: 10.1016/j.trc.2021.103123
- Guo, F., & Lv, Y. (2017). An effective algorithm for the airline crew scheduling problem based on adaptive large neighborhood search. *Journal of Air Transport Management*, 58, 78-85. doi: 10.1016/j.jairtraman.2016.09.006
- He, F., & Qu, R. (2012). A constraint programming based column generation approach to nurse rostering problems. *Computers & Operations Research*, 39(12), 3331-3343.
- Heimerl, C., & Kolisch, R. (2010). Scheduling and staffing multiple projects with a multi-skilled workforce. *OR Spectrum*, 32(2), 343-368.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.
- Huang, K.-Y. (2021). An adaptive optimization algorithm for airline crew scheduling. *Journal of Air Transport Management*, 94, 101946. doi:

10.1016/j.jairtraman.2021.101946

- Ip, W. H., Wang, D., & Cho, V. (2013). Aircraft ground service scheduling problems and their genetic algorithm with hybrid assignment and sequence encoding scheme. *IEEE Systems Journal*, 7(4), 649-657.
- Jacquillat, A., & Odoni, A. R. (2015). An integrated scheduling and operations approach to airport congestion mitigation. *Operations Research*, 63(6), 1390-1410.
- Kim, H.J., Kim, M.S., & Kim, T. (2020). An optimization model for scheduling crew with preferential bidding system in airline. *Journal of Scheduling*, 23(4), 341-354. doi: 10.1007/s10951-020-00641-w
- Jacquillat, A., & Odoni, A. R. (2015). An integrated scheduling and operations approach to airport congestion mitigation. *Operations Research*, 63(6), 1390-1410.
- Kim, H.J., Kim, M.S., & Kim, T. (2020). An optimization model for scheduling crew with preferential bidding system in airline. *Journal of Scheduling*, 23(4), 341-354. doi: 10.1007/s10951-020-00641-w
- Kim, S., Ko, Y., Uhm, S., & Kim, J. (2013). An Efficient Method for Nurse Scheduling Problem using the Genetic Algorithm. In IST 2013, *ASTL*, 23, 252- 255.
- Kundu, S., Mahato, M., Mahanty, B., & Acharyya, S. (2008). Comparative performance of simulated annealing and genetic algorithm in solving nurse scheduling problem. In *Proceeding of International Multiconference of Engineers and Computer Scientists*, 1, (pp. 19-21).
- Kyngäs, N., Goossens, D., Nurmi, K., & Kyngäs, J. (2012). Optimizing the unlimited shift generation problem. In: Di Chio, C. et al. (Eds.). *Applications of evolutionary computation. EvoApplications 2012*. Lecture Notes in Computer Science, 7248. Springer, Berlin, Heidelberg.

- Laporte, G., Nobert, Y., & Biron, J. (1980). Rotating schedules. *European Journal of Operational Research*, 4(1), 24-30. 10.1016/0377-2217(80)90036-3.
- Lau, H. C. (1996). On the complexity of manpower shift scheduling. *Computers & Operations Research*, 23(1), 93-102.
- Leksakul, K., & Phetsawat, S. (2014). Nurse scheduling using genetic algorithm, *Mathematical Problems in Engineering*, 2014.
- Liu, T., & Gao, Y. (2018). An efficient hybrid algorithm for the airline crew scheduling problem with multi-airport constraint. *Journal of Advanced Transportation*, 2018, 1-14.
- Li, H., & Yin, Y. (2018). A two-phase algorithm for airline crew scheduling under complex constraints. *Journal of Air Transport Management*, 67, 60-69. doi: 10.1016/j.jairtraman.2017.11.011
- Liu, J. (2018). An integrated algorithm for airline crew scheduling with flight connection. *Journal of Advanced Transportation*, 2018, 1-11. doi: 10.1155/2018/1392865
- Liu, J., & Gao, Y. (2019). A distributed algorithm for airline crew pairing problem with a large number of constraints. *Journal of Advanced Transportation*, 2019, 1-13. doi: 10.1155/2019/2714027
- Liu, H., & Gao, Y. (2021a). A hybrid algorithm for airline crew scheduling problem. *Journal of Industrial and Production Engineering*, 38(3), 241-252. doi: 10.1080/21681015.2020.1775849
- Liu, J., & Gao, Y. (2021b). Airline crew scheduling: A hybrid algorithm combining genetic algorithm and simulated annealing. *Journal of Advanced Transportation*, 2021, 1-16. doi: 10.1155/2021/5575166
- Mallick, B. K., & Mishra, R. K. (2018). Revenue Management in Airlines: An Overview.

- Journal of Revenue and Pricing Management, 17(1), 1-19. doi: 10.1057/s41272-017-0113-3
- Misir, M., Smet, P., & Vanden Berghe, G. (2015). An analysis of generalised heuristics for vehicle routing and personnel rostering problems. *The Journal of the Operational Research Society*, 66(5), 858-870.
- Montoya, C., & Mejía, G. (2006). Heuristic algorithm for workforce scheduling problems. *Brazilian Journal of Operations & Production Management*, 3(2), 35-48.
- Morris, J. G., & Showalter, M. J. (1983). Simple approaches to shift, days-off and tour scheduling problems. *Management Science*, 29(8), 942-950.
- Morz, M., & Musliu, N. (2004). Genetic algorithm for rotating workforce scheduling problem. In *Proceeding of Second IEEE International Conference on Computational Cybernetics, 2004. ICCCC 2004*. (pp. 121-126). IEEE.
- Musliu, N. (2006). Heuristic methods for automatic rotating workforce scheduling. *International Journal of Computational Intelligence Research*, 2(4), 309-326.
- Musliu, N., Gärtner, J., & Slany, W. (2002). Efficient generation of rotating workforce schedules. *Discrete Applied Mathematics*, 118(1-2), 85-98.
- Muter, İ., Birbil, Ş. İ., Bülbül, K., Şahin, G., Yenigün, H., Taş, D., & Tüzün, D. (2013). Solving a robust airline crew pairing problem with column generation. *Computers & Operations Research*, 40(3), 815-830.
- Mutingi, M., & Mbohwa, C. (2014). Health-care staff scheduling in a fuzzy environment: A fuzzy genetic algorithm approach. In *Proceeding of International Conference on Industrial Engineering and Operations Management* (pp. 303-312).
- Odoni, A. R., & Barnhart, C. (2003). Optimization Techniques for Airline Schedule

- Planning. *Transportation Science*, 37(3), 267-291. doi: 10.1287/trsc.37.3.267.16022
- Özğüven, C., & Sungur, B. (2013). Integer programming models for hierarchical workforce scheduling problems including excess off-days and idle labour times. *Applied Mathematical Modelling*, 37(22), (9117-9131).
- Pan, S., Akplogan, M., Létocart, L., Touati, N., & Calvo, R. W. (2016). Solving a Multi-Activity Shift Scheduling Problem with a Tabu Search Heuristic. In *PATAT 2016: Proceedings of the 11th International Conference of the Practice and Theory of Automated Timetabling* (pp. 317-326).
- Pan, S., Akplogan, M., Touati, N., Létocart, L., Calvo, R. W., & Rousseau, L. M. (2018). A hybrid heuristic for the multi-activity tour scheduling problem. *Electron Notes Discret. Mathematics*, 69, 333-340.
- Peterson, E. B., Neels, K., Barczi, N., & Graham, T. (2013). The economic cost of airline flight delay. *Journal of Transport Economics and Policy*, 47(1), 107-121.
- Pinheiro, R. L., Landa-Silva, D., & Atkin, J. (2015). A variable neighbourhood search for the workforce scheduling and routing problem. In: *Advances in Nature and Biologically Inspired Computing* (pp. 247-259). Springer, Cham.
- Pot, A., Bhulai, S., & Koole, G. (2008). A simple staffing method for multiskill call centers. *Manufacturing & Service Operations Management*, 10(3), 421-428.
- Rodič, B., & Baggia, A. (2013). Dynamic airport ground crew scheduling using a heuristic scheduling algorithm. *International Journal of Applied Mathematics and Informatics*, 7(4), 153-163.
- Shen, Y., Peng, K., Chen, K., & Li, J. (2013). Evolutionary crew scheduling with adaptive chromosomes. *Transportation Research Part B: Methodological*, 56, 174-185.
- Souai, N., & Teghem, J. (2009). Genetic algorithm based approach for the integrated airline

- crew-pairing and rostering problem. *European Journal of Operational Research*, 199(3), 674-683.
- Sze, S. N., Chiew, K. L., & Sze, J. F. (2010). Multi-trip vehicle routing and scheduling problem with time window in real life. In *AIP Conference Proceedings. The 15th International Conference of Hong Kong Society for Transportation Studies*. American Institute of Physics.
- Sze, S. N., Chiew, K. L., & Sze, J. F. (2012). An Efficient Algorithm for Manpower Scheduling Problem with Time Windows and Multiple Trips. In *AIP Conference Proceedings*. 1479(1), (pp. 1151-1154). American Institute of Physics.
- Sze, S. N., Phang, M. H., & Chiew, K. L. (2017). Real-life faculty examination timetabling to utilise room used. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*. 9(3-11), 51-54.
- Tanomaru, J. (1995). Staff scheduling by a genetic algorithm with heuristic operators. In *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, 3, (pp. 1951-1956). Telhada, J. (2014). Alternative MIP formulations for an integrated shift scheduling and task assignment problem. *Discrete Applied Mathematics*, 164, 328-343.
- Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., & De Boeck, L. (2013). Personnel scheduling: A literature review. *European Journal of Operational Research*, 226(3), 367-385.
- Wang, Z., & Wang, H. (2019). A new variable neighbourhood search algorithm for airline crew scheduling. *Journal of Air Transport Management*, 78, 87-98.
- Wei, W., & Zhao, Y. (2022). A Fuzzy-Based Algorithm for Airline Crew Scheduling Problem. *Journal of Ambient Intelligence and Humanized Computing*, 1-13.

<https://doi.org/10.1007/s12652-021-03302-w>

- Wei, J., & Yuan, J. (2017). A branch-and-price algorithm for the airline crew pairing problem with duty constraints. *Journal of Air Transport Management*, 65, 23-32.
- Zhang, X., & Jin, S. (2020). A new optimization algorithm for airline crew scheduling based on improved tabu search. *Journal of Air Transport Management*, 85, 101742.
- Zhang, Guangyu & Wang, Hongbo & Zhao, Wei & Guan, Zhiying & Li, Pengfei. (2021). Application of Improved Multi-Objective Ant Colony Optimization Algorithm in Ship Weather Routing. *Journal of Ocean University of China*. 20. 45-55. 10.1007/s11802-021-4436-6.
- Zhongyuan Liang, Mei Liu, Peisi Zhong, Chao Zhang, Xiao Wang. Hybrid Algorithm Based on Genetic Simulated Annealing Algorithm for Complex Multiproduct Scheduling Problem with Zero-Wait Constraint. *Mathematical Problems in Engineering*, vol. 2021, Article ID 9951995, 21 pages, 2021. <https://doi.org/10.1155/2021/9951995>

APPENDICES

Appendix A: Journal Publications

1. **Lee, K.W.** & Sze, S. (2019) Real-Life Optimum Shift Scheduling Design. *Journal of ICT Research and Applications*, Vol. 13, No. 1, 2019, 19-35. DOI: <https://doi.org/10.5614/itbj.ict.res.appl.2019.13.1.2>
2. **Lee, K.W.**, Sze, S., & Phang, K. (2017). Heuristic method for optimum shift scheduling design. Proceedings of the 6th International Conference on Computing and Informatics, ICOCI 2017 25-27April, 2017 Kuala Lumpur. Universiti Utara Malaysia.