MODIFIED HARMONY SEARCH ALGORITHMS FOR THE AIRCRAFT LANDING PROBLEM

OMAR SALIM ABDULLAH

THESIS SUBMITTED IN FULFILMENT FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

FACULTY OF INFORMATION SCIENCE AND TECHNOLOGY
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2019

ALGORITMA GELINTARAN HARMONI TERUBAHSUAI UNTUK MASALAH PENDARATAN PESAWAT

OMAR SALIM ABDULLAH

TESIS YANG DIKEMUKAKAN UNTUK MEMPEROLEH IJAZAH
DOKTOR FALSAFAH

FAKULTI TEKNOLOGI DAN SAINS MAKLUMAT
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2019

DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged.

September 2019

OMAR SALIM ABDULLAH
P74197

ACKNOWLEDGMENT

In the name of Allah, the most Gracious and most Merciful. I give thanks to Allah, the creator of all that exists, for this blessing of completing my thesis. A thesis like this would not have been possible without the guidance and support of several individuals who contributed to the preparation and completion of this study. I would like to extend my sincere gratitude to all of them.

First, my sincere gratitude to my main supervisor, Prof. Dr. Salwani Abdullah for her continuous support, patience, and motivation during my Ph.D. study and related research. Her guidance helped me in my research and the writing of this thesis. I could not have imagined having a better supervisor and mentor for my Ph.D. study. Also, I would like to thank my co-supervisor, Assoc. Prof. Dr. Mohd Zakree Ahmad Nazri for his suggestions. I acknowledge the financial support from Universiti Kembangan Malaysia (UKM). I appreciate the Data Mining and Optimization Research Group (DMO) and all postgraduate staffs of FTSM for their support.

I address my special thanks to Dr. Abdelaziz Isam Hammouri for being my friend and providing much-needed discussions and support. I also would like to express appreciation to all my friends for their encouragement, support, and prayers. I am indeed grateful to you all. I address my special thanks to my friend, Dr. Bashir Elhub for his support and help. Last but not least, I would like to thank my family: my father (Salim) and mother, my brother (Qusay), my wife and sisters for their endless love, support, and encouragement throughout this study and my life in general.

ABSTRACT

The Aircraft Landing Problem (ALP) is a challenging task for air traffic controllers in an airport. ALP is a non-deterministic polynomial-time hard (NP-hard) problem that deals with assigning an available runway and a landing time to an arriving aircraft. The landing time of each aircraft must be within a stipulated target landing time. If the actual landing time deviates from the target landing time, an additional cost will be imposed. This cost is determined by the amount of earliness or lateness with respect to the actual landing time. ALP can be divided into static and dynamic problems. The static ALP (s-ALP) occurs when all the information on aircraft are fixed and there is no change in the information when the scheduling process commences. On the other hand, the dynamic ALP (d-ALP) considers changes in the information that occur during the scheduling process as new aircraft may appear in the radar range. Solving both static and dynamic ALPs aim to minimize the overall cost i.e., the deviation from a preferred target time of each aircraft. The complexity of the ALP draws the attention of researchers from various research domains to generate a robust system that supports air traffic controllers in making the landing decision. Many heuristic and metaheuristic approaches have been developed to derive a highly effective solution for this complicated problem. The research work presented in this thesis aims to build upon the state-of-the-art search methodologies for aircraft scheduling problems by investigating the use of the Harmony Search Algorithm (HSA) as a population-based algorithm for s-ALP and d-ALP. The research first investigates HSA and incorporates a number of rules to control the neighborhood structures employed during the optimization process. This contribution systematically avoids the slow convergence problem in the traditional HSA algorithm as a result of the random strategy deployed therein. Afterward, a hybridization between HSA and Variable Neighborhood Search (VNS) is proposed to improve the ability of the HSA in exploring an unvisited region in the search space. Moreover, Simulated Annealing (SA) is used to avoid the local optimum. The newly proposed algorithm is coded as VNHSA. The VNS replaces the pitch consideration rate condition in the improvisation step in the original HSA which ensures the algorithm is not stuck in the local optimum. Due to the fact that unpredictable changes might occur during the course of an ongoing scheduling process, the applicability of the VNHSA is tested on d-ALP. The proposed approaches are tested using well-known datasets from OR-library with a range of 10 to 500 aircraft and 1 to 5 runways. Computational results based on standard benchmark datasets demonstrate the effectiveness of the proposed algorithm. Further evaluations are made through comparisons with the best results from other approaches in the scientific literature and statistical tests. The results show that the proposed approaches are able to obtain competitive results and can be deployed in practice.

ALGORITMA TAMBAHBAIK PENCARIAN HARMONI UNTUK MASALAH PENDARATAN PESAWAT

ABSTRAK

Masalah Pendaratan Pesawat (MPP) merupakan tugas penting bagi pengawal trafik udara di lapangan terbang. MPP adalah masalah NP-sukar yang berkaitan dengan penugasan pesawat yang tiba ke landasan yang ada dan waktu pendaratan. Masa pendaratan untuk setiap pesawat mesti berada dalam jarak masa yang merangkumi waktu pendaratan sasaran. Jika masa pendaratan sebenar menyimpang dari masa pendaratan sasaran, kos tambahan akan dikenakan yang ditentukan oleh jumlah keterawalan dan keterlambatan waktu pendaratan sebenar. MPP boleh dibahagikan kepada masalah statik dan dinamik. Masalah statik ialah apabila semua maklumat tentang pesawat adalah tetap dan tiada perubahan maklumat apabila proses penjadualan bermula. Sebaliknya, masalah dinamik menganggap perubahan kepada maklumat berlaku semasa proses penjadualan iaitu pesawat-pesawat baru mungkin muncul dalam julat radar. Kedua-dua jenis statik dan dinamik MPP (s-MPP dan d-MPP) bertujuan untuk meminimumkan kos keseluruhan, iaitu penyimpangan dari masa sasaran pilihan setiap pesawat. Kerumitan MPP menarik perhatian penyelidik dari pelbagai bidang penyelidikan untuk menghasilkan sistem yang mantap untuk menyokong pengawal trafik udara dalam membuat keputusan pendaratan. Banyak pendekatan heuristik dan metaheuristik dalam kesusasteraan telah dibangunkan untuk menghasilkan penyelesaian yang berkualiti tinggi untuk masalah ini, kerana tahap komplikasi tinggi MPP. Kerja penyelidikan yang dibentangkan dalam tesis ini bertujuan untuk membina metodologi carian untuk masalah penjadualan pesawat dengan menyiasat penggunaan Algoritma Carian Harmoni (ACH) sebagai algoritma berasaskan populasi untuk s-MPP dan d-MPP. Penyelidikan pertama menonjolkan penyiasatan ke atas ACH, di mana beberapa petua diperkenalkan untuk mengawal struktur kejiranan yang akan digunakan semasa proses pengoptimuman berbanding dengan strategi rawak dalam ACH asal yang menyebabkan penumpuan algoritma yang lambat. Kedua, untuk meningkatkan keupayaan ACH dalam menerokai rantau yang tidak diterokai di ruang carian, hibridisasi antara ACH dan Carian Kejiranan Pembolehubah (CKP) diselidiki (dikodkan sebagai CKPH). CKP menggantikan keadaan Kadar Pertimbangan Pitch dalam langkah improvisasi dalam ACH asal yang membantu mengelakkan daripada mudah terjebak dalam optimum tempatan. Oleh sebab perubahan tidak dapat diramalkan berlaku semasa proses penjadualan yang berterusan, akhirnya, penerapan CKPH diuji pada d-MPP. Pendekatan yang dicadangkan diuji pada dataset yang terkenal dari OR-perpustakaan dengan bilangan pesawat antara 10 hingga 500, dan bilangan landasan pesawat antara 1 hingga 5. Hasil komputasi berdasarkan data penanda aras standard menunjukkan keberkesanan pendekatan yang dikaji di sini pada masalah pendaratan pesawat statik dan dinamik. Perbandingan dengan pendekatan lain dalam kesusasteraan saintifik menunjukkan bahawa pendekatan yang dicadangkan dapat memperoleh hasil yang kompetitif dan dapat dianggap sebagai metodologi yang sesuai untuk digunakan.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

COP	Combinatorial Optimization Problems
TSP	Traveling Salesman Problem
FCFS	First Come First Served
ALP	Aircraft Landing Problem
MIP	Mixed-Integer Programming
B&B	Branch and Bound
S-metaheuristic	Single based Solution Metaheuristic
P-metaheuristic	Population based Solution Metaheuristic
HSA	Harmony Search Algorithm
PSO	Particle Swarm Optimization
SS	Scatter Search
HM	Harmony Memory
s-ALP	Static ALP
d-ALP	Dynamic ALP
GHSA	Guided Harmony Search Algorithm
VNHSA	Variable Neighborhood Harmony Search Algorithm
ATM	Air Traffic Management
NM	Nautical Miles
SA	Simulated Annealing
VNS	Variable Neighborhood Search
VND	Variable Neighborhood Descent
ILS	Iterated Local Search
BKV	Best Known Values

GA	Genetic Algorithm
TS	Tabu Search
CA	Cellular Automation
RO	Relaxation Operator
ACO	Ant Colony Optimization
DE	Differential Evolution
SD	Simple Descent
RH	Rolling Horizon
BA	Bat Algorithm
HBA	Hybrid Bat Algorithm
EO	Extremal Optimization
CPS	Constrained Position Shifting
FAA	Federal Aviation Administration
ICAO	International Civil Aviation Organization
SbO	Simulation-based Optimization
VNTS	Variable Neighborhood Tabu Search
HMCR	Harmony Memory Consideration Rate
PAR	Pitch Adjustment Rate
RC	Random Consideration
HMS	Harmony Memory Size
NI	Number of Improvisations
GHS	Global Best Harmony Search
VRPTW	Vehicle Routing Problem with Time Windows
DVRPTW	Dynamic Vehicle Routing Problem with Time Windows

ADHS	Adaptive Dynamic Harmony Search
CSA	Clonal Selection Algorithm
DCHS	Dynamic Clustering Harmony Search
FCM	Fuzzy c-means Algorithm
BBO	Biogeography Based Optimization
BHS	Biogeographic Harmony Search

CHAPTER I

INTRODUCTION

1.1 MOTIVATION AND BACKGROUND

In most activities, human intends to expand the benefit or income and limit the expenses or the outcome. To achieve that, optimization approaches are required for better usage of the available resources, especially when the resources are limited. Optimization is used in engineering, medicine, decision making, modelling, designing, scheduling and many other applications. One of the most common optimization problems is combinatorial optimization (Koret & Vygen, 2014). In this class of optimization problems, there are a large number of solutions and it is very difficult and sometimes impossible to find the best solution among the existing solutions. Most Combinatorial Optimization Problems (COP), cannot be solved by an exact approach. Furthermore, evaluating all the existing solutions is impossible due to the limitation of time and memory; an example of such is Traveling Salesman Problem (TSP). Thus, the approximation approach is suitable to solve this class of problems (Talbi, 2009).

One of the foremost vital examples of optimization application is in transportation. Over the years, different transportation problems arise as optimization problems where there are limited resources available to serve many beneficiaries. Transportation problems include vehicle routing, bus transport systems, train transfer problem, aircraft take-off and aircraft landing scheduling problem. Among these problems, the airport operations problem is considered as an important optimization problem due to the high cost associated with the disturbance of the take-off or landing of an aircraft. In cases where trips are cancelled, the airline companies compensate the costumers by changing the trips or booking new trips. These problems are costly especially in the busy airports. Additionally, the airports’ managers attempt to achieve the maximum capacity of runways usage by landing many aircraft in less time to acquire more income. In busy airports, the task of generating an efficient schedule for aircraft landing is a difficult task because of air traffic and time limitation (Bennell et al., 2013).

Over the last decades, air traffic experienced tremendous growth. Millions of passengers and tons of cargos are conveyed through airspace every year. For example, according to Dubai airport official website in January 2016, 7,327,637 passengers travel through Dubai airport and 201,483 tons of cargos at their ports ("http://www.dubaiairports.ae/corporate/media-centre" 2018). Comparing these with the record in the last twenty years, there is a huge increase in the numbers of passengers and cargo. This growth in air traffic makes the embarkation and disembarkation very critical tasks in the airports. Expanding the existing infrastructures in the airports is challenging because of many reasons such as economic, spatial, and political reasons.

Airports managements seek to optimize runways capacity by generating landing schedule for the waiting aircraft having short waiting time as much as possible while considering safety requirements. In the last decades, the airport management teams followed the First Come First Served (FCFS) approach. This approach fairly schedules the landing orders according to the predefined landing time. However, it is not always practical, especially in the cases when there are several aircraft waiting to land as it may lead to a conflict in the landing order. When more than one aircraft takes the same landing time. This results in a long waiting time and increased total cost. Thus, researchers use other approaches to generate efficient landing schedule for aircraft landing problem based on metaheuristic optimization algorithms. This maximizes the throughput of the existing resources (runways) and reduces the cost of long waiting time.

Aircraft landing problem (ALP) received special attention in research recently. The first research with a simple formulation and single runway (Roger G. Dear, 1976), considered ALP as an optimization problem. ALP occurs when there is a list of aircraft waiting for permission to land. In such situations, each aircraft must be assigned a certain landing time on the runway. The optimal sequence is to land all the waiting aircraft considering aircraft having the shortest waiting time to land. Each pair of aircraft landing on the same runway should be given sufficient separation time to ensure safety. ALP can be solved statically when all the aircraft information is available and there is no update during the scheduling. On the other hand, it is solved dynamically when new aircraft appear during the scheduling process (Bennell et al., 2013). In the last few years, different methods have been proposed to solve the ALP such as dynamic programming, Mixed-Integer Programming (MIP) Formulation, Branch and Bound algorithms (B&B) and Heuristic based-approaches. Using metaheuristic algorithms is an effective approach to solving the ALP. This is because such problems are usually large. Furthermore, compared with other approaches for solving the ALP, it exhibits an efficient performance with a reasonable complexity.

A metaheuristic algorithm is a high-level search methodology that uses the heuristic procedures to comprehensively explore the search space (Talbi, 2009). Metaheuristic algorithms are divided into two groups: single-based solution and population-based solution. Single-based solution metaheuristics (S-metaheuristic) are a class of algorithms that improves one solution during the optimization process by waking through the neighborhoods of the search space. The wake is iterated from one solution to another to get a better solution. Hill climbing, simulated annealing, Tabu search, iterated local search, and guided local search are examples of single-based solution. Population-based solution metaheuristic (P-metaheuristic) starts with a set of solutions and it iteratively improves the population of solution. Genetic algorithm, scatter search, ant colony, and particle swarm optimization are examples of population-based solution algorithms.

Metaheuristic and exact methods have been used for solving ALP in the literature. Generally, the exact methods are used to solve optimization problems with limited sizes (Chong, 2012). Hence, exact methods are not preferred for solving most optimization problems with larger sizes. Conversely, metaheuristic algorithms are the suitable choice to handle such optimization problems. Metaheuristic algorithms used to solve ALP includes Genetic Algorithm (Hu & Di Paolo, 2008), Ant Colony (Farah et al., 2011a), Scatter Search (Pinol et al., 2006), and Iterated Local Search (Sabar & Kendall, 2015). According to (Lieder et al., 2015), the ALP with multiple runways is a highly challenging problem and there is no known efficient algorithm to solve this problem. This motivated us to investigate the possibility of new metaheuristic algorithm to solve the ALP with multiple runways.

Relatively, there is a new population-based algorithm known as Harmony Search Algorithm (HSA), which can explore the search space. This algorithm has been successfully deployed in different combinatorial optimization problems such as timetabling (Al-Betar et al., 2012), job shop scheduling (Gao et al., 2016), and vehicle routing (Yassen et al., 2015). The most effective approach used to improve the performance of harmony search is the modification of parameters tuning methods and its hybridization with other algorithms (Alia & Mandava, 2011). HSA has proved to be an efficient method in many combinatorial optimization problems (Manjarres et al., 2013).

The significance of ALP and the efficiency of HSA motivated us to investigate the possibility of using HSA to solve ALP. We conducted a preliminary investigation on the use of HS algorithm to solve ALP and examine its performance with respect to dynamic ALP. The weakness and strength of the algorithm were reported, and an improvement procedure was proposed.

1.2 PROBLEM STATEMENT

In the last few decades, ALP attracted more attention in air transportation due to the safety and efficiency required for this task. Also, the annual growth rate in air transportation is expected to be between three and five percent in spite of the short-term economic recession (Mesgarpour, 2012). The increasing traffic in air transportation directly affects the airport operations. Also, the growth of air transportation presents a significant challenge, especially in the congested airports. ALP is associated with assigning an arriving aircraft to an available runway. The landing time for each aircraft must be within a time interval encompassing a target landing time. If the actual landing time deviates from the target landing time, an additional cost will be imposed. This cost is determined by the amount of earliness and lateness of the actual landing time.

ALP is a challenging task for air traffic controllers in airports due to the limited runways, short time availability, and safety constraints. ALP is a non-deterministic polynomial-time hard (NP-hard) problem (Beasley et al., 2003; Girish, 2016; Sabar & Kendall, 2015). Due to the complexity of ALP, even in medium-sized instances, there is no exact method suitable to solve it in large size instances. According to (Lieder et al., 2015), " *To date, no efficient methods have been proposed in the reviewed literature for the multi-runway ALP that are capable of solving large problem instances*". Therefore, finding the optimal solution by exploring the huge number of possible solutions is very difficult if not impossible. This is because exploring the whole solution space is a computational task that involves a prohibitive amount of computational time.

The significance of ALP in real-life applications necessitates developing new optimization techniques. Hence, metaheuristic techniques will be appropriate to find a high-quality solution within reasonable resources. HSA is a recent meta-heuristic population-based optimization algorithm introduced by (Geem et al., 2001). HSA has been utilized to tackle several optimization problems and it has continued to be of particular interest to researchers (Elyasigomari et al., 2017; Esfahani et al., 2016; Keshtegar et al., 2017) for several reasons. In comparison to Genetic Algorithm (GA), HSA can overcome the drawback associated with the building block theory of GAs by considering all the existing solutions instead of considering only two solutions (parents) in its reproduction. Also, it does not require crossover and mutation operators, thus it needs less computational effort, in terms of memory and runtime. Additionally, HSA is more flexible and has a well-balanced mechanism to improve both global and local exploration capabilities in contrast to heuristic techniques.

Despite significant achievements made to improve the performance of HSA technique, researchers still need to overcome its weaknesses such as the HSA's random mechanism in selecting the decision variables in solution improvisation process (Al-betar, 2010). Unlike PSO which handles a solution vector by unitary rule, HSA adjusts each variable independently (Ouyang et al., 2018). This can have a negative effect on the convergence speed and the quality of new solutions. Therefore, a solution may be discarded iteratively because its quality is worse than the existing solutions in the HM. This problem affects the performance of the algorithm as the algorithm continually discards the new solution; implying that HM will not be updated. Slow convergence speed is another common weakness of the metaheuristic algorithms. This problem is highly influenced by the value of the decision variables in improvising a new solution (Siddique & Adeli, 2016).

Combinatorial optimization problems have extremely difficult constraints and variable neighborhood is required to overcome this difficulty efficiently. The neighborhood structure directly affects the performance of the optimization algorithm for solving any optimization problem. The key challenge of creating neighborhood solution is the violation of one or more constraints while solving the optimization problem (Hosny & Mumford, 2010). Therefore, a controlled neighborhood search operator is essential to guide the search toward unvisited locations within the search space. According to (Burke et al., 2010), using more than one neighborhood structure during the search offers a great opportunity to escape from local optima. Deeper knowledge about the specifics of the problem at hand can help to design very efficient neighborhood moves and avoid the unnecessary moves which thereby ensures that there are smaller neighborhoods (Yuan et al., 2013). Although the HSA is successfully implemented in numerous scheduling problems such as job-machine scheduling (Yuan et al., 2013), careful design is required to generate neighborhood solutions. Another challenge with using HSA is that it accepts the newly generated solution only if it is better than all the existing solutions in the harmony memory. This can make the algorithm to stuck in a local optimum. Therefore, an investigation of the HSA that accepts the worst solution with certain probability advancing from the principle of simulated annealing algorithm is required.

. The majority of the research works in evolutionary computation focuses on optimization of static problems. However, many real-world optimization problems are actually dynamic, and optimization methods capable of continuously adapting the solution to a changing environment are needed. Although the success of metaheuristic algorithms lies in its ability to tackle various optimization problems, the majority of the works on ALP only focus on the static type of the problem. The difficulty of solving dynamic optimization problems in conventional metaheuristic algorithm can be tackled once the algorithm converges the solution (Ertenlice & Kalayci, 2018). Solving dynamic optimization problems is very challenging due to the fact that the parameter values change dynamically during the solving process (Nseef et al., 2016). Moreover, the frequency and magnitude of an environmental change pose challenges to dynamic optimization. According to the literature, d-ALP has not received as much attention as s-ALP. Therefore, this research aims to investigate the performance of the proposed metaheuristic algorithm in solving d-ALP.

1.3 RESEARCH QUESTIONS

i. How can the guided HSA (coded as GHSA) improve the quality of the solution in static ALP, compared with the standard HSA?
ii. How can the multiple neighborhood structures and acceptance of a worse solution (coded as VNHSA) improve the performance of the GHSA in static ALP by avoiding being easily trapped in local optimum?
iii. Is the improved HSA (i.e., VNHSA) applicable to dynamic ALP?

1.4 RESEARCH OBJECTIVES

The main goal of this thesis is to develop an efficient metaheuristic algorithm based on HSA for solving the ALP. To achieve this goal, the following objectives are outlined:

i. To develop a guided HSA (GHSA) that can improve the quality of the solution in static ALP.
ii. To replace the pitch adjustment rate (PAR) with a variable neighborhood search and the probability of accepting a worse solution which avoids local optimum in static ALP to produce VNHSA.
iii. To add an active window to the VNHSA for dynamic ALP.

Table 1.1 Mapping between research questions, research objectives and contribution in this thesis.

Research Question	Research objective	Contribution
How can the guided HSA (coded as GHSA) improve the quality of the solution in static ALP, compared with the standard HSA?	To develop a guided HSA (GHSA) that can improve the quality of the solution in static ALP.	An improved HSA (Guided HSA) with better quality solution (Chapter IV).
How can the multiple neighborhood structures and acceptance of a worse solution (coded as VNHSA) improve the performance of the GHSA in static ALP by avoiding being easily trapped in local optimum?	To replace the pitch adjustment rate (PAR) with a variable neighborhood search and the probability of accepting a worse solution which avoids local optimum in static ALP to produce VNHSA.	An improved VNHSA with an acceptance rate of worse solution and ability to avoid the local optimum solution (Chapter V).
Is the improved HSA (i.e., VNHSA) applicable to dynamic ALP?	To investigate the applicability of the VNHSA to dynamic ALP.	VNHSA solver for d-ALP (Chapter VI).

1.5 RESEARCH SCOPE

This research is concerned with the development of an alternative metaheuristic algorithm (HSA) to tackle the Aircraft Landing Problem (ALP). The development was made by using a guided memory to select the decision variables for improvising a new solution. Further improvement was done by using multiple neighborhood structures and accepting the worse solution. The capability of the improved algorithm was tested on s-ALP and d-ALP. The proposed approach was verified on well-known benchmark datasets from OR-library site. These datasets have been used in several research articles in the literature (e.g., Beasley et al., 2003, 2004; Ghizlane Bencheikh, Boukachour, et al., 2013; Girish, 2016). The results from the proposed approaches were compared with state-of-the-art results and statistical tests were carried out to determine the significant difference in the obtained results.

1.6 OVERVIEW OF THE THESIS

This thesis consists of eight chapters. Chapter I introduces the background of this research, the research problem, the problem statement, the research questions, the research objectives, the scope of research, and finally overviews this thesis. Chapter II reviews the literature on ALP, the methods used to solve the ALP and the available datasets of ALP. It also discusses the metaheuristic algorithms and their variants. Harmony search algorithm and the modifications of harmony search algorithm are examined. The findings from the literature review are also highlighted.

Chapter III describes the methodology used in this thesis including the characteristics of the dataset used.

Chapter IV presents the implementation of the basic and guided harmony search algorithm for static ALP. This chapter also presents the result of the preliminary experiment conducted to determine the appropriate parameter values for HSA. Besides, the modified HSA and the rule-based approach are described. This modification improves the convergence speed of HSA and the quality of the solution.

Chapter V describes the second modification on the HSA achieved by hybridization with variable neighborhood search (VNS) and worst solution acceptance rate. In this modification, the pitch adjustment rate (PAR) which represents a local improvement of the newly constructed solution in HSA is replaced with VNS. Moreover, in the standard HSA, only solutions of higher quality will be accepted, which may lead to a local optimum solution. In this chapter, the HSA algorithm is improved by including the acceptance rate for a worse solution to prevent it from being trapped in the local optimum solution. Different neighborhood operators were used to visit new regions in the search space and to avoid being stuck in local optimum. The modified algorithm is codded as VNHSA.

Chapter VI investigates the applicability of VNHSA to solving the dynamic ALP. In this chapter, the s-ALP is presented, and the formulation of the problem is described. The results of VNHSA in s-ALP is reported as well.

Chapter VII presents the analyses of the proposed approaches. The evaluation of the proposed approaches in this thesis is based on the comparison of the modified version of the proposed algorithms with state-of-the-art results. Statistical tests are also utilized to investigate the significance of the results.

Finally, Chapter VIII summarizes this thesis and indicates future direction.

CHAPTER II

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter presents the background information of the aircraft landing-scheduling problem as it relates to the research questions in the previous chapter. The problem description and the approaches used to solve the aircraft landing-scheduling issue are reviewed. The analyses of recent works in this chapter reveal that a number of issues need to be attended to before the research problem is addressed.

The rest of this chapter is organized as follows: the ALP variants are defined in Section 2.2; the formulation of ALP variants is described in Section 2.3; the optimization approaches used for ALP are detailed in Section 2.4; findings from the literature review on ALP research are revealed in Section 2.5; a review of the HSA and its modifications is given in Section 2.6; the justification for choosing HSA for ALP is described in Section 2.7 and finally, the summary of the chapter is presented in Section 2.8.

2.2 AIRCRAFT LANDING PROBLEM (ALP)

The air traffic management (ATM) system is responsible for the flow of traveling aircraft between the take-off landing areas of two airports. The ATM in an *airport traffic control tower* controls an area of about 5 nautical miles and 3000 ft above ground level from the airport. The responsibilities of the control tower are clearance delivery, gate hold, ground control, ground planning, and runway control.

The next controlling area is the *terminal airspace control centre*, also known as the *approach control airspace*, which handles departures and arrivals of up to 40 nm and 10,000 ft from the airport. Meanwhile, the *en-route control airspace* area handles the traffic flow outside the terminal manoeuvring area. All these control areas are shown in Figure 2.1. This research focuses on the scheduling of the arriving aircraft in the traffic control tower area. This is because of the need for intelligent scheduling techniques to support the decision made by air traffic controllers in order to generate better schedules for landing aircraft.

The scheduling of aircraft in airport operation is considered a challenging task since the 1960s (Soykan & Rabadi, 2016). As such, researchers in the industry have spent considerable effort to find efficient methods to tackle these scheduling operation problems. There are several distinct scheduling tasks in airport operation such as staff scheduling, aircraft take-off scheduling and aircraft landing scheduling. The focus of this thesis is on aircraft landing scheduling.

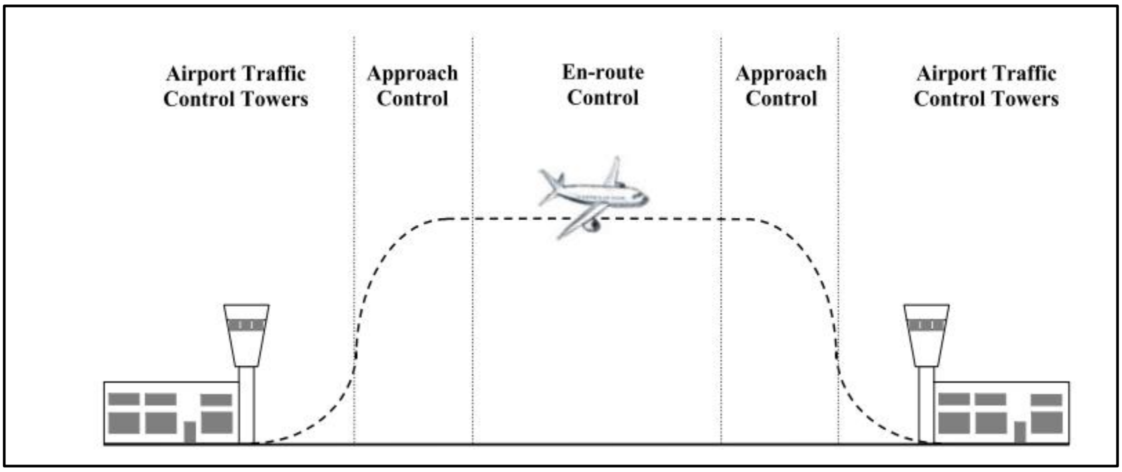

Figure 2.1 Air traffic control areas (Bennell et al. 2013).

Aircraft Landing Problem (ALP) entails the assignment of landing time on specific runways to a sequence of aircraft waiting to land on the terminal area. The runway is a key point in the aircraft-landing problem. Due to economic, environmental and political reasons, it is not always possible to build new runways. Thus, maximizing the runway throughput by reducing the delay associated with the circling aircraft remains the most viable option. Some airports have only one runway (single runway) while others have more (multiple runways). Some studies consider a single runway operation as being particular to landing or take-off while other studies consider both scenarios simultaneously.

Aircraft landing is scheduled at certain time intervals to ensure that safety requirements are met. The need for proper time separation necessitates that the different characteristics of the aircraft should be considered. For example, large size aircraft need more separation time than small size aircraft. The separation time differs for different types of aircraft. The aircraft in the landing sequence make wake turbulence from their engines while landing on the runway. Trailing aircraft should be far enough from the preceding aircraft to avoid the risk associated with wake turbulence. This separation time between the aircraft is handled based on the aircraft' sizes. An illustration of aircraft with different sizes in the landing sequence is shown in Figure 2.2. In the figure, the separation time between the aircraft depends on the aircraft size. The aircraft in the figures are classified into three weight categories (as light, medium and heavy weight) and the separation requirements depend on the categories of the aircraft. A heavier aircraft trailing a lighter aircraft needs less separation when compared with the reverse order as seen in wake vortex separation in Nautical Miles (NM) for different weight categories.

The landing of aircraft is managed by air traffic controller in the terminal area. Each aircraft has a predefined target-landing time predicted based on the flight speed. The landing time must be bounded within a predetermined earliest and latest time values named as Time Window. An aircraft can reach maximum airspeed during its cruise and, in such a case, the aircraft can land at an early landing time. On the other hand, if the aircraft's speed is dependent on its flight fuel-efficient airspeed, it will land at the latest landing time and can hold for a maximum holding time.

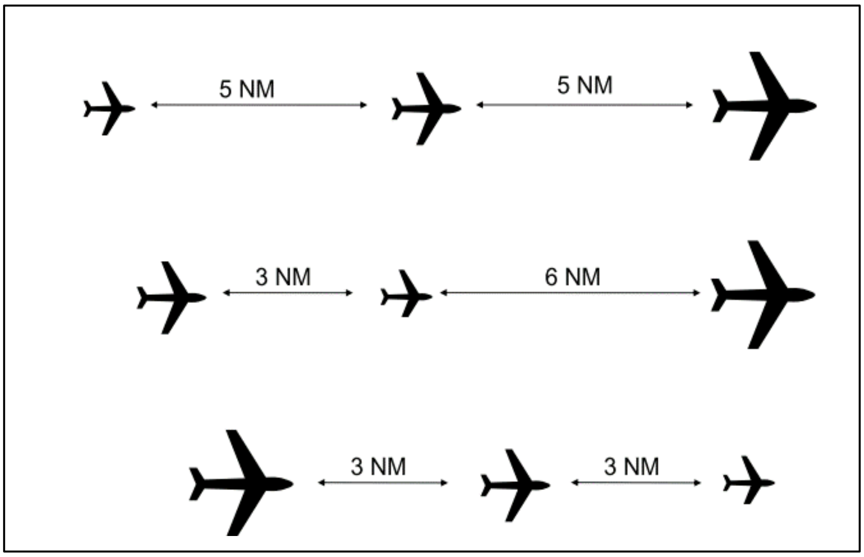

Figure 2.2 Separation time for different types of aircraft.

A change in the predefined information due to change in the cruise speed or hold will lead to an extra cost charged as penalty. This penalty is calculated according to the deviation from the predefined target or (preferred) landing time as shown in Figure 2.3. In the figure, three times (referred to as Earliest, Target and Latest) control the penalty cost of the aircraft. The penalty cost depends on the degree of deviation of the aircraft's landing time from its target time. The penalty cost is zero when the aircraft lands at its target time. However, when the aircraft lands earlier or later, a penalty cost will be charged.

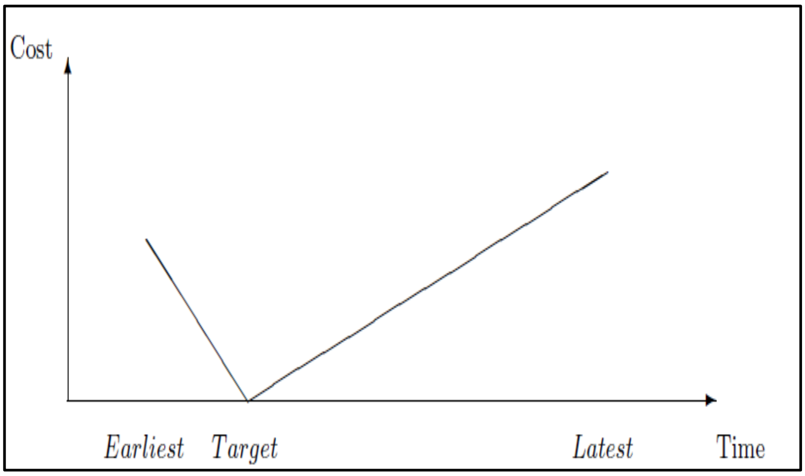

Figure 2.3 Relation between target time and objective function.

Research on ALP has two main directions: the first one is focused on finding an efficient algorithm to generate the landing schedule, while the second is directed at the overall strategies of scheduling automation. This thesis focuses on the first direction i.e., generating the landing schedule using metaheuristic algorithms. ALP is regarded as a non-deterministic polynomial-time hard (NP-hard) problem (Awasthi et al., 2013). ALP is a combinatorial optimization problem that can be solved as a static (off-line) or dynamic (on-line) problem.

2.2.1 Static ALP (s-ALP)

In s-ALP, there is complete knowledge about the set of aircraft waiting to land. The complete information about target landing times and time windows for a given set of aircraft is called the schedule. In the planning horizon, schedules are generated for all the aircraft at once. Furthermore, no changes are expected in the landing times and the constraints will not change during the scheduling process. Most of the works in the literature solved the ALP for the static case (Beasley et al., 2003; Pinol et al. 2006)).

2.2.2 Dynamic ALP (d-ALP)

In d-ALP, the number of aircraft landing is unknown; the decision of assigning landing time and the runway for each aircraft is made as time passes. A new aircraft may appear in the radar at the same time when another aircraft is landing, and no further scheduling can be made (i.e., frozen). This means that the approaching aircraft is available for scheduling at the time of its appearance. Also, it is active for rescheduling up to the moment when its landing time is too close to its current time. At this time, no further changes are possible, and the aircraft must be fixed in the schedule. Research on d-ALP are few as compared with s-ALP. For a better understanding of d-ALP, we refer readers to (Beasley et al., 2004; Ghizlane Bencheikh et al., 2016; Moser & Hendtlass, 2007).

2.3 ALP FORMULATION

As mentioned in the previous section, the aircraft landing scheduling problem is a combinatorial optimization problem consisting of three main components namely aircraft, runways, and landing time. The objective function is to minimize the total penalty associated with the delay of the circling aircraft waiting to land. The problem is represented by a set of arriving aircraft and a number of runways. Each aircraft has a target-landing time according to its most economical speed and it is assigned to a specific runway.

2.3.1 s-ALP Formulation

The formulation of s-ALP follows the static single and multiple interdependence runway version of ALP. The problem formulation in this thesis conforms with the work of Beasley et al. (2000).

a. Notations

The meanings of the notations used in this thesis are as follows:

- n: the number of the arrival aircraft.
- m: the number of runways.
- S_{ij}: the separation time ($S_{ij} > 0$) between aircraft i and j when they are assigned to same runway.
- T_i: the preferred landing time (target time) of aircraft i.
- E_i: the earliest landing time of aircraft i.
- L_i: the latest landing time of aircraft i.
- C_{1i}: the incurred penalty per unit of time for late landing of aircraft i.
- C_{2i}: the incurred penalty per unit of time for early landing of aircraft i.

The most commonly used objective function in the aircraft landing-scheduling problem is to minimize the total penalty by reducing the deviation from the target landing time with respect to landing before or after the target landing time. The objective function is as specified in Equation (2.1).

$$Min\ f\ = \sum_{i=1}^{n}(a_i C1_i + b_i C2_i) \tag{2.1}$$

Subject to:

$$E_i < x_i <= L_i \qquad i = 1, 2 \ldots, n \tag{2.2}$$

$$(x_j - x_i) \geq s_{ij}\delta_{ij} + t_{ij}(1 - \delta_{ij}) - My_{ji}, \tag{2.3}$$

$$i, j = 1, 2 \ldots, n, \qquad i \neq j$$

$$y_{ij} + y_{ji} = 1, \qquad i, j = 1, 2 \ldots, n, \qquad i \neq j \tag{2.4}$$

$$\delta_{ij} \geq y_{ir} + y_{jr} - 1, \qquad i, j = 1, 2, \ldots, n, \qquad i \neq j, \tag{2.5}$$

$$r = 1, 2, \ldots, m$$

$$\sum_{r=1}^{m} y_{ir} = 1 \qquad 1, 2, .., n \tag{2.6}$$

$$y_{ij}, y_{ir}, \delta_{ij} \in \{0,1\} \tag{2.7}$$

$$i, j = 1, 2, \ldots, n, \qquad r = 1, 2, \ldots., m$$

$$x_i, a_i \ , \ b_i \geq 0, \qquad i = 1, 2, \ldots. n \tag{2.8}$$

b. Decision variables

- x_i: the assigned landing time of aircraft *i* (1, 2... *n*).
- y_{ij}: equals 1 if aircraft *i* is assigned to land before aircraft *j*. Otherwise it takes 0.
- y_{ir}: equals 1 if aircraft *i* is scheduled to land on a runway *r*(*r*= 1,2,3…,*m*). Otherwise, it takes 0.
- δ_{ij}: equals 1 if aircraft *i* and *j* are scheduled to land on the same runway. Otherwise, it takes 0.
- a_i : the tardiness of landing when an aircraft *i* scheduled to land after the target time, $a_i = \max(0, x_i - T_i)$.
- b_i: the earliness of landing when an aircraft *i* is scheduled to land before the target time, $b_i = \max(0, T_i - x_i)$.

Each aircraft must land within the time window as specified in constraint (2.2). Constraint (2.3) ensures that the separation time is respected if aircraft *i* and aircraft *j* landed on the same runway. Constraint (2.4) indicates whether aircraft *i* lands before *j* or aircraft *j* lands before aircraft *i*. Constraint (2.5) ensures that when both aircraft *i* and *j* are assigned to the same runway, the runways must be identical. Constraint (2.6) ensures that each aircraft is allocated to only one runway. Constraint (2.7) ensures that the decision variables y_{ij}, y_{ir}, and δ_{ij} are binary. Constraint (2.8) ensures that x_i, a_i, and b_i are nonnegative. In the above-mentioned formulation, when the number of the runways m =1, the ALP will be formulated as a single runway and there is no option of selecting multiple runway. This is depicted in Figure 2.4.

Aircraft ID	1	2	3	…	…	N-1	N
Landing Time (X)	150	160	782	…	…	X(N-1)	X_N

Figure 2.4 Single Runway Solution Format

The first line in Figure 2.4 represents the aircraft's *ID* starting from 1 to *N*, where *N* is the total number of aircraft. The second line represents the landing time (*x*) of each aircraft in the first line. When the number of runways is more than one, the sequence of the solution structure for the runway number of each aircraft will be different. This is shown in Figure 2.5, where the first line represents the aircraft *ID* starting from 1 to *N*, the second line represents landing time (*x*), and the third line represents the runway number *R*.

Aircraft ID	1	2	3	…	…	N-1	N
Landing Time (X)	150	160	782	…	…	X(N-1)	X_N
Runway No. (R)	1	2	3	…	…	R-1	R

Figure 2.5 Multiple Runways Solution Structure.

The number of runways is an important parameter in the literature. ALP is solved as a single and multiple runway-scheduling problem. In the case of a single runway, all aircraft must be scheduled to land on one runway. Separation time must be considered between each pair of the aircraft depending on the aircraft type or class. In the case of multiple runways, the arriving aircraft are scheduled to land on multiple runways. Different runway selection mechanisms are considered by different researchers. Here, the type of the runway refers to the construction of the runway and the operation that can be served on the runway. There are two essential types of runways: interdependence and heterogeneous (Lieder & Stolletz, 2016). With interdependence runways, the operations on one runway restrict the operations on other runway(s). On the other hand, with heterogeneous runways, not all operations can be performed on all the runways. Other information about the ALP will be detailed in the following subsections.

2.3.2 d-ALP Formulation

In the dynamic ALP (d-ALP), the formulation of the problem differs from the static case as the number of the aircraft is unknown a priori. The objective function of both cases (s-ALP and d-ALP) is the same i.e., to minimize the total penalty by reducing the deviation from the target landing time. The same notations are used in both s-ALP and d-ALP, except for some additional notations and constraints highlighted as follows:

a. Notations

- *A*: appearance time of aircraft *i*.
- *C*: current time.
- *Fz*: freezing time.

b. Constraints

In addition to the constraints in s-ALP, d-ALP has the following additional constraints which are related to the dynamic environment:

$$i = \begin{cases} 1 & if \quad A_i = C_i \\ 0 & if\ A_j \neq C_i \end{cases} \tag{2.9}$$

$$C_i - x_i < f_{zi} \tag{2.10}$$

Constraint (2.9) ensures that the aircraft will be active to scheduling if its appearance time A_i is equal to the current time C_i. Constraint (2.10) ensures that aircraft *i* must be frozen when its freezing time f_{zi} is close to the current time. In other words, aircraft *i* will be frozen with there is no swapping or time shifting. In Constraint (2.10), x_i is the landing time of aircraft *i*, fz_i is the freezing time of aircraft *i* and C_i is the current time of aircraft *i*.

2.4 APPROACHES APPLIED TO s-ALP AND d-ALP

This subsection reviews the approaches used for solving the s-ALP and d-ALP. According to the literature, the approaches used for s-ALP and d-ALP can be mainly classed as exact approaches and metaheuristic approaches. The literature review in this thesis covers the research articles that applied their approaches using OR-Library dataset introduced by Beasley (1990). Several researches work in the literature applied their methods on this dataset. Additionally, works that used other datasets are reviewed. In the next subsections detail these approaches.

2.4.1 Approaches Applied to s-ALP on OR-Library Dataset

Here, we review previous studies on s-ALP and highlight the main methodologies used. A summary of the methods that have been applied to s-ALP using OR-Library dataset is given in Figure 2.6.

a. Exact Methods

In exact methods, the proposed algorithms search through the entire search space of the problem. The main classes of the exact methods are dynamic programming, branch and bound, and constraint programming (Talbi, 2009). These methods are used when the algorithm can visit all the search spaces of the problem.

Dynamic programming is an optimization approach that converts a complex problem into a sequence of simpler problems. Its essential characteristic is the multistage nature of the optimization procedure. Dynamic programming provides a general structure for considering problems of diverse forms. Within this structure, a range of optimization methods can be deployed to solve specific aspects of a more general construction.

Branch and bound algorithm is based on an implicit enumeration of all the solutions for the considered optimization problem (Talbi, 2009). This class of methodologies is widely used to solve optimization problems within an exact routine. In branch and bound methods, the search space is explored by dynamically constructing a hierarchy tree, where the source node represents the problem being solved and its whole related search space. The leaf nodes are the possible solutions and the inside nodes are the sub-problems of the entire solution. The size of the sub-problems is gradually reduced as one approach the leaves.

Ernst et al. (1999) deployed the specialized simplex algorithm as well as the branch and bound method to solve s-ALP for single and multiple runways. The specialized simplex algorithm was used to find the landing time for each aircraft while the branch and bound found the upper and lower bounds. The branch and bound algorithm were then used to generate solutions to both single and multiple runway problems. The experiments were conducted using 44 aircraft (instances number 1–8).

ALP was formulated as a mixed-integer zero–one MIP program. It proffers solution statically using tree search in Beasley et al. (2000). A single runway formulation was initially developed and later extended to multiple runways. The impact of such formulation on the derived solution and the solution's quality was deeply discussed. In addition, a heuristic algorithm was used to solve the problem. In both cases, 50 aircraft were considered in the evaluation of the proposed algorithms.

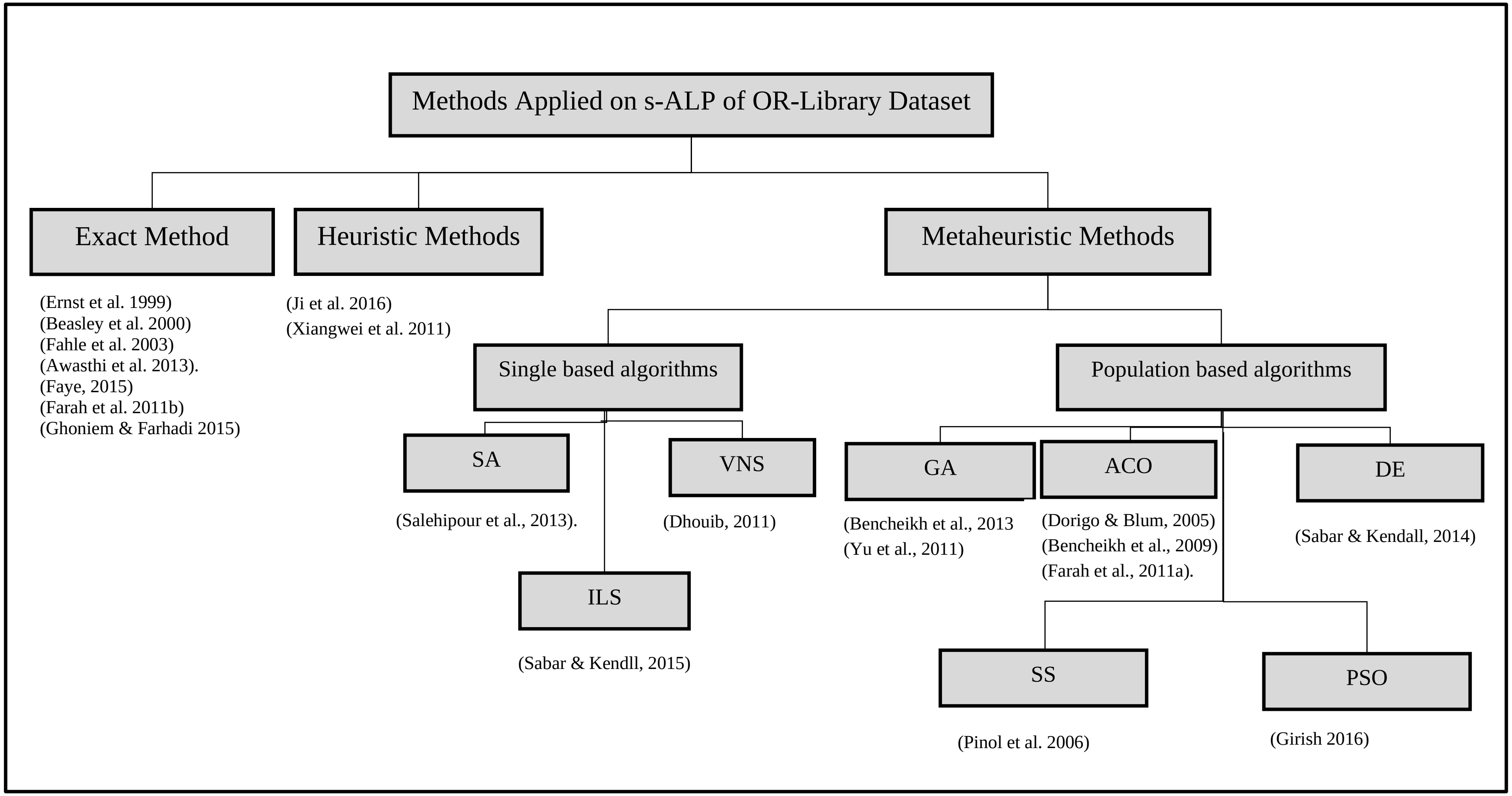

Figure 2.6 A summary of methods applied for s-ALP problem of OR-Library dataset

Fahle et al. (2003) employed several exact methods for ALP. Particularly, two integer programming models, Constraint Programming formulation and Satisfiability problem, were used in this work. The results of these methods were compared with each other and with local search methods. Approximately fifty aircraft were used in the experiments performed. Results confirmed the superiority of the ASP models with multiple runways.

The work of Farah et al. (2011b) consists of two parts. A new modelling method was presented for the static single runway ALP based on a quadratic model in the first part. The second part used an exact method based on the branch and bound method to solve the problem. In the context of modelling, binary variable was used to show the first aircraft in the landing sequence while integer variable represented the arrival time. The branch and bound method used for the ALP consists of three steps: the initialization of the solution, application of the principles of the branch and bound method, and attainment of the best solution. Small sized instances were considered as only 44 aircraft were used in the experiment.

Another exact method, named a polynomial exact algorithm, was used for single and multiple static ALP in the work by Awasthi et al. (2013). The proposed exact algorithm achieves a feasible landing sequence on single or multiple runways and solves only the landing time assignment part of the problem. After the proposed algorithm completes the initialization step, it reduces the landing time of the aircraft in form blocks while maintaining the safety constraints. The authors defined a mathematical procedure to continually reduce the assigned landing time. An exception to this procedure is when one of the aircraft in the block is scheduled on its early landing time or the safety constraint is violated. This experiment considers single and multiple runways separately. Approximately 500 aircraft and runways ranging from 1 to 5 (in the case of multiple runways) were used in this work. The results obtained were compared with the results of Pinol et al. (2006).

Faye proposed an approximation of the separation time matrix and on the time discretization approach to solve ALP (Faye, 2015). In his approach, the separation time is approximated by a rank two matrix. The ALP is stated as a 0-1 integer problem. When the separation time matrix is not a rank two matrix, an approximation is made. This provides lower bounds or upper bounds depending on the choice of the approximating matrix. These bounds were used in a constraint generation algorithm to, optimally or heuristically, solve the problem. The results obtained using this exact method were compared with the results of Beasley's work in Pinol et al. (2006). Based on the comparison, this approach outperformed Beasley's work in some scenarios.

Ghoniem and Farhadi (2015) investigated the computational tractability and the relative merits of a 0-1 MIP. A set partitioning formulation was solved using the column generation approach for ALP. The proposed method deploys an objective function which is slightly different from the other approaches where the objective function is to minimize the total weighted start-times. The dataset used includes 10 to 500 aircraft and 1 to 5 runways. The result, compared with the result of Pinol et al. (2006), shows a clear improvement over Pinol et al. (2006).

b. Heuristic Methods

A heuristic is a logic-based algorithm considered to work rapidly and deliver very efficient solutions but may not be globally optimal. A heuristic based on sliding window algorithm adapted from receding horizon control was proposed for ALP (Xiangwei et al., 2011). The algorithm starts by sequencing the aircraft (in waiting to land) in the FCFS order based on their target landing time. The aircraft are defined in a window and scheduled optimally in terms of their landing time. The landing time in this work is a key factor of the optimization algorithm. About 500 aircraft and 5 runways were used in the experiment. The result, compared with the work in Pinol et al. (2006), showed that the sliding window algorithm outperforms the scatter search algorithm and bionomic algorithm presented in previous literature in terms of execution times and solution values.

Ji et al. (2016) developed an algorithm named: a sequence searching and evaluation algorithm. In this study, the authors classified the different formulation of the ALP studied in previous literature. The algorithm tackled the ALP in two parts.

First, a group of sequences is generated by ignoring the constraints temporarily. In the second part, the landing time is optimized, and the solution is evaluated. Fifty aircraft and a runway were used in this study. This work was not compared with the best-known result in the literature at that time.

c. Metaheuristic Methods

Metaheuristic algorithm is the most successful class of optimization algorithms. In the last 30 years, metaheuristic algorithms have been utilized in several optimization problems in areas such as engineering design, medical applications and transportation. Metaheuristic algorithms are nature-inspired or bio-inspired phenomena. The nature-inspired algorithms mimic the behaviour of nature while bio-inspired algorithms mimic behaviours of humans and lower animals. The two main classes of metaheuristic algorithms are single-based solution and population-based solution algorithms.

The vital features in any metaheuristic algorithm are the abilities to explore the search space regions and exploit the specific area for local improvement. Exploration and exploitation (some researchers refer to them as diversification and intensification, respectively) are the most crucial characteristics in any optimization algorithm. Any algorithm with a balanced rate between these two characteristics performs well. There are different factors affecting these characteristics from one algorithm to another. The single-based algorithm is highly effective in intensifying the search area but poor in diversifying it. Conversely, the population-based algorithm performs efficiently with respect to search space diversification, but not intensification. Different procedures have been proposed to guarantee a balance between these characteristics because the absence of one of these characteristics will lead to a convergence problem in the algorithm.

i. Single-Based Metaheuristic Algorithms

In single-based metaheuristic algorithm (also known as Local Search), one solution is improved iteratively to solve the targeted optimization problem. The search for a better solution is represented as a walk through the search space using the search operators.

The general procedure of the single-based solution algorithms is to iteratively replace the current solution when a better solution is obtained through the search process. The search operators generate a neighborhood to visit different areas in the search space. Next, we review the works on S-ALP using single-based metaheuristic algorithms.

Simulated Annealing

Simulated Annealing (SA) is an approach which originates from statistical mechanics. It is based on a Monte Carlo model that was used by Metropolis et al. (1953) to simulate energy levels in cooling solids (Kirkpatrick et al. 1983). Starting from a random point in the search space, a random move is made. A move to a neighbouring point from the present solution s_0 is recognized if it either enhances the value of the current fitness (objective) function or ignores it. If a solution having the worst fitness value is generated, the solution is accepted with the probability, $P_{(t)}$ = e-δ/t, that depends on the magnitude of the deterioration δ (which the difference in the objective function values i.e., δ = f (s)-f (s_0)) and temperature *t*. The function $P_{(t)}$ has a value close to 1 at the beginning (because of high temperature), but gradually decreases to zero (when t = 0), with the cooling of a solid. Initially, any move is accepted, but as the "temperature" reduces, the probability of accepting a negative move is lowered. Negative moves are essential to escape from the local minima. However, too many negative moves will simply diverge from the global minimum. Unlike the random search, SA handles only one candidate solution at a time and does not build up an overall picture of the search space. No information is saved from the previous moves to guide the selection of new moves.

SA starts with the initial solution x and improves on it by generating a candidate solution x'. If the quality of solution x' is better than *x*, then it is accepted. Otherwise, solution x' will be accepted based on the acceptance criterion that depends on a certain probability r (r is a random number between 0 and 1). If r is less than the value of $e^{-\Delta E/T}$ (where $-\Delta E$is the difference between the candidate and the current solutions and T is a control parameter called temperature), then the candidate solution is accepted, and the temperature is lowered gradually according to a cooling schedule (β is the cooling rate). This process is repeated until the termination criterion is reached.

A combination of SA with Variable Neighborhood Descent (VND) and Variable Neighborhood Search (VNS) was introduced for S-ALP with multiple runways in the work of Salehipour et al. (2013). In this work, different neighborhood structures were used to improve the solution quality. Both the VND and VNS algorithms were used with the neighborhood structure which was controlled by the SA framework to avoid trapping in local optima. The dataset used in this work ranged from 10 to 500 aircraft and 1 to 5 runways. Furthermore, the s-ALP was solved using an exact method while using CPLEX software to measure the performance of the proposed metaheuristic algorithms. The results obtained from the combined algorithms (SA+VND and SA+VNS algorithms) and the CPLEX result were compared with the Scatter Search result from Pinol et al. (2006). According to the author, that was the first time an optimal solution for 100 aircraft was obtained. Prior studies had only achieved optimal solutions for a maximum of 50 aircraft.

Variable Neighborhood Search

Variable Neighborhood Search (VNS) is a recent metaheuristic algorithm successfully verified to tackle combinatorial optimization problems (Gutman et al., 1997). Its basic idea is to randomly or systematically explore a set of pre-defined neighborhoods, typically arranged one after the other, both in finding local optima as well as escaping from them. VNS basically uses the fact that a global optimum links to a local optimum for a specific neighborhood. When it deals with different neighborhoods, different landscapes in the search space will be generated to improve the chances of finding the optimal solution.

Multi Variable Neighborhood Search was used to solve the ALP with multiple runways (Dhouib, 2011). The author followed the procedure of Pinol et al. (2006) to represent the solution . Three types of movements were used to improve on the most recent solution. These are modifying the landing time of the aircraft, modifying the sequence order of the aircraft on each runway, and modifying the runway allocation for the aircraft. The objective function used is similar to most related works i.e., to minimize the total delay of the aircraft with respect to its target landing time. In this work, multi-start techniques were used to overcome the problem of diversification in local search algorithms. Another improvement on the proposed algorithm was implemented by employing memory to prevent the algorithm from visiting neighborhoods that have already been visited with no improvement. The average deviation from the best-known result taken from Pinol et al. (2006) is 2.3%, and 2.1% and 1.7 % for Scatter Search and Bionomic Algorithm, respectively.

Iterated Local Search

Iterated Local Search (ILS) is a single-solution metaheuristic algorithm that iteratively finds an optimal solution (Stützle & Ruiz 2018). ILS starts with an initial solution then uses a local search algorithm to find a local optimum. Therefore, it is a perturbation operator to modify the current local optimum in order to avoid the local optimum and to move to alternative point in the search space.

Sabar and Kendall (2015) proposed an Iterated Local Search with multiple perturbation operators and time-varying perturbation strength to solve ALP. The success of the proposed algorithm depends on the local improvement. Local optima problem is a common drawback for most of the local search algorithms. To solve this problem, the author proposed four perturbation operators with a time-varying perturbation strength. Variable Neighborhood Decent (VND) algorithm was used for the local search procedure. In this work, the initial solution was generated using the Randomized Greedy (GR) heuristic. Then, four neighborhood structures were used for the local search result improvements while considering swaps and moves. Afterward, the Perturbation phase was invoked. This phase involves four different perturbation operators including swaps and moves. The dataset instances used in this work range from 10 to 500 aircraft and 1 to 5 runways. The results of this work were compared in terms of the effectiveness of the perturbation operators on the performance of the proposed ILS, where the ILS was tested with each perturbation operator separately. Additionally, the result was compared with different state-of-the-art references by (A. T. Ernst et al., 1999a; Salehipour et al., 2013) in terms of the gap from the best-known values (BKV) works such as Pinol et al. (2006). The author reported that this work gives new best results in 16 out of 49 instances. A comparatively equal performance was achieved for the remaining 33 instances.

ii. Population-Based Metaheuristic and Hybrid Algorithms

In population-based metaheuristic algorithms (also named as Global Search), the search algorithm starts by populating the solutions and iteratively improving the quality of this population. This is achieved by generating a new population and integrating it with the old one based on search operators. Names of the algorithms usually come from a specific behaviour or phenomena. Genetic algorithms, evolutionary algorithms, ant colony, scatter search, particle swarm optimization and bee colony are examples of this class of algorithms.

Recently, another class of optimization had emerged from the combination of two or more algorithms known as hybrid algorithms. In the hybridization of two or more algorithms, an algorithm is employed to do a specific procedure in cooperation with another algorithm. The idea behind the hybridization is to exploit the strength of an algorithm to treat the weakness of the other algorithm. An example of hybridization is the use of a local search algorithm to increase the intensification of the global search algorithm. In this section, the frequently used algorithms and research works on s-ALP using the population-based metaheuristic algorithms are methods discussed. Besides, an exposition is given on the hybrid methods.

Genetic Algorithm

Genetic Algorithm (GA) represents one of the most known metaheuristic algorithms and was introduced by Goldberg & Holland (Holland 1992). GA has been very popular as it is one of the earliest discovered methods capable of generating efficient solutions for highly complicated optimization algorithms within a reasonable time. The main principle of GA is the survival of the fittest, sometimes called natural selection, where the best solutions are allowed to breed with each other. The general procedure of GA begins either with a population (P) of solutions created arbitrarily or via a specific version of heuristic algorithms. The fitness of these solutions is then calculated. Subsequently, the process of selection according to the fitness value is carried out to choose two solutions as the parents.

Bencheikh et al. (2013) proposed a hybrid of two metaheuristic algorithms to solve the static ALP with multiple runways. The author reported that the best solution for ALP was obtained by hybridizing algorithms and not by using a single algorithm. Genetic Algorithm (GA) and Tabu Search (TS) algorithm were used in the proposed work with different hybridization scenarios which included the use of TS to improve the solution obtained after the selection stage in GA. In addition, crossover and mutation process were also utilized to improve the solutions. The data used in the experiment ranged from 10 to 500 aircraft and 1 to 5 runways. The result obtained from the proposed approach was compared with the work by Pinol et al. (2006).

GA with Cellular Automation (CA) was introduced for modelling the ALP (Yu et al., 2011). This work was achieved in several steps. CA and GA were used to generate and optimize the landing sequence, respectively. The GA deploys a mutation operator for optimization. Also, a Relaxation Operator (RO) was used to generate the landing time for the obtained landing sequence from CA. A total of 13 datasets were used with a single runway in the experiment. The result obtained was compared with the result of Pinol et al. (2006). For the large size instances (Instance 9- 13), some of the instances performed better than Pinol et al. (2006).

Ant Colony Optimization

Ant Colony Optimization (ACO) is a population-based metaheuristic algorithm inspired by the movements of ants during their search for food (Dorigo & Blum, 2005). The concept of the algorithm entails cooperation and the sharing of information to find the shortest path towards the source of food. The ants use a chemical pheromone trail on the route to the source of food so as to guide the other ants to go along the same way and to avoid unknown routes. The level of the chemical pheromone indicates the distance of the route towards the food source. As a metaheuristic algorithm, ACO initializes the population of solution (ants) either randomly or heuristically. The pheromone is initially set as a small constant value and is updated according to the quality of solution achieved during the search process. This process is iterated until the stopping criteria are achieved.

In the study of Bencheikh et al. (2009), two population-based metaheuristic algorithms were combined to solve the ALP namely, the ACO algorithm and the GA. The ACO was used to generate the initial population and GA was used for the improvement process. The purpose of using ACO is to generate high-quality initial population because the GA performance strongly depends on the quality of the initial population. The author formulates the ALP as a job shop scheduling problem. In this respect, static ALP with multiple runways was solved. The result was evaluated using 50 aircraft (instances 1-8). The obtained results were compared with the optimal solution reported in earlier works.

ACO algorithm was proposed to solve the ALP for multiple runways (Farah et al., 2011a). In the thesis, a static single runway problem was solved. The approach deploys the quadratic model to describe the problem. It was solved using ACO, modelled as a set of ants (solutions) moving through the ALP stats. Each solution has two vectors representing the landing sequence and the landing time. The objective function was to minimize the total delay of the arriving aircraft. The total number of aircraft used in the work is approximately 150 aircraft (instances Airland1 to Airland10).

Differential Evolution

Differential Evolution (DE) is a population-based algorithm introduced by Storn and Price (1997). DE was first proposed to deal with continuous optimization problems or to optimize real parameter and real valued functions. The principle of DE is based on two main operators (mutation and crossover). The mutation operator selects different solutions from the population and integrates them to produce a new solution. Then, the produced solution is combined with the initial solution based on the crossover operator. If the quality of the new solution is better than the initial solution, it will be added to the new population.

In the study of Sabar and Kendall (2014), the Hybrid Differential Evolution (DE) and Simple Descent (SD) Algorithms were implemented to solve the ALP. The nature of the DE algorithm is that it deals with continuous optimization problems and thus cannot be used directly to solve the ALP. In this respect, the authors proposed a new representation method for the decision variables of the ALP. The integer part represents the number of runways and the fraction represents the order by which the aircraft land on the runway. The first step involves representing the solution using DE. This necessitates that the number of runways are represented using decision variables. Note that the aircraft land on these runways randomly. In the next step, the aircraft sequence on each runway was sorted in ascending order and was given landing time based on their target landing time.

Commonly, DE improvement depends on the crossover and mutation which may negatively affect the quality of the solution in ALP as small movements in the order of aircraft can cause significant changes. Thus, after deploying DE for crossover and mutation, SD was used to improve the solution. The improvement in SD was implemented by examining the neighborhoods using move operators. By combining the proposed algorithms, the DE explores the search space efficiently while SD exploits it. The proposed approach was tested using all the instances of the dataset in the 13 files with 500 aircraft and 5 runways. The obtained results were compared in two facets. In the first, the DE performance was compared with DE-SD. The second aspect of the comparison was made to the works by Salehipour et al. (2013) and Pinol et al. (2006) .Note that the BKV result was taken from Pinol et al. (2006). The authors indicated that this work yielded new best results in some instances.

Scatter Search Algorithm

Scatter Search (SS) is a deterministic method that has been utilized successfully on several continuous and combinatorial optimization problems. SS was introduced for the first time by Glover (1977). As an evolutionary and population metaheuristic, SS recombines solutions taken from a reference set to create more solutions. Firstly, an initial population (Pop) that fulfils all the diversity and quality criteria is generated. Next, a reference set (RefSet) with a moderate population size is created by selecting good representative solutions from the population. The combination of the selected solutions is then used as the starting solutions for an S-metaheuristic based improvement procedure. Based on the results of the procedure, an update to the reference set and even the population of solutions is carried out to include high-quality and diversified solutions. The process is iterated until the stopping criterion is fulfilled.

Pinol et al. (2006) introduced two different heuristic techniques (scatter search and bionomic algorithm) for the s-ALP problems. These heuristic techniques were formerly used by researchers but have not been applied to ALP before this research. Mixed-integer zero-one formulation was used for the multiple runways s-ALP. The authors deployed the algorithms on about 500 aircraft and 5 runways. The results of the algorithm were compared with each other and with the result from FCFS. The results showed that the scatter search algorithm outperformed bionomic algorithm.

Particle Swarm Optimization

Particle swarm optimization (PSO) is another nature-inspired swarm intelligence algorithm which imitates the flock of birds (Eberhart & Kennedy 1995). In PSO, each candidate solution is represented by a particle, and the flock of particles moves in the search space to find the global optima. To direct the search to the optimum part in the search space, the swarm of particles is preserved during the search procedure, and the particles cooperate by sharing their archives among others. The particles move in multi-dimensional search space with a specific velocity (*v*) and trail the recent best particles. Each particle changes its position according to its prior searching knowledge and that of other particles. The particles relocate their positions to the next positions with a velocity using particles fittest solution and global best solution values. The particles' positions change depending the best positions found by the particle itself (*pbest*) and the best position found by the whole swarm (*gbest*). The algorithm performs local search with global search techniques to achieve well balanced exploitation and exploration.

PSO algorithm with local search was proposed for s-ALP with multiple runway by Girish (2016). In the proposed algorithm, the authors separately dealt with single and multiple runways in the construction of the initial solution and the optimization process. Girish also proposed the rolling horizon (RH) framework, which represents an online optimization strategy. RH optimizes a problem for a fixed time horizon framework based on the available information about the aircraft. The proposed approach was evaluated by comparing it with the following eight papers from the literature. (Awasthi et al., 2013; Bencheikh, et al., 2013; Sabar & Kendall, 2015; Salehipour et al., 2013; Xie et al., 2013; Yu et al., 2011: Zhou et al. 2018). The number of aircraft used ranged from 10 to 500 and the runways ranged from 1 to 5. The result of the proposed HPSO reflected the efficiency of the approach.

Bat Algorithm

Bat Algorithm (BA) is a population-based metaheuristic algorithm introduced by Yang (2010). It is based on the echolocation activity of bats in the natural world. Echolocation is the making of very loud sound waves and echoes to recognize where objects are in space. When sound waves sent by bats hit an object, they generate echoes, which return to the bat's ears. Bats listen to the echoes to understand where the object is, its size and its character. Bats have this ability in the darkness.

Bat Algorithm combined with local search was introduced for s-ALP with multiple runways (Xie et al., 2013). Hybrid Bat Algorithm (HBA) consists of the population-based algorithm BA and two local search algorithms. In this work, the solution representation has an important role in the improvement process. The purpose of the local search is to improve the global best solution. An improvement procedure was implemented by randomly selecting a runway from the solution, randomly selecting an aircraft on this runway, and assigning the selected aircraft to a different runway. This was done in one local search. Another improvement procedure was implemented by mutating the current global best solution to ensure the diversity of the population. The process was iterated until the termination criterion was met. The dataset used in this work ranged from 10 to 500 aircraft and 1 to 5 runways. The result obtained by HBA was compared with the results obtained by Pinol et al. (2006) and Bencheikh et al. (2011).

2.4.2 Approaches Applied to d-ALP on OR-Library Dataset

Research works on d-ALP consider the scheduling of aircraft in the landing sequence when new aircraft appear over time, which requires that the landing of the previous aircraft should be rescheduled. In the next subsections, the research works on d-ALP are reviewed.

a. Exact Method

Beasley et al. (2004) proposed a new method named d-ALP, to solve ALP dynamically as a displacement problem. In d-ALP, the landing time of the aircraft is assigned while considering the time pass and change in the environment. While solving the displacement problem, new constraints or values may change during the process. This requires that a new decision must be made while taking the former decision into consideration. In d-ALP, a new data is added to the problem such as the appearance of a new aircraft or the landing of the new aircraft. The new schedule must have a back link to the previous one. Tree search method was used to dynamically generate the solution for ALP with multiple runways. Small and large datasets (up to 500 aircraft) were used in the experiment.

b. Heuristic Method

In Extremal Optimization (EO) dynamic single runway ALP (Moser & Hendtlass, 2007), ALP is solved in two parts by finding the sequence of the aircraft and optimizing the landing time for the aircraft' sequence. EO is an optimization technique which depends on the mutation operator to improve the solution. The landing sequence resulted from the EO solver used as input for the algorithm in the landing time assignment. There were 500 aircraft used in the experiment with a single runway. The result obtained was compared with the result from Beasley et al. (2004).

c. Metaheuristic Methods

Bencheikh et al. (2016) proposed the ant colony optimization algorithm combined with the local search for solving ALPs in scenarios with dynamic multiple runways. In the d-ALP, new aircraft may appear as time passes. The proposed algorithm begins by initializing the ants constructively, starting with aircraft selection, runway allocation, and landing time assignment. The solution was represented following the bi-level graph. The graph starts with dummy nodes representing its input and output. Dataset instances used in the evaluation of the proposed algorithm ranged from 10 to 50 aircraft and 1 to 4 runways (instances 1 to 8). The result of the proposed approach was compared with the result of Beasley et al. (2004).

2.4.3 Research Works on ALP using Real-World Problems

This section reviews the most common optimization techniques for ALP using real world datasets collected from airports. This review will also tackle the departure problem caused by the high similarity between the departing and landing aspects as an optimization problem. The review is summarized in Figure 2.7.

Since the fifties, there have been attempts to improve the capacity of runways in airports by improving the management of air-traffic controller tasks. Atypical example can be found in the work by Blumstein (1959), where an analysis was carried out in the terminal area of the New York airport. This analysis proved the possibility of achieving significant improvements through the reduction of separation at the gate. Blumstein introduced an analytical model for determining the landing capacity for any single runway.

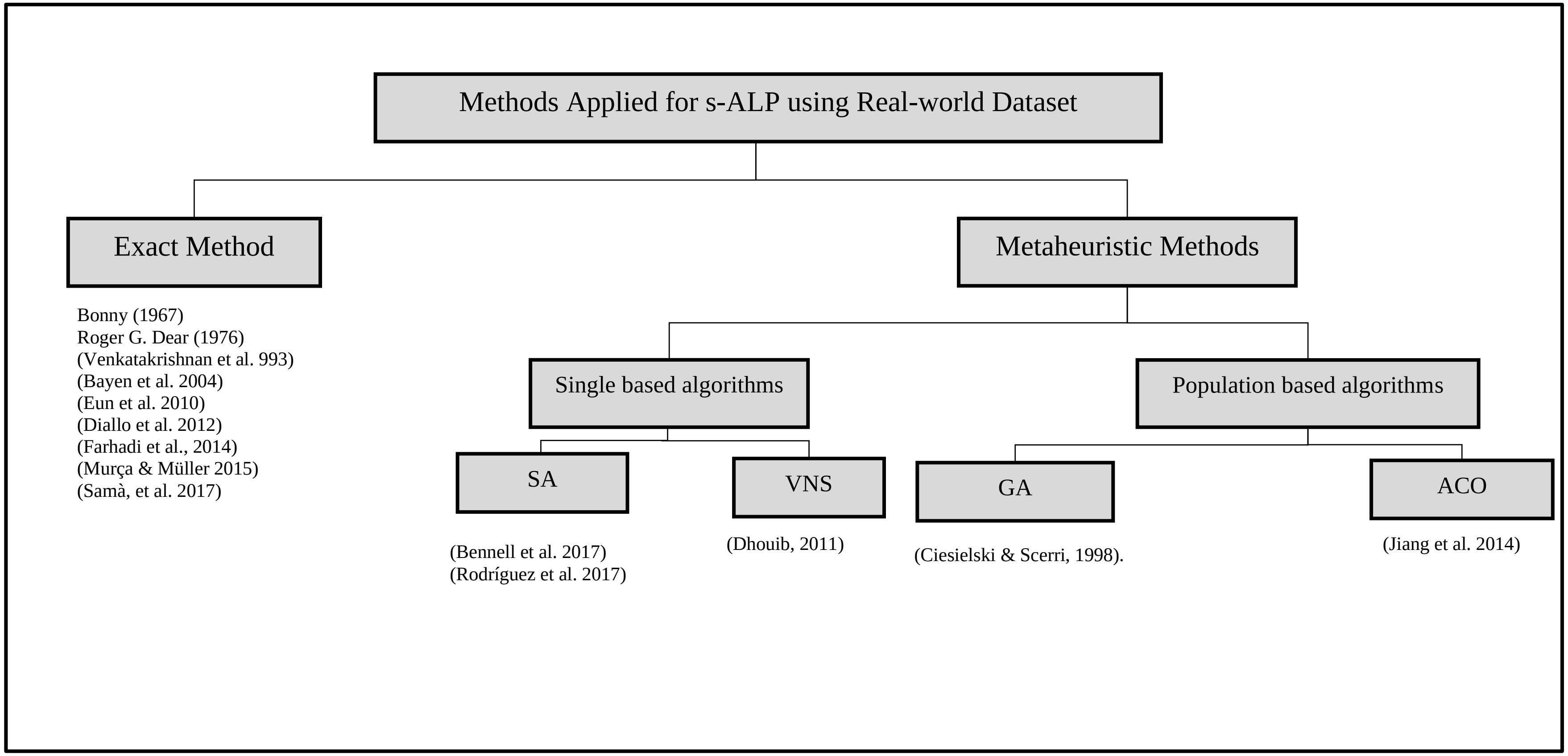

Figure 2.7 A summary of methods applied for s-ALP problem of real-world dataset

a. Exact methods

Bonny (1967) introduced a computer-assisted sequencing for the task at Heathrow London airport. This is one of the most prominent works which established the use of computers in assisting the controller in the radar area. The system was deployed in the Royal Radar Establishment.

Roger G. Dear (1976) stated that the performance of the decision makers in the control area would not be efficient enough and the maximum throughput will not be achieved by depending on the FCFS discipline. Thus, a new decision methodology was proposed in this work named Constrained Position Shifting (CPS). The idea behind this methodology ensures that aircraft do not shift away from the FCFS order.

Venkatakrishnan et al. (1993) developed a statistical model to deal with ALP at Boston's Logan Airport where two problems had emerged. The first problem was the large gap between the aircraft in the arrival stream while the second was the sequencing problem in the terminal area. The model developed in this work managed to tackle these problems by controlling the time interval between the arriving aircraft.

Bayen et al. (2004) proposed a dynamic programming method to improve the spacing of arriving aircraft at the Dallas/Fort Worth International Airport. The problem studied is a little different from those presented in other research papers. The problem requires computing the maximum and minimum spacing for the landing aircraft. The result of this work was compared with the result of the Trajectory-Cantered Simulation method developed by NASA.

Eun et al. (2010) proposed a branch-and-bound algorithm with linear programming (LP) and Lagrangian dual decomposition. Aircrafts were categorized as heavy, large and small to effectively formulate the problem while considering the safety time between the aircraft. The objective function is the total cost of all the delayed or advanced aircraft from their target landing time. The algorithm starts branching based on the tree search. The bounding of each branch was calculated using the linear programming-based mathematical formulation. The algorithm performance was evaluated using real life data from Gimpo Airport (GMP) which is the biggest airport in South Korea. Results show that the proposed algorithm has significant computational performance, particularly for jammed airspace states.

Diallo et al. (2012) used another exact method to schedule the arrival of aircraft. They considered a single runway using the mixed-integer 0-1 formulation. Akin to most previous works, the objective functions used were the minimization of the total delay of the holding aircraft and an assigned landing time very close to the target time. The proposed algorithm was evaluated using real life data from the Léopol Sédar Senghor (LSS) airport in Dakar, Senegal. The result was compared to the ASECNA system, which is used in the management of air traffic. The algorithm was able to maximize the throughput of the runway system, hence minimizing the cost of deviation from the target times.

A recent empirical study was carried out to assess the runway capacity in AL-Doha airport in Qatar (Farhadi et al., 2014). This study deployed another exact method, the Mixed-Integer formulation, for both landing and take-off operations. Three factors were considered to increase the runway capacity, namely the runway configuration, the scheduling approach, and the separation interval standard between the aircraft (see Figure 2.8).

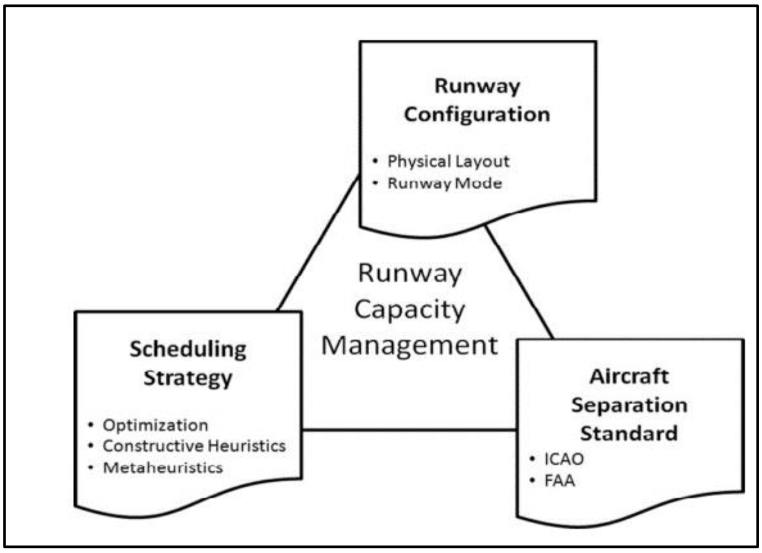

Figure 2.8 Runway capacity factors (Farhadi et al. 2014)

The novel contribution in this study is the consideration of the runway configuration as another factor for increasing the runway’s capacity. This study also investigated the two-separation approach based on the rules of the Federal Aviation Administration (FAA) and the International Civil Aviation Organization (ICAO). In addition to the aforementioned investigations, single and two runway cases were also studied: the single runway case in Doha International Airport, DOH and two runway cases in Hamad International Airport, HIA. The Mixed-integer method was used to present the notation of the problem and the objective function was hinged on the total fuel cost resulting from the deviation of the aircraft from their target landing time.

The result of the proposed approach indicated that fairness was achieved in the aircraft scheduling. It was also observed that shifting the FCFS sequence beyond two possessions at the Doha International Airport is not required. Additionally, the proposed approach yielded optimal or near-optimal solutions. This corresponds to a considerable optimization of fuel usage and decreased delay. The result also proved that international airports such as the Hamad International Airport can benefit more from using the FAA aircraft separation standard instead of the ICAO rules.

Murça and Müller (2015) proposed another approach that is slightly different from other scheduling approaches (in which the arrival routes in the terminal are considered as the optimization factor). In this paper, a mixed integer linear programming model was proposed to consider the approach routes for scheduling and sequencing aircraft’ arrival. This means that the model governs the related discrete trades or time advances to substitute arrival routes and/or holding procedures that must be appointed to each aircraft to avoid a clash. The proposed approach aims to minimize the total delay in aircraft’ arrival in the trial area presented in Beasley et al. (2000) However, the formulation is different in terms of the objective function as this work focuses on the arrival routes. To evaluate the performance of the approach, real-world data was taken from the Sao Paulo/Guarulhos International Airport in Brazil. The results demonstrated that an optimal solution can be reached in reasonable time using the CPLEX solver, if the developed dynamic approach is executed in real-time situations. Moreover, the results proved that the delay can be reduced by 35% for situations close to the actual procedures of the Sao Paulo/Guarulhos International Airport.

An exact method based on the Mixed-Integer programming formulation was proposed for aircraft scheduling by Samà, et al. (2017). In this study, the trade-off between various performance indicators was investigated considering the safety constraints. The proposed model utilized real-life data from two Italian airports: Milano Malpensa and Roma Fiumicino. The result showed that the proposed approach outperformed FCFS solutions.

b. Metaheuristic Methods

The formulation of real-world ALP problems is a challenging task for the researchers using the metaheuristic algorithms. In the next subsections, the research works on ALP using metaheuristic algorithms and the real-world problems will be reviewed.

Genetic Algorithm

In 1998, an evolutionary approach based on genetic algorithm was introduced to solve the long computational time at Sydney Airport (Ciesielski & Scerri, 1998). The data was collected from the airport on the busiest day of the year. Two sets of data were generated: the first contained 28 aircraft arriving in 37 minutes while the second had 29 aircraft arriving in 38 minutes. Standard binary GA and seeding GA versions were tested in the approach. The fitness function considered in this approach was contingent on the penalty of the solutions which occurs as result of invalidity, clashed aircraft, early aircraft, close proximity of aircraft, late aircraft, and adjacent or crossed aircraft. The ALP treats this as a rearrangement or permutation problem. According to the author, there are a number of drawbacks resulting from this approach such as invalid solutions generated in some runs, chromosome size problem, and long-time execution in the crossover and mutation operators.

Ant Colony Optimization

Jiang et al. (2014) proposed the ACO for ALP with two objective functions. The proposed approach was used to investigate the ability of the ACO algorithm to reduce the aircraft delay as well as achieving fairness among the airline companies. A real-life dataset from a large airport in China was used for the performance evaluation. The result indicated a 42.22% reduction in delay time and 38.64% fairness among the airlines, based on the data used.

Simulated Annealing

Rodríguez et al. (2017) proposed the use of Simulated Annealing (SA) algorithm (which has a low computational time) for mixed-operation runway in order to minimize the delay in aircraft' arrival and departure time. In addition to the safety constraints considered in solving the problem, Constrained Position Shifting restrictions were used to guarantee fairness between the aircraft schedules such that each aircraft can only shift to a limited position. The numerical data used in the performance evaluation of this algorithm was obtained from the London Gatwick airport. The result reflected the efficiency of the proposed approach in terms of computational time and reduction in the delays of aircraft in the holding area.

Bennell et al. (2017) proposed an optimization approach for s-ALP based on hybridization of algorithms. The approach was updated with rolling horizon approach for d-ALP. In this approach, multi-objectives take into consideration the runway throughput and fuel consumption. Dynamic programming, iterated descent and Simulated Annealing local search were utilized for the ALP. The proposed approach was evaluated using two types of data: a real-life dataset taken from Heathrow airport, and a randomly generated dataset. The result of the static problem indicates that the proposed approach could achieve a high throughput for the runway while minimizing the delay and fuel expenditure. For dynamic scheduling problem, the proposed approach performed efficiently with a lower computation cost. Overall, the computational results proved that the proposed algorithms can obtain better solutions than the FCFS result.

Others Search Heuristic Approaches

Beasley et al. (2001) investigated a population heuristic (PH) algorithm to improve the scheduling of aircraft waiting to land at London's Heathrow Airport. Similar to GA, a population of solutions is represented by real numbers. Each individual consists of early, target and latest landing time. The objective function of each individual is measured by the total deviation of aircraft in the individual from their target landing time (preferred time). The selection of parents in this algorithm employed a binary tournament selection mechanism, where two individuals are selected randomly from the population and the best fit will be chosen as the first parent. This procedure is repeated to get the second parent. The next step involves the application of crossover. In this step, a uniform crossover is implemented. The authors evaluated the PH algorithm using controller sequencing decisions with the minimum possible separation time. The result reflects the efficiency of the proposed algorithm in terms of computational speed and the effective schedule generated.

A fast dynamic local search algorithm based on the job-shop scheduling technique was proposed by Bianco et al., (2006) for aircraft landing scheduling. The procedures discussed in the paper include the multiple runway and multiple approaches and leaving procedures. The constraints of the ALP are analysed carefully. The fast heuristic algorithm introduced in this work is also capable of dealing with some operational restrictions that produces, in real time, a new schedule each time a new aircraft enters the TMA for landing/departing. The proposed approach was tested using real data sets from Milan-Malpensa and Rome-Fiumicino airports. Results of the proposed approach showed a 40% reduction in delays and a 30% increase in capacity of the terminal area.

Samà et al. (2015) proposed an optimization approach based on job-shop scheduling for landing and take-off scheduling problems. This work considered two objective functions which are to minimize the maximum delay of aircraft and the total travel time spent in the terminal control area. This paper is distinct in literature as it considers the route path of the aircraft in the terminal area as another factor for solving scheduling problems. For result evaluations, the authors obtained real world instances from the Roma Fiumicino airport (the largest airport in Italy). For comparison, the two objective functions were considered. CPLEX solver was used to compute the result to prove the performance of the approach.

Kepir et al. (2016) studied another aspect in the ALP by considering the aircraft fleet utilization and the waiting times of transfer passengers as two objectives for the AnadoluJet Turkish Airlines. The paper focused on finding balance in the trade-off between these two objectives. The authors proposed a mathematical model which was further extended to heuristic algorithm. This was to expand fleet utilization, decrease waiting times for most transfer travellers, and produce flight schedules that are subjected to numerous constraints. Both the objective function and the constraints of the problem were represented in the mathematical model. The result showed that the proposed approach obtained an optimal solution and reduced the scheduling time as compared to manual scheduling.

Kapolke et al. (2016) proposed a pre-tactical optimization technique for runways where the arrival time of aircraft are uncertain. A unique approach which eliminates all irrelevant parameter was used in this paper. The mathematical representation of the problems was generated and the uncertainty of aircraft' arrival was incorporated into the mathematical model. An important factor in the proposed model is that several aircraft can be assigned to the same time window which minimizes the problem's complexity. The developed model was tested using real-life dataset taken from a large airport in Germany. Computational study showed that the proposed optimization model, with pre-knowledge about the uncertainty, could affect the performance of the entire model.

Soykan and Rabadi (2016) proposed a hybrid metaheuristic algorithm with a simulation-based optimization (SbO) approach for multi-objective runway scheduling problem based on the Scatter Search (SS) algorithm. In the proposed approach, a greedy algorithm was used to generate the initial population and a reference set was then selected from the initial population. The proposed algorithm was evaluated based on real data taken from the Washington Dulles International Airport (IAD). In this data, the aircraft were categorized as heavy, large and small aircraft. The results of the computational experiments demonstrate that the use of an SS-based metaheuristic algorithm within a SbO framework is encouraging.

Samà, et al. (2017) proposed the hybridization of Tabu Search Algorithm and Variable Neighborhoods search which produces Variable Neighborhood Tabu Search (VNTS) for aircraft scheduling. The VNS algorithm was used to explore different neighborhoods in the search space while the TS algorithm was used to avoid cycling and escaping from the local optimum solution. The proposed algorithms were evaluated based on real-life dataset taken from the Milan Malpensa terminal control area (MXP). The overall performance of the proposed hybrid algorithm showed a fast computational performance; an effective minimization of the time required to obtain quality solutions; a better achievement in terms of solution quality and computation time, particularly for the hardest cases including disruptions; and the calculation of new best recognized solutions for some instances where the optimum solution are yet to be obtained.

Apart from the above research works, several other studies have used different datasets. The static and dynamic cases of the problem as well as the optimization approaches were investigated. These works will be reviewed in the next subsection.

2.4.4 Research Works on ALP using Other Datasets

A number of researchers have used datasets other than those used in the previous subsections. In this subsection, a review of these papers is presented with an explanation of the data used. This is shown in Table 2.1.

Table 2.1 Others ALP Dataset

References	**Technique**	**Dataset Source**
(Abela et al., 1995)	GA	Randomly generated
(Cheng et al., 1999)	GA	Randomly generated
(Caprì & Ignaccolo, 2004)	GA	System aspects and optimization models in air traffic controller planning.
(Bäuerle et al., 2006)	Mathematical model proposed by author	Randomly generated
(Artiouchine et al., 2008)	Mixed-Integer Programming (MIP)	http://www.lix.polytechnique.fr/~baptiste/flight_scheduling_data.zi

to be continued …

… continuation

(Tang et al., 2008)	Differential Evolution (DE)	Randomly generated
(Soomer & Franx, 2008)	Local Search proposed by author + MIP	Randomly generated
(Hu & Di Paolo, 2008)	GA	Randomly generated
(Zhan et al., 2010)	Ant Colony Optimization (ACO)	(Hu & Chen 2005) and (Hu & Di Paolo 2008)
(Tavakkoli-Moghaddam et al., 2012)	fuzzy programming approach	Randomly generated
(Shidong et al., 2012)	ACO	Randomly generated
(H. Zheng et al., 2013)	GA	(Hansen 2004) and (Pinol & Beasley 1999)
(Briskorn & Stolletz, 2014)	Exact method developed by the author	(Bianco et al. 1999) and (Beasley et al. 2000)
(Ghoniem et al., 2015)	Branch & bound Algorithm	(Farhadi et al. 2014) and randomly generated data
(Ghaith Rabadi, 2016)	Tabu Search TS	(Ghoniem et al. 2015)

2.5 FINDINGS FROM LITERATURE REVIEW ON ALP RESEARCH WORKS

In this subsection, the main findings from the literature review are discussed to highlight the challenges and weaknesses in the optimization algorithms used to solve the ALP. These challenges can be summarized in three main categories as follows: the convergence speed of the algorithm, the neighborhood structures followed to improve the current solution, and the adaption of algorithms to solve the d-ALP. Each challenge will be discussed in detail in the following subsubsections.

2.5.1 Convergence Speed

The convergence speed indicates the efficiency of an algorithm when used to solve an optimization problem. Solution construction forms an important step in solving a combinatorial optimization problem using metaheuristic algorithms. The convergence speed is affected by the solution construction procedure during the generation of the new solution (Rodríguez-Díaz et al., 2017). In ALP, the solution construction is considered a critical task due to the rigid nature of the problem. This is because small changes in the solution component may lead to inefficient solutions when the construction is fully random (Bencheikh et al., 2009). This is as a result of limited resources such as limited number of available runways and the time window for landing. In solving an optimization problem using a metaheuristic algorithm, the algorithm generates new solution based on its nature. For example, in GA, the solution generation at each time depends on the crossover and mutation. The selection of the decision variable for these procedures (i.e., crossover and mutation) is accomplished randomly or using a heuristic procedure. In other metaheuristic algorithm such as HSA, the procedure of generating a new solution (i.e., improvisation of a new solution) depends on HMCR and PAR procedures. These procedures are similar to crossover and mutation in GA. Therefore, the selection of the decision variables in HMCR is a critical step when dealing with a problem like ALP.

The limitation in the standard HSA is that the pitch adjustment operator is mainly designed for mathematical and engineering optimization problems. In these problems, the examined decision variables' values, that meet the PAR probability. are replaced by the neighbouring values by modifying the decision variable (Al-betar 2010). Therefore, in ALP, a fully random selection of the decision variables may lead to slow convergence and a low-quality solution. Thus, an aircraft can be in conflict with another one during the course of generating the landing sequence or the landing time (Girish, 2016). The problem with solution construction in ALP is the selection of unfeasible decision variables during the construction of new solutions. Therefore, a guided procedure must be considered when generating a new solution to improve the convergence speed, avoid premature algorithmic convergence, and increase the solution quality. Guided procedure can improve the performance, and reduce the effort and time needed to obtain high-quality solutions and proper convergence speed.

2.5.2 Neighborhood Structures

One of the crucial factors in determining the efficiency of a metaheuristic is the neighborhood operators provided by the user. Note that the best alternative for a problem domain can only be formalized by an expert (Boussaïd et al., 2013). Also, the