COM6910-001
150113245

Individual Assessed Work Coversheet

Assessment Code: COM6910

Description: Final Dissertation

Staff Member Responsible:

Due Date: 07-09-2016 15:00:00

☑ **I certify that the attached is all my own work, except where specifically stated and confirm that I have read and understood the University's rules relating to plagiarism.**

I understand that the Department reserves the right to run spot checks on all coursework using plagiarism software.

Student Registration Number:

150113245

Assessment Code:

COM6910-001

Optimal scheduling of planes at an airport

Prepared by: Yue Xiao (150112345)

Supervised by: Dr. Joab Winkler

COM6910 Dissertation Project

03 September 2016

This report is submitted in partial fulfillment of the requirement for the degree of Msc Information Systems

ACKNOWLEDGMENT

First I would like to express my sincere gratitude to my supervisor Dr. Joab Winkler for his advice and inspiration to my dissertation project. Then many thanks to all people who ever help me and teach me during my master study. Last but not the least, to my families and friends for their consistent help, support and motivate.

DECLARATION

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Name: Yue Xiao

Signature:

Date: 5th September 2016

ABSTRACT

Aircraft scheduling in an airport is always a crucial issue for air traffic and a focus of operational research. The aircraft landing problem is now a bottleneck which causes delay and limits throughput of an airport and efficient landing sequences must be acquired to improve the airport performance. Therefore, to develop approaches to find the optimal sequence of aircraft landing is a meaningful research.

The aim of this research project is to develop a new algorithm for optimal scheduling aircraft landing in airports. This research develops an adaptation of the Ant-Q algorithm to solve aircraft landing problem and makes several improvements to the generic Ant-Q algorithm. Computational results show that the Ant-Q algorithm can solve the aircraft landing problem approximately well and improvements made significantly increase the performance of the generic Ant-Q. This research successfully introduces a new direction of research to solve the aircraft landing problem.

Key words: **Optimization Aircraft Landing Problem Ant-Q algorithm**

Contents

Chapter 1 Introduction

1.1 General introduction

Air traffic grows rapidly in recent decades and it plays a critical role in the global passenger transport. As a example, London Heathrow International Airport is one of the best connected hub airports in the world, with over 80 global airlines operating regular scheduled flights to almost 250 destinations. With 75.0 million passengers in 2015, Heathrow is the fifth busiest airport around the world and it consequently suffers from congestion at specific times of the day (Heathrow 2015) . As time goes by, capacity of airports may become a significant bottleneck of transport capacity of air traffic, in which the limitation of runway is key during the airport operation.

The capacity of of an airport is often measure as the total throughput which is the number of aircraft land and take-off per hour. Due to the limitation of runways and constraints of landing and take-off(for instance, Heathrow only has two runways), and building more infrastructure like runways seems not an option realistic because of investment costs and time, how to sequence and schedule the allocation of runways is the most important task for Traffic Control(ATC). When an aircraft is approach and waiting to land or take off, the aircraft traffic controllers need to get information of the aircraft and assign a proper runway and a time lot to it. For convenience and simplicity, the controllers used to schedule the runway using a way called "First Come First Serve"(FCFS), which mean the aircraft comes easier has the priority to land and take off. However this method can not fit the demand of throughput. Meanwhile, the comfort of customers,the safety, the economical efficiency of aircraft scheduling should be also considered. As a result, an efficient Aircraft Traffic Management(ATM) system is needed to advise controllers for better sequencing and scheduling.This problem shall be summarized as Aircraft Scheduling Problem(ASP).

The ASP can be divided into Aircraft Landing Problem(ALP) and Aircraft Take-off Problem(ATP) which try to find a optimal landing or take-off sequence for aircraft in single or multiple runway airports for different objectives(throughput,delay,cost,etc.), considering operational constraints(safety,fairness,etc.).There are two main challenges for the ASP. Both ALP and ATP are NP-hard which means that computational time to find an optimal solution will be enormously great for large problems and the ATM system is a real-time system.The first challenge is to develop a algorithm that can find a optimal solution in polynomial time so that it can be implemented in the real ATM system. Second, due to the constraints of real world, a good algorithm may satisfy the objectives but not always be possible to be implemented in practice, which must consider safety, efficiency, robustness, fairness, pollutions and other issues (Bennell et al. 2011). To conclude, the two basic nodus are efficiency and practicability.

A huge amount of research have been presented to solve the ALP and ATP. This thesis will focus on investigate and solving the ALP,Aircraft Landing Problem. Many approaches were proposed such as Dynamic Programming, Branch and Bound, Genetic Algorithm and achieved remarkable results. In recent twenty years, Ant Colony Optimization(ACO) was adopted in ALP and the greatest strengths of ACO for ALP are its robustness and simplicity for expressing and modifying. Meanwhile, machine learning especially Reinforcement learning is one of the hottest areas in computer science research however machine learning is hardly used in operational research, the decision model of aircraft scheduling could be can be viewed as a Markov decision process, which exactly is the basic concept of Q-learning. So attempts to apply some learning perspectives in ALP should be interesting, meaningful and may be able to find a new idea to solve ALP.

1.2 Aims and objectives

The aim of this thesis is to develop a more comprehensive formulation of ALP and find a new algorithm based on Ant Colony Optimization with a machine learning perspective. The aim includes several objectives as follow：

1. To define a appropriate mathematical formulation for Aircraft Landing Problem with more comprehensive set of considerations of constraints in practice.
2. To develop a new algorithm based Ant Colony Optimization with learning perspectives to solve
3. To improve the algorithm to achieve higher computational effort and better solution
4. To implement the algorithms in experimental environment
5. To evaluate the computational results and compare the performance of the proposed algorithms.

1.3 Structure of the report

The rest part of this thesis is organized as follows. Section 2 describes the the Aircraft Landing Problem in detail including some basic concepts, objectives and constrains.The literatures of previous works focus on ALP will be reviewed in Section 3 and the review is presented approach by approach.Then Section 4 demonstrates the mathematic formulation for the ALP and a new algorithm is provided in Section 5. Section 6 will discuss further improvements and extensions of the new algorithm and Section 7 contains implementation and experiments on test case. Detailed computational results and evaluations will be demonstrated in Section 8. Finally, Section 9 will present conclusions and suggestion for further research on the ALP.

Chapter 2 Problem description

In this section, basic concepts of Aircraft Landing Problem as well as its objectives and constraints will be discussed.

2.1 Basic Concepts

Aircraft Landing Problem could be described as a decision problem to sequence, schedule aircrafts landing and assign runways. This first one is to give queue the aircraft to land and the second is to assign a proper specific time for each aircraft

Scheduling of the process of aircraft landing can be divided into three stages: creating an initial landing sequence,adjustment the sequence, and freezing the sequence(Mesgarpour 2012)

The decision processes of the ALP can be divided in three stages, and the model is shown in the following figure:

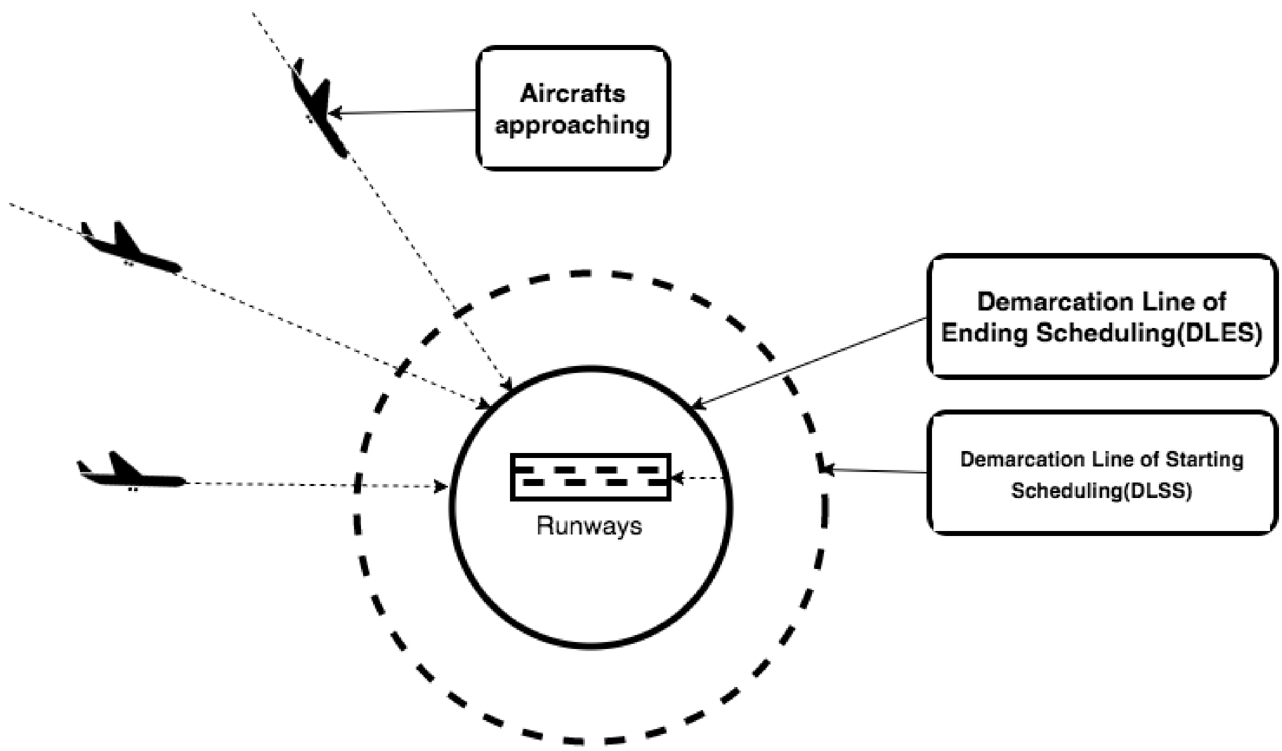

Figure 1 The spatiotemporal model of airport terminal areas for the ALP

First, When a set aircrafts arrive in the radar areas of ATC tower for starting scheduling in a specific time, a initial sequence for scheduling will be generated, this time for sequencing is called **Demarcation Line of Starting Scheduling(DLSS).** Basically the initial sequence is mostly built following a first-come-first-served(FCFS) rule. This stage is the sequencing stage

Then If aircrafts pass through the DLSS, the controller tower will start to adjust the landing sequence according to the **Estimated Landing Time** and other aspects. This stage is the scheduling stage.

The **Demarcation Line of Ending Scheduling(DLES)** is the time point when aircrafts in sequence are ready to land(2-3 minutes before landing). When aircraft pass through this line, the landing sequence is frozen and could not be modified any landing for the the distance from aircraft and runway is too short(Neuman and Erzberger 1991). This stage is the freezing stage.

2.2 Constraints

In the real ATM system, controller must make decisions subject to several technical and operation constraints for the reason of safety, fairness,fuel limitation and so on.

2.2.1 Time windows

The actual process of landing time allocation is that, when aircrafts enter the radar area of ATC, they may ask for a landing time from control tower. However, this is is a time window rather than a specific time point. The time window is provide via a earliest landing time (ELT)and latest (LLT)landing time.The time windows may differ from different flight and can be overlapped in practice or in theory. The earliest landing time is related to the fastest speed of each aircraft and an aircraft can land at the ELT using its fastest speed but it will cause fuel waste due to the speeding up. On the other hand, the LLT is related to the fuel load.

There is a cost-optimal speed for every aircraft, which is called cruising speed. The average fuel cost per hour is minimized when a aircraft flies in the cruising speed. The estimated landing time must fall in the interval of time window and either earlier or latter landing is uneconomic and the cost grows with speeding up or deceleration. Hence, in the time perspective, the fuel overhead increases with increased time deviation from the estimated landing time.

2.2.2 Constrained Position Shifting(CPS)

Considering fairness of every aircraft await to land, the FCFS seems the fairest algorithm for ALP and Dear (1976) also claimed that the landing sequence should not show relatively large difference from the FCFS sequence. Hence, he first introduced *constrained position shifting* (CPS) for the Aircraft to prevent excessive deviations from the optimal landing sequence.(Roger G. Dear 1976)

CPS means there is a maximum number of positions that an aircraft can move in the landing sequence from the FCFS sequence. In detail, for k-CPS, aircraft in initial landing sequence can not shift more than k units of position. For example, the first came aircraft can not shift to position 1+k when imposing the k-CPS constraint. CPS can not only maintain the fairness between aircraft but also reduce the solution space(J. A. Bennell et al. 2011). Therefore many approaches posted imposed CPS as a method to improve computational efficiency.

Thus it can be seen that CPS is not only a constraint for the ALP but also a useful idea to reduce the scale of the problem

2.2.3 Seperations(security interval)

The time difference for every two aircrafts' landing time, called minimum aircraft separation. This separation is basically considered for safety reasons which is the most crucial issue of ATM system and its main theoretical basis is aerodynamics. For instance, Beasley (2000)states that a Boeing 747 could create a strong wake vortex when it is flying and other aircrafts follow it or fly pass by may be influenced by the wake vortex, loosing their aerodynamic stability. Actually a large number of aircraft accidents were caused this phenomenon. For this safety reason, there must be a long enough time separation between two aircraft landings.Comparatively speaking, larger aircrafts may cause a stronger wale vortex. So the International Civil Aviation Organization(ICAO) sets a standard of security intervals under no wind condition between two aircrafts according to the Maximum Take-Off Weight. The ICAO generally divides the aircrafts into six classes and a simple version of "Heavy, Medium and Light". When scheduling for maximizing the throughput, the time separation is often the most significant constraint.

2.3 Objectives

(J. A. Bennell et al. 2011) claimed that objectives of aircraft landing problem differ from a number of different stake holder and summarized the objective listed as follows:

ATC	Airline	Airport	Government
maximizing the runway throughput	minimizing operating costs	maximizing punctuality relative to the operating schedule	minimizing environmental effects
minimizing the approach	minimizing engine run	minimizing the need	

time of aircraft before landing	times before take-off	for gate changes due to delays	
minimizing air traffic controllers' workload	maximizing punctuality with respect to landing time in published timetables		
maximizing fairness among the aircraft	minimizing total passenger delays		
minimizing the aircraft taxi-in time	maximizing adherence to airline priorities within their own flights		
minimizing the arrival delay	maximizing the connectivity between incoming and outgoing flights.		
minimizing deviations from an appropriate balance between arrivals and departures.			

Table 1 Objectives of ALP from different perspectives(J. A. Bennell et al. 2011)

Chapter 2 Problem description

Most commonly used objective functions in previous literature are minimizing the arrival delaymaximizing the runway throughput and minimizing operating costs. Here in this research, we shall use a total penalty cost to minimize as our objective function which is presented in the next sections.

Chapter 3 Literature review

In this section, literature on aircraft landing problem(ALP) will be reviewed. Many different ways of solutions were established during the past 40 years. As the days progressed, several approaches were introduced or adapted for the problem. In general, this section reviews the literature in 3 main categories of algorithms:Dynamic programming, Branch-and-bound and Genetic Algorithm. In addition, it will place particular emphasis on discussing Ant Colony Optimization approach due to the objectives of this thesis.

3.1 Dynamic Programming

Dynamic programming is a common approach for combination optimization problems, it was developed from the principle of optimality introduced by R.E.Bellman(1950).The dynamic programming transforms a multistage decision process into sub decision-making processes and every subprocess has its own optimal solution.When the subproblems are recursively solved, the optimal solution of the original problem will be achieved. The ALP is at first normally modeled as Dynamic programming problem because the original set of the aircraft waiting for landing can be easily divided into partial sets with same structure.

Dear (1976) introduced one of the first dynamic programming algorithm for the single runway ALP imposing the constrained-position-shifting (CPS)

Psaraftis (1978)then adapted Dynamic programming for ALP with "the exploitation of special structure". He first reviewed the dynamic programming approach for TSP and claimed that ALP was tightly related with TSP but with different objective functions, special constraints, and some special factors. He tried the ALP by classifying the original set of planes into a small number of distinct categories, this enormously decreased the scale of the problem. At last, he developed

Dynamic Programming algorithms for three versions of static ALP with two objectives:last landing time and total passenger delay. First, for the single-runway unconstrained case, the algorithm has a time complexity of O(CnC), where C and n represent the number of aircraft classes and the number of all aircraft . Then, he considered the CPS(Constrained Position Shifting) in a single-runway case with same computational effort. Finally, he discussed the algorithm for a double-runway case with two parallel runways.

Their research proves that dynamic programming can be used in ALP but due to the limited computational effort, those approaches are "inefficient" for large size problem. Approaches using DP have been being continually developed improved and many markable solutions were present.

Bayen et. al. formulated the ALP as a single machine scheduling problem with two objectives to minimize:(1)Sum of landing times(2)The last landing time, considering all aircraft are in a same "holding loops". They presented an approximation algorithm and the ALP was divided into two subproblems, one of them is to solve 5 and 3-factor case using DP. The other part of the algorithm used linear programming relaxation and rounding.

Balarishana and Chanran (2006) presented a dynamic programming-based approach to maximize the total runway throughput. They imposed several constraints including CPS, FAA-specified minimum inter-arrival spacing restriction, presence relations as well as time window available for each aircraft. With the computational experience based on the implementation on real test cases from Denver International Airport(single runway), it is known that their algorithm is fast enough for both static case and a real-time implication in practice. They also extended the approach to other cases like multiple runway problems and take-off scheduling. Balarishana and Chanran (2006)that the significant contribution is that the algorithm can the APL with precedence constraints. Based on the previous work, Balarishana and Chanran (2007) latter present a DP approach considering the same set of constraints which could compute the tradeoff curve between runway throughput and probability of the deviations of aircraft from the all constraints. The computational complexity of their algorithm is O(n(L/ε)3), where n is the number of all aircraft, L is the largest time window for landing of all aircraft, and ε is the probability of accuracy with constraints. Their work is innovative because it provides an approach to consider the tradeoff between reliability and throughput and reliability is vital in the real aircraft scheduling system.

Lee and Balakrishan (2008) proposed a dynamic programming algorithm to compute various different tradeoffs as an extension of previous work posted by Balarishana and Chanran (2007). It was argued that minimizing total average delay was more beneficial for a real-world problem. The paper demonstrated tradeoffs between minimizing total delay, minimizing fuel cost, and maximizing throughput. By analyzing the experimental result on real-world schedules based Dallas-Fort Worth Airport, it is argued that minimizing fuel costs or operating costs generally did not result in significant decreases in the throughput of the schedule and the average delay can be significantly improved through decreasing the throughput (Lee & Balakrishnan 2008). Lee and Balakrishan(2008) also suggested that a time advance up to 3 minutes is optimal in a realistic case. The results presented by Lee and Balakrishan(2008) deepened the cognizance of the tradeoffs between different objectives in ALP. Balarishana and Chanran (2010) finally summarized previous posts (Balakrishnan & Chandran 2006; Chandran & Balakrishnan 2007; Lee & Balakrishnan 2008)and proposed a full framework for ALP and algorithms under Constrained Position Shifting. They extended the framework by including a more general cost functions and mixing the take-off into landing operations with discrete-time models. As a result, it is proved that the algorithms can solve ASP in polynomial time and is efficiency enough to be used in the real-time implementation.

Briskorn and Stolletz (2014) developed an improved version of the MIP based on the previous work by Beasley et al. (2000) and proposed a Dynamic programming approach for the static multiple-runway ALP. However, computational results for their approach were not provided.

Lieder (2015) then present an improved dynamic programming algorithm for multiple runway airports based on the DP approach posted by Briskorn and Stolletz (2014). Using a state-space reduction method, this algorithm is able to solve a case of 100 aircraft optimally within seconds. The "domain criterion" Lieder(2015) significant improved the performance of DP approach applied in ALP while maintaining optimality, which was backed by the computational results.

Most recently, Bennell et.al. (2016) developed a multi-object formulation for single-runway ALP, which considered constraint of fairness(CPS), minimum separation. The approach used the similar technique to that proposed by Lieder (2014)to improve computational effort of DP, it eliminated the state of dominated partial landing schedules in initialization and next iteration stage. According to the experimental results based on both random test data and data from Heathrow International Airport, the improved DP is capable of finding solutions for static problems as well as a dynamic one.

3.2 Branch-and-bound

Branch and bound was first invented by Land and Doig (1960) . This algorithm develops a search tree according to available solutions and finds searches each branch to find the best solution. Branch and bound are also able to solve mixed integer programming. For linear programming, it can divide the non-integer decision variable into two approximate integers to from branches and calculate upper and lower bound of objective function then the best solution can be found. Branch and bound are an essentially kind enumeration with special technique exploiting the special structure of different problems. So it can be widely used but requires strong skills. For different problems, Branch and bound algorithms are applied in highly distinct forms.

The main fundamental idea of Branch and bound is to search the best solution from the tree of feasible solutions with specific restrictions. When branch and bound are solving problems, it continually separates space of feasible solutions into branches and calculates upper and lower bound for each subset of solutions. Those branches exceed the bounds then do no need to be considered in search and to be divided next and it achieves to prune the solution tree. As we can see, Branch and bound can effectively reduce the solution space so it mostly can find the optimal solution within reasonable computational time.

It is Brinton (Brinton 1992)who first developed Brach and bound adaptation for the aircraft landing problem. He considered the ALP as a dynamic problem at the first place and the objective function was a total cost calculated from estimated landing time and scheduled landing time subject to the security interval. His research introduced three different kinds of limitation:Static limiting, Dynamic Limiting, and Depth Limiting to apply Branch and bound. Although no detailed computational results were presented, the approach was successfully used in practice at Denver Air Route Traffic Control Center.

Another approach of Branch and bound was introduced latter and the ALP was formulated as a mixed integer programming problem (Abela et al. 1993). As it mentioned above, Branch and bound is a suitable algorithm for the MIP. The approach also used minimizing costs from the deviation between scheduled landing time and estimated landing time as the objective function. Compared with the GA approach proposed in the same paper, the branch and bound approach showed a better ability to solve large problems , which could provide a good feasible solution in a reasonable computing time. Beasley(2000) summarize the mathematical formulations of MIP model for the ALP and presented a strong formulation based on the one posted by Abela (1993).

Ernst et. al. (1999)then combined simplex algorithm with Branch and bound to calculate lower bounds. And a new problem space heuristic was also developed to provide good upper bounds in Branch and bound. minimizing penalty cost was used as the objective function in their research. They also developed an extension version of the algorithm for the multiple runway problems. The computational results proved that the branch and bound approach was effective as well as a heuristic, however, the branch and bound algorithm depended heavily on the preprocessing methods.

The literature above mostly used minimizing the penalty cost as the problems' objectives. which is a cost linearly related to a deviation between scheduled landing time and estimated landing time. Cause all aircraft have a most efficient speed, either easier landing or later landing would produce penalty cost since speeding up will cost more fuel and longer holding time obviously the same. The speeding-up and holding penalty cost in the time window is shown in the following figure：

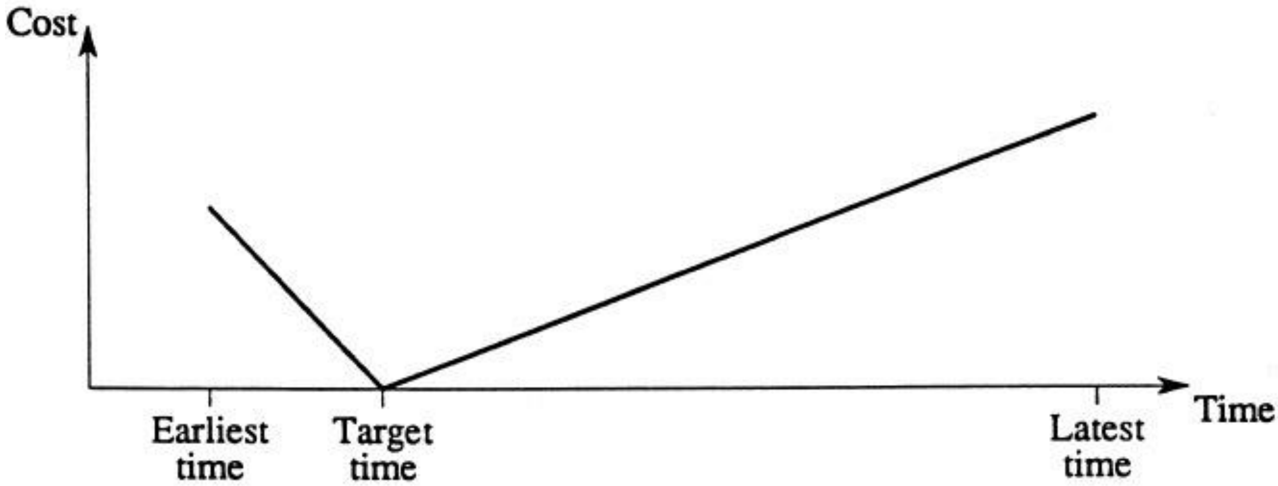

Figure 2 Variation in cost for a plane within its time window.(Beasley 2000)

The linear relation before the estimated landing time (Beasley 2000)shows the penalty cost per time unit when aircraft landing earliest and the right one is for the later landing. This thesis will also set minimizing total penalty cost as the objective function.

3.3 Genetic Algorithm

The Genetic Algorithm is a class of bionic algorithms inspired by the evolutionary laws(natural selection) in biology. Genetic algorithms are commonly used to solve optimization problems. As a meta-heuristic algorithm, it was first introduced by J.Holland (1992). The GA initially constructs a population of all feasible solution of the problem. A population contains a number of individuals with different chromosomes. There are various genes on chromosomes and chromosomes reflects the behavior or performance of individuals which can be calculated to fitness. To apply the GA, the first important issue is to code gene to map solutions to chromosomes. Then the algorithm will select better individuals from the population by evaluating their fitness and the genetic operators make such as "crossover" and "mutation" may produce new populations. The total fitness of all population will get better generation by generation. Finally, the best approximate will be found by decoding. Most of the meta-heuristic algorithms used in the ALP are GA.

Abela et.al. (1993)developed a heuristic for the ALP firstly using a genetic algorithm. They compared the solutions obtained by GA heuristic and Branch and bound and found that the first simplest GA performed well for the small problem and exact algorithm achieved better solutions in larger problem although took more time.

Then Stevens introduced another approach based on GA to solve static aircraft landing problem in his thesis. The objective of his research is to minimize the total penalty cost caused by landing advance and delay, which is subject to security intervals and time windows. Computational results showed that the GA application was able to deal with a set of over 40 aircraft. But for the big problem, his static model showed an apparent defect that planes always landed too early.

Ciesielski and Scerri (1998) then proposed an improved algorithm from GA. It is proved in their research that a GA based scheduling system could be used in real-time aircraft scheduling with high-quality solutions. Two implements including Seeding and specialized crossover and mutation operators were made and former one led to improving the fitness and a bigger number of feasible solutions however the latter one did not achieve significant improvement over the standard operators.

Cheng et al. (1999)developed four different genetic algorithms formulations applied to the multiple runway aircraft landing problems. The first one designed two separate chromosomal for sequence scheduling and runway assignment to minimize the total delay of aircraft. The second algorithm used only on chromosomal to map both sequencing and assigning runways with the same objective as the former one. The second one performed better the first in a number of generations. Next, the third formulation used same chromosomal as the second one. The difference was the third one used a "fitness-based probabilistic selection process" for evaluation which made it even faster than the second formulation. The fourth formulation differed from the all three above, using a different principle of chromosomal representation and the results were not good as the first three.

Beasley introduced a Population Heuristic approach to solving aircraft landing problem. The algorithm was tested based the real data from London Heathrow International airport. He compared the computational results with the real operations in Heathrow to find out how much cost was saved. Beasley (2001)proposed “Unfitness” in his paper as another norm to evaluate selections. Later Pinol and Beasley developed Scatter Search and Bionomic Algorithm with Unfitness based on the previous post(Beasley et al. 2001). Computational results showed that both two algorithms could achieve optimal solutions within a reasonable amount of computing time.

Hu and Chen (2005) proposed a new improvement:Receding Horizon Control, into GA for the dynamic version of the ALP. The objective of their formulation is minimizing the airborne delay time rather than total penalty cost. Their computational results showed that the RHC based GA not only performed better than a genetic GA but also had higher computational efficiency.

3.4 Ant colony optimization

The Ant Colony Optimization is a family of meta-heuristic algorithms to solve optimization problem inspired by the behavior of ants. Detailed introduction of the ACO will be in Section 4.

Randall (2002) developed one of the first adaptation of Ant Colony System to solve a static ALP with an objective related to the deviations from estimated landing time. The adaptation was able to solve a problem with up to 50 aircraft.

Benchieikn et al. (2009)then modeled the ALP as a job shop scheduling problem and propose a hybrid method combing GA and ACO. The computational results show that the hybrid approach performs better in both objective values and computing time than a generic GA.

Wang et al. (2010) defined a new transition probability rule and 2-opt local search strategy for ACO but the conclusion is just that the improved ACO can overcome the FCFS's shortcoming.

In the same time, Zhan et.al. (2010) developed an ACO approach combined with the RHC (RHC-ACS) to solve single-runway static APL and the computational results showed that the RHC-ACS over-performed the ACS and GAs. Meanwhile, it is claimed that algorithm performance could be further enhanced by using 2-opt local search(Wang et al. 2010)

Then Bencheikh et al. (2011)proposed an adjustment method to improve ACS but without detailed mathematical formulation. The computational results showed that the improved ACS could provide optimal solutions in 80% of the test case. And the remain 20% were with an average 5% deviation from optimal solutions. It was proved that the adjustment method could significantly improve the generic ACS to solve the static ALP.

In addition, Farah et al. (2016) presented a new formulation for static single-runway ALP based on TSP formulation and solved it using ACS. But just a few computational were presented.

Most recently, Bencheikh el al. (2016) developed an extension of the previous formulation for static case (Bencheikh et.al. 2011) to solve the dynamic ALP. Comparisons have been made between the improved CO and three approached in the previous post by Beasley et .al(2006)

3.5 Other approaches

Cellular automation (Yu et al. 2011), Queueing theory (B a uerle et al. 2006) and branch-and-price algorithm (Wen 2005) are also used to solve different versions of ALPs.

3.6 OR-Library

Beasley (2000) presented the test cases for the ALP in OR-Library, which has thirteen problems, eight of them are small problems with optimal solutions (Pinol & Beasley 2006). The data set of ALP in OR-Library for each aircraft contains the approaching time, earliest landing time, estimated landing time, last landing time ,penalty cost per time unit for deviation and security intervals. Therefore, using OR-Library as test cases in this thesis seems an appropriate choice.

Chapter 4 Mathematical formulation

In this section, a new mathematical formulation of the ALP based on the classical formulation proposed previously Beasley(2000). The formulation is for static case of the single runway ALP.

4.1 Notations

N: The number of aircrafts waiting to land,

E_i: The earliest landing time for aircraft i,

L_i: The latest landing time for aircraft i,

T_i: The estimate landing time(target) for aircraft i,

S_{ij}: The minimum time separation between aircraft i and j if i lands before j,

Pg_i: Penalty cost per time unit for aircraft i if it lands before estimate landing time

Ph_i: Penalty cost per time unit for aircraft i if it lands before estimate landing time,

t_i: The actual scheduled landing time for aircraft i,

ta_i: The advance made by aircraft i,

tt_i: The tardiness made by aircraft i,

$$\lambda_{ij} = \begin{cases} 1 & if\ aircraft\ i\ lands\ before\ j \\ 0 & otherwise \end{cases}$$

4.2 Objective function

The objective of this formulation is to minimize the overall penalty cost of deviation between scheduled landing times of all aircrafts and their estimated landing times.

$$min \sum_{i=1} (Pg_i ta_i + Ph_i tt_i) \qquad (1)$$

4.3 Constraints

There are several constraints according to the Section 2.

4.3.1 Landing time window

The scheduled landing time must be in the time window, interval $[E_i, L_i]$ which is described as:

$$E_i \leq t_i \leq Li \forall i = 1, \dots, N \qquad (2)$$

And for aircraft i landing before aircraft j or j before i, there are:

$$\lambda_{ij} + \lambda_{ji} = 1 \forall i, j = 1, \dots, N; i \neq j \qquad (3)$$

$$\lambda_{ij} \in -1,0 \forall i, j = 1, \dots, N \qquad (4)$$

4.3.2 Seperation

The separation constraint must be subjected to, and for aircraft i landing before aircraft j there is:

$$t_j \geq \lambda_{ij} S_{ij} + t_i \forall i, j = 1, \dots, N; i \neq j \qquad (5)$$

And for aircraft i landing after aircraft j, the equation become:

$$t_i \geq \lambda_{ji} S_{ji} + t_j t_j \forall i, j = 1, \dots, n; i \neq j \qquad (6)$$

So we can conclude from (5) and (6) that:

$$\lambda_{ij} t_j + \lambda_{ji} t_i \geq \lambda_{ij} S_{ij} + \lambda_{ij} t_i + \lambda_{ji} S_{ij} + \lambda_{ji} t_i t_j \forall i, j = 1, \dots, N; i \neq j \qquad (7)$$

As $t_i < L_i$ and $t_j > E_j$ when aircraft i landing after aircraft j, (5) can be extended to:

$$t_j \geq \lambda_{ij} S_{ij} + t_i + (E_j - Li) \lambda_{ij} \forall i, j = 1, \dots, N; i \neq j \qquad (8)$$

It can be proved that the equation (8) always holds when aircraft i lands before j or after. Then, the deviation before and after the estimate landing time for aircraft i is expressed as $t_i = T_i - ta_i + tt_i$ and the constraints are showed by (9), (10):

$$ta_i \geq T_i - t_i \forall i = 1, \dots, N \tag{9}$$

$$tt_i \geq ti - Ti \forall i = 1, \dots, N \tag{10}$$

4.3.3 CPS(additional)

For k-CPS, let i, j be the order of two aircrafts(i, j) enter the scheduling area, there are:

$$x_{ij} = 0 \quad if i - j - k > 0; \forall i, j = 1, \dots, N; i \neq j \tag{11}$$

To conclude, the algorithm for solving this formulation is to minimize (1) under constrains (2)-(4),(7-10). If CPS is imposed, the **Equation 11** should also be subjected to.

Chapter 5 Ant-Q algorithm

This Section will review Ant Colony Optimization and introduce Ant-Q algorithm then apply Ant-Q algorithm on the aircraft landing problem.

5.1 Review of Ant Colony Optimization(ACO)

The Ant Colony Optimization is a family of meta-heuristic algorithms to solve optimization problem. The first framework of ACO was proposed by Dorigo et al. (1996)as Ant System and many work has been done on it during the years. The ACO is inspired by the behavior of ant. When ants forage from nest, they will leave a kind material called Pheromone on their routes to exchange information and share experience.Commonly ants will search randomly and then follow the pheromone on the track. The knowledge of food source will spread through the pheromone. With the time, the pheromone on the shortest route to food will become strongest eventually and most of ants shall follow this route.

The most significant feature of ACO is to keep experience from the previous solutions when search heuristically. Moreover, the ACO keeps search the routes of bad solution perviously with a certain probability, which can avoid premature convergence locally. Ants make choicely using two mechanisms:Attractiveness and Trail Level.Attractiveness represents influence from current environment(route) itself, which is sometimes called “Sight” and represented by Heuristic Value. On the other hand, Trail Level namely is pheromone level. The weights of Attractiveness and Trail Level are different while solving different problems.

weights of Attractiveness and Trail Level are different while solving different problems.

The general concepts of ACO are showed as follows:

6. Solution Construction: Let ants make their choices and construct solutions of routes according to the pheromone and information of routes. The solutions will be constructed iteration by iteration to the optimal approximation(complete solution).

7. Heuristic Value: This value represents the attractiveness of route itself. Greedy thoughts is used in attractiveness which means an ant may choose a currently better action without a long-term thinking.

8. Pheromone Update: Adjusting probability of choices by laying and evaporation of pheromone. The route of a better solution usually has more pheromone. As time goes on, pheromone will evaporate to prevent excess of pheromone which can lead to premature convergence.The strategies for pheromone updating may vary from case to case.

Since Ant Colony Optimization was first introduced, it has been being developing for decades. The main concepts of ACO remain the same however calculation of heuristic value and probability or strategies for pheromone updating have a lots changes and improvements. The Ant Colony System was proposed by Dorigo et. al(1997). Compared with Ant System, it has a pseudo-random-proportional rule when making choices which can strike a balance between the probability of choices with Attractiveness and Trail Level. Meanwhile, Dorigo et. al(1997)divided the Pheromone Update into two mechanisms:Global update and Local update. Global update accelerated convergence and Local update prevents local premature convergence. Stutzle and Hoos (2000) improved AS and developed Max-Min Ant System(MMAS) by setting upper and lower limits of *pheromone* concentration and the MMAS only allows ants who achieved currently best solution to lay *pheromone.The ACO is normally used in solving TSP/ALP and this research will adapt a variant approach of AS:* **Ant-Q algorithm** *to solve ALP.*

5.2 Ant-Q algorithm

The Ant-Q algorithm was first introduced as a reinforcement learning approach to solve the traveling sales man problem by Gambardella and Dorigo(1995). Ant-Q was found able to solve TSP especially the difficult asymmetric TSP. In this sub section, the Ant-Q will be adapted to solve the ALP formulation presented in Section 3 using similar idea as solving TSP.

Like in TSP, the sales man choose a node to make a visiting sequence. The aircrafts waiting to land can be regard as nodes in a Graph. Given a set of aircrafts N=n, the ALP is given by a fully connected graph(N,E) and E(i,j) presents the action a aircraft i lands before aircraft j. However, the ALP graph does not have a value like distance between cities in TSP so that we should find a new heuristic value for the adaption of Ant-Q.

5.2.1 Notations

Here are basic notations of Ant-Q:

m: Number of ant agents,

k: Serial number of ant agents,

$AQ(r,s)$: Ant-Q-value

$\Delta AQ(r,s)$: Reward for choosing aircraft j landing after aircraft r,

$HE(r,s)$ ：Heuristic value for choosing aircraft j landing after aircraft r,

$P_k(r,s)$: Transition probability, probability of hoosing aircraft j landing after aircraft r,

$J_k(r)$ ：For agent k, allowed next choice after aircrafts in one iteration

α: Learning rate,

β: Weight of heuristic value,

θ: Weight of, Ant-Q-value,

γ: Discount factor,

TPC_k: The total penalty cost of a land sequence from agent k.

Detailed explanations are distributed in following sections.

5.2.2 Solution construction

First, Ant-Q needs a value to store and share knowledge as a replacement of pheromone and counterpart of Q-learning Q-values. Let $AQ(r,s)$ be a positive real value associated to the action choosing aircraft s to land after aircraft r, which indicates the greedy value of this choice. Let $HE(r,s)$ be a heuristic value associated to the action choosing aircraft s to land after r which allows an heuristic evaluation of which choice costs less. An agent k scheduled aircraft s landing s using the a pseudo-random-proportional rule(Gambardella & Dorigo 1995) :

$$s = \begin{cases} \underset{u \in J_k(r)}{arg\,max}\{[AQ(r,s)^{\theta}] \cdot [HE(r,s)^{\beta}]\} & if q \leq q_0 \\ S & otherwise \end{cases} \tag{12}$$

where and θ and β are parameters weight the importance between the Ant-Q-values and the heuristic values.

And q is a value chosen randomly with uniform probability in [0,1], $q_0 (0 \leq q_0 \leq 1)$ is a parameter for possibly to make random choice, for instance a higher q_0 could result more random choices.

$$q \sim U[0,1] \tag{13}$$

S is random and selected following a rule of probability distribution given by a function of the $AQ(r,s)$ and $HE(r,s)$, with $u \in J_k(r)$:

$$P_k(r,s) = \begin{cases} \frac{[AQ(r,s)^{\theta}]\cdot[HE(r,s)^{\beta}]}{\sum_{u\in J_k(r)}[AQ(r,s)^{\theta}]\cdot[HE(r,s)^{\beta}]} & ifs \in J_k(r) \\ 0 & otherwise \end{cases} \qquad (14)$$

Equation 14 shows the probability that ant agent k transit from state r to state r(choose aircraft r to land after aircraft r) where s must not be in the $J_k(r)$, a set of aircrafts already scheduled to land before.The probability to choose s is determined by both Ant-Q-value and Heuristic value on edge(r,s).The higher $[AQ(r,s)^{\theta}] \cdot [HE(r,s)^{\beta}]$ the higher the probability to choose aircraft s landing after aircraft r. θ and β represent the weights of Ant-Q-value and Heuristic value which control the proportion of the exploration and exploitation of agents.

5.2.3 Reinforcement of Ant-Q-Value

Two kinds of reinforcement learning aspect used in Ant-Q.First, in one iteration of ant agent k, it can learn from other agents and the knowledge from later state can be passed to the earlier decision-making process.Ant-Q-value are updated by the following equation :

$$AQ(r,s) \leftarrow (1-\alpha)\cdot AQ(r,s) + \alpha \cdot [(\Delta AQ(r,s) + \gamma \cdot \underset{z\in J_k(r)}{Max} AQ(s,z)] \qquad (15)$$

The update rule is composed of both the reinforcement term and the discounted evaluation of the next state. Parameters α and γ are the learning rate and the discount factor which can prevent premature convergence. The reinforcement $\Delta AQ(r,s)$ is always zero except after every agent has finished scheduling all aircrafts so called “delayed reinforcement”.

The delayed reinforcement shows the ability that agents can learn from the last best solution and there two kinds of way to reward. First, agents learn from the current best solution from all agents of all iterations which called

$Global - best - reinforcement$:

$$\Delta AQ(r,s) = \begin{cases} \frac{W}{TPC_{k_{gb}}} & if(r,s) \in tour\,done\,by\,agent\,k_{gb} \\ 0 & otherwise \end{cases} \qquad (16)$$

W is a parameter to determined by tests and k_{gb} is the agent who made the globally best solution with lowest total penalty cost. Actions belong to the globally best solution will receive reward.

Likewise, $iteration - best - reinforcement$ means that actions from the best solution of current single iteration will receive reinforcement which is showed as follow:

$$\Delta AQ(r,s) = \begin{cases} \frac{W}{TPC_{k_{ib}}} & if(r,s) \in tour\,done\,by\,agent\,k_{gb} \\ 0 & otherwise \end{cases} \qquad (17)$$

Where k_{ib} is the maker of best solution in one iteration.

5.2.4 **Heuristic value**

The ALP is similar to the TSP but it does not have a value like distance from city to city. This research develops a Heuristic value for Ant-Q according the feature of the ALP:

$$HE(r,s) = \frac{Pg_s}{T_s - t_r}. \qquad (18)$$

When aircraft r land at time t_r, the bigger $(T_s - t_r)$ means that the aircraft s has more chance to land at its estimate landing time, so it can be chosen later. And an aircraft with higher penalty cost per time unit may be chosen in higher probability for reducing total penalty cost. Actually it is the reciprocal of penalty cost of an action. Since $Pg_s == Ph_s$ in this research, there is no need for another formula with Ph_s . Note that this heuristic value is just proposed for the integrity. Other kinds of heuristic value shall be used and tested in the later implementation.

How to assign aircraft land times remains a problem. Here agents assign estimated landing time respecting the landing time window, for aircraft s landing after r:

$$t_s = max(Ts,\ \max_{r \in J_k(r)} (t_r + S_{rs})) \qquad (19)$$

Where $\hat{J}_k(r)$ is the set of aircrafts which have been assigned a landing time already.

5.2.5 Solution procedure

Figure 3 reports the main procedures of the Ant-Q algorithm.

There four main stages in the Ant-Q algorithm.First, it is in the Initialization stage where basic values are initialized such as AQ matrix and first aircraft to land for each agent. Then, in the Solution Construction stage, the solutions from agents are constructed and Ant-Q-Values are continually updated without reinforcement. Next, when every ant agent has finished the solution construction, it comes to the Delayed Reinforcement stage. Here, the actions on best solution route will receive reinforcement as reward and Ant-Q-Values are also updated. After these,the algorithm stops and outputs or return back to Solution Construction stage using a judgment on the termination condition.

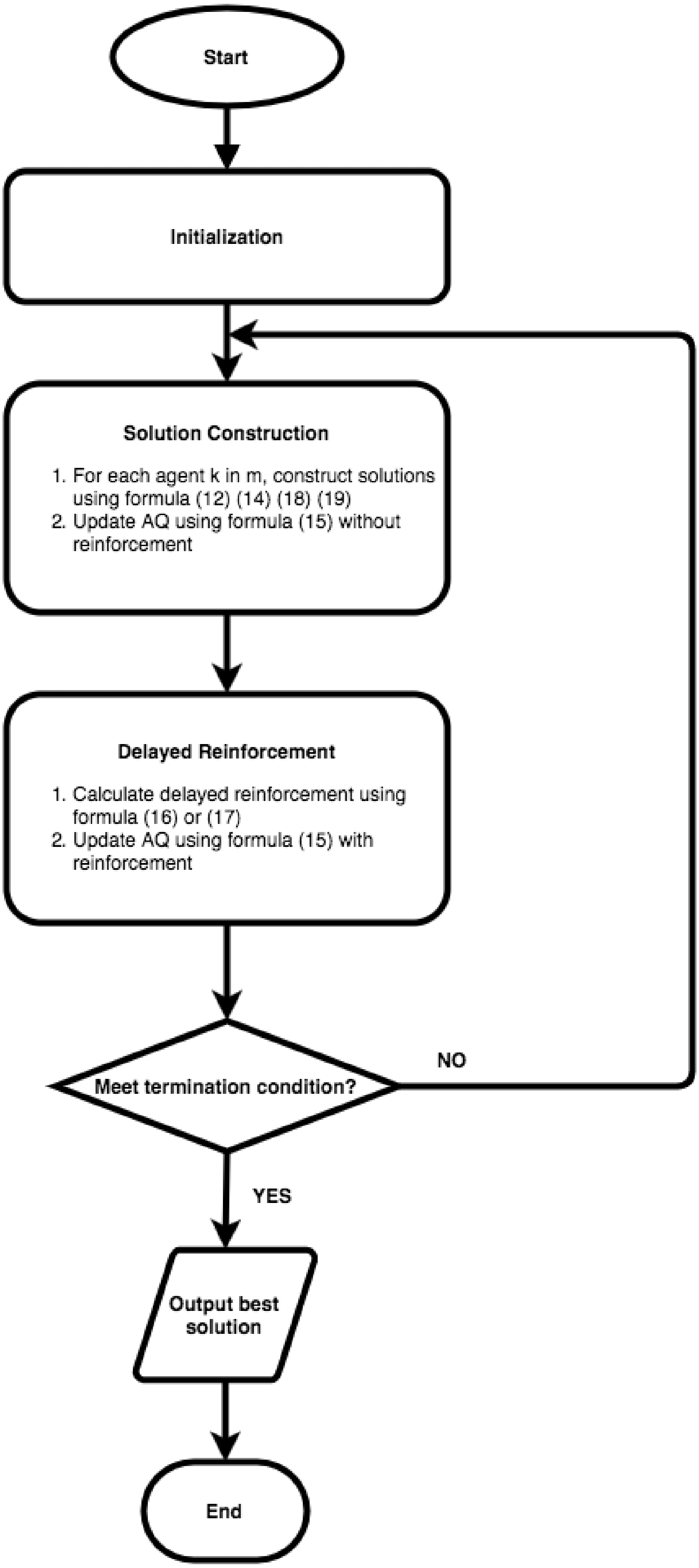

Figure 3 Flow chart of the Ant-Q algorithm

The Pseudocode showed as follow in List 1:

```
/* 1.Initialization phase */
For each edge(r,s)
        // Initialize AQ
        AQ(r,s):=AQ0
End-for

For k:=1 to m do
        Let rk1 be the first aircraft for agent k
        Jk(rk1):= {1, ..., n} - rk1 // Remove rk1 from allowed-next set
        rk:=rk1
        Assign landing time trk for aircraft rk using formula (19)
End-for

/* 2.Solution construction phase */
For i:=1 to n do
        If i≠n Then
                For k:=1 to m do
                        Caculate HE using formula (18)
                        Choose the next city sk according to formula (15)
                        Assign landing time tsk for aircraft sk using formula (19)
                        If i≠n-1 Then           // Update allowed-next set
                                Jk(sk):= Jk(rk) - sk
                        If i=n-1 Then
                                Jk(sk):= Jk(rk) - sk + rk1
```

```
                    Actionk(i):=(rk,sk)
                    // record action for agent k at step i.
            End-for
        Else
        // the last aircraft to schedule
            For k:=1 to m do
                    sk := rk1
                    Assign landing time tsk for aircraft sk using formula (19)
                    Actionk(i):=(rk,sk)
            End-for
            For k:=1 to m do
                    Update AQ using formula (15)
                    // Note here ΔAQ == 0
            rk :=sk
        End-for
End-for
3./* Delayed Reinforcement phase */
For k:=1 to m do
        Compute TPCk
End-for


Choose best solution Actionk to give delayed reinforcement


For each edge (r,s) in Actionk
        Compute the delayed reinforcement ΔAQ(r,s) using formula (16) or (17)
```

```
End-for
Update AQ-values applying a formula (15)
/* 4.Termination phase */
If (Termination_condition == True) then
        Output current globally best solution
else
        goto Step 2
```

Listing 1 Pseudocode of the Ant-Q algorithm

In detail, some calculations and update must be done in different stages like calculating HE values, updating allowed-next set and so on.

To conclude, in this section we present a adaption of the Ant-Q algorithm for the ALP. However this generic Ant-Q does not exploits the special structure of the ALP. In next section, improvements will be made on Ant-Q algorithm combining with some characteristic techniques.

Chapter 6 Improved Ant-Q algorithm

In this section, we will improve Ant-Q algorithm presented in last Section from 2 direction which exploit the special structure of the ALP.

6.1 Assigned landing time adjustment.

If we dig into formula (19)(Equation of assigning landing time), a defect can be easily found that landing at estimated landing time greedily for every aircraft may be not suitable for a optimal solution.Since the penalty cost per unit of each aircraft is not the same, an aircraft with a penalty cost per time unit, currently lands on its estimated landing time can be adjusted to land easier in order to let the landing time deviation with next aircraft with higher penalty cost per time unit down. This kind of adjustment can reduce total penalty cost and shows an idea that a local optimal solution may not be a part of the optimal result.

Next, the case 1 of 10 aircrafts from OR-Library will be discussed for exploring the adjustment method.

Aircraft Number	Approaching Time	Earliest Landing Time	Estimated Landing Time	Latest Landing Time	Penalty before target time	Penalty after target time
1	54	129	155	559	10	10
2	120	195	258	744	10	10
3	14	89	98	510	30	30

4	21	96	106	521	30	30
5	35	110	123	521	30	30
6	45	120	135	576	30	30
7	49	124	138	577	30	30
8	51	126	140	573	30	30
9	60	135	150	591	30	30
10	85	160	180	657	30	3

Table 2 Aircraft data of case 1

And security intervals are showed in following the table

Aircrafts	1	2	3	4	5	6	7	8	9	10
1	/	3	15	15	15	15	15	15	15	15
2	3	/	15	15	15	15	15	15	15	15
3	15	15	/	8	8	8	8	8	8	8
4	15	15	8	/	8	8	8	8	8	8
5	15	15	8	8	/	8	8	8	8	8

6	15	15	8	8	8	/	8	8	8	8
7	15	15	8	8	8	8	/	8	8	8
8	15	15	8	8	8	8	8	/	8	8
9	15	15	8	8	8	8	8	8	/	8
10	15	15	8	8	8	8	8	8	8	/

Table 3 Security landing intervals

Using the Ant-Q algorithm presented in Section 5 with heuristic value regarding the earliest landing time, the landing sequence achieved is showed in the flowing table, obviously it is not the optimal solution with a total penalty cost of 2860:

Aircraft landing sequence	3	4	5	6	7	8	9	1	10	2
Landing Time	98	106	123	135	143	151	159	174	189	258

Table 4 A typical landing sequence regarding the earliest landing time

Then we will simulate the scheduling process artificially step by step to discover the a adjustment method . First of all, the first 3 aircrafts 3,4,5 do land at their estimated landing time therefore this no need for adjustment. So, we just assign landing time as usual using formula (19),then it comes to the stage showed as follow:

Aircraft landing sequence	3	4	5	6	7					
Landing Time	98	106	123	135	143					

Table 5 Step 5 of the solution for case 1

If we follow the rule of formula (19), aircraft 7 will land at 143 which is later than its estimated landing time. The penalty cost per time unit of aircraft 6 and other scheduled aircrafts are same, so adjustment seems necessary for it can not bring any decrease men in total penalty cost. However, landing as earliest as possible within the same penalty cost is fruitful as well, we accept this adjustment. And the adjustment can be made by 4 time units since the penalty cost change is $4 * 30 - 3 * 30 = 0$ and the interval between aircraft 5 and aircraft 6 gets to the limitation $S_{56} = 8$. The stage then comes to:

Aircraft landing sequence	3	4	5	6	7					
Landing Time	98	106	123	131	139					

Table 6 Adjusted Step 5 of the solution for case 1

To describe this adjustment method, we establish a model called "Pushing forward" which is very interesting and easy to understand. This model regards the adjustment as aircraft scheduled latter pushing backward aircrafts scheduled previously. As it showed in **Figure 4**, each aircraft has two value:strength and energy. Force comes from its penalty cost per time unit and energy is the tardiness. The aircrafts with energy can push forward aircrafts with 0 energy(stopped) and aircrafts can only push aircrafts with Weight in same value comparing with their total strength.When a aircraft stops, it gets a weight comes from penalty cost per time unit. Aircrafts can move forward until they meet the further front aircraft, the distance is subject to the earliest time time and security interval.

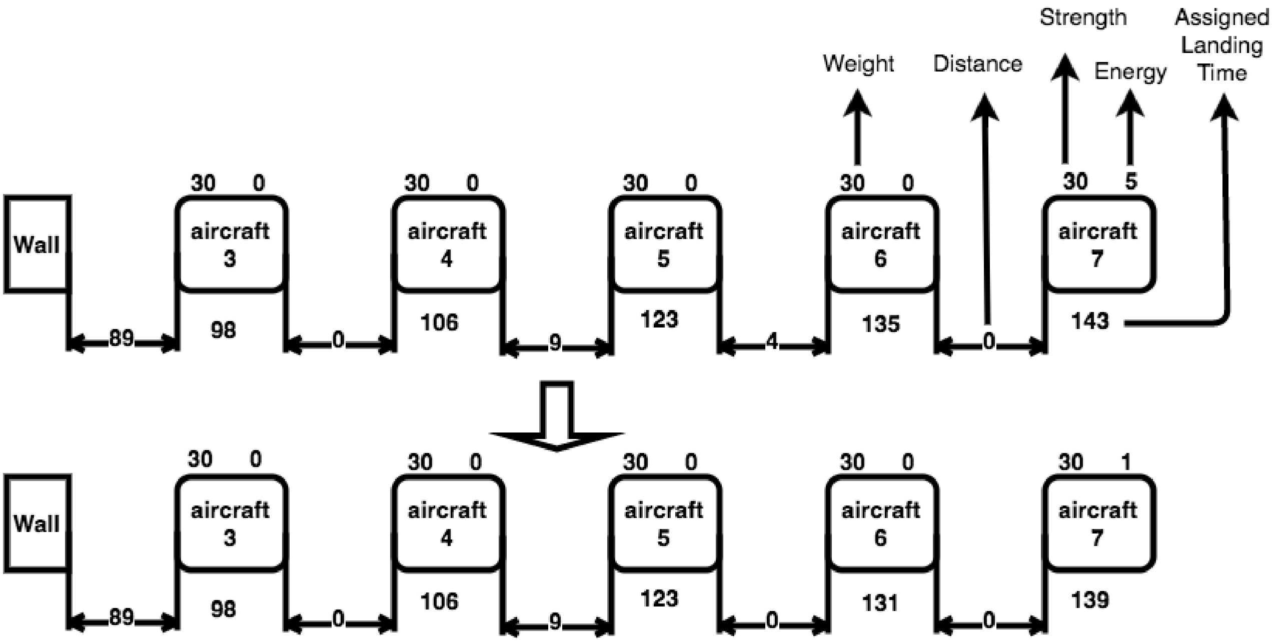

Figure 4 Pushing backward adjustment model

For example, **Figure 2**reports the adjustment process on stage 6. For aircraft 7, its strength is 30 and energy is 5(143-138). Obviously, distance between aircraft 7 and aircraft 6 is 0 as the landing time assigned to aircraft 7 is bigger than estimated landing time. Then, distance between aircraft 5 and aircraft 6 is $4 = 123 - (106 + 8)$, which respects security interval. As it is showed that the aircraft 7 is facing one aircraft to push since aircraft 6 can still move in distance 4 and i2t can push aircraft 6 with strength 30 equal to the weight of aircraft 6. After moving for 4 units, aircraft 7 consume 4 units energy and aircraft 6 meet with aircraft 5(distance =0). Now, aircraft 7 is facing two aircraft to push but it can’t, because aircraft 7 does not have enough strength to push aircrafts with weight 60(60<30). Finally, the adjustment finishes and the result is the same with the one shown in **Table 5**.

Then when aircraft 8 comes, other adjustment is made. As it is shown in **Figure 5**, aircraft 7 and aircraft 8 have enough strength 60 to push aircraft 5 and with weight of 60. After 1 time unit adjustment, aircraft 7 runs out its energy and stops. Then aircraft 7 can't push 3 aircraft backward so the adjustment is accomplished.

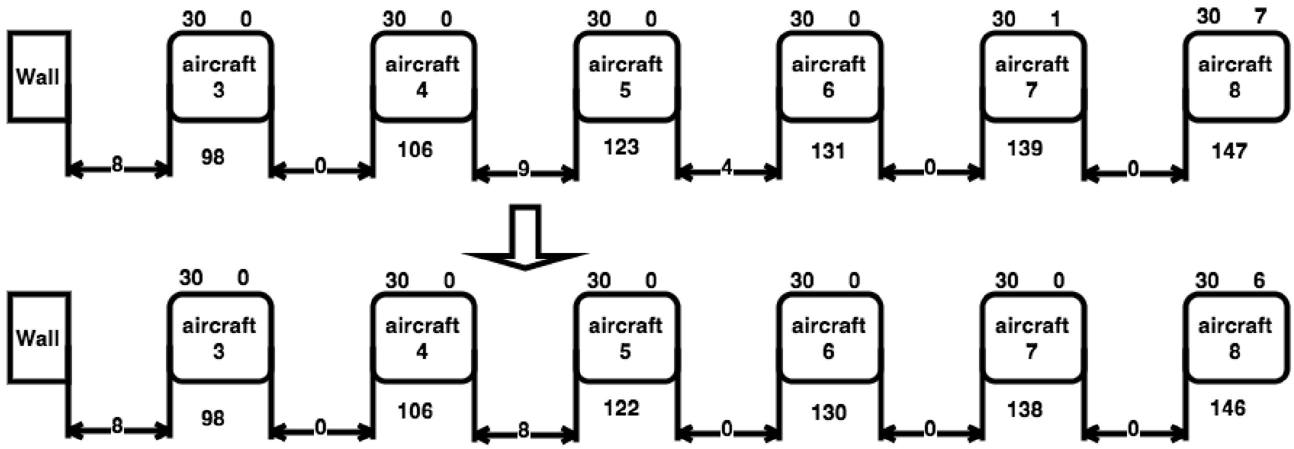

Figure 5 Adjustment process on stage 6

The adjustment processes are shown in following tables:

Aircraft landing sequence	3	4	5	6	7	8				
Landing Time	98	106	122	130	138	146				

Table 7 Adjusted Step 6 of the solution for case 1

Aircraft landing sequence	3	4	5	6	7	8	9			
Landing Time	98	106	122	130	138	146	153			

Table 8 Adjusted Step 7 of the solution for case 1

Aircraft landing sequence	3	4	5	6	7	8	9	1		
Landing Time	98	106	120	128	136	144	152	167		

Table 9 Adjusted Step 8 of the solution for case 1

Aircraft landing sequence	3	4	5	6	7	8	9	1	10	
Landing Time	98	106	120	128	136	144	152	165	180	

Table 10 Adjusted Step 9 of the solution for case 1

Aircraft landing sequence	3	4	5	6	7	8	9	1	10	2
Landing Time	98	106	12	128	136	144	152	165	180	258

Table 11 Adjusted Step 10 of the solution for case 1

The total penalty cost is: 700 which equals to optimal solution calculated by CPLEX software package(Beasley 2000). Next the mathematical expression will demonstrated,for each aircraft i landing sequence :

1.Strength

$$Str_i = Pgh_i; \qquad (20)$$

2.Energy

$$Eng_i = tt_i \qquad (21)$$

3.Weight

$$W_i = Pgh_i \qquad (22)$$

4.Distance between aircraft j landing after aircraft (j>i)

$$Dij = \begin{cases} t_i - (T_i + S_{ij}) & if|i-j| = 1 \\ D_{i(i+1)} + D_{(i+1)j} & otherwise \end{cases} \quad (23)$$

5. Aircrafts do pushing in one adjustment: for aircraft i in landing sequence, if $Pu_i = 1$, it is in the pushing set

$$Pu_i = \begin{cases} 1 & if Eng_i > 0 \ and \exists n(n > 0) \ (Pu_{i+n} = 1 \ and D_{i(i+n)} = 0) \\ 1 & else \ if Eng_i > 0 \ and \ aircraft \ i \ is \ the \ last \ one \ of \ current \ landing \ sequence \\ 0 & otherwise \end{cases} \quad (24)$$

6.Aircrafts to be pushed in one adjustment:for aircraft j, if $BPu_j = 1$, it is in the set to be pushed

$$BPu_i = \begin{cases} 1 & if Eng_i = 0 \ and Pu_{i+1} = 1 \ and D_{i(i+1)} = 0 \\ 1 & else \ if Eng_i = 0 \ and tPu_{i+1} = 1 \ and D_{i(i+1)} = 0 \\ 0 & otherwise \end{cases} \quad (25)$$

Let the pushing set be PuS and the set to be pushed $BPuS$.

7.Adjustment, time shifting for every aircraft i in PuS and $BPus$ follows the rule:

$$t_i = \begin{cases} t_i - \min_{u \in PuS} Eng_u & if \sum_{v \in PuS} Str_v \geq \sum_{w \in BPuS} W_w \\ ti & otherwise \end{cases} \quad (26)$$

Here is a complex example(not from the case) shown as follow:

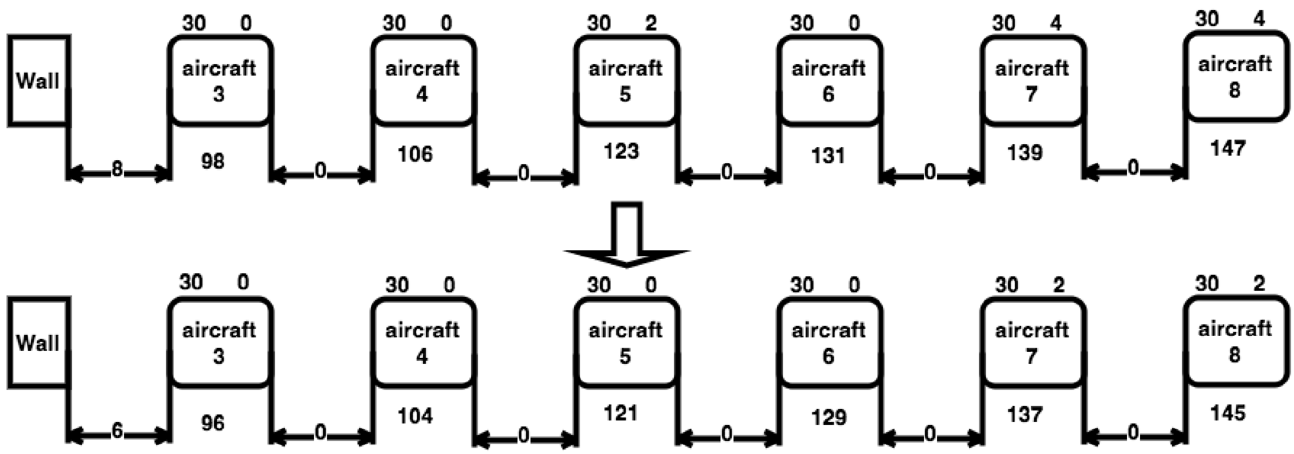

Figure 6 A complex example of time shifting

In **Figure 4**, when aircraft 8 joins the landing sequence, according to the formula (24), aircraft 8 is in PuS. Then, because $D_{78} = 0; D_{58} = 0; Pu_8 = 1$,aircraft 5 and 7 are also in PuS . According to the formula (25), in $BPuS$ there are aircraft 3,4,6. The total Strength of PuS is 90 and the total weight of $BPuS$ is 90, so the adjustment is accepted. The time shift calculated by formula (26), is minimum energy from PuS :2. Finally, the adjustment is made by shifting back for 2 time units.

To improve the AntQ algorithm, we add the adjustment method presented above to the AnQ. The improved Ant-Q algorithm can be called “Ant-Q with Pushing Backward Model”(PBM-Ant-Q).

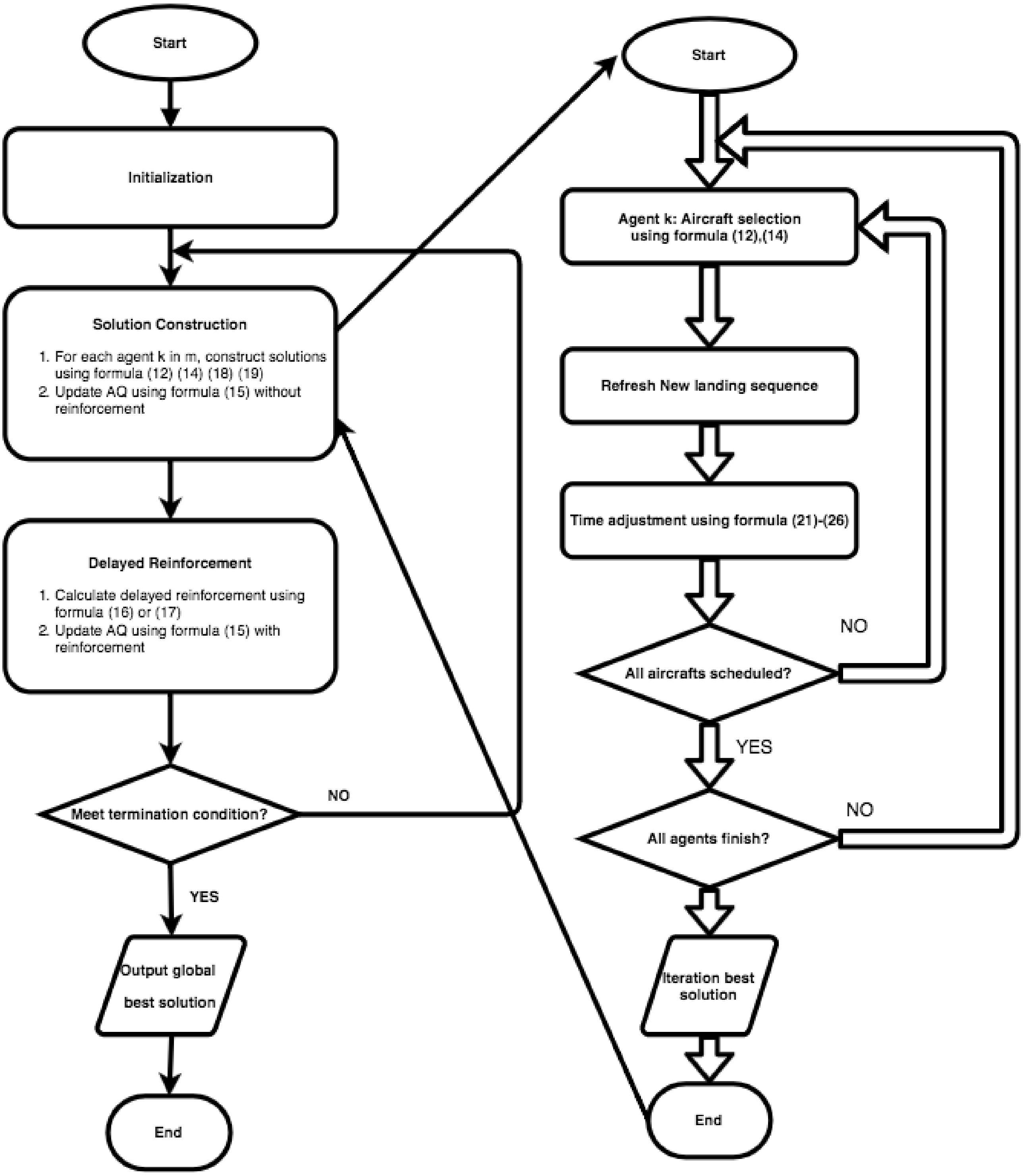

Figure 7 Flow chart PBM-Ant-Q

As it can be seen in **Figure 5**, PBM-Ant-Q add a time adjustment process after aircraft selection process.

To conclude, in this sub-section, we improve the Ant-Q by presenting PBM-Ant-Q for more optimal solution. Then it is the computational effort should be concerned in next section.

6.2 CPS running

Dear (1976)claimed that the landing sequence should not show relatively large difference from the FCFS sequence. There for set a constrained position shifting can reduce the scale of solution space without decrease the quality of solutions. Balakrishnan and Chandran(2006a)

To use CPS as a pruning method, we propose two ways: pre-calculation of Ant-Q-values matrix and tabu list.

6.2.1 Pre-calculation of Ant-Q-values

Since the CPS respects the sequence that comes, we can apply the CPS by calculating a initialized AQ_0 . The AQ_0 is like a terrain for ant agents, and agents tend to avoid the path with a higher gradient.

We can constrain position shifting according to approaching time(FCFS), earliest landing time or estimated landing time. The calculation rules are shown as follows:

$$AQ_0(r,s) = \begin{cases} Q_h & if Es - Er > X \\ Q_l & elseif Es - Er < 0 \\ Q_n & otherwise \end{cases} \quad (27)$$

In formulation, Qh represents the higher gradient and Ql is more flat. X is a threshold for interval of earliest landing times. When the interval between r and s is greater than X, it means that s has a great chance to break CPS constraint so the AQ(r,s) has got a Q_h. For k-CPS, X shall be k+1 times of average interval:

$$X = (k+1)Avg(\textit{intervals of ealiest landing time}) \quad (28)$$

$$X = (k+1)Avg(\textit{intervals of approaching landing time}) \quad (29)$$

$$X = (k+1)Avg(\textit{intervals of estimated landing time}) \quad (30)$$

Formula (28),(29),(30) calculate the threshold. And here is the AQ_0 for case 1:

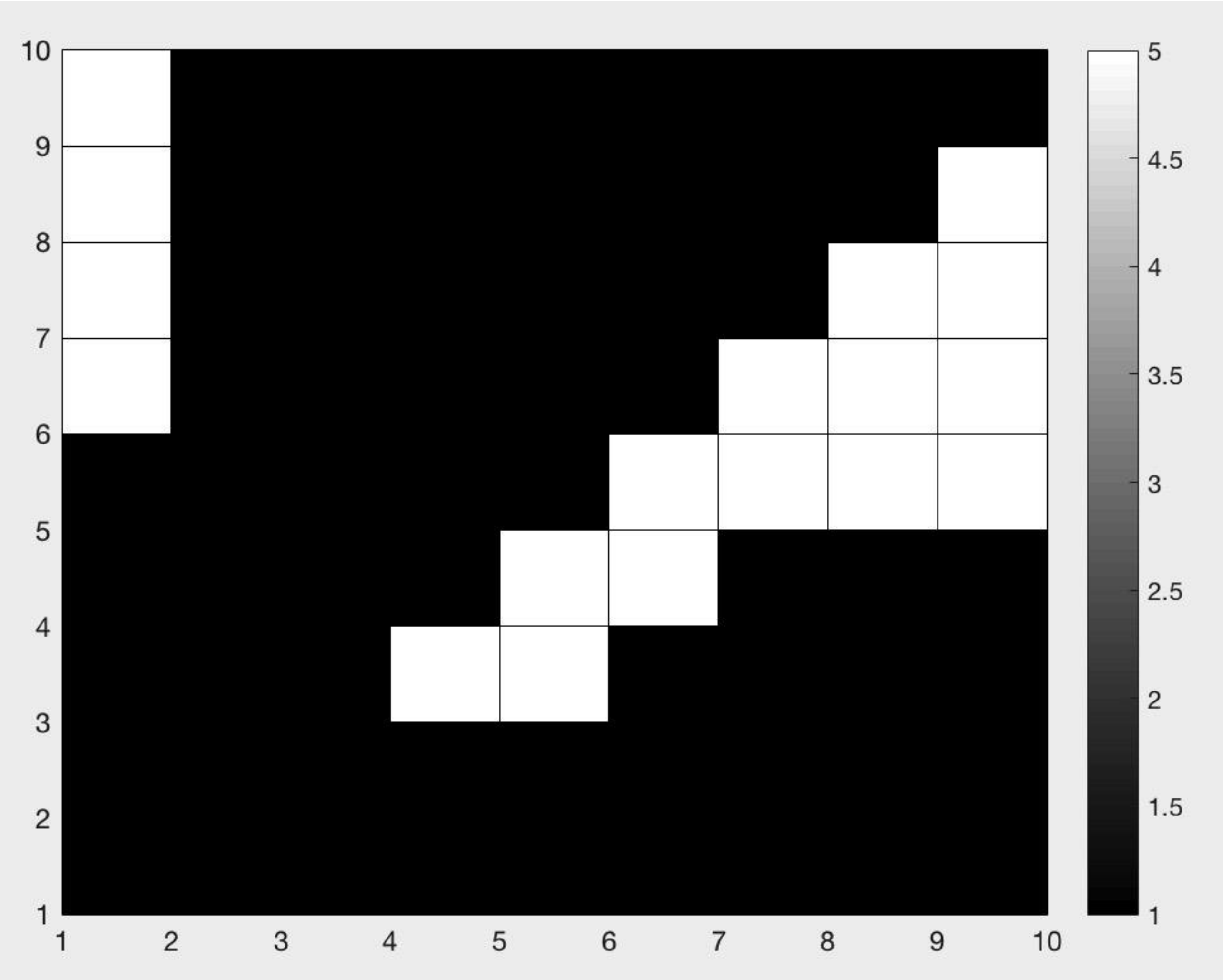

Figure 8 AQ0 AQ_for case 1

It can be seen clearly that for every aircraft there are a set of aircraft with higher priority to choose(shown as write sure) and another set unlikely to be chosen. This AQ_0 can narrow the initial solution space in order to increase the computational effort.

6.2.2 Tabu list

Meanwhile, we can forbid some sub sequence using a technic called Tabu list.

For k-CPS, the position shift is limited in k positions. For example, in case 1 showed before, Tabu list of 1-CPS considering earliest landing time will forbid some path when choose aircraft 1 to landing at first place. The example is shown as follow:

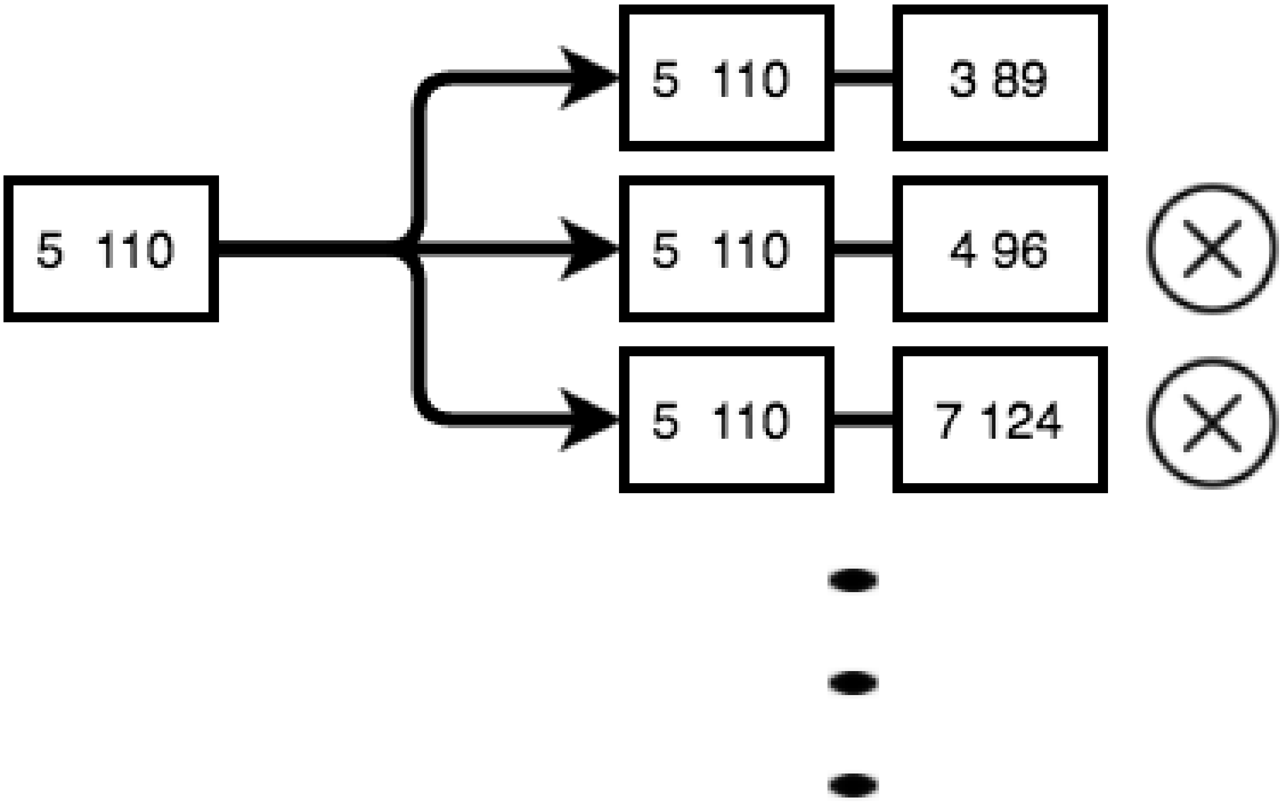

Figure 9 A Tabu example of 1-CPS

As we can see in **Figure 9**, the easiest landing time of aircraft 3 is 89 and it is at the first position in the order of earliest time. For 1-CPS, the aircraft 3 could not shift more than one position so the for step 2, all aircrafts expect aircraft 3 are forbidden and in the Tabu list. Therefore, for k-CPS, at step n, if aircraft with time order O has not landed yet, the Tabu list is:

$$Tabu(agent, step_n, aircraft_i) = \begin{cases} 0 & if \exists u \in J_k : O_u + k \leq n \; and \, i = u \\ 1 & if \exists u \in J_k : O_u + k \leq n \; and \, i \neq u \\ 0 & otherwise \end{cases} \quad (31)$$

For aircraft i, $Tabu(agent, n, i) = 1$ means that agent at step n must not choose aircraft i to land next. Note that the Tabu list is dynamic and updated step by step rather than set in the initialization phase.

To conclude, in this section we proposed two improvements for Ant-Q to optimize solutions and to improve computational efficiency.

Chapter 7 Computational results

In this experiment we used 2.7 GHz Intel Core i5 processor, 8 GBAM. These algorithms have been tested on cases from OR-Library(http://people.brunel.ac.uk/~mastjjb/jeb/info.html)(Beasley 2000).

The following numeric results are the best solutions from 100 round running the algorithms:

Case number	Optimal	FCFS	Ant-Q	PBM-Ant-Q	iterations
1	700	1790	1150	700	10
2	1480	2610	1720	1480	10
3	820	2930	1610	820	10
4	2520	6290	4480	2520	10
5	3100	8370	5040	3100	10
6	24442	24442	24492	24442	10
7	1550	1550	5441	1550	10
8	1950	26835	3585	1950	10

Table 12 Test results by Ant-Q and PBM-Ant-Q

Results in **Table 12** shows that Ant-Q can just solve the ALP approximately and the during the tests, it can be found that for those case whose optimal solution equals to FCFS sequence, the

Ant-Q 's solutions may have a bit of deviation from the optimal solution. However for most cases, the Ant-Q can obvious overcome FCFS and the PBM can significantly improve the Ant-Q to the optimal solution. The PBM can improve the Ant-Q from an approximation algorithm to an nearly exact algorithm. Note that all the best solutions are solved within 10 interactions which means the computational efficiency of the Ant-Q is fairly not low.

The average values of solution in 100 rounds by the generic Ant-Q are compared with best solution by the generic ACS(Bencheikh et al. 2011) as follows

Ant-Q best	Ant-Q average	ACS
1150	1205	1150
1720	2001	1840
1610	1994	2540
4480	4746	4820
5040	6360	6260
24492	27361	64001
5441	6040	53342
3585	7230	13840

Table 13 Comparation between the Ant-Q and the ACS

As we can see, even the average solution can beat ACS in 70% of cases especially large problems. Therefore, the Ant-Q is more suitable to solve the ALP and the main objectives of this thesis has been achieved by now.

As the PBM-Ant-Q can solve the cases optimally in seconds, solutions the generic Ant-Q with CPS improvements are

Case number	Ant-Q	1-CPS	2-CPS	3-CPS
1	1150	1150	1150	1150
2	1720	1722	1720	1720
3	1610	1610	1610	1610
4	4480	4480	4480	4480
5	5040	4800	5040	5040
6	24492	24492	24492	24492
7	5441	5840	5441	5441
8	4230	4230	4443	4243

Table 14 Test results by Ant-Q with CPS

CPS improvements do not lead to significant decrease on values of best solution even sometimes higher the the solutions by the generic Ant-Q. The reason why CPS improvements are not so functional may be the case problems are too small and the optimal solution basically in FCFS sequence.In the future, CPS improvements shall be tested by larger cases to prove their ability.

In a addition, the following **Figure 10** shows the final Ant-Q-values from case 8.

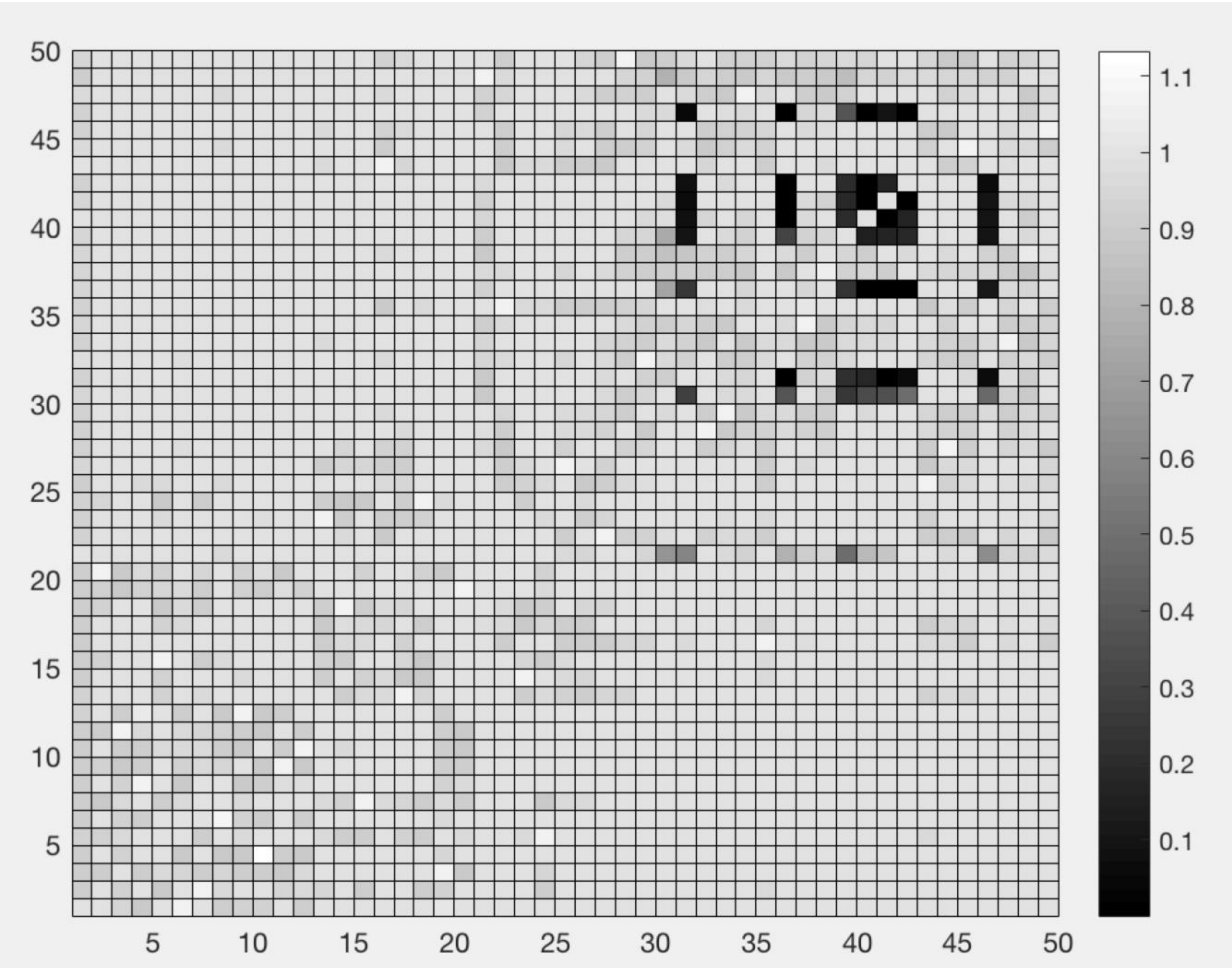

Figure 10 Ant-Q-values of best solution from case 8

The Ant-Q-values does not show much about the best solution but does prune branches at the interactions of the beginning and the end.

The result calculated by the Ant-Q algorithm are showed and discussed in this Section and it proves that the Ant-Q indeed is a good alternative approach to solve and research the ALP.

Chapter 8 Conclusion

This thesis develop an adaption algorithm of the Ant-Q to solve aircraft landing problem. The Aircraft landing problem is a very important issue in reality. The problem researched in this thesis is to find a optimal landing sequence subjects to several constraints, minimizing the total penalty cost.

The Ant-Q is first introduced to solve TSP and its “relative” Ant Colony Optimization was found able to solve aircraft landing problem. Since the ALP is similar to the TSP and there is no previous work using the Ant-Q to solve ALP, applying Ant-Q on the ALP is an interesting and proper choice to make. However, like the Ant System and As, the asymptotic rate of convergence of generic Ant-Q is low and the solutions are easy to fall into local optimum. Moreover, along with the growth of problem scale , the computational time gets higher and higher to a enormous value. Hence, improvements must be made to the Ant-Q algorithm. We introduction two different improved to optimize solutions and to increase computational efficiency. In the end, the proposed algorithms are implemented in Matlab and tested by OR-Library, the computational results are then evaluated.

First of all, Section 2 describes the aircraft landing problem in detail and discusses main constraints exist in scheduling aircraft landing. It is found that constraints for aircraft landing problem are security intervals, time windows and constrained position shifting. And the objective of the ALP in this search is set as the total penalty cost.

Then, many previous work on aircraft landing problem are reviewed in Section 3. Dynamic programming, Branch and bound and GA are found able to solve ALP well in the previous posts. To give a new idea to solve the ALP, perspective of reinforcement learning is considered to be applied to the ALP. Ant Colony Optimization is good entry point to achieve this and the ACO algorithms on the ALP are reviewed in more detail.

Chapter 8 Conclusion

In Section 4, the aircraft landing problem is formulated in mathematical way. Formulas to describe basic concepts, constraints and objective of the ALP are presented accurately which provide good foundation for developing the Ant-Q adaption for the ALP.

The generic adaption of Ant-Q algorithm is presented in Section 5 and it provide a full framework based on learning perspective to solve the ALP. Then improvements are made in Section 6. The most significant improvement is the pushing backward model we propose. The PBM-Ant-Q can adjust landing sequence by time shifting , following the model we present. Moreover, methods to exploit CPS constraint are also present in two way: initial Ant-Q-values and Tabu list.

Next,various versions of the Ant-Q algorithms are implemented and tested by running the test cases from the OR-Library. The computational shows that Ant-Q adaption shows its ability to solve the ALP very well and the PBM significant improvement on the objective value of the best solution. However, the improved algorithm still shows sometimes the problem of premature local convergence and CPS methods seems not able to deal with this problem.

To conclude, this research has achieved the main objectives proposed in the Sections. First, it managed to make a proper mathematical define a appropriate mathematical formulation for static, single-runway Aircraft Landing Problem with comprehensive set of considerations of constraints in practice. Second, the Ant-Q algorithm is successfully adapted to solve the ALP. Finally improvements made for the Ant-Q are proved to be able to boost the performance of the Ant-Q algorithm, both in objective value of best solution and the computational efficiency.

Chapter 9 Future work

This research managed to provide a study ideal to solve aircraft landing problem. However, more improvements still can be made:

* Extensions: The framework shall be extended to solve the ALP in multi-runway airport and should be able to deal with dynamic cases.

* Parameter optimization: Choosing proper parameter values is the biggest problem in the Ant-Q and the most difficult challenge in this research. Due to limitation of author's ability, the parameters used in the implementation of this research are selected by hands and the best parameters for different test cases seems different. The premature local convergence is still a problem for the Ant-Q if the parameters are not appropriate and adjusting parameter blindly without a clear rule is the biggest limitation of this thesis. Further work should be done in developing a formal framework of parameters choosing for the ALP.

* More improvements on learning algorithms: Q-learning is currently a popular research field as machine learning comes to center stage today. More and more research are posted to improve the Q-learning and reinforcement learning. Those newest research findings can be used in the ALP since we introduce the learning method to it.

* Computational efficiency : the method we proposed in Section 6.2.1 sets a initial Ant-Q-Value for the ALP. Although the results are not very good. The AQ initialization must be a possible way to improve the Computational efficiency. Rules shall be found to calculate more proper initial AQ by better exploiting the structure of the ALP.

* Real time implementation:Due to the learning perspective of the ALP, the ant agents can be trained in advance using vast amounts of cases. Trained agent may be more suitable to deal with the scheduling in real time.

* Parallel programming: The Ant-Q is naturally can be implemented using Parallel programming. This work may greatly improve the computational performance of the Ant-Q.

Bibliography

Abela, J. et al., 1993. Computing Optimal Schedules for Landing Aircraft. *The 12th National Conference of the Australian Society for Operations Research*.

Approach, A.D.P., The, T.O. & Sequencing, A., 1978. Harilaos N . Psaraftis FTL Report R78-4 October 1978 Flight Transportation Laboratory Massachusetts Instittue of Technology Cambridge , Massachusetts.

B a uerle, N., Engelhardt-Funke, O. & Kolonko, M., 2006. On the waiting time of arriving aircrafts and the capacity of airports with one or two runways. *European Journal of Operational Research*, 177(2).

Balakrishnan, H. & Chandran, B., 2006a. Scheduling aircraft landings under constrained position shifting. *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference 2006*, 4.

Balakrishnan, H. & Chandran, B., 2006b. Scheduling aircraft landings under constrained position shifting . *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference 2006*, 4.

Balakrishnan, H. & Chandran, B.G., 2010. Algorithms for Scheduling Runway Operations Under Constrained Position Shifting. *Operations Research*, 58(6).

Beasley, J.E., 2000. Scheduling Aircraft Landings The Static Case. *Transportation Science*, 34(2).

Beasley, J.E., Sonander, J. & Havelock, P., 2001. Scheduling aircraft landings at London Heathrow using a population heuristic. *Journal of the Operational Research Society*, 52(5).

Bibliography

Bencheikh, G. & Boukachour, J., 2009. Hybrid method for aircraft landing scheduling based on a Job Shop formulation. *International Journal of {\ldots}*, 9(8).

Bencheikh, G., Boukachour, J. & Alaoui, A., 2011. Improved Ant Colony Algorithm to Solve the Aircraft Landing Problem. *International Journal of {\{}{\ldots}{\}}*, 3(2). Available at: http://www.ijcte.org/papers/309-G442.pdf.

Bencheikh, G., Boukachour, J. & El Hilali Alaoui, A., 2016. A memetic algorithm to solve the dynamic multiple runway aircraft landing problem. *Journal of King Saud University - Computer and Information Sciences*, 28(1).

Bennell, J., Mesgarpour, M. & Potts, C., 2011. Airport Runway Scheduling. *4OR - A Quarterly Journal of Operations Research*, 9(2).

Bennell, J.A., Mesgarpour, M. & Potts, C.N., 2011. Airport runway scheduling. *4OR*, 9(2). Available at: http://link.springer.com/10.1007/s10288-011-0172-x.

Bennell, J.A., Mesgarpour, M. & Potts, C.N., 2016. Dynamic Scheduling of Aircraft Landings. *European Journal of Operational Research*. Available at: http://dx.doi.org/10.1016/j.ejor.2016.08.015.

Brinton, C.R., 1992. An implicit enumeration algorithm for arrival aircraft. *[1992] Proceedings IEEE/AIAA 11th Digital Avionics Systems Conference*.

Briskorn, D. & Stolletz, R., 2014. Aircraft landing problems with aircraft classes. *Journal of Scheduling*, 17(1). Available at: http://link.springer.com/10.1007/s10951-013-0337-x.

Chandran, B. & Balakrishnan, H., 2007. A Dynamic Programming Algorithm for Robust Runway Scheduling. *2007 American Control Conference*.

Bibliography

Cheng, V.H.L., Crawford, L.S. & Menon, P.K., 1999. Air traffic control using genetic search techniques. *Proceedings of the 1999 IEEE International Conference on Control Applications (Cat. No.99CH36328)*, 1.

Ciesielski, V. & Scerri, P., 1998. Real time genetic scheduling of aircraft landing times. *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*.

Dorigo, M. & Gambardella, L.M., 1997. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1).

Dorigo, M., Maniezzo, V. & Colorni, A., 1996. Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 26(1).

Ernst, a T., Krishnamoorthy, M. & Storer, R.H., 1999. Heuristic and exact algorithms for scheduling aircraft landings. *Networks*, 34.

Farah, I. et al., 2016. Ant Colony Optimization for Aircraft Landings.

Gambardella, L.M. & Dorigo, M., 1995. Ant-Q: A Reinforcement Learning approach to the traveling salesman problem. *Proceedings of ML-95, Twelfth Intern. Conf. on Machine Learning*.

Heathrow, A. report from, 2015. Heathrow Airport Holdings Limited Annual report and financial statements.

Holland, J.H., 1992. Genetic Algorithms - Computer programs that ‘evolve’ in ways that resemble natural selection can solve complex problems even their creators do not fully understand.

Hu, X. & Chen, W., 2005. Genetic Algorithm Based on Receding Horizon Control for Real Time Implemenetations in Dynamic Environment. *Control*.

Land, A.H. & Doig, A.G., 1960. An automatic method of solving discrete programming problems. *Econometrica*, 28(3). Available at: http://www.jstor.org/stable/1910129.

Lee, H. & Balakrishnan, H., 2008. A study of tradeoffs in scheduling terminal-area operations. *Proceedings of the IEEE*, 96(12).

Lieder, A., Briskorn, D. & Stolletz, R., 2015. A dynamic programming approach for the aircraft landing problem with aircraft classes. *European Journal of Operational Research*, 243(1).

Mesgarpour, M., 2012. Airport Runway Optimization. *Ph.D thesis FACULTY OF SOCIAL AND HUMAN SCIENCES SCHOOL OF MATHEMATICS UNIVERSITY OF SOUTHHAMPTON*.

Pinol, H. & Beasley, J.E., 2006. Scatter Search and Bionomic Algorithms for the aircraft landing problem. *European Journal of Operational Research*, 171(2).

Randall, M.C.C., 2002. Scheduling Aircraft Landings using Ant Colony Optimisation. *PROCEEDINGS OF THE IASTED INTERNATIONAL CONFERENCE ARTIFICIAL INTELLIGENCE AND SOFT COMPUTING*, pp.129–133. Available at: http://www.actapress.com/Abstract.aspx?paperId=26022.

Roger G. Dear, 1976. The synamic scheduling of aircrafting in the near terminal area.

St u tzle, T. & Hoos, H.H., 2000. MAX-MIN Ant System. *Future Generation Computer Systems*, 16(8).

Wang, S., Zhong, S. & Yu, H., 2010. The Optimization of Flight Landing Sequence. *Journal of System and Management Sciences {\ldots}*, 1(3).

Wen, M., 2005. Algorithms of Scheduling Aircraft Landing Problem.

Yu, S.P., Cao, X.B. & Zhang, J., 2011. A real-time schedule method for Aircraft Landing Scheduling problem based on Cellular Automation. *Applied Soft Computing Journal*, 11(4).

Zhan, Z.H. et al., 2010. An efficient ant colony system based on receding horizon control for the aircraft arrival sequencing and scheduling problem. *IEEE Transactions on Intelligent Transportation Systems*, 11(2).