# Review on computational techniques in solving aircraft landing problem

**Aminurafiuddin Zulkifli, Nor Azlina Ab. Aziz, Nor Hidayati Abdul Aziz**
*Faculty of Engineering and Technology, Multimedia University, 75450 Melaka, Malaysia*

**Zuwairie Ibrahim**
*Faculty of Electrical and Electronics Engineering, Universiti Malaysia Pahang, Pekan, 26600 Pahang, Malaysia*

**Norrima Mokhtar**
*Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia*

*Email: aminurafiuddin@gmail.com, azlina.aziz@mmu.edu.my, hidayati.aziz@mmu.edu.my, zuwairie@ump.edu.my, norrimamokhtar@um.edu.my*

**Abstract**

The problem of sequencing and scheduling arriving aircraft landing is commonly known as aircraft landing problem (ALP). This problem, due to various constraints such as the number of arriving aircrafts, the number of runways, the mode of runway operation, the type of arriving aircrafts, the minimum separation between each arriving aircraft, and the weather condition, is considered to be a NP-hard problem. Therefore, it is almost impossible to compute every possible solution and computational intelligence methods had been adopted to solve ALP. In this paper, we review the computational intelligence techniques used in ALP. The main techniques include the evolutionary algorithms namely; genetic algorithm, genetic programming, scatter search and bionomic algorithm, the swarm intelligence algorithms like particle swarm optimization and ant colony optimization and also other methods such as the constrained position shifting and dynamic programming.

Keywords— aircraft landing problem; computational intelligence; evolutionary algorithms; swarm intelligence; scheduling; runway operation

## 1. Introduction

Over the years, many studies had been conducted in the field of air traffic control. Since the aviation industry is expanding, a need for closer attention to this field has arisen. The complexity of air traffic operation is motivated by the increase in air traffic volume every year. In Europe and US, increase in air traffic demand is expected to double in the next 15 years [1]. In Malaysia, according to Malaysian Ministry of Transport an increase of 2.7% was reported in term of total commercial aircraft movements handled by Malaysian airport in 2015 compared to 2014 [2].

One of the important task in air traffic control is managing the air traffic operations for aircraft take-off and landing. This task is handled by air traffic controllers. The controllers determine which aircraft is taking off from, or landing on, available runway at airports, subject to operational constraints. To handle the landing and taking-off of an aircraft, is a very challenging process. It is highly related to safety, efficiency, robustness, and competitiveness issues. The common practice is to tackle the task using first come first serve (FCFS) basis. This method may not be efficient but it is certainly the simplest way to manage the operation.

However, with rapid development in computing, researchers have been developing multiple ways to aid the process of scheduling and sequencing aircrafts. This include the adaptation of computational intelligence. In this paper, we focus on the application of computational intelligence in solving the problem of sequencing and scheduling landing aircrafts, which is commonly known as aircraft landing problem (ALP).

The remainder of this paper is organized as follows. Basic concept of aircraft landing problem is briefly explained in section II. Section III describes the computational techniques used in order to optimize ALP. Section IV concludes this work.

## 2. Aircraft Landing Problem

ALP can be defined as a process of sequencing and scheduling the arriving aircrafts. Nowadays with airports operating with more than one runways, ALP can be viewed as two objectives problem, which are sequencing the optimal landing order of the arriving flights and scheduling a runway for each arriving aircraft [3]. In [1], the objectives are listed as; a) maximizing runway throughput, b) minimizing the approach time of aircraft before landing, c) minimizing the arrival delay, d) minimizing air traffic controller's workload. Thus, one of the challenges for the researchers in this field is to develop a solution capable of dealing with a variety of objectives. This raise the issue of which objective function to adopt. By far ALP's objective function causes the most discussion among researchers. Arguments can convincingly be made for many different objective functions. Different users will, for perfectly legitimate reasons, use different objective functions [4].

ALP is a dynamic problem as the controller need to operate in real-time. The controller must generate an updated schedule for the set of arriving aircraft to be sequenced and scheduled both periodically and in response to aperiodic events, while the length of the periodic cycle is related to the basic radar update time interval, which is 4–12 seconds long [5].

In most studies, the common structure of ALP (static case) has $n$ number of landing aircrafts, which the aircrafts denoted as $i = 1, 2, \ldots, n$, and the estimated earliest landing time is $E$, and the latest landing time is $L$. Therefore, the time window for an aircraft can is bounded within $[E(i),L(i)]$. Controller will assign a target landing time for every landing aircraft, $T$, and it must be in between $E$ and $L$, hence, $E(i) \leq T(\mathrm{i}) \leq L(i)$. The controller must also obey the minimum separation time constraint between leading aircraft and trailing aircraft. This is a mandatory safety measure to avoid complications like the wake turbulence and aircraft collision.

### *2.1 Problem modelling*

Psaraftis [6] models ALP into a simple version. He sums up the landing times of the arriving aircraft as cost/delay, which is described as Total Passenger Delay (TPD) while the throughput is measured based on the landing time of the last aircraft in the sequence, described as Last Landing Time (LLT). The objective is to minimize both TPD and LLT. Psaraftis also stated that aircraft sequencing problem is the same, as the NP-hard Travelling Salesman Problem (TSP), where, the cost of the graph represents the landing cost and the route represents the aircraft landing sequence. However, Psaraftis did not consider the aircraft latest landing, *L,* which allows the aircraft to land at any given time.

On the other hand, Bayen et al. [7] focuses on formulating holding time using two approaches. The scheduling process is optimized when the holding time is minimized and, therefore, minimize the sum of arrival time. They formulated ALP as a single machine job scheduling problem. However, they consider that all arriving flights in one large class. Thus, the required separation between landings is independent of the aircraft type.

Cheng et al. [8] emphasis on runway assignment for ALP. They discuss sequencing and scheduling a number of arrivals to a number of runways. The paper focuses on static case of ALP. The data set used contains the estimated time of arrivals (ETA). The aircraft performances and the flight pattern are also taken into consideration. The main objective is to minimize the delay, which is the difference between the scheduled time of arrival (STA) and the earliest ETA in all the runways for an aircraft. Hansen [9] improves the implementation made by Cheng with an experiment on larger set of aircrafts made up to the possible realistic level. The same formulation is also adopted in various work [10],[11] ,[12] ,[13].

Another study regarding to the static case of ALP was carried by Krishnamoorthy et al. [14] and then improved by Beasley et al. [4]. The study comprises both single-runway and multiple-runway cases. The objective is to minimize the total cost, where the cost is linearly related to deviation from target landing time for landing planes. Figure 1 shows how the cost of aircraft landing is introduced in the paper. Although the cost function shown is nonlinear, it can be linearized by decomposing the two linear portions and formulate the problem with a linear objective function. The paper includes various mathematical constraints to manifest every possible condition of ALP. This study has been cited by many ALP researchers as it presented a general description of the model, goals, and mathematical formulation of the landing planning for one or more runways [15], [16], [17], [18],[19],[20].

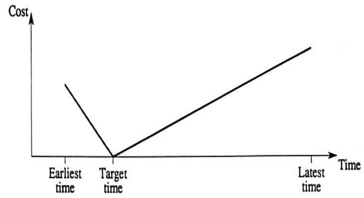


Figure 1. Cost variation in time window during flight

## 3. Computational Intelligence on ALP

This section reviews the literature related to ALP. The subsections are organized according to the computational intelligence method used.

### *3.1 Evolutionary algorithms*

Genetic algorithm (GA) is a famous evolutionary algorithm. It has been used in various problems including ALP. In previous work of Cheng et al. [8] the runway assignment is carried using genetic search algorithm. Four schemes of genetic search were used to solve a simple scenario involving 12 flights in 3 runways. The result shows application of genetic search gives an impressive result. The method is later improved by Hansen [9] with a larger data set, which is nearer to the realistic case. Hansen proposed the improved version of genetic algorithm called Genetic Programming (GP). GP shows a better performance than GA when dealing with larger set of arriving aircrafts. Liu [11] compared GA with newly developed Genetic Local Search (GLS), which is an extension of GA in solving ALP. Liu uses Hansen's case as well to experiment with GLS and it is proven to be more efficient in term of both optimality and time cost than GA.

Wang et al [3] adopted GA to solve ALP on parallel runways, which include independent approach and relevant approach. It is tested on 40 arriving flights.

Abela et al. [14] discussed two different solution for ALP which are using GA and branch-and-bound. Both approaches are tested with a data set and GA is proven to be effective in small problem. Hu and Di Paolo [13] have integrated GA with receding horizon (RH) strategy to tackle ALP. The chromosomes in GA are constructed using the arriving order and/or arriving time of each aircraft.

Pinol and Beasley [15] introduce a hybrid method using another evolutionary algorithms, scatter search and bionomic algorithm for ALP. Both methods are combined and used for problem instances up to 500 aircrafts and 5 runways. In linear ALP cases, scatter search performs better than bionomic algorithm while in non-linear cases, bionomic algorithm performs better than scatter search. These methods, are flexible to be adapted and able to change objective between linear and non-linear ALP cases.

### *3.2 Swarm Intelligences*

Other than evolutionary algorithms, swarm intelligence algorithms are also used in solving ALP. Among the Swarm intelligence algorithms used are particle swarm optimization (PSO), ant colony optimization (ACO) and gravitational search algorithm (GSA).

A PSO hybridized with local search (LS) on rolling horizon (RH) is proposed by Girish [21] to minimize the deviation cost of aircraft landings by using Beasley's formulations [4]. It is then compared with another two PSO variants. The RH framework with hybrid PSO-LS is proven to be more efficient and has shorter computational time.

Benceikh [19] uses ACO to solve ALP and the Beasley's formulation is improved by incorporating dynamic cases, for example, flight cancellation and runway closing. They also improved ACO with more robust heuristic named Improved ACO (IACO) to reduce the penalty cost.

Kazem et al [10] introduce GSA in their paper for the purpose of solving ALP. GSA is a swarm intelligence algorithm which was inspired by Newton's law of gravity [22]. Firstly, random sequence of aircrafts with their allocated runway are generated as initial solutions and each solution's fitness is computed. Using the Newton's law of gravity, each solution is improved on the next iteration. The study uses Hansen's data set and compared with genetic search algorithm, GA, scatter search and bionomic algorithm, and GSA is found to provide better solution in shorter amount of time.

### *3.3 Other techniques*

Dear [23] describes ALP as a dynamic problem and introduced the constrained position shifting (CPS) method. CPS works as the sequencing mechanism where the order of landing aircraft is only shifted by limited ***k*** positions from its FCFS position. Psaraftis [6] further improved Dear's study using backward dynamic programming algorithms that use the number of aircraft from each classes that has not yet been scheduled to land and the class of the last aircraft to land as the state variables. His method is also adaptable using CPS without increasing the time complexity. However, Psaraftis method assumes there is no landing restriction which means, the aircraft can land at any time. Balakrishnan et al [24] develop the work of Psaraftis's CPS by simultaneously handling precedence constraints, landing restriction and CPS operational constraints, which reduces the computational time and make it easy to be adopted in real-time for both static and dynamic cases. The updated work on CPS by Rodriguez-Diaz et al [25] show how CPS is adapted on single mixed-mode runway operation. They use simulated annealing (SA) to experiment with CPS on a very large data set. The data considered is up to 200 flights and 2000 instances. The study emphasizes on how the focus of research should include mixed-mode operation in their problem modelling.

Beasley et al. [4] on the other hand approach the problem from a different perspective using mixed-integer zero-one formulation which is adopted using a basic tree search strengthen with linear programing (LP) relaxation. A scenario of 50 aircraft and four runways is tested and the result showed the LP-based tree search are able to work with various constraints that commonly encountered in practice. However, since the problem is NP-hard, it is likely for the computation time to grow exponentially with the number of flights.

Bayen et al. [7] use dynamic programing (DP) and linear programming relaxation (LP) with rounding approach. DP approaches have the performance ratio of 5 for the sum of arrival times of all aircraft and LP relaxation with rounding has the performance ratio of 3 for the landing time of the last aircraft.

Lieder et al. [26] introduce DP in solving ALP with different aircraft classes on multiple runways with positive target landing times and limited time windows. A new dominance criterion is developed order to improve the performance of DP approach. The criterion based on the formulation by Briskorn et al [27].

Table 1 comprises all the techniques discussed on this section. The techniques are tabulated according to the methodology, the formulation and the instances used in their respective studies.

## 4. Conclusions

This paper provides an overview on how computational intelligence is used in ALP. From the review, it can be seen that computational intelligence techniques are popular in solving ALP. The problem modelling of ALP is also observed not to be uniform across the literature. For example, in some literatures, ALP is viewed as static case while other view it as dynamic case. For our future work, we will investigate on how to solve ALP using computational intelligence approach that are more robust and dynamic similar to the real problems faced by air traffic controllers.

**Acknowledgement**

This research is funded by the Fundamental Research Grant Scheme (FRGS/1/2015/ICT02/MMU/03/1), which is awarded by Ministry of Higher Education Malaysia to Multimedia University.

**References**

[1] J. A. Bennell, M. Mesgarpour, and C. N. Potts, "Airport runway scheduling," *Ann. Oper. Res.*, vol. 204, no. 1, pp. 249–270, 2013.
[2] M. of Transport, "Transport Statistics Malaysia 2015," 2015.
[3] L. L. Wang and Q. L. Gu, "Optimization model for sequencing arrival flights on parallel runways," *Proc. - 2014 Int. Conf. Inf. Sci. Electron. Electr. Eng. ISEEE 2014*, vol. 1, no. 61179042, pp. 489–492, 2014.
[4] J. E. Beasley, M. Krishnamoorthy, Y. M. Sharaiha, and D. Abramson, "Scheduling Aircraft Landings — The Static Case,"

*Transp. Sci.*, vol. 34, no. 2, pp. 180–197, 2000.
[5] H. Erzberger and E. Itoh, "Design Principles and Algorithms for Air Traffic Arrival Scheduling," no. May, pp. NASA/TM—2014–218302 Design, 2014.
[6] H. N. Psaraftis, "A Dynamic Programming Approach to The Aircraft Sequencing Proble," 1978.
[7] A. M. Bayen and C. J. Tomlin, "An Approximation Algorithm for Scheduling Aircraft with Holding Time †," pp. 2760–2767, 2004.
[8] V. H. L. Cheng, L. S. Crawford, and P. K. Menon, "Air traffic control using genetic search techniques," *Proc. 1999 IEEE Int. Conf. Control Appl. (Cat. No.99CH36328)*, vol. 1, pp. 249–254, 1999.
[9] J. V. Hansen, "Genetic search methods in air traffic control," *Comput. Oper. Res.*, vol. 31, no. 3, pp. 445–459, 2004.
[10] K. Dastgerdi, N. Mehrshad, and M. Farshad, "A new intelligent approach for air traffic control using gravitational search algorithm," *Sadhana*, vol. 41, no. 2, pp. 183–191, 2016.
[11] Y. H. Liu, "A genetic local search algorithm with a threshold accepting mechanism for solving the runway dependent aircraft landing problem," *Optim. Lett.*, vol. 5, no. 2, pp. 229–245, 2011.
[12] A. Salehipour, L. M. Naeni, and H. Kazemipoor, "Scheduling aircraft landings by applying a variable neighborhood descent algorithm : Runway-dependent landing time case," *J. Appl. Oper. Res.*, vol. 1, pp. 39–49, 2009.
[13] X. B. Hu and E. Di Paolo, "An efficient genetic algorithm with uniform crossover for air traffic control," *Comput. Oper. Res.*, vol. 36, no. 1, pp. 245–259, 2009.
[14] G. M. M. Krishnamoorthy, J. Abela, D. Abramson, A. De Silva, "Computing Optimal Schedules for Landing Aircraft," in *The 12th National Conference of the Australian Society for Operations Research*, 1993, pp. 1–15.
[15] H. Pinol and J. E. Beasley, "Scatter Search and Bionomic Algorithms for the aircraft landing problem," *Eur. J. Oper. Res.*, vol. 171, no. 2, pp. 439–462, 2006.
[16] J. E. Beasley, M. Krishnamoorthy, Y. M. Sharaiha, and D. Abramson, "Displacement problem and dynamically scheduling aircraft landings," pp. 54–64, 2004.
[17] F. Farhadi, A. Ghoniem, and M. Al-Salem, "Runway capacity management - An empirical study with application to Doha International Airport," *Transp. Res. Part E Logist. Transp. Rev.*, vol. 68, pp. 53–63, 2014.
[18] K. Dastgerdi, N. Mehrshad, and M. Farshad, "Journal of Soft Computing and Decision Support Systems A New Intelligent Approach to Aircrafts Take-off / Landing Planning at Congested Single Runway Airports," vol. 2, no. 2, pp. 17–25, 2015.
[19] G. Bencheikh, J. Boukachour, and A. Alaoui, "Improved Ant Colony Algorithm to solve the aircraft landing problem," *Int. J. Comput. Theory Eng.*, vol. 3, no. 2, pp. 224–233, 2011.
[20] I. Moser and T. Hendtlass, "Solving dynamic single-runway aircraft landing problems with extremal optimisation," *Proc. 2007 IEEE Symp. Comput. Intell. Sched. CI-Sched 2007*, pp. 206–211, 2007.
[21] G. B.S., "An efficient hybrid particle swarm optimization algorithm in a rolling horizon framework for the aircraft landing problem," *Appl. Soft Comput.*, vol. 44, pp. 200–221, 2016.
[22] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A Gravitational Search Algorithm," *Inf. Sci. (Ny).*, vol. 179, pp. 2232–2248, Jun. 2009.
[23] Roger G. Dear, "The dynamic scheduling of aircrafting in the near terminal area," pp. 1–325, 1976.
[24] H. Balakrishnan and B. G. Chandran, "Algorithms for Scheduling Runway Operations Under Constrained Position Shifting," *Oper. Res.*, vol. 58, no. 6, pp. 1650–1665, 2010.
[25] A. Rodriguez-Diaz, B. Adenso-Diaz, and P. L. Gonzalez-Torre, "Minimizing deviation from scheduled times in a single mixed-operation runway," *Comput. Oper. Res.*, vol. 78, pp. 193–202, 2017.
[26] A. Lieder and R. Stolletz, "Scheduling aircraft take-offs and landings on interdependent and heterogeneous runways," *Transp. Res. Part E Logist. Transp. Rev.*, vol. 88, pp. 167–188, 2016.
[27] A. Lieder, D. Briskorn, and R. Stolletz, "A dynamic programming approach for the aircraft landing problem with aircraft classes," *Eur. J. Oper. Res.*, vol. 243, no. 1, pp. 61–69, 2015.

Table 1. Overview of related literatures

| *Source* | *Methodology* | *Type of case* | *Objective* | *Problem instances* |
|---|---|---|---|---|
| Dear (1978) | Constrained Position Shifting | Static/dynamic | Max throughput<br>Min Σ total delay | Random generated instances |
| Psaraftis (1978) | Constrained Position Shifting | Static | Min Σ costs/min makespan | Random generated instances |
| Balakrishnan et al. (2006) | Constrained Position Shifting | Static | Min Σ costs/min makespan | Denver airport real data |
| Rodriguez-Diaz (2017) | Constrained Position Shifting | Static | Min Σ delay | Random generated instances, Beasley (2000) and Gatwick airport |
| Beasley et al (2000) | Linear programming | Static | Min Σ penalty cost | Beasley (1990) |
| Bayen (2004) | Linear programming and dynamic programming | Static | Min Σ arrival time<br>Min last aircraft landing time | No instances (theoretical analysis) |
| Cheng et al (1999) | Genetic algorithm | Static | Min Σ delay | Sampling data |
| Hansen (2004) | Genetic Algorithm, Genetic programming | Static | Min Σ delay | Cheng et al (1999) |
| Wang (2014) | Genetic Algorithm | Static | Min Σ delay | Sampling data |
| Liu (2011) | Genetic local search | Static | Min Σ penalty cost | Beasley (2000) |
| Abela et al (1993) | Genetic algorithm, Branch and bound | Static | Min Σ deviation cost | Randomly generated test data |
| Hu and Di Paolo (2009) | Genetic algorithm | Static | Min Σ delay<br>Max arrival queues | Hansen (2004) |
| Pinol and Beasley (2006) | Scatter search and Bionomic algorithm | Static | Min Σ penalty cost | Beasley (2000) |
| Girish (2016) | Particle swarm optimization with local search | Static | Min Σ penalty cost | Beasley (2000) |
| Benceikh et al (2011) | Ant Colony Optimization | Dynamic | Min Σ penalty cost and displacement function | Beasley (2000) |
| Kazem et al (2016) | Gravitational search algorithm | Static | Min Σ total delay | Hansen (2004) |