This is a post-peer-review, pre-copyedit version of a chapter published in Huang, DS., Bevilacqua, V., Premaratne, P., Gupta, P. (eds) Intelligent Computing Theories and Application. ICIC 2017. Lecture Notes in Computer Science, vol 10361. Springer, Cham. The final authenticated version is available online at: https://doi.org/10.1007/978-3-319-63309-1_50.

Aircraft scheduling considering discrete airborne delay and holding pattern in the near terminal area

K.K.H. Ng[1], C.K.M. Lee[1]

[1] Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University,
Hung Hom, Hong Kong, China
kkh.ng@connect.polyu.hk, ckm.lee@polyu.edu.hk

Abstract. In this paper, a constructive heuristic using the artificial bee colony algorithm is proposed to resolve the aircraft landing problem considering speed control for airborne delay and holding pattern in the near terminal area. Safety is a top priority in civil aviation management, and air traffic control has to consider handling air traffic promptly. The degree of conservatism in dealing with airborne and terminal traffic should be increased to maintain a high level of resilience for the runways system, enhance the robustness of landing schedule, and reduce the workload of air traffic controllers. The computational results show that the proposed algorithm can resolve the problem in a reasonable amount of time for practical usage.

Keywords: Aircraft Landing Problem, Discrete Airborne Delays, Holding Pattern, Swarm Intelligence, Artificial Bee Colony Algorithm

1 Introduction

The Air Traffic Control (ATC) tower plays a major role in maintaining smooth air traffic and balancing the airborne and airport traffic. The growing demand for air transport increases the pressure on the efficiency of ATC, especially during peak hours. Most of the runways in the international airports are foreseen to reach the maximum runway capacity. Airport capacity expansion is urgently needed to avoid the consequences of exceeding capacity and enhance resilience on managing airport resources. Overcrowded air traffic is a serious issue in managing passengers' satisfaction and comfort and affects airport's reputation. Also, the authorities have an obligation to attempt to resolve safety and delay issues arising at the turnaround and terminal control. Such "alarms" become critical in the future, as most airports foresee a strong growth in the aviation sector. ATC require a robust delay and risk management system in handling daily air traffic and maintaining modern aviation safety standards. The airport capacity in controlling turnaround free-flow progress is limited by a scarce resource – the runway [1, 2]. The planning and construction of a new runway require a long lead-time. Besides the need for runway expansion, aviation authorities are seeking the computational intelligence to reduce the workload of the ATC tower and utilise the current airport resources.

adfa, p. 1, 2011.

The Aircraft Sequencing and Scheduling Problem (ASSP) has been well studied in the current literature with different configurations and model objectives leading to variants of the ASSP model. The ASSP model includes the Aircraft Landing Problem [3, 4], Aircraft Take-off Problem [5, 6] and mixed-mode aircraft sequencing operation [7, 8]. Depending on the airport design configuration, the runway system can be heterogeneous or interdependent. As for heterogeneous runways systems, the flight can have a significant difference in the estimated time of arrival (ETA) when runways are located in different position [9]. An interdependent runway system refers to the aircraft scheduling operation using a pair of adjacent runways. The approaching flight generates vortices and may affect the nearby flights from other runways and trailing flights from the same runways [10]. The literature adopts a static approach without consideration of speed control and holding pattern.

In civil aviation, ATC is required to manage the air traffic and safety issues in airborne and airport traffic in the Terminal Manoeuvring Area (TMA). ATC need to determine the scheduled time of landing by priorities and the possible arrival time landing by priority and possible arrival time. Under free flow situation, the speed restriction can be cancelled by the ATC (e.g. flight CX710, no speed restrictions / resume normal speed). Flight speed needs to be maintained when there is a high volume of airborne traffic. ATC restricts the speed of the flight within the TMA to keep a safe distance between flights via speed command. Besides, ATC has the authority to command a particular flight speed during high traffic situations in near terminal area (e.g. Flight BER456, reduce speed to 210 knots.). Moreover, the engine type, flight weight and vertical altitude at standard pressure are also the factors affecting the possible upper bound and lower bound speeds. Therefore, the speed profile of each flight is a class-dependent and flight-level-dependent set.

In this research, we aim to reduce the ATC workload and maintain a high level of resilience for the runway system by the adopting the computational intelligence to obtain an aircraft scheduling solution considering discrete airborne delay and holding patterns. The complexity of the ASSP model is a Non-deterministic Polynomial hard (NP-hard) problem [11]. Therefore, swarm intelligence is proposed to reduce computational effort while ensuring the quality of the solution with close-to-optimal performance.

2 An aircraft scheduling considering discrete airborne delays and holding pattern

As mentioned in the previous section, aircraft scheduling considering discrete airborne delays and holding pattern is considered in the model. The runway is configured as an aircraft landing problem, in which the runways are solely for approaching flights only. The objective is to minimise the total tardiness (airborne delays and delay time caused by holding) of all flights. In order to avoid unnecessary workload and confusion in voice-communication-based command between ATC and pilots, the design of the system is intended to enhance the resilience of the aircraft schedule via speed control and the number of aeronautical holdings in the TMA, as shown in Figure 1.

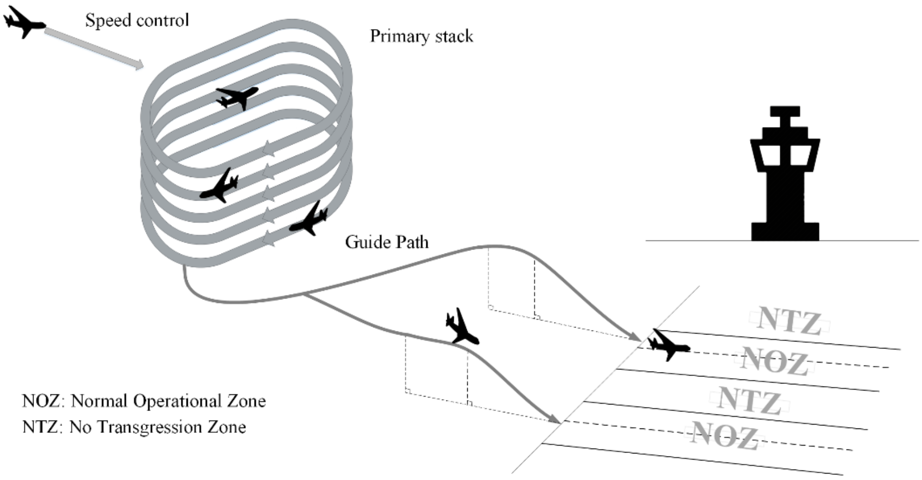

Fig. 1. The aircraft scheduling problem taking into account of speed control and the number of holding on fixed stack

2.1 Problem formulation

Table 1 shows the notation and decision variables in the model. A flight is denoted as $i, i = (1,2, \dots, n)$ and the total number of flights is n. A multiple runway system is considered in the model. Each runway is denoted as $r, r = (1,2, \dots, m)$, where m is the maximum number of runways in the airport. The decision variable x_{ir} determines the runway assignment of each flight i, while y_{ijr} denotes the sequence of flight i and j (not necessarily immediately) on the same runway r.

Table 1. Notation and decision variables

Notations	Explanation
n	The number of aircraft
m	The number of runways
ETA_i	The estimated landing time of aircraft i
STA_i	The scheduled landing time of aircraft i
l_i	The latest landing time of aircraft i
S_{ij}	The separation time between aircraft i and j scheduled on the same runway, $S_{ij} \geq 0$
h	The completion time of an oval course flown on aeronautical holding stack
K	The maximum number of aeronautical holdings $K = \max(t_i)$
D_i	A set of discrete airborne delay, $D_i = \{d_i^1, d_i^2, \dots, d_i^{Q_i}\}$
Q_i	The number of elements of the set D_i
M	Large number associated with the artificial variable
Decision variables	**Explanation**

t_i	The cumulative number of completing aeronautical holding(s) of aircraft i, $(i = 1,2,\ldots,n)$
d_i	The airborne delay time of aircraft i, $(i = 1,2,\ldots,n)$
x_{ir}	1, if aircraft i is assigned to runway r, $(r = 1,2,\ldots,m)$ 0, otherwise
y_{ijr}	1, if aircraft i is scheduled to land before aircraft j on runway r 0, otherwise

The safety regulation enforced is that any pair of consecutive landings flights on the same runway must be separated with a time buffer – separation time S_{ij}, where flight i is the leading flight and flight j is the trailing flight. S_{ij} is a flight class-dependent value. The detailed separation requirement is shown as Table 2.

Table 2. Separation time (in seconds) between two consecutive flights in aircraft landing problem

Separation time (sec)		Trailing flight (Arrivals)		
		SSF	MSF	LSF
Leading flight (Arrivals)	SSF	82	69	60
	MSF	131	69	70
	LSF	196	157	96

SSF = Small size flight; MSF = Medium size flight; LSF = Large size flight

The estimated time of arrival ETA_i of flight i is a predetermined/roughly calculated arrival time found from the distance between the departure airport and destination airport. However, the formulation of ETA_i does not consider the air traffic pattern, queue length in a specific time interval and runway capacity of the destination airport. The final approaching time STA_i is usually assigned by ATC when the flight is ready to enter the TMA. The assigned airborne delay time d_i is from a set of discrete value $D_i = \{d_i^1, d_i^2, \ldots, d_i^{Q_i}\}$, where Q_i is the maximum element in the set. Besides controlling the speed of approaching flights, ATC arranges flights on the queue for landing by utilizing aeronautical holding in TMA when the airspace in TMA is congested. A completed time for an oval course flown in aeronautical holding is defined as h, and the cumulative number of completing holdings is denoted as t_i, where t_i is less than the maximum number of aeronautical holdings K. Therefore, the total holding time of flight i is calculated by $t_i \times h$.

The model minimises the total delay from the estimated time of arrival in absolute value for the worst case directly. The completed mathematical formation for aircraft scheduling considering the airborne and holding pattern is shown as below:

$$\min f = \sum_{i=1}^{n}(|d_i| + ht_i) \tag{1}$$

s.t.

$$STA_i = ETA_i + d_i + ht_i, \forall i \tag{2}$$

$$STA_i \leq l_i, \forall i \tag{3}$$

$$STA_j - STA_i \geq S_{ij} - M(1 - y_{jir}), \forall i, j, i \neq j \tag{4}$$

$$y_{ijr} + y_{jir} \leq 1, \forall i, j, r, i \neq j \tag{5}$$

$$\textstyle\sum_{r=1}^{m} x_{ir} = 1, \forall i \tag{6}$$

$$x_{ir}, y_{ijr} \in \{0, 1\}, \forall i, r \tag{7}$$

$$d_i \in D_i = \{d_i^1, d_i^2, \dots, d_i^{Q_i}\} \tag{8}$$

$$t_i \in \mathbb{Z}, 0 = \min(t_i) \leq t_i \leq \max(t_i) \tag{9}$$

The objective function (1) is used to minimise the airborne delay in absolute values and the total holding time of all flights. Constraint (2) computes the scheduled time of arrival STA_i of flight i by the sum of estimated time ETA_i of arrival of flight i, airborne time $d_i$via speed control and the total holding time ht_i.Constraint (3) limits the scheduled time of arrival STA_i before the latest time of arrival l_i. Constraint (4) guarantees that the scheduled time of arrival STA_j of flight j can only approach when the landing procedure of flight i is completed with separation time S_{ij}. Constraint (5) ensures that the landing sequence on the same runway r, either flight i before flight j or flight j before flight i. Each flight i is restricted to being assigned on only one runway r by constraint (6). Constraint (7) confirms that decision variables x_{ir}, y_{ijr} are binary numbers. Considering the travel time by speed up or slowdown is a discrete operation in aviation management, and the controlled airborne delay d_i is a discrete value from a set of speed profiles D_i of flight i by constraint (8). Constraint (9) denotes the minimum and maximum rounds of aeronautical holding in integer values.

3 Resolution Procedure for Aircraft Scheduling

3.1 Proposed MIP-based Artificial Bee Colony Algorithm

The Artificial Bee Colony (ABC) algorithm is considered as a Swarm Intelligence based (SI-based) algorithm in optimisation problem [12]. The major advantage of using the ABC algorithm in optimisation problem is that the design of the algorithm focuses on balancing the exploration and exploitation during searching. Exploitation refers to the ability of searching from a known solution, while exploration refers to the ability of escaping from local optimal. Three major features foster the optimisation process effectively and efficiently. These include: decentralisation, self-organizing and collective behaviour [13]. The notation and the process flowchart of ABC algorithm are shown in Table 3 and Figure 2 correspondingly. The ABC algorithm constructs the aircraft scheduling and sequencing solution, while the mixed integer programming is involved in obtaining the discrete airborne delay and number of aeronautical holdings.

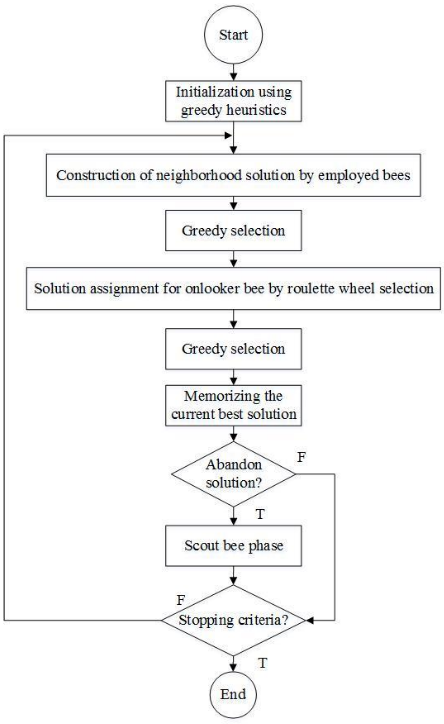

Fig. 2. The process flowchart of proposed artificial bee colony algorithm

Table 3. Notation of Proposed Artificial Bee Colony Algorithm

Notations	Explanation
CS	The size of bee colony
SN	The number of candidate solutions
D	The dimension of an independent solution
$c_{i,} i = 1,2,\ldots,SN$	The position of each solution in bee colony
$\text{fun}(c_i)$	The objective value of solution c_i
$\text{fit}(c_i)$	The fitness value of solution c_i
$Prob(c_i)$	The probability of an individual solution c_i among the entire colony in term of fitness value
$\overline{c_i}$	The neighbour solution of an individual solution c_i
$\text{trial}(c_i)$	The accumulated trial value of an individual solution c_i, which cannot be enhanced the quality of solution in terms of its objective value
$limit$	The maximum tolerance of $\text{trial}(c_i)$

p	Random number, $0 \leq p < 1$

3.2 Constructive heuristic in the Initialization Phase

Compared with randomised initialization, constructive heuristic provides a fairly good starting point for optimal searching from a promising solution region. Given a satisfactory initial solution with high quality, the algorithm is able to reduce the convergence time and computational burden. A simple constructive heuristic is proposed.

The initial solution is constructed from a sequential order of a set of ETA_i. The objective function is to minimize the airborne delay and holding delay. The construction from a First-come-first-serve sequence provides a good initial solution for further exploitation. In order to maintain diversity of the solution sets by constructive heuristic, a random runway assignment is considered from a sorted FCFS sequence in ascending order of ETA_i. The flight with the earlier time in ETA_i will be assigned an earlier position with a random runway assignment, and vice versa.

3.3 Employed Bee Phase

In each iteration, an employed bee performs neighbourhood search operators to generate a neighbourhood solution $\overline{c_i}$ from a known solution $c_i = 1,2,\dots,SN$. The greedy method is applied to obtain better solution quality by comparing the objective value of the known solution $fun(c_i)$ and neighborhood solution $\overline{fun(c_i)}$. Two operators are considered in this phase: The swap operator and the insert operator. The swap operator randomly selects two flights from different runways and performs swapping of the position of the two elements. The insert operator aims to reassign the randomly selected flight to another runway at a particular position. Any unsuccessful update in the candidate solution will be cumulated by the parameter $trial(c_i)$.

3.4 Onlooker Bee Phase

The onlook bee further enhances the solution quality by the neighbourhood search operators. The selection criterion of candidate follows the fitness probability distribution using equation (10). The high value in fitness approximation $\mathrm{fit}(c_i)$ denotes a high solution quality in terms of the objective value across the population. The selective probability of each solution is calculated by equation (11).

$$\mathrm{fit}(c_i) = {}^{1}\!/_{1 + fun(c_i)}, \forall \mathrm{i} \tag{10}$$

$$\mathrm{Prob}(c_i) = {}^{fit(c_i)}\!\Big/_{\sum_{i=1}^{SN} fit(c_i)}, \forall \mathrm{i} \tag{11}$$

3.5 Scout Bee Phase

Excessive exploitation may result in a local optimal trap. In the ABC algorithm, the scout bee will evaluate the possibility of being a trap in a local optimal by considering the parameter $trial$. A higher value in $trial$ implies a higher probability of being trapped in a local optimal. The scout bee will initialize the solution candidate when the $trial(c_i)$ is excess the maximum tolerance of unsuccessful update $limit$.

4 Computational Experiment

4.1 Instance Description

The test instance is randomly generated with the number of flights n is 20, and the number of runways r is 2. The $ETA_i, \forall i$ is randomly assigned from a uniform interval of $[240{,}600]$ to represent a high traffic situation. The detailed description of the test instance is shown as Table 4.

Table 4. Instance description of aircraft scheduling model ($n = 20, m = 2$)

Flight ID	ETA	Flight size	Speed profile	Flight ID	ETA	Flight size	Speed profile
0	491	Small	D_α	10	589	Medium	D_α
1	375	Small	D_β	11	449	Medium	D_β
2	371	Medium	D_α	12	506	Medium	D_β
3	388	Medium	D_γ	13	265	Medium	D_β
4	534	Medium	D_α	14	474	Medium	D_β
5	371	Medium	D_α	15	454	Medium	D_β
6	327	Medium	D_β	16	436	Medium	D_β
7	291	Medium	D_α	17	366	Large	D_α
8	424	Medium	D_β	18	371	Large	D_β
9	491	Medium	D_γ	19	499	Large	D_α

$D_\alpha = \{-180, -120, -90, 0, 90, 120, 180\}$; $D_\beta = \{-120, -90, 0, 90, 120\}$; $D_\gamma = \{-90, 0, 90\}$;

4.2 Effectiveness of the proposed algorithm

To measure the effectiveness of the proposed ABC algorithm, an exact method by *IBM ILOG CPLEX Optimization Studio 12.6.3* and the original ABC algorithm are also applied as a baseline for comparison. The optimal solution by the exact method is shown in Figure 3. The configuration of the computational environment is Intel Core i7 3.60 GHz CPU and 16 GB RAM under Window 7 Enterprise 64-bit operating system. The algorithms are written in C# language with visual studio 2015. In our preliminary study, the parameters of the proposed ABC algorithm are set as follows: $\mathrm{CS} = 40, \mathrm{SN} = {}^{CS}\!/_{2}, limit = SN \times m \times n$. The computational time of exact method by

IBM ILOG CPLEX is limited to an hour, while the ABC algorithm was given a maximum computational time of 300 seconds for resolution. Each algorithm is repeated 10 times to obtain the average performance. Table 5 indicates that proposed ABC algorithm obtain a fairly good approximation solution compared with the original ABC algorithm.

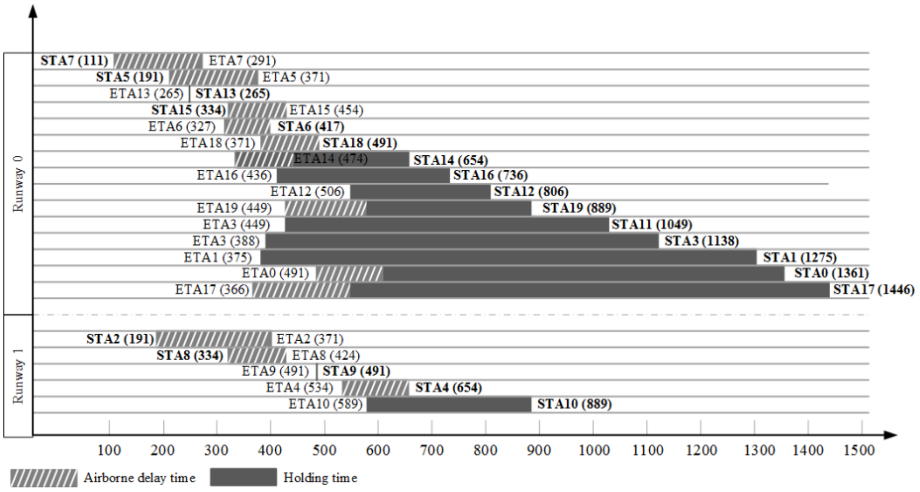

Fig. 3. Gantt charts for the solution obtained by exact method with one-hour computational limit

Table 5. Computational performance by exact method and ABC algorithm

CPU = 3600 seconds	CPU = 300 seconds					
MIP w/ CPLEX	ABC algorithm			Proposed ABC algorithm		
Optimal	Avg obj	Best obj	Gap (Opt.)	Avg obj	Best obj	Gap (Opt.)
6990	9546	8430	36.57%	7785	7500	11.37%

Avg obj = Average objective value; Best obj = Best objective value; Gap (Opt.) = Deviation between average objective value and optimal by *ILOG CPLEX*

5 Conclusion

Due to the increase of air traffic, the workload is increasingly affecting the resilience of ATC. During heavy air traffic situations and in a dynamic environment, the landing schedule may be adjusted from time-by-time. Pilots may be confused when ATC provides an excessive voice-communication-control on flight speed and landing time by rescheduling. To reduce the workload of ATC, a swarm intelligence algorithm is proposed to solve the aircraft scheduling problem considering discrete airborne delays and holding patterns for daily operation.

References

[1] A. Ghoniem, H. D. Sherali, and H. Baik, "Enhanced models for a mixed arrival-departure aircraft sequencing problem," *INFORMS Journal on Computing,* vol. 26, pp. 514-530, 2014.

[2] H. Balakrishnan and B. G. Chandran, "Algorithms for scheduling runway operations under constrained position shifting," *Operations Research,* vol. 58, pp. 1650-1665, 2010.

[3] J. Beasley, J. Sonander, and P. Havelock, "Scheduling aircraft landings at London Heathrow using a population heuristic," *Journal of the operational Research Society,* pp. 483-493, 2001.

[4] G. Bencheikh, J. Boukachour, A. E. H. Alaoui, and F. Khoukhi, "Hybrid method for aircraft landing scheduling based on a job shop formulation," *International Journal of Computer Science and Network Security,* vol. 9, pp. 78-88, 2009.

[5] G. Hancerliogullari, G. Rabadi, A. H. Al-Salem, and M. Kharbeche, "Greedy algorithms and metaheuristics for a multiple runway combined arrival-departure aircraft sequencing problem," *Journal of Air Transport Management,* vol. 32, pp. 39-48, 2013.

[6] J. A. Atkin, E. K. Burke, J. S. Greenwood, and D. Reeson, "On-line decision support for take-off runway scheduling with uncertain taxi times at London Heathrow airport," *Journal of Scheduling,* vol. 11, pp. 323-346, 2008.

[7] J. A. Bennell, M. Mesgarpour, and C. N. Potts, "Airport runway scheduling," *4OR,* vol. 9, pp. 115-138, 2011.

[8] A. Lieder and R. Stolletz, "Scheduling aircraft take-offs and landings on interdependent and heterogeneous runways," *Transportation Research Part E: Logistics and Transportation Review,* vol. 88, pp. 167-188, 2016.

[9] Y.-H. Liu, "A genetic local search algorithm with a threshold accepting mechanism for solving the runway dependent aircraft landing problem," *Optimization Letters,* vol. 5, pp. 229-245, 2011.

[10] J. E. Beasley, M. Krishnamoorthy, Y. M. Sharaiha, and D. Abramson, "Scheduling aircraft landings—the static case," *Transportation science,* vol. 34, pp. 180-197, 2000.

[11] L. Bianco, P. Dell'Olmo, and S. Giordani, "Scheduling models and algorithms for TMA traffic management," in *Modelling and simulation in air traffic management,* ed: Springer, 1997, pp. 139-167.

[12] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical report-tr06, Erciyes university, engineering faculty, computer engineering department2005.

[13] S. Zhang, C. Lee, K. Choy, W. Ho, and W. Ip, "Design and development of a hybrid artificial bee colony algorithm for the environmental vehicle routing problem," *Transportation Research Part D: Transport and Environment,* vol. 31, pp. 85-99, 2014.