A Modified Variable Neighborhood Search for Aircraft Landing Problem

K.K.H. NG[1], C.K.M.LEE[1]
[1]Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong, China
kkh.ng@connect.polyu.hk, ckm.lee@polyu.edu.hk

Abstract – **Aircraft landing problem is a crucial operation in air traffic flow management. Appropriate and efficient landing sequencing, runway assignment and scheduled landing time are of great importance to maintain flight safety and minimize the overall tardiness of all flights. The heuristic approaches mainly focus on providing a fast and feasible solution in a reasonable amount of time. The proposed modified variable neighbor search heuristic shows its robustness in searching the optimal result. The computational study for analyzing the algorithm indicates the effectiveness in handling time sensitive aircraft landing model. The performance of the proposed algorithm is analyzed in comparison with the result from the literature and it is capable of accommodating the perturbation strengthen iteratively to escape from the local optimum trap and reduce the computational burden in branch-and-bound algorithm using CPLEX optimizer.**

Keywords – **Aircraft landing problem, airside operation, meta-heuristics, variable neighborhood search**

I. INTRODUCTION

Aircraft Sequencing and Scheduling Problem (ASSP) is one of the most important aspects of air transport management, which considers the traffic of air transportation, and the landing and takeoff sequences. It consists of two major operations: Aircraft Landing Problem (ALP) and Aircraft Takeoff Problem (ATP). The major objective of ASSP is to ensure on-time performance or slight effect on airport tardiness and interruption of air ground operations, such as ground handling service, flight gate assignment, aircraft maintenance routing, etc. The conventional ASSP model applied the First-Come-First-Serve (FCFS) principle to arrange the aircraft landing or takeoff sequence based on the order appeared on the radar system. Aircraft generates wake vortices during landing that causes adverse effect and turbulent conditions on the neighboring landing aircraft. A considerable separation time measured by the size of two succeeding flights is considered in order to reduce the possibility of passenger discomfort and possible damages to the subsequent aircraft.

The objective of ALP is to minimize the total tardiness of all flights from the target landing time. This problem considers the flight sequencing, runway assignment and decision of landing time with practical constraints, such as time window, runway and airport facility capacity and other interrelated airside operations. In computational complexity theory, ALP is known as the NP-hard combinatorial problem, which requires a significant amount of computational time to acquire an optimal level when the number of nodes increases [1]. The complexity of ALP model has been widely studied by different proposed meta-heuristic algorithms with intricate constraints. The predominant focus on solving ALP model with the meta-heuristics is to construct flight sequencing and runway assignment in the first phase. The reduction of problem size in the first phase significantly alleviates the computation effort in calculating the landing time with exact methods, including Linear Programming (LP) and Branch-and-Bound (B&B) algorithms. ALP model is not solely an expert system in resolving real life aircraft arrival sequencing model, but is incorporated with other decision support system to maintain high flexibility and service level, and safety requirement in traffic flow management. Meta-heuristics performs as a powerful tool to explore searching progress in the feasible region, and let the landing time decision become the continuous decision variables that could be solved by the exact method in a speedy way. Although the meta-heuristics does not constantly reach to optimum sometimes, but is able to handle large size problems and provide fairly good solutions in a reasonable amount of time.

Exact and heuristic methods have been developed for the ALP model. Sakehipur, Modaares & Naeni introduced a hybrid meta-heuristic method by combining Simulated Annealing (SA) with Variable Neighborhood Search (VNS) and Variable Neighborhood Descent (VND). Both of the algorithms outperform a commercial mathematical optimizer *Cplex* [1]. Vadlamani & Hosseini have further evaluated the performance of the SA+VNS algorithm and proposed an Adaptive Large Neighborhood Search (ALNS) algorithm to resolve the single runway ALP problem [2]. Hancerligogullari et al. presented a randomized priority search (meta-RaPS) to improve the initial solution construction [3]. Sabar & Kendall suggested multiple perturbation operators for Iterated Local Search (ILS) that efficiently enhance the solution quality and reduce the computational time [4]. In the extant literature, few studies have implemented a dynamic perturbation operator to expedite the searching and escape from local optimum trap, except Sabar & Kendalls' work. In their proposed algorithm, the number of iterations for applying the perturbation operator is determined by perturbation strength and current number of iterations. The control of diversification of current solution by mixed perturbation operators is simply designed by the number of unsuccessful variables in neighborhood search. As a result, less computational effort is involved to resolve the ALP model. According to the actual operation, the decision of landing sequence must be made within 1000 seconds. However, the model complexity limits the optimality in small size instances. Meta-heuristics are proven to be effective and efficient in yielding near-optimal solutions with less computational effort in other combinatorial problem. This research attempts to expedite the related meta-heuristic research in airport landing problem. The proposed modified variable neighborhood search algorithm is developed for ALP model to further enhance the solution's quality and convergence speed.

II. PROBLEM FORMULATION

ALP model is recommended to decide the optimal aircraft landing time and runway assignment with the objective of minimizing the total tardiness time associated with penalty cost. Furthermore, the separation constraints act as an essential factor to maintain flight and passenger safety. While meeting the safety regulations, the actual landing time may deviate from the target landing time. It is complex to handle the scheduling manually by air traffic control planners. ALP model works as an expert system to determine and suggest a possible solution considering the number of approaching flights and the runway traffic.

TABLE I
Notation and Decision Variable

Notations	Explanation
n	The number of aircrafts
m	The number of runways
E_i	The earliest landing time of aircraft i
T_i	The target landing time of aircraft i
L_i	The latest landing time of aircraft i
C_i^E	The incurred penalty cost per time unit associated with early arrival of aircraft i
C_i^L	The incurred penalty cost per time unit associated with late arrival of aircraft i
S_{ij}	The separation time between aircraft i and j scheduled on the same runway, $S_{ij} \geq 0$
s_{ij}	The separation time between aircraft i and j scheduled on the different runway, $s_{ij} \geq 0$
Decision variables	**Explanation**
x_i	The scheduled landing time of aircraft i, $(i = 1,2,\dots,n)$
y_{ir}	1, if aircraft i is assigned to runway r, $(r = 1,2,\dots,m)$ 0, otherwise
y_{ij}	1, if aircraft i is scheduled to land before aircraft j 0, otherwise
δ_{ij}	1, if aircraft i and j are scheduled to land on the same runway r 0, otherwise
a_i	The accumulated penalty cost of late arrival after the target time. $a_i = \max(0, x_i - T_i)$
b_i	The accumulated penalty cost of late arrival after the target time. $b_i = \max(0, T_i - x_i)$

The ALP model can be modeled as a job shop scheduling in matching flight landing, sequencing and runway assignment. The settings and notations of ALP are described in Table I. The number of aircraft is known beforehand in the static ALP model, with the predefined earliest, target and latest landing time. The value of separation time S_{ij} is uniquely defined for a pair of aircraft, rather than type dependent value, on the same runway. The separation time of two consecutive flights on different runways is defined as s_{ij}. The partially linear penalty cost is expressed as a linear function of early landing and late arrival.

$$\min f = \sum_{i=1}^{n}(a_i C_i^L + b_i C_i^E) \tag{1}$$

s.t.

$$E_i \leq x_i \leq L_i, \forall i \tag{2}$$

$$x_j - x_i \geq S_{ij}\delta_{ij} + s_{ij}(1 - \delta_{ij}) - M y_{ji}, \forall i,j, i \neq j \tag{3}$$

$$y_{ij} + y_{ji} = 1, \forall i,j, i \neq j \tag{4}$$

$$\delta_{ij} \geq y_{ir} + y_{jr} - 1, \forall i,j,r, i \neq j \tag{5}$$

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

$$y_{ir}, y_{ij}, \delta_{ij} \in \{0,1\}, \forall i, r \tag{7}$$

$$x_i, a_i, b_i \geq 0, \forall i \tag{8}$$

Objective function (1) is designed to minimize the overall penalty cost associated with variation between actual landing time and target landing time. Flights usually fly at its cruise speed, which indicate the target landing time. Fuel consumption will be increased when a flight flies with the fastest speed requested by air traffic controller. When all runways reach the capacity limit, nearby aircraft are directed to hold on the air until the release of available runway. Extra fuel consumption of aircraft being on hold in the airspace becomes the additional cost for avoiding flight conflict and airport congestion.

Constraint (2) guarantees that flight i must land within the time window. The earliest landing time indicates the maximum speed needed to direct the landing position, while the late landing time points out the last-minute landing before the situation of fuel exhaustion. Constraint (3) calculates the minimum separation time of two succeeding flights. Separation time S_{ij} occurs when two flights are approaching to the same runway. Otherwise, the buffer time on other runway usage is limited to the separation time s_{ij}. Constraint (4) confirms that either flights i or j must land first in the flight sequencing model. Constraint (5) shows that δ_{ij} equals to 1 when flights i and j are planned to land on the same runway. Otherwise, δ_{ij} takes 0. Constraint (6) demonstrates that each flight is assigned to only one runway. $y_{ir}, y_{ij}, \delta_{ij}$ are binary values in the ALP system in constraint (7). The scheduled landing time x_i and the ahead of landing time before and overdue time after target landing time a_i, b_i must be larger than or equal to 0 and it is shown in constraint (8).

III. ITERATIVE VARIABLE NEIGHBORHOOD SEARCH HEURISTIC

The proposed method, named modified variables neighborhood search (MVNS), has a better exploitation ability in obtaining nearly optimal or even reach to optimal value in a reasonable amount of time. The conventional VNS algorithm applies a random operator to select one variable to evaluate all possible neighborhood structures in order to find a better solution. Although VNS algorithm shows its significant contribution in searching neighborhood structure to escape from local optimum, local optimum trap may happen by using single swap of two variables. The iterative searching process may fall into the same local optimum trap due to the greedy nature of the searching neighborhood structure in VNS, when there are no random operators applied in the initialization phase.

Table II demonstrates the overall MVNS algorithm architecture with a process flowchart in Figure *1*. Extra parameters are introduced, for example, the maximum tolerance operator $limit$, which denotes the maximum resistance of local searching, and $trial$, which measures the number of unsuccessful neighborhood structure searching to indicate the possibility of being trap in local optimum. It is supposed that the solution will be improved iteratively, however, certain neighborhood structure has been reviewed several times without further improvement. In order to decrease the computational burden, the algorithm will terminate the neighborhood structure searching when the $trial$ run reaches the parameter $limit$. It significantly imposes a dynamic parameter to escape from local optimum trap and restarts from another possible solution. The stopping criterion herein is that either the number of iteration $iter$ reaches the maximum iteration $MaxIter$ or the best known solution is found. The parameter $limit$ is simply defined as the maximum number of iterations by preliminary test.

TABLE II
A modified variable neighborhood search algorithm

Algorithm Architecture
Introduce maximum tolerance operator $limit$ as the maximum number of iteration $MaxIter$
$limit = MaxIter$
Define the stopping criteria as $iter$ reaching the $MaxIter$ or best known value BKV is found
Set $iter = 0$
Set $trial = 0$
Call function: Initialization Phase
Do
Call function: Modified Neighborhood Search Phase
Call Function: Perturbation Phase
Until stopping criterion is met, $iter > MaxIter \;\|\|\; f\ (best) \leq BKV$

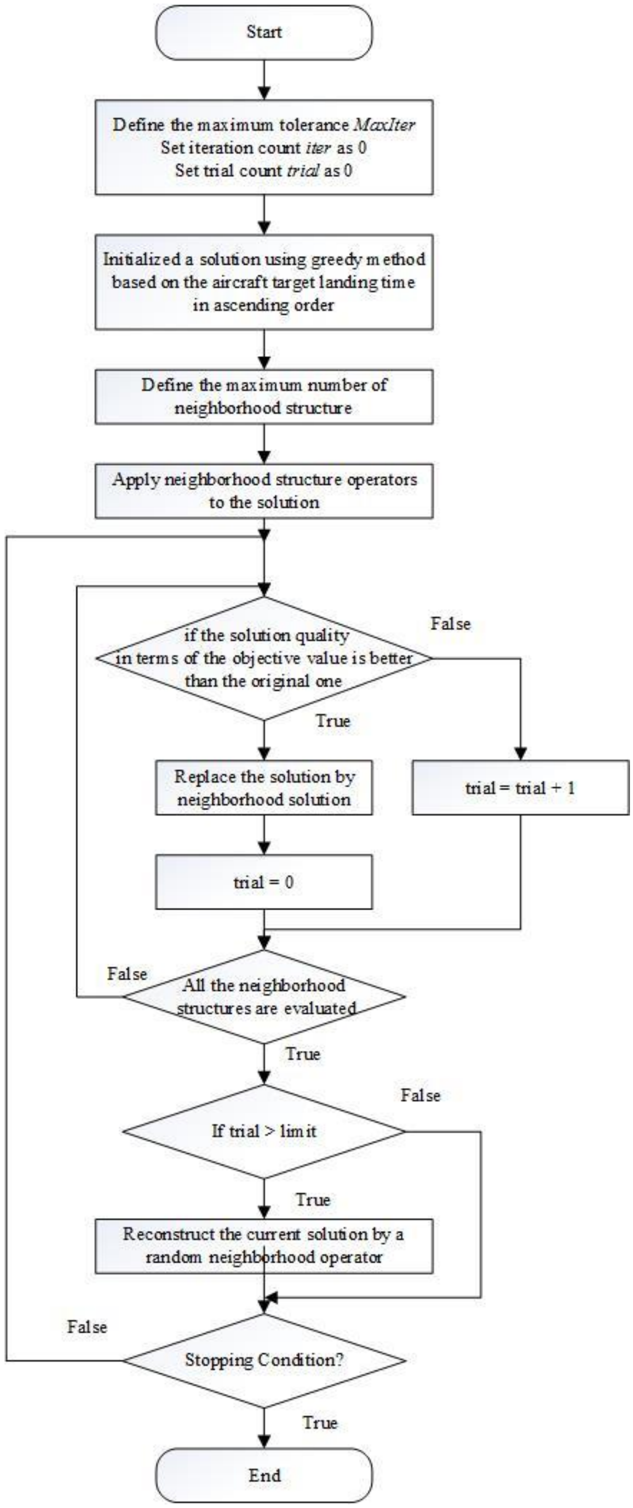

Figure 1. Process flowchart of modified variable neighborhood search

The initial solution is initialized by the sorting of aircraft target landing time sequences in ascending order as shown in Table III. The objective is to minimize the time deviation between scheduled and target landing time.

Following the suggested target landing time to design the initial solution is possible to enhance the robustness of searching optimal solution and limit the search in favorable and feasible region in solution space, noted that the possibility of being trap in local optimum. The solution to resolve this problem is presented in perturbation phase. After the construction of aircraft sequencing, the runway assignment is based on the greedy method to allocate all flights to the corresponding runways. The first aircraft is set to land on the first runway by default. The runway assignment is measured by the target time difference between two successive flights and separation time of flight i and j. If the target landing time of flight i and separation time S_{ij} is greater than the target landing time of flight j, then flight j is assigned to land on the next runway. Otherwise, both flight i and j are allocated to the same runway. This greedy approach gives a feasible solution at the beginning.

TABLE III
Construction heuristics for aircraft landing problem

Initialization Phase

Sort the aircraft landing sequencing in ascending order based on the target landing time T_i in permutation array $permu$
$permu = \{i_1, i_2, \ldots, i_{n-1}, i_n\}$, where $T_{i_1}, T_{i_2}, \ldots, T_{i_{n-1}}, T_{i_n}$
 	
Let $r = 1$
Construct the runway assignment using greedy method in runway permutation array $permu\ runway$
Assign aircraft i_1 in runway r
For $k < n$
    Evaluate the deviation of target landing time between two consecutive flights i annd j
    If $T_{i_k} + S_{i_k i_{k+1}} \geq T_{i_{k+1}}$
    Then
        Assign flight k into the another runway
        $r := r + 1$
        $permu\ runway_k = r$
    Else
        Assign the flight $k + 1$ after the position of flight k into the same runway
        $permu\ runway_k = r$
End for
Generate an initial solution $Sol_{initial}$ and pass into the current solution
$Sol_{ori} \leftarrow Sol_{initial}$

The local search process will be taken in modified neighborhood search phase as shown in Table IV. The neighborhood solution is generated iteratively by each neighborhood structure operator NS. The number of iterations of each NS is determined by the number of possible neighborhood structure. A random variable will be selected and local search is applied until all possible solution will then be updated by the best neighborhood structure and the control parameter $trial$ I is set to be 0. If no improvement is found, the $trial$ is increased by one to record the number of neighborhood search failure. Four neighborhood searches are considered in our proposed algorithm.

NS_{Swap1}: This operator randomly selects a flight x_i and measures all feasible swap of the two elements on the **same** runway to evaluate the improvement by the objective function and then replace the current solution by the improved neighborhood solution.

NS_{Swap2}: This operator randomly selects a flight x_i and measures all feasible swap of the two elements on **different** runways to evaluate the improvement by the objective function. The current solution is then replaced by the improved neighborhood solution.

$NS_{Insert1}$: This operator randomly selects a flight x_i and measures all feasible insert position on the same runway to evaluate the improvement by the objective function and then replace the current solution by the improved neighborhood solution.

$NS_{Insert2}$: This operator randomly selects a flight x_i and measures all feasible insert position on the different runways to evaluate the improvement by the objective function. The current solution is replaced by the improved neighborhood solution.

TABLE IV
Modified neighborhood search phase

Modified Neighborhood Search Phase

Do
Define the maximum number of neighborhood structure as K
Initial the iteration count
Apply a neighborhood structure operator, NS
    For $k < K$
        Generate a neighborhood solution S_k from current solution S_o using the neighborhood structure operator, NS
        If $f(Sol_k) < f(Sol_o)$
        Then
            Replace the current solution by neighborhood solution
            $Sol_o \leftarrow Sol_k$
            Record the current best objective value
            $f(best) = f(Sol_k)$
            $i = 1$
            $trial = 0$
        Else
            $i = i - 1$
            $trial = trial + 1$
    End for
Until All neighborhood structure operators are applied

Perturbation phase is a novel design to escape from local optimum in the proposed algorithm. Steps are shown in Table V. At first, the initialized solution is evaluated by the VNS algorithm. The sequencing order and runway assignment are based on the target landing time and separation time of all succeeding landing flights. VNS algorithm is able to escape from local optimum, but the evaluation time of neighborhood structure is in slow progress. A randomized reconstruction heuristics is considered to tune the initialized feasible solution with a diversification strategy, which leads to a fast convergence to obtain the global optimum. The process of perturbation phase examines the neighborhood solution with four NS operators simultaneously to reach another feasible solution, but not necessarily a better solution than the previous one. The VNS algorithm then performs the neighborhood search phase that possibly reduces the likelihood in stepping revisited neighborhood structure and achieves diversified neighborhood searching process.

TABLE V
Randomized reconstruction heuristics for searching feasible solution

Perturbation Phase

IF $trial > limit$
Then
  Do
    Do
      Generate an initial solution $Sol_{initial}$ using construction heuristics
      $Sol_{ori} \leftarrow Sol_{initial}$
      $Sol_n \leftarrow$ re-construct the current solution Sol_{ori} by neighborhood search operators simultaneously
      $trial = 0$
    Until All neighborhood search operators are executed
  Until a feasible solution is found
$Sol_{ori} \leftarrow Sol_n$

IV. RESULT AND DISCUSSION

TABLE VI
Modified Variable neighborhood search algorithm

Instance	Number of aircrafts, n	Number of runway, m	Instance Node
Airland1	10	1	1
		2	2
		3	3
Airland2	15	1	4
		2	5
		3	6
Airland3	20	1	7
		2	8
		3	9
Airland4	20	1	10
		2	11
		3	12
		4	13
Airland5	20	1	14
		2	15
		3	16
		4	17

The proposed MVNS algorithm is evaluated with benchmark ALP instances from OR-library. The ALP instances were proposed by Beasley et al. [4]. Any readers who are interested in the instance detail are referred to the online source from the OR-library (http://people.brunel.ac.uk/~mastjjb/jeb/orlib/airlandinfo.html). Table VI describes the major characteristic of these instances. There are total 13 ALP instances from the website. Small size instances from airland1 to airland5 are selected to analyze the algorithm performance. The proposed MVNS algorithm was coded in *C++ language* with *visual studio 2013* and *IBM CPLEX Optimizer library* on a computer with Intel Core i7 3.60 GHz CPU and 16.0GB ram under *Window 7 Enterprise 64-bit* operating environment. The maximum number of iterations equals the multiplication of fixed integer number 100 and the number of aircraft ($MaxIter = 100n$). Each instance was run 40 times to summarize an average performance and compared with the results from other literatures.

In table VII, the best-known solution (BKV) and best computational time (CPU) are extracted from 40 runtimes, and compared with the best-known solution and solutions from the literature [1] [4]. The results demonstrate that the proposed MVNS algorithm has the ability to escape from the local optimum trap and further exploit the result from moving to neighborhood structure. Less computational effort is required to reach the optimum. In Table VIII, the average performance of the proposed algorithm are indicated. The results denoted that the proposed algorithm is capable to converge to the optimal solution within a minute in 40 runways from node 1 to 14.

TABLE VIII
The experimental result of ALP instances from OR library

Instance	Node	m	Max	Min	Avg	Avg Time
Airland1	1	1	700	700	700	0.13
	2	2	90	90	90	0.29
	3	3	0	0	0	0.24
Airland2	4	1	1480	1480	1480	0.48
	5	2	210	210	210	0.40
	6	3	0	0	0	2.84
Airland3	7	1	820	820	820	0.97
	8	2	60	60	60	6.23
	9	3	0	0	0	0.62
Airland4	10	1	2520	2520	2520	0.69
	11	2	640	640	640	21.57
	12	3	130	130	130	14.72
	13	4	0	0	0	15.81
Airland5	14	1	3100	3100	3100	0.97
	15	2	2250	650	844.5	36.74
	16	3	3070	170	437.25	214.07
	17	4	60	0	3.75	80.22

TABLE VII
The experimental result of ALP instances from OR library

Instance	Instance Node	CPLEX [a]		SA + VND [a]		SA + VNS [a]		ILS [b]		Proposed MVNS	
		BKV	CPU	BKV	CPU	BKV	CPU	BKV	CPU	BKV	CPU
Airland1	1	700	0.66	700	0	700	0	700	0	700	0.05
	2	90	0.12	90	0	90	0	90	0	90	0.08
	3	0	0.1	0	0	0	1.38	0	0	0	0.06

Airland2	4	1480	0.49	1480	1.59	1480	1.65	1480	0	1480	0.12
	5	210	0.24	210	1.66	210	1.91	210	0	210	0.19
	6	0	0.28	0	1.98	0	1.73	0	0	0	0.16
Airland3	7	820	0.39	820	1.78	820	4.22	820	0	820	0.21
	8	60	0.3	60	3.12	60	5.11	60	0.8	60	0.33
	9	0	0.36	0	3.29	0	2.85	0	0.1	0	0.24
Airland4	10	2520	5.12	2520	1.98	2520	3.94	2520	1.7	2520	0.21
	11	640	12.62	640	3.56	640	5.05	640	1.9	640	0.70
	12	130	0.84	130	3.74	130	7.15	130	2	130	0.32
	13	0	0.41	0	4.06	0	1.89	0	2.3	0	0.33
Airland5	14	3100	20.44	3100	1.85	3100	4.84	3100	1.3	3100	0.20
	15	650	15.03	650	3.04	650	4.92	650	2.4	650	0.31
	16	170	1.47	170	4.11	170	3.04	170	3.7	170	0.35
	17	0	0.33	0	4.35	0	2.14	0	3.1	0	0.30

[a] obtained from [1]

[b] obtained from [4]

V. CONCLUSION

In this research, a novel perturbation operator in variable neighborhood search is proposed. The algorithm initially starts from a randomized solution to ensure the diversity along the searching space. The adaptive control parameter $trial$ represents the number of unsuccessful updated by neighborhood structure operators. A randomized reconstruction of neighbor solution maintains the solution diversity to converge well solution in exploitation. The static ALP model does not truly reflect the actual operation, but provide insightful idea in aircraft scheduling. As for the future directions, dynamic ALP research on soft window target landing time would be a new research direction. Aircraft landing is not restricted in a point of time, but a period of time. The main concern in aircraft landing is to reliably schedule the landing and maintain a low turnaround time before the next departure, as flight travel time and service time has stochastic in nature. The problem considered herein can be extended by including the schedule of turnaround operation to conduct a more realistic representation of real life operation.

ACKNOWLEDGMENT

The research is supported by The Hong Kong Polytechnic University. The authors would like to thank the research committee and the Department of Industrial and Systems Engineering of the Hong Kong Polytechnic University for support in this project (RU8H).

REFERENCES

[1] A. Salehipour, M. Modaares and L. M. Naeni, "An efficient hybrid meta-heuristic for aircraft landing problem," *Computers & Operations Research,* vol. 40, no. 1, pp. 207-213, 2013.

[2] S. Vadlamani and S. Hosseini, "A novel heuristic approach for solving aircraft landing problem with single runway," *Journal of Air Transport Management,* vol. 40, pp. 144-148, 2014.

[3] G. Hancerliogullari, G. Rabadi, A. H. AI-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.

[4] N. R. Sabar and G. Kendall, "An iterated local search with multiple perturbation operators and time varying perturbation strength for the aircraft landing problem," *Omega,* vol. 56, pp. 88-98, 2015.

[5] J. E. Beasley, M. Krishnamoorthy, Y. M. Sharaiha and D. Abramson, "Scheduling aircraft landings—the static case," *Transportation science,* vol. 34, no. 2, pp. 180-197, 2000.