Int. J. Nonlinear Anal. Appl. 13 (2022) 2, 435–445
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2021.23255.2510

# A new improved gray wolf optimization algorithm to solve the aircraft landing problem at Mashhad Shahid Hasheminejad International Airport

Manizheh Teimoori[a], Houshang Taghizadeh[a,*], Morteza Honarmand Azimi[a], Jafar PourMahmoud[b]

[a]*Department of Management, Tabriz Branch, Islamic Azad University, Tabriz, Iran*

[b]*Department of Applied Mathematics, Azarbaijan Shahid Madani University, Tabriz, Iran*

*(Communicated by Ali Jabbari)*

**Abstract**

Air traffic management is a sensitive and stressful job with various daily problems and obstacles. The aircraft landing problem is one of the most important issues addressed currently in flight surveillance. This issue has several optimal local points. Gradient-based algorithms cannot produce an optimal solution in a reasonable time to solve this problem. Meta-heuristic algorithms are used to solve such problems. Since landing earlier or later than the scheduled time will lead to higher costs for each aircraft, this article aims at minimizing the time deviation from the originally scheduled landing time of each flight. The Gray Wolf Algorithm is a new meta-heuristic algorithm inspired by wolf behaviour. However, it has a problem with global and local searches. To solve this problem, the fitness of each wolf is assigned a weight and the new member is obtained using those weights. In addition, in order to increase the local and global search capability of the algorithm, if a condition with a probability of 0.3 is met, a random search is performed around the position of the best wolf. Otherwise, by setting a condition with a probability of 0.1, a global search is performed as the mutation operator around a selected wolf. This improves the algorithm's ability to search globally and locally. In order to evaluate this algorithm in solving the problem, its result is compared with the algorithms of particle swarm optimization, firefly and the common Gray Wolf. The results show a very high performance of this algorithm compared to other similar algorithms.

Keywords: Aircraft Landing Problem (ALP), Metaheuristic Algorithms, Improved Gray Wolf Algorithm
2020 MSC: 65K10

## 1 Introduction

Over the last few decades, aviation has grown exponentially. Therefore, the air traffic control and landing schedule of each aircraft have become very important. Airport authorities need to be able to strike a balance between airline satisfaction and passenger satisfaction in order to attract airlines, reduce costs, and increase revenue. The most important goal of the ALP (Aircraft Landing Problem) is to minimize the early and late arrival times based on the

*Corresponding author
*Email addresses:* Teimoori1397@gmail.com (Manizheh Teimoori), taghizadeh@iaut.ac.ir (Houshang Taghizadeh), third.author@email.address (Morteza Honarmand Azimi), fourth.author@email.address (Jafar PourMahmoud)

*Received:* February 2021 *Accepted:* April 2021

predetermined time in the flight schedule. Because the early arrival or late arrival of any aircraft will lead to an increase in costs of fuel and passenger dissatisfaction. In this problem, in addition to this goal, the order of landing of aircraft should be considered according to the size and weight and observing the minimum separation distance required to maintain flight safety. If these conditions are not met, there is a possibility of an accident for consecutive aircraft. Therefore, proper scheduling of the landing of each aircraft is crucial. Aircraft scheduling management is an NP-hard problem. Traditional gradient-based methods are not suitable for solving this problem due to falling in local optimizations. Therefore, there is a need to use methods based on random and intelligent methods. So far, various methods have been proposed in the research literature to solve this problem. In [25], a CPLEX method has been introduced to solve the aircraft landing problem with respect to minimizing early and late arrival times. In [24] a real-time scheduling method for landing aircraft based on cellular automation has been proposed. Salehipour et al. [17] used the simulated annealing method to solve the aircraft landing problem. They solved this problem for 100 aircraft. [28] provides an overview of computational techniques in the aviation landing problem in which the methods proposed in the research literature include problem modelling, evolutionary algorithms, intelligent algorithms and other methods for solving aircraft landing problem are presented. [8] reviews the research literature on the aircraft landing problem and discusses the advantages and disadvantages of each method. Zheng et al. proposed a hybrid simulated annealing and reduced variable neighbourhood (RVN) search to solve an aircraft scheduling and parking [27]. [7] considers a two-stage stochastic programming approach for aircraft landing problems was considered for airport runway scheduling under the uncertainty of arrival time on a single runway. The first stage determines the sequence of aircraft weight class. In the second stage, the delay time of all aircraft was considered. To solve this problem, a sample average approximation (SAA) algorithm is developed. In [10, 11, 12] a multi-objective genetic algorithm with non-dominated sorting was used to solve the problem of aircraft landing problem. In these articles, in addition to minimizing aircraft delays, other objectives were considered, including minimizing fuel costs and other costs due to early and late aircraft arrival. In [21], the NSGA-II multi-objective optimization algorithm was used for aircraft arrival and departure schedules on multiple runways. This article aims at minimizing the delay time of flights and idle time of the runways considering also the relevant constraints. In [14], an optimal data-splitting algorithm was used for aircraft scheduling on a single runway to maximize throughput. In [20], a novel heuristic approach called adaptive large neighbourhood search was introduced to solve solving aircraft landing problems with a single runway. Genetic algorithms were used to solve aircraft landing problems in [3, 19]. The Ant Colony algorithm was used to solve aircraft scheduling problems in [1, 23]. Lee et al. [13] solved the problem of aircraft sequencing and scheduling problem under the uncertainty of arrival and departure delays using a novel efficient artificial bee colony algorithm. In [26], Zhang et al. (2020) aim to solve the aircraft landing problem (ALP) considered a multi-objective optimization problem. They used an Imperialist Competitive Algorithm (ICA) to solve the ALP. In 2016, Benell et al. [2] presented the issue of dynamic scheduling of aircraft landings. In [5], the researchers propose a heuristic approach based on optimistic planning to solve the aircraft landing problem. model the ALP as an environment of states, actions, transitions and costs, then explore the resulting search tree so as to identify a near-optimal sequence of actions within a limited time budget. They investigate a baseline model based on linear regression, and two different machine learning (ML) models trained on a large number of optimized solutions. These models can quickly and accurately estimate the cheapest-sequence cost, which helps the search to identify a near-optimal branch more efficiently. The aircraft maintenance routing problem was presented for each operational aircraft using a reinforcement learning-based algorithm in [16]. Evaluation of this method showed that it could produce high-quality solutions in large and medium-scale aerial databases.

[22] proposes an optimization model for the integrated aircraft flight scheduling and routing problem, which allows a simultaneous determination of the departure time of each flight trip and assignment of a set of aircraft located at different airports to perform all flight trips. This model envisages that each flight trip is covered by its own particular aircraft type or a larger aeroplane. In addition, departure and arrival times of each flight trip are within a flexible time window in its aircraft's route and origin/destination airports, and the number of aeroplanes firstly distributed in the base airports is fully accounted for in the model. The model not only can effectively minimize weighted operation costs for the number of aeroplanes and the total idle time for adjacent flight trips covered by an aircraft, but also can maximize the number of transported passengers. This model has been able to obtain a feasible flight trip timetable. The model is applied to a case study to design the integrated aircraft flight scheduling and routing plan for a real airline in China.

In [4] dynamic feasible programming was proposed to optimize long-term aircraft maintenance programming. The validation of this model was done based on a European database.

[15] studies a branch-and-bound embedded genetic algorithm for resource-constrained project scheduling problems with a resource transfer time of aircraft moving assembly line. It aims to minimize the makespan of the project while respecting precedence relations and resource constraints. Several experimental tests revealed that the proposed

algorithm outperformed other existing algorithms in finding high-quality solutions. In [18] the maintenance task scheduling problem for an aircraft fleet was studied with a hybrid simulation-optimization approach. the maintenance tasks delegated to a shop should be scheduled in such a way that sufficient aircraft are available on time to meet the demand of planned missions. A robust formulation has been proposed so that the duration of the maintenance task is subject to unstructured uncertainty due to environmental and human factors. Experimental results confirmed the satisfactory performance of the proposed method in the face of uncertain scenarios.

Considering the aforementioned material, various methods have been applied in the research to solve the aircraft landing problem. Nevertheless, most of these articles have used small-scale data. In addition, no research has been done on the aircraft landing problem (ALP) at Mashhad International Airport. This article considers the aircraft landing problem (ALP) at this airport. This is a problem of deciding when to land each aircraft at the airport at a specific time so that each aircraft lands within a predetermined time window and the landing conditions of each aircraft are set. To maintain flight safety, each aircraft should have a minimum separation distance from its preceding aircraft. Based on these conditions and circumstances, the goal is to minimize the earliest possible time for landing and the latest possible time for landing. This goal can be considered as the absolute value of the difference between the predicted landing time and the predetermined landing time and then it can be minimized using appropriate optimization methods. This article considers the aircraft landing problem at Shahid Hasheminejad Airport in Mashhad. It should be noted that in this article, a static mode is considered. This means that the results are obtained offline for a specific period. Therefore, the goal is to optimize the landing schedule of each aircraft according to its size and weight so that there is no time interference for aircraft landing. Since this is a complex problem, and improved Gray Wolf intelligent optimization algorithm has been proposed to solve it.

## 2 Describing the Problem

From the moment the aircraft enters the airport radar range, the air traffic control tower must allocate a landing time to increase airport efficiency and minimize delays. In this article, it is assumed that the airport is a single runway. The allotted landing time is bounded by the earliest landing time and the latest landing time known as the time window. The earliest possible time for each aircraft to land is considered, based on the maximum speed at which each aircraft can fly at that speed, depending on its type. The latest possible time is determined by the amount of fuel that allows the flight to be in the flight phase.

The assigned landing time will change depending on the flight path, size and speed of each aircraft, so the landing time may be longer. To maintain flight safety, the time between the landing of a particular aircraft and the landing of each subsequent aircraft must be greater than a specified minimum separation, which depends on the size and weight of the aircraft involved. For example, an aircraft following a Boeing 747 (which is large in size and weight) requires a greater minimum separation distance compared with the time when an aircraft is following an MD aircraft (which is considered average in terms of aircraft size and weight). The reason for this is the wake turbulence that occurs behind an aircraft due to the aerodynamic shape of the aircraft wings and the cracking of the air layers. This turbulence leads to serious aerodynamic instability in a closely following aircraft. Therefore, this is the minimum separation time required between flights to create and maintain flight safety.

The landing time of each aircraft is within the limits of a time window depending on the characteristics and capabilities of the aircraft. When an aircraft can land directly, even after arriving at the airport, while flying at the highest speed allowed, it is the earliest landing time. The latest landing time (upper range) is when an aircraft can fly at the speed at the most effective fuel consumption level, and maintain that speed for as long as possible. Among these goals, minimizing landing delays is of paramount importance.

In the aircraft landing problem, in the airport air traffic control space, a series of items such as separate schedules, the latest possible landing time, the earliest possible landing time and the cost function are considered. To solve this problem, considering the safe landing conditions of the aircraft, a cost function is introduced in which the goal is to minimize the difference between the pre-determined time and the time predicted by the proposed algorithm. To minimize this cost function, an improved Gray Wolf optimization algorithm is used.

### 2.1 Decision Variables

This section introduces the decision variables in aircraft landing problem (ALP). Decision variables in [10] have been used to formulate the main equations of the problem.

$SLT_i$: The landing time for each aircraft $i$ predicted by trajectory synchronizer equipment after entering the aircraft into the radar range. This time is obtained by the proposed Gray Wolf algorithm.

$ELT_i$: The expected (or target) landing time of aircraft $i$, based on the assigned time slot which is normally specified in flight plan, considering the type of aircraft.

$TELT_i$: Aircraft type $i$ in size category based on three different types of aircraft in small, medium and large.

$\Delta_{ij}$: The minimum time separation between aircraft $i$ and $j$, if aircraft $i$ land before aircraft $j$.

$ea_i$: The allowed earliness for aircraft $i$ to land before $ELT_i$, from the moment the wheels touch the ground to reach the parking lot (including moving along taxiways).

$da_i$: The allowed lateness for aircraft $i$ to land after $ELT_i$, from the moment the wheels touch the ground to reach the parking lot (including moving along taxiways).

$E_i$: Aircraft earliness, meaning how much earlier than the predetermined time the aircraft has landed. If it is zero, it means that the aircraft landed later than the preset time. It can be calculated by the following formula:

$$E_j = \max(ELT_i - SLT_j, 0)$$

$T_j$: Aircraft lateness, meaning how much later the aircraft has landed. If it is zero, it means that the aircraft landed earlier than the preset time. This variable is calculated as follows:

$$T_j = \max(SLT_i - ELT_j, 0)$$

$X_{ij}$: If aircraft $i$ land before aircraft $j$, this value is 1, and if aircraft $i$ land after aircraft $j$, this value will be zero.

### 2.2 Cost function

As mentioned, the goal is to assign a landing schedule for each aircraft so that the objective function of this problem is minimized in the face of given conditions. The constraints of this problem are considered according to the earliest possible time and the latest possible time for landing. Optimization of the absolute value between the predicted landing time and the predetermined time and consideration of the mentioned conditions may help solve the problem. By minimizing this objective according to paper [10], we will also achieve the goal of minimizing the earliness and lateness of aircraft. As a result, additional fuel costs and other costs (such as parking costs) incurred by aircraft late arrival or early arrival are reduced. According to the above explanations, the cost function can be defined as follows:

$$\min \sum_{i=1}^{n} |SLT_i - ELT_i| \tag{2.1}$$

where $n$ is the number of aircrafts whose landing scheduling is determined by this cost function. There are several operational restrictions on ALP. Considering the real world, the most practical ones for use in a band are mentioned. In general, all scheduled landing times (SLT) should be determined and calculated based on the following restrictions.

- Runway Use Restrictions: Each runway can be used by only one aircraft at the same time. Thus, aircraft $i$ land before aircraft $j$ or vice versa.

$$X_{ij} + X_{ji} = 1,\ \forall ij = 1, 2, ..., n \tag{2.2}$$

  In this relation, if aircraft $i$ land before $j$, $X_{ij} = 1,\ X_{ji} = 0$, and if aircraft $j$ land before $i$, then: $X_{ij} = 0,\ X_{ji} = 1$. This condition implies that under no circumstances will a specific landing time be allotted to two aircraft. In the event of such a problem, the program re-calculates another time for the aircraft to land until this condition is met.

- Guarantee limit for minimum separation distance: An aircraft following another aircraft should be in a safe distance from the other aircraft to avoid wake turbulence created by the aircraft ahead of it.

$$(SLT_i - SLT_j) \geq \Delta_{ij} \tag{2.3}$$

  Here, $SLT_i$ is the predicted landing time for the $i^{th}$ aircraft and $SLT_j$ is the expected landing time for the $j^{th}$ aircraft. The expression $\Delta_{ij}$ indicates the minimum separation required for a safe landing between aircraft

$i$ and $j$. This condition depends on the size and weight of the aircraft $i$ and $j$. Given these constraints and formulations mentioned above, the aircraft landing problem is considered an NP-hard optimization problem, so the complexity of this problem, makes finding the optimal solution difficult, even for medium-sized samples. In addition, it is very important to have a flexible schedule based on an acceptable time.

There are two states with regard to the problem. A static case in which the number of aircraft, the constraints and the corresponding time must be predefined in the table. In the latter case, which is dynamic, the rules of updating are constantly applied and it is possible to manage a new aircraft outside the schedule. Different cost functions related to these two cases can be applied. In this article, the static case is considered. It is also assumed that there is one flight path. The following is a description of the scheduling structure using the improved Gray Wolf algorithm.

### 2.3 Scheduling structure

The proposed approach tries to provide the optimal schedule using the improved Gray Wolf optimization algorithm. Scheduling by the Gray Wolf algorithm requires a structure to provide scheduling. This structure is the basis of all calculations performed in this algorithm. Therefore, first, the structure is introduced. To schedule, a set containing $n$ aircraft with an array of length $n$ is used, where the contents of each cell will be a number that is the landing time of aircraft $i$. To clarify, assume that a set of 10 aircraft is available for scheduling. A schedule for this data set can be shown as Table 1.

Table 1: An example of ten aircraft scheduling

| | |
|---|---|
| 1 | 10.12 |
| 2 | 10.21 |
| 3 | 10.30 |
| 4 | 10.18 |
| 5 | 10.24 |
| 6 | 10.37 |
| 7 | 10.40 |
| 8 | 10.33 |
| 9 | 10.50 |
| 10 | 10.45 |

For example, in this landing table, the aircraft number 7 is scheduled for 10.40. Obviously, this structure can help to create different schedules.

## 3 Scheduling with improved Gray Wolf algorithm

In the proposed method, the improved gray wolf algorithm is used for scheduling. The Gray wolf algorithm (GWO) is a meta-heuristic algorithm inspired by the hierarchical structure and social behavior of wolves while hunting. This algorithm is population-based, has a simple configuration process, and can easily be generalized to large-scale problems [9]. The Gray Wolf algorithm was introduced by Mirjalili et al [9], in 2014. According to the results obtained in [9], this algorithm has shown more exploitation ability compared to differential evolution (DE) algorithm and particle swarm optimization (PSO) algorithm. The results also showed better performance in high local optima avoidance in both constrained and non- constrained problems. Therefore, in this research, this algorithm has been considered.

### 3.1 Hierarchical structure and social behavior of gray wolves

Gray wolves are at the top of the food chain and have a social life. Grey wolves prefer to live in a pack whose size is 5-12 on average. There are 4 main ranks in each pack, modeled as a pyramidal structure as shown in Figure 1:

1. Leader wolves are called alpha group, and they can be male or female. These wolves dominate the pack and manage items such as resting places or hunting grounds.
   In addition to the dominant behavior of alpha wolves, a kind of democratic structure is also seen in the group.
2. Beta wolves assist alpha wolves in the decision-making process and are also prone to being selected in their place.
3. Delta wolves: Lower than beta wolves and include old wolves, predators and baby wolves.

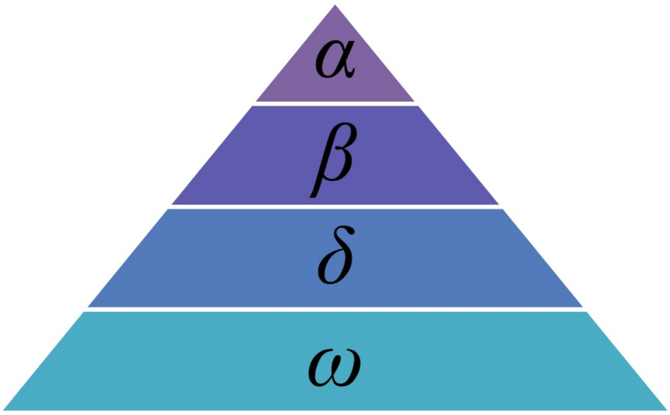


Figure 1: Hierarchy of grey wolf (dominance decreases from top down).

4. Omega wolves: The lowest rank in the hierarchical structure has the least rights over the rest of the group. After all, they eat and do not participate in the decision-making process.

The main phases of gray wolf hunting are as follows:

1. Tracking, chasing, and approaching the prey
2. Pursuing, encircling, and harassing the prey until it stops moving
3. Attack towards the prey

In this research, the hierarchical structure and social behavior of wolves during the hunting process are mathematically modeled and used to design an algorithm for optimization.

**A. Hierarchical structure modeling (power pyramid)**

Optimization is done using alpha, beta and delta wolves. A wolf, alpha, is assumed to be the main leader of the algorithm, and a wolf, beta, and another one, delta, also participate in hunting. The other wolves are considered their followers.

**B. Modeling the process of encircling the prey**

The following two modeling equations are used:

$$D = |CX_p(t) - X(t)|\,,\ C = c_1 r_2 \tag{3.1}$$

where $A,\ C$ are coefficient vectors, $X_p$ is the spatial position vector of the prey, and $X$ is the spatial position vector of each wolf, and $t$ indicates the current iteration.

$$X(t+1) = X_p(t) - AD,\ A = 2a(t)r_1 - a(t) \tag{3.2}$$

Where components of a decrease linearly from 2 to 0 over the course of iterations and $r_1,\ r_2$ are random vectors in the range [0,1].

**C. Modeling the process of hunting**

Grey wolves have the ability to estimate the location of prey. For mathematical modeling, this process is performed as follows:

In the initial search, we have no idea about the location of the prey. We suppose that the alpha, beta and delta have better first-hand knowledge about the potential location of prey (optimum). The position of these three best candidate solutions is determined using the following formulas:

$$D_\alpha = |C_1 X_\alpha(t) - X(t)|\,,\ X_1 = X_\alpha - A_1 D_\alpha; \tag{3.3}$$

$$D_\beta = |C_2 X_\beta(t) - X(t)|\,,\ X_2 = X_\beta - A_2 D_\beta; \tag{3.4}$$

$$D_\delta = |C_3 X_\delta(t) - X(t)|\,,\ X_3 = X_\delta - A_3 D_\delta; \tag{3.5}$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \tag{3.6}$$

**D. Modeling the process of attacking**

Once the prey is surrounded by wolves and stops moving, the attack begins led by Alpha Wolf. In order to mathematically model attacking the prey we decrease the value of vector $a$. Since $A$ is a random vector in the interval $[-2a, 2a]$, if a decreases, the coefficient vector $A$ also decreases.

If $|A| < 1$, the alpha wolf will approach the prey (and the rest of the wolves) and if $|A| > 1$ the alpha wolf will move away from the prey (and the rest of the wolves).

The GWO algorithm obliges all wolves to update their position according to the position of alpha, beta and delta wolves.

**E. Searching (exploring) for the prey**

The search process is exactly the opposite of the attack process, that is, while searching, wolves move away from each other to search for prey ($|A| > 1$), but after tracking the prey, in the attack phase, the wolves approach each other to attack prey. ($|A| < 1$). This process is called divergence when searching and convergence when attacking.

**The role of C Vector:** The $C$ vector is considered as the effect of obstacles in nature that slow down wolves to approach prey. $C$ Vector provides random weights for prey and makes it more inaccessible to wolves. This vector, unlike $a$, does not decrease linearly from 2 to 0.

To sum up, the algorithm can be summarized as follows:

1. The fitness of all the solutions is calculated and the top three solutions are selected as alpha, beta and delta until the end of the algorithm.
2. In each iteration, the top three solutions (alpha, beta, and delta wolves) are able to estimate the location of prey, and this is done using equation (3.6) in each iteration.
3. In each iteration, after determining the position of alpha, beta and delta wolves, the rest of the solutions, following these wolves, update their position.
4. In each iteration the vectors $a$ (and consequently $A$) and $C$ are updated.
5. At the end of the iterations, the position of the alpha wolf is considered as the optimal point.

## 4 Improved gray wolf algorithm

In [6], a modified gray wolf algorithm for multilevel thresholding is proposed in which the new position of wolves in Equation (3.6) is calculated by weighting coefficients in Equation (4.1);

$$\vec{X}(t+1) = w_1 \cdot \vec{X}_1 + w_2 \cdot \vec{X}_2 + w_3 \cdot \vec{X}_3 \tag{4.1}$$

where $w_1, w_2, w_3$ are the corresponding weights that are calculated by the following equations:

$$w_1 = \frac{f_1}{F},\ w_2 = \frac{f_2}{F},\ w_3 = \frac{f_3}{F} \tag{4.2}$$

where $f_1, f_2, f_3$ represent the value of the objective function of the problem. $F = f_1 + f_2 + f_3$. In this study, the main modification to GWO is the application of weighting coefficients (4.2) to the equation of updating search agents. Since for each wolf a different fitness function is obtained, placing the weight as equation (4.2) will increase the accuracy and precision of the algorithm.

In order to improve the global and local search of the algorithm, a series of random searches around the position obtained from all three alpha, beta and delta wolves are performed according to Equation (4.1), that is, a random number is considered, if this number is greater than 0.4, equation (4.1) is executed, and if this number is between 0.1 and 0.4, the following equation is executed:

$$\vec{X}(t+1) = w_1 \cdot \vec{X}_1 + w_2 \cdot \vec{X}_2 + w_3 \cdot \vec{X}_3 + (2rand_i(5) - 5) \tag{4.3}$$

In Equation (4.3), an integer from one to 5 is calculated using the $rand_i$ command. As a result, we will have a number between $-3$ and 5 in parentheses.

Otherwise, if it is less than 0.1, Equation (13) is applied.

$$\vec{X}(t+1) = w_1 \cdot \vec{X}_1 + w_2 \cdot \vec{X}_2 + w_3 \cdot \vec{X}_3 + (a)rand_n(1, nVar) \tag{4.4}$$

Here, $rand_n$ is a random expression added to the new position and controlled by the algorithm search coefficient $a$. When a is large, the focus is on global search, and when $a$ is small, the focus is on local search. The performance of this algorithm in air traffic flow management will be examined.

## 5 Discussion and conclusion

To conduct the experiments, the data set collected on the flights of Shahid Hasheminejad Airport in Mashhad was used. Experiments on real data sets related to time flights from 3/5/2018 to 3/7/2018 are included. This data set includes flight date, aircraft ID, origin, destination, flight register, aircraft type, flight altitude, aircraft size, flight path, scheduled landing time, flight speed and so on. The size of this data is 4368. From this data set, data including flight date, aircraft type, and scheduled landing time are required to be used in the aircraft landing problem.

Evaluations are made using the improved gray wolf algorithm on the cost function (1). To carry out the evaluations, first 10% of the data is considered and for this amount of data, the results are obtained by methods of improved gray wolf (IGWO), common gray wolf (GWO), particle swarm (PSO), and the firefly algorithm (FA). After recording the results of the evaluation of each method, 20% of the data set is considered for evaluation and this time all four methods for this amount of data are examined and the results are recorded.

This procedure continued until 100% of the data set was evaluated. All algorithms are run 20 times and the results for the mean, best and worst cost function found are stored. The comparative results of the algorithms for different data are shown in Table 2 and Figure 2.

Table 2: Comparative results of the cost function value obtained between IGWO, GWO, PSO, FA algorithms after 20 runs

| **Algorithm** | | **10%** | **20%** | **30%** | **40%** | **50%** | **60%** | **70%** | **80%** | **90%** | **100%** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **IGWO** | The best | 1002 | 2007 | 3137 | 4074 | 5137 | 6107 | 7103 | 8255 | 9180 | 10325 |
| | Mean | 1061 | 2085 | 3188 | 4143 | 5262 | 6211 | 7240 | 8265 | 9300 | 10372 |
| | The worst | 1085 | 2142 | 3260 | 4238 | 5392 | 6345 | 7401 | 8301 | 9399 | 10412 |
| **GWO** | The best | 1415 | 2706 | 4127 | 5259 | 6873 | 8297 | 8796 | 10986 | 12204 | 13606 |
| | Mean | 1422 | 2754 | 4203 | 5531 | 6968 | 8470 | 9766 | 11058 | 12437 | 13843 |
| | The worst | 1451 | 2807 | 4300 | 5733 | 7084 | 8617 | 9902 | 11170 | 12661 | 13964 |
| **PSO** | The best | 1873 | 3748 | 5802 | 7382 | 9320 | 11191 | 13075 | 14737 | 16986 | 18708 |
| | Mean | 1985 | 3839 | 5791 | 7540 | 9476 | 11391 | 13203 | 14881 | 17068 | 18824 |
| | The worst | 2101 | 3931 | 5892 | 7672 | 9690 | 11429 | 13444 | 15078 | 17251 | 19031 |
| **FA** | The best | 1233 | 2614 | 3887 | 5146 | 6434 | 7633 | 8935 | 10238 | 11555 | 12849 |
| | Mean | 1287 | 2645 | 3948 | 5212 | 6562 | 7725 | 9093 | 10300 | 11636 | 12924 |
| | The worst | 1313 | 2681 | 3994 | 5312 | 6679 | 7849 | 9146 | 10335 | 11674 | 13082 |

The corresponding numbers in Table 2 indicate the time difference between the time predicted by different algorithms and the time predefined in the schedule. According to Table 2, it can be seen that the proposed improved gray wolf algorithm (IGWO), which has the best, Mean and worst cost function in 20 times of program running, has achieved better results than other algorithms. For example, in the case of 10% of the data, the best value function of cost obtained from the IGWO algorithm is 1061, which is a better result, compared to the GWO algorithm, with a numeric value of 1415, PSO algorithm with a numeric value of 1873, and FA algorithm with a numeric value of 1287. Moreover, in the case of 100% of the data, the proposed IGWO algorithm has a numerical value of 10372 in terms of the Mean of best cost function in 20 times, which is a lower value compared to GWO algorithm, with a numerical value equal to 13843, PSO algorithm with a numerical value equal to 18824, and FA algorithm with a numerical value equal to 12924.

Overall, it can be seen that all the numbers related to the value of the cost function obtained from the improved Gray Wolf algorithm are lower than the other algorithms. Therefore, the high efficiency and performance of this algorithm in solving the problem of flight scheduling is confirmed. Figure 2 shows a comparison curve for the Mean cost function obtained from the algorithms for 100% of the data.

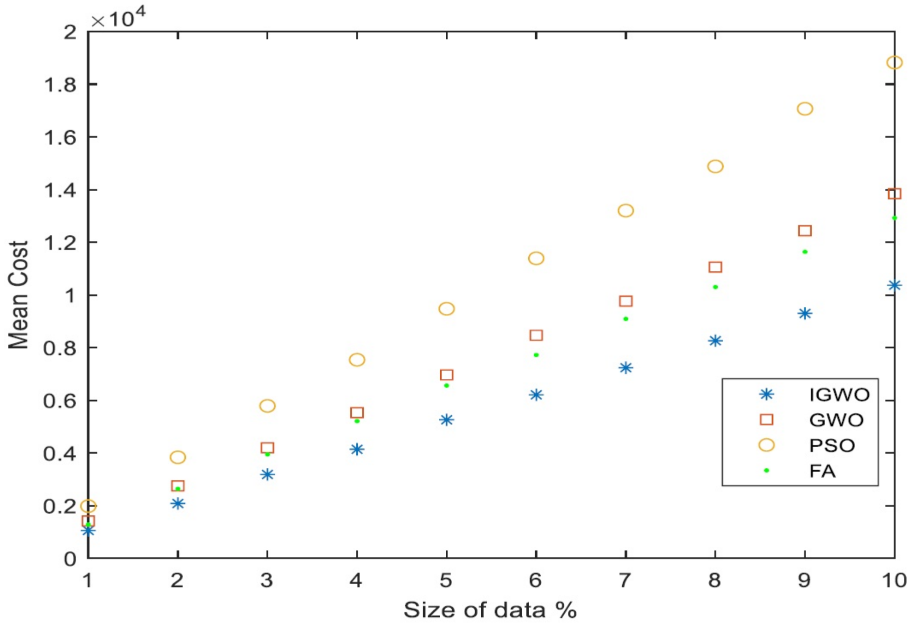


Figure 2: Mean curve of cost function in terms of data size using IGWO, GWO, PSO, FA algorithms

Figure 2 shows the points representing the Mean of the best cost found in terms of data size. For example, the point [3 0.6] with a circle shape in this curve is related to the PSO optimization algorithm, that is, the value 5791 for this algorithm which is obtained from Table 2. The star shaped points are related to the IGWO algorithm. Point shaped, square shaped, and circle shaped points correspond to the FA, GWO, and PSO optimization algorithms, respectively. For all the different data sizes, the proposed algorithm was able to obtain smaller values compared to point, square, and circle points. According to Table 2, it can be seen that in general, the proposed improved Gray Wolf (IGWO) algorithm has performed better in terms of the Mean cost function. Among other algorithms, the performance of the firefly (FA) algorithm is better than the other two algorithms, but the running speed of this algorithm is low. Overall, based on the results, it is concluded that the proposed improved Gray Wolf algorithm has performed better in terms of cost function minimization in all cases related to different data sizes compared to other algorithms.

## 6 Conclusion and suggestions

In this article, an approach was presented to solve the aircraft landing problem (ALP) related to the real data set of Shahid Hasheminejad Airport in Mashhad using an improved Gray Wolf algorithm. The proposed improved Gray Wolf algorithm is designed to be able to schedule flights in such a way as to achieve the goal of minimizing the difference between the predicted and scheduled time, taking into account the problem conditions. In order to increase the efficiency of the algorithm, a mutation operator and a random search were performed around the best wolf. In order to improve this algorithm, based on the amount of fitness obtained for each type of alpha, beta and delta wolves, a weight was assigned to them and then a new position was obtained based on this weighting. Simulations were performed on 10 percent to 100 percent of the data. In order to evaluate the performance of the proposed method, the results were compared using particle swarm optimization, common Gray Wolf and firefly algorithms. The results showed that the proposed algorithm had significant performance and efficiency compared to other similar algorithms. As a suggestion, in future work, the proposed algorithm can be used in applications similar to the aircraft landing problem.

## References


[1] G. Bencheikh, J. Boukachour and A.E.H. Alaoui, *A memetic algorithm to solve the dynamic multiple runway aircraft landing problem*, J. King Saud Uni.-Comput. Inf. Sci. **28** (2016), no. 1, 98–109.

[2] J.A. Bennell, M. Mesgarpour and C.N. Potts, *Dynamic scheduling of aircraft landings*, European J. Oper. Res. **258** (2017), no. 1, 315–327.

[3] S. Caprí and M. Ignaccolo, *Genetic algorithms for solving the aircraft-sequencing problem: the introduction of departures into the dynamic model*, J. Air Transport Manag. **10** (2004), no. 5, 345–351.

[4] Q. Deng, B.F. Santos and R. Curran, *A practical dynamic programming based methodology for aircraft maintenance check scheduling optimization*, Eur. J. Oper. Res. **281** (2020), no. 2, 256–273.

[5] S. Ikli, C. Mancel, M. Mongeau, X. Olive and E. Rachelson, *Coupling mathematical optimization and machine learning for the aircraft landing problem*, ICRAT 2020, 9th International Conference for Research in Air Transportation, 2020.

[6] L. Li, L. Sun, J. Guo, J. Qi, B. Xu and S. Li, *Modified discrete grey wolf optimizer algorithm for multilevel image thresholding*, Comput. Intell. Neurosci. **2017** (2017).

[7] M. Liu, B. Liang, F. Zheng, C. Chu and F. Chu, *A two-stage stochastic programming approach for aircraft landing problem*, 15th Int. Conf. Serv. Syst. Serv. Manag. (ICSSSM), IEEE, 2018, pp. 1–6.

[8] A.A. Mahmud and W. Jebersen, *Review on dynamic aircraft scheduling*, Int. J. Pure Appl. Math. **117** (2017), no. 21, 753–767.

[9] S. Mirjalili, S.M. Mirjalili and A. Lewis, *Grey wolf optimizer*, Adv. Engin. Software **69** (2014), 46–61.

[10] S. Mokhtarimousavi, H. Rahami and A. Kaveh, *Multi-objective mathematical modeling of aircraft landing problem on a runway in static mode, scheduling and sequence determination using NSGA-II*, Iran Univ. Sci. Technol. **5** (2015), no. 1, 21–36.

[11] S. Mokhtarimousavi, H. Rahami, M. Saffarzadeh and S. Piri, *Determination of the aircraft landing sequence by two meta-heuristic algorithms*, Int. J. Transport. Eng. **1** (2014), no. 4, 271–284.

[12] S. Mokhtarimousavi, D. Talebi and H. Asgari, *A non-dominated sorting genetic algorithm approach for optimization of multi-objective airport gate assignment problem*, Transport. Res. Record **2672** (2018), no. 23, 59–70.

[13] K.K.H. Ng, C.K.M. Lee, F.T. Chan and Y. Qin, *Robust aircraft sequencing and scheduling problem with arrival/departure delay using the min-max regret approach*, Transport. Res. Part E: Logist. Transport. Rev. **106** (2017), 115–136.

[14] R. Prakash, R. Piplani and J. Desai, *An optimal data-splitting algorithm for aircraft scheduling on a single runway to maximize throughput*, Transportation Research Part C: Emerging Technologies, 95 (2018), 570–581.

[15] Y. Ren, Z. Lu and X. Liu, *A branch-and-bound embedded genetic algorithm for resource-constrained project scheduling problem with resource transfer time of aircraft moving assembly line*, Optim. Lett. 14 (2020), no. 8, 2161–2195.

[16] J.H. Ruan, Z.X. Wang, F.T. Chan, S. Patnaik and M.K. Tiwari, *A reinforcement learning-based algorithm for the aircraft maintenance routing problem*, Expert Syst. Appl. **169** (2021), 114399.

[17] A. Salehipour, M. Modarres and L.M. Naeni, *An efficient hybrid meta-heuristic for aircraft landing problem*, Comput. Oper. Res. **40** (2013), no. 1, 207–213.

[18] H. Shahmoradi-Moghadam, N. Safaei and S.J. Sadjadi, *Robust maintenance scheduling of aircraft fleet: a hybrid simulation-optimization approach*, IEEE Access **9** (2021), 17854–17865.

[19] K. Sylejmani, E. Bytyçi and A. Dika, *Solving aircraft sequencing problem by using genetic algorithms*, Intell. Decision Technol. **11** (2017), no. 4, 451–463.

[20] S. Vadlamani and S. Hosseini, *A novel heuristic approach for solving aircraft landing problem with single runway*, J. Air Transport Manag. **40** (2014), 144–148.

[21] M. Wei, B. Sun, W. Wu and B. Jing, *A multiple objective optimization model for aircraft arrival and departure*

*scheduling on multiple runways*, Math. Biosci. Eng. **17** (2020), no. 5, 5545–5560.

[22] M. Wei, L. Zhao, Z. Ye and B. Jing, *An integrated optimization mode for multi-type aircraft flight scheduling and routing problem [J]*, Math. Biosci. Eng. **17** (2020), no. 5, 4990–5004.

[23] B. Xu, *An efficient ant colony algorithm based on wake-vortex modeling method for aircraft scheduling problem*, J. Comput. Appl. Math. **317** (2017), 157–170.

[24] S.P. Yu, X.B. Cao and J. Zhang, *A real-time schedule method for aircraft landing scheduling problem based on cellular automation*, Appl. Soft Comput. **11** (2011), no. 4, 3485–3493.

[25] J. Zhang, P. Zhao, C. Yang and R. Hu, *A new meta-heuristic approach for aircraft landing problem [J]*, Trans. Nanjing Univ. Aeronautics and Astronautics **37** (2020), no. 2, 197–208.

[26] J. Zhang, P. Zhao, Y. Zhang, X. Dai and D. Sui, *Criteria selection and multi-objective optimization of aircraft landing problem*, J. Air Transport Manag. **82** (2020), p. 101734.

[27] S. Zheng, Z. Yang, Z. He, N. Wang, C. Chu and H. Yu, *Hybrid simulated annealing and reduced variable neighbourhood search for an aircraft scheduling and parking problem*, Int. J. Product. Res. **58** (2020), no. 9, 2626–2646.

[28] A. Zulkifli, N.A.A. Aziz, N.H.A. Aziz, Z. Ibrahim and N. Mokhtar, *Review on computational techniques in solving aircraft landing problem*, Proc. Int. Conf. Artific. Life Robotics, Oita, Japan, 2018.