Journal of Optimization in Industrial Engineering, Vol.11, Issue 1,Winter and Spring 2018, 77-90
DOI: 10.22094/JOIE.2018.234

A Bi-Objective Airport Gate Scheduling with Controllable Processing Times Using Harmony Search and NSGA-II Algorithms

Morteza Khakzar Bafruei[a,*], Sananz khatibi[b], Morteza Rahmani[c]

[a] *Assistant Professor, Department of Industrial Engineering, Technology Development Institute (ACECR), Tehran, Iran*
[b] *Ph.D. Student, Department of Industrial Engineering, Technology Development Institute (ACECR), Tehran, Iran*
[c] *Associate Professor, Department of Industrial Engineering, Technology Development Institute (ACECR), Tehran, Iran*

Received 20 June 2015; Revised 11 May 2016; Accepted 30 June 2016

Abstract

Optimizing gate scheduling at airports is an old, but also a broad problem. The main purpose of this problem is to find an assignment for the flights arriving at and departing from an airport, while satisfying a set of constraints.A closer look at the literature in this research line shows thatin almost all studies airport gate processing time has been considered as a fix parameter. In this research, however, we investigate a more realistic situation in which airport gate processing time is a controllable. It is also assumed that the possible compression/expansion processing time of a flight can be continuously controlled, i.e. it can be any number in a given interval.Doingsohas some positive effectswhich lead to increasing the total performance at airports' terminals. Depending on the situation, different objectives become important.. Therefore, a model which simultaneously (1) minimize the total cost of tardiness, earliness, delay andthe compression as well as the expansion costs of job processing time, and (2) minimize passengers overcrowding on gate is presented. In this study, we first propose a mixed-integer programming model for the formulated problem. Due to complexity of problem, two multi-objective meta-heuristic algorithms, i.e. multi-objective harmony search algorithm (MOHSA) and non-dominated sorting genetic algorithm II (NSGA-II) are applied in order to generate Pareto solutions. For calibrating the parameter of the algorithms, Taguchi method is used and three optimal levels of the algorithm's performance are selected. The algorithms are tested with real-life data from Mehrabad International Airport for nine medium size test problems. The experimental results show that NSGA-II has better convergence near the true Pareto-optimal front as compared to MOHSA; however, MOHSA finds a better spread in the entire Pareto-optimal region.Finally, it is possible to apply some practical constraints into the model and also test them with even large real-life problems instances.

Keyword: Gate scheduling problem, Multi-objective decision making, Harmony search algorithm, NSGA-II, Controllable processing times.

1. Introduction

Because of the growth of air transport traffic, which has roughly doubled since the early 1980s, techniques for efficiently managing and allocating of airport and airlines resources have receiveda considerable attention. The increased competition among airlines and the demand for more comfort have brought the requirement of new models and methods. The scheduling problems faced by airport and airline managers are even more complex than most other traditional scheduling problems (Dorndorf et al., 2007). According to the literature, the most important topic which has been considered by experts in recent years is the airport gate scheduling problem. The aim is to find an assignment of flights to suitable positions at the airport terminal so that certain criteria are met. The stand positions of aircrafts are commonly referred to as "gate", and they are designed for the passenger'sembarkment or disembarkment (Dorndorf et al., 2008). In addition, the schedule should be appropriate for airport services and comfortable for passengers. Gate assignment is an important decision problem in daily operations at main airports all over the world.

The main input for gate scheduling is the flight timetable that consists of the flight arrival and departure times. Other important parameters are the number of gates and passengers, the aircraft types, the flight processing time and special handling procedures. It is clear that the input data is subject to uncertainty and may change over time (Dorndorf et al., 2012). The uncertainty takes place in several situations such as gate breakdown, flight delay, flight cancelation, flight emergency, bad weather condition and so forth. Hence the management must reschedule the gates to minimize extra delays due to undesirable conditions.

With respect to real-life situation, almostall of the classical gate scheduling models assume that airport's gates processing times are fix; however, the processing times depend on the amount of resources such as equipment, capabilities of facilities, and human resources,

---

*Corresponding author Email address: khakzar@acecr.ac.ir which might over time Obviously, this assumption are in contrast to the real-world problems.

The controllable processing time means that each job can be processed in a shorter or longer time, depending on its objective function, by reducing or increasing the available resources such as equipment or human resources. Whenever the job processing times are controllable, the scheduling performance is increased and the cost of manufacturing is decreased (Zarandi and Kayvanfar, 2015).

Scheduling problem with controllable processing time can be seen in some context such as metallurgy and chemical industries. For example, in chemical industry, the processing time of job is reduced by using catalyzer or increase by an inhibitor. An inhibitor is any agent that interferes with the activity of enzymes. Enzyme inhibitors are molecules that bind to enzymes and decrease their activity (Zarandi and Kayvanfar, 2015, Wang et al., 2003, and Sørensen et al., 2004).

In this study, we consider more realistic situations where the processing time of the airport gate is controllable and the arrival and departure time can be changed, i.e., for the small flights the processing time can be compressed. This means that the total time, which is compressed, can be assigned for servicing the other flights. For example, by increasing the number of buses for picking up the passenger from gates to the apron site,or by increasing the number of the persons who control the flight's card, more passengers can be serviced in shorter times. In addition, for (big) crowded flights the processing time can be expanded, which means that passengers can embark the airplane in a more comfortable way; it reduces the passengers' congestion in the terminal's hall areas and increases passengers' satisfaction.

Moreover, these assignments have to fulfill some constraints and optimize different objectives. Thus, we proposed a bi-objective model which (1) minimize total cost of tardiness, earliness, delay as well as compression and expansion costs of job processing time and (2) minimize the passengers overcrowding on gate.

Since, the delay, earliness, tardiness, and passengers crowding (the mentioned objectives) are not desirable, and because one can represent airport authority concerns and other one may represent the passenger concerns; it motivated us to present a model which simultaneously minimizes all. In addition, it is assumed that the normal processing time of flights can be reduced or increased to a predefined limit.

The main contribution of this research could be summarizedas follows:

- To the best of the authors' knowledge, there exists no accomplished research for airport gate scheduling problem in which such criteria have been considered together while the job processing times are controllable. In this research, controllable processing times mean the flights could be either compressed or expanded up to a certain limit.
- Two well-known multi-objective meta-heuristic algorithms, i.e., multi-objective harmony search algorithm (MOHSA) and non-dominated sorting genetic algorithm II (NSGA-II) are applied for solving such a bi-objective problem.

The rest of the paper is organized as follows: Section 2 goes over the related literature. In Section 3, the problem definition including assumptions, notations, and the mathematical model are described. In Section 4, the proposed meta-heuristic algorithms are explained. Section 5 consists of computational results, and finally, Section 6 includes conclusions and future works.

2. Literature Review

Since the early 1970s, there have been a number of papers on airport and airline management. Here, only the literature concerned with the gate assignment problem is reviewed. Babic et al. (1984) formulated the gate assignment as a 0-1 Integer-Programming (IP) problem and used a branch and bound algorithm for finding an optimal solution for the GAP where transfer passengers were not considered. Later, Mangoubi and Mathaisel (1985) took this requirement into account and used an LP relaxation as well as greedy heuristics in order to solve the GAP. Haghani and Chen (1998) suggested a heuristic that assigned parking successive flights at the same gate when there is no overlapping. In the situation where there was overlapping, flights were assigned based on the shortest walking distance coefficients. Yan and Chang (1998) proposed a network model which was formulated as a multi-commodity network problem. Xu and Baily (2001) proposed a Tabu search meta-heuristic in order to solve the problem. The algorithm exploited special properties of several types of neighborhood moves and created highly efficient candidate list strategies. Yan and Hou (2001) employed a fix buffer time in their static gate assignment problems between two continuous flights assigned to the same gate in order to absorb the stochastic flights' delay. Dorndorf (2008) concurrently maximized flight gate preference scores, minimized dummy gate assignments and minimized the number of tows, which were required if an aircraft was assigned to different gates for arrival and departure procedures. The problem was solved using the truncated branch and bound algorithm. Ding et al. (2004, 2005) presented a GAP in which the number of flights exceeded the number of available gates. They addressed both the objectives of minimizing the number of open (unassigned) flights and total connection times. Also, a two-staged algorithm, which used both a greedy strategy and a Tabu search meta-heuristic improved by a new neighborhood search technique were proposed for solving the problem. Lim et al. (2005) studied gate assignment with the time windows. Dorndorf et al. (2007) presented a detailed review of gate assignment problem and considered minimizing an absolute deviation of the new gate assignment from a so-called reference schedule as a part of their objective function. The problem was also solved using the truncated branch and bound algorithm. Contrary to Dorndorf et al. (2002),Pintea et al. (2008) proposed a hybrid Ant-local search system for the airport gate assignment problem denoted by HAS-AGAP. Nikulin and Drexl (2008) added a third objective of maximizing flight gate preference scores. The problem was solved by Pareto-simulated annealing. Das (2009) formulated an integer-programming (IP) formulation with the objective of minimizing total walking distances of passengers, which was subject to different constraints relevant to a practical situation. Jaehn (2010) considered the problem of gate assigning flights to airport with the only objective of maximization of flight gate preference scores. A dynamic programming also was proposed as a solution approach solving the problem in linear time. Niklin and Drexl (2007, 2010) focused on the multi-criteria aspect and solved the problem using Pareto Simulated Annealing. Dorndorf et al. (2012) addressed the way the flight gate scheduling problem modeled as a clique partitioning problem can be extended for solving a multiple period problem.Diepen et al. (2012) presented a completely new linear integer programming (LIP). The model took into account all realistic constraints and the column generation was exerted as a solution approach. Based on the ideas of Dorndorf (2002),

Bouras et al. (2014)surveyed the state of art of these problems and various methods completely. The survey covered both theoretical and real aspect of gate assignment problem with the description of mathematical formulations and resolution methods. They also classified the solution's methods based on exact, heuristic, and meta- heuristics. Marinelli et al. (2015) proposed a method based on the Bee Colony Optimization (BCO) to find an optimal flight gate assignment for a given schedule. Two main objectives were considered: minimization of passenger total walking distance and remote gate usage.

Guepet et al. (2015) formulated stand allocation problem in assigning aircraft activities to aircraft stands as Mixed Integer Problem (MIP). They considered maximizing the number of passengers at contact stands and minimizing the number of towing movements as objective functions while respecting set of operational and commercial requirement. In addition, they proposed two heuristic algorithms based on actual data from major European airports.

Hidayatno et al. (2015) studied gate assignment problem for Soekarno Hatta International Airport. They considered the mathematical models which were proposed by Drexl and Nikulin (2008). The objective function was minimizing the number of Un-gated Flights. They exerted a meta-heuristic approximation approach namely simulated annealing to solve the problem. The result showed that the number of Un-gated Flights has been decreased.

Palmisano et al. (2015) considered multi criteria objective function for gate assignment problem. They minimized the number of flights assigned to remote terminals and the total walking distance as objective function. As a solution approached, they applied a hybrid Biogeography-based Bee Colony Optimization.

Mercedes et al. (2015) studied robust gate assignment problem in order to tackle arrival delays. They considered a simulation-based experimental approach that evaluated the minimum amount of stands at the terminal necessary to cope with arrival/departure pattern traffic increased and a causal model to evaluate shortages and benefits of different policies and strategies for gate assignment to mitigate undesirable consequences.

Scheduling problems with controllable processing times have received increasing attention during the last decades. Most of the earlier studies on controllable processing times addressed the single machine environment because of complexity of working on other machine environment (2010). Vickson (1980) studied one of the first models on controllable processing time scheduling problems, with goal of minimizing the total flow time and the total processing cost incurred due to job processing time compression.

In the last few decades, a lot of researchers studied multi-objective parallel machine scheduling problem. Of them, we can refer to Shmoys and Tardos (1993),who studied unrelated parallel machine scheduling to minimize the makespan subject to the total flow time. Shabtay and Steiner (2007)accomplished a complete survey on scheduling with controllable processing time. Gurel and Akturk (2007) surveyed the identical parallel CNC machines with controllable processing time in which a time/cost trade-off consideration wasconducted.

Li et al. (2011) considered the identical parallel machine scheduling problem to minimize the makespan with controllable processing times in which the processing times were linear decreasing functions of the consumed resource. Kayvanfar et al. (2015) investigated the identical parallel machine with controllable processing times where each job could be either compressed or expanded to a given extent, while in most of other researches job could be only compressed.They tried to simultaneously minimize total the cost of tardiness, earliness and compression as well as expansion costs of job processing times, and maximum completion time or makespan. For solving such a problem they proposed NSGA-II and NRGA.

Airport gate scheduling problem is equivalent to the parallel machine scheduling with m parallel machines in which generally a flight is served once by an idle gate. In order to make our problem more realistic, we are going to apply controllable processing time's concept for modeling this problem, in Section 3.

3. Problem Definition and Modeling

Consider an airport gate assignment problem as follow: assume that N aircrafts are scheduled at an airport during the next planning horizon. All arrival Ai and processing times P_i, due dates du_i, and maximum amount of flight compression/expansion $P1_i$and $P2_i$ are known in advance. Each flight $i$has the number of passengersPas_i and a start-time window $[TA_i, TB_i]$, i.e., $TA_i$is the earliest start time and $TB_i$is the latest start time. Fig 1 shows the start-time window. It is assumed that M gates are available for giving service to the flights. We assume that the gates are heterogeneous it means that some of the gates may not be permissible for certain flights, or may not provide the desired service level. In addition, we divided the day into six time slots T in order to consider the whole schedule in different time intervals. It is assumed that the flights are in ascending order of their arrival times so that $i>j$ as long as $A_i \geq A_j$.

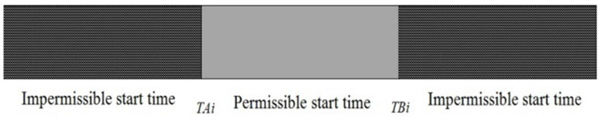


Fig. 1.Hard start-time window for flights

The considered problem is formulated according to the following assumptions.

3.1 Assumptions

- ❖ Airport layout
- The layout of airport terminal is taking into account.
- The buffer time isthe required time period at the gate between two consecutive flights. For simplicity, the buffer time and ground service were not explicitly considered in this work.
- The planning period is set to be one day [00:01 to 23:59].
- ❖ Aircraft
- There are 3 aircraft sizes "small, "medium", and "large" (particularly, B747s, and wide body). It is assumed that if a large aircraft is assigned to one of the gates, the other gates cannot be processed and aircraft of size "medium" or "large"
- ❖ Processing time
- Processing a flight at its normal processing time incurs no additional processing cost.
- The normal processing time could be compressed by an amount of O1i or expanded by an amount O2i which necessitates a unit cost of compression or expansion, respectively.
- ❖ Gate
- No breakdown is allowed, i.e., all gates are available throughout the scheduling period.
- Each gate could process only one flight at time.

3.1 Notations

3.1.1 Subscripts

$i,j \in F$	Index for flight $i,j = 1,2,...,N$
$k,l \in G$	Index for gate $k,l = 1,2,...,M$
$t \in T$	Index for time $t = 1,2,...,6$
$N = \lvert F \rvert$	Number of flights
$M = \lvert G \rvert$	Number of gates

3.1.2 Input parameters

A_i	Arrival time of flight *i*
Pas_i	Number of passenger of flight *i*
P_i	Normal processing time of flight *i*
$P1_i$	Crash (minimum allowable) processing time of flight *i*
$P2_i$	Expansion (maximum allowable) processing time of flight *i*
Cr_i	The cost of delay for flight *i*
c_i	Compression unit cost of flight *i*
$c_i^{'}$	Expansion unit cost of flight *i*
α_i	The earliness unit penalty of flight *i*
β_i	The tardiness unit penalty of flight *i*
$L1_i$	Maximum amount of flight *i* compression, $L1_i = P_i - P1_i$
$L2_i$	Maximum amount of flight *i*expression, $L2_i = P2_i - P_i$
du_i	Due date of flight *i*
M	Very big number
ε	Very small number

3.1.3 Decision variables

$Y_{i,k}$	1: If flight *i*assign to gate *k*; otherwise 0.
$X_{i,j,k}$	1: If flight *i*and*j* assign to gate *k*; otherwise 0.
S_i	The start time of service for flight *i*
$O1_i$	Amount of flight *i*compression, $0 \leq O1_i \leq L1_i$
$O2_i$	Amount of flight *i*expansion, $0 \leq O2_i \leq L2_i$
EE_i	Earliness of flight *i*
TT_i	Tardiness of flight *i*
λ_i	A binary variable to prevent the synchronization
$W_{t,k}$	Total number of passengers on gate *k* in time interval *t*

3.2 The mathematical model

Before the constraints are explained, the objectives functions are described. In this model, objective function (1) minimizes the total difference between the maximum number of passenger in time interval *t* and minimum number of passenger in time interval *t* which is a non-linear objective function. It means that it control crowd congestion on gates and use equal share of capacity of each gates during the whole day. This means that each gate gets its equal share of passengers in each time slot. Objective function (2) considers the minimization of tardiness, earliness, delay penalties and the cost of flights compression as well as expansion processing time.

Equation (3) guarantees that one flight is assigned to one and only one gate. Equation (4) shows that the service time of flight *I* should greater or equal to the arrival time of flight *i*. Equations (5, 6) assure that two flights cannot be simultaneously assigned to the same gate due to overlapping schedule. Equations (7, 8) define the earliness and tardiness of flight *i*whereboth constraints must gate minimized. Equations (9, 10) limit the amount of compression and expansion for each flight. Equations (11, 12) consider that only one of the variables $O1_i$ and $O2_i$ can be positive simultaneously. Equation (13) ensures that if flight *j* come after flight *i*then it will not possible that the flight *i*comes after flight *j*.Equation (14) forces that if flight *j* is after flight *i*on gate *k* then it will not possible be after the other flights on the other gates. Equations (15, 16) state that $X_{i,j,k}$ is 1 if and only if $Y_{i,k}$ and $Y_{j,k}$ are 1 i.e. if both flight *i* and *j* are assigned to gate *k*. Equations (17, 18) are the time window constraint. It means that the flight must receive service between the earliest and latest time.Equation (19) calculates the total passengers on gate *k* in time interval *t*. Finally equations (20) till (27) define the non-negativity and binary requirements for decision variables.

$$F1 = Min \sum_{t=1}^{T} \left(\left[\underset{k}{Max}(W_{t,k}) - \underset{k}{Min}(W_{t,k}) \right] \right) \tag{1}$$

$$F_2 = Min \sum_{i=1}^{n} \alpha_i .EE_i + \beta_i .TT_i + c_i O1_i + c'_i O2_i + \sum_{i=1}^{n} Cr_i (S_i - A_i) \tag{2}$$

$$\sum_{k=1}^{M} Y_{i,k} = 1 \ \forall i \tag{3}$$

$$S_i \geq A_i \ \forall i \tag{4}$$

$$S_j \geq (S_i + P_i - O1_i + O2_i - M(X_{i,j,k} - 1) + 2M(Y_{i,k} + Y_{j,k} - 2)) \ \forall i,j,k, i \neq j \tag{5}$$

$$S_i \geq (S_j + P_j - O1_i + O2_i - M(X_{i,j,k}) + M(Y_{i,k} + Y_{j,k} - 2)) \ \forall i,j,k, i \neq j \tag{6}$$

$$Si + P_i - O1_i + O2_i - du_i \leq TT_i \ \forall i \tag{7}$$

$$du_i - Si - P_i + O1_i - O2_i \leq EE_i \ \forall i \tag{8}$$

$$P_i - P1_i \geq O1_i \ \forall i \tag{9}$$

$$P2_i - P_i \geq O2_i \ \forall i \tag{10}$$

$$M\lambda_i \geq O1_i \ \forall i \tag{11}$$

$$M(1-\lambda_i) \geq O2_i \ \forall i \tag{12}$$

$$\mathrm{X}_{i,j,k} + \mathrm{X}_{j,i,k} \leq 1 \ \forall \mathrm{i,j,k}, i \neq j \tag{13}$$

$$\sum_{k=1}^{M} X_{i,j,k} \leq 1 \ \forall i,j,k, i \neq j \tag{14}$$

$$\mathrm{X}_{i,j,k} \leq \mathrm{Y}_{i,k} \ \forall i,j,k, i \neq j \tag{15}$$

$$\mathrm{X}_{i,j,k} \leq \mathrm{Y}_{j,k} \ \forall i,j,k, i \neq j \tag{16}$$

$$\mathrm{TA}_i \leq \mathrm{S}_i \ \forall i \tag{17}$$

$$\mathrm{S}_i \leq TB_i \ \forall i \tag{18}$$

$$W_{t,k} = \sum_{i=1}^{M} U_{i,t} Pas_i Y_{i,k} \ \forall t,k \tag{19}$$

$Y_{i,k} \in \{0,1\} \ \forall i,k$ (20) $X_{i,j,k} \in \{0,1\} \ \forall i, \mathrm{j}, k$ (21) $S_i \geq 0 \ \forall i$ (22)

$\mathrm{W}_{\mathrm{t,k}} \geq 0 \ \forall t,k$ (23) $O1_i \geq 0 \ \forall i$ (24) $O2_i \geq 0 \ \forall i$ (25)

$TT_i \geq 0 \ \forall i$ (26) $EE_i \geq 0 \ \forall i$ (27)

4. Solution Approaches

In this section, two multi-objective meta-heuristic methods, is used to solve the problem, including NSGA-II and multi-object harmony search algorithm (MOHSA). The solution representation and fitness functions of both algorithms are completely the same.
In the following, we will focus first on NSGA-II and explain the steps of it, later MOHSA will be covered.

4.1. NSGA-II algorithm

NSGA-II is the second version of famous non-dominated sorting genetic algorithm (NSGA) which was proposed by Deb et al. (2000) for solving multi objective optimization problems. The general description of NSGA-II is explained as following.
The population is initialized as genetic algorithm. The major difference is occurred once the population is sorted based on non-dominated sorting into each front. Each individual in each front are assigned rank (fitness) values i.e. individuals in first front are given a fitness value of one and individuals in second front are assigned fitness value as two and so on. In addition to fitness value, another new measure called crowding distance is calculated for each individual. The larger value of this measure will result in better diversity in the population.
For generating offsprings (for new population) using crossover and mutation operators, the population is selected by binary tournament selection. Then the current population with current offsprings is merged, the whole population is sorted again based on non-dominated sorting and only the best N individuals (the population size) are selected which is based on rank and crowding distance (Seshadri 2006). The procedures are repeated till maximum iteration. The general procedure of NSGA-II is shown in Fig 2.
In the following section, the organization of a chromosome used in the algorithm is explained.

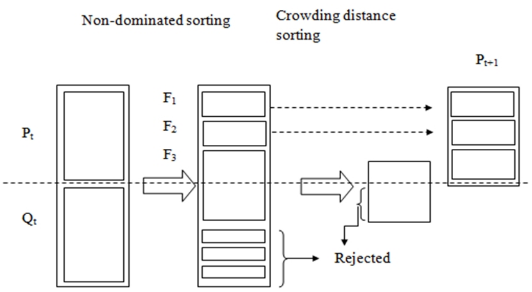


Fig. 2. The general procedure of NSGA-II (Deb 2000)

4.1.1. Solution representation

In the following, the organization of a chromosome (Harmony) used in the survey is demonstrated. The chromosome has two rows. The length of chromosome (in both rows) is taken to be same as the number of flights. For example, if the number of flights is 8, then the length of the chromosome is also 8. In the first row,each gene value of the chromosome is generated randomly through generating random numbers and generated number is scaled to the number of gates. For example, if the number of gates is assumed to be 4, then the generated random number is scaled to a discrete number between [1, 4]. This means that each gene position represents a flight number and each gene value represents the gate number which processes the flight. Thus, against each gate, there is a queue of flights to be processed.
In the second row, each gene value of the chromosome is generated randomly number which is scaled to a number (continues) between [-2, 2]. This means that if the numbers which is generated be in [-2,-1) and (1, 2] intervals then it is converted to 0; otherwise it is considered without any changes. For those have been given zero numbers, the normal processing time is considered; however, for the ones which have negative numbers, the processing time will be compressed. For positive ones the processing time is expanded. This structure is represented in Fig 3.
In order to run the crossover for the first row, two types of discrete crossover, single-point crossover and two-point crossover are implemented over the selected values. Offspring values are then added to the pool of offsprings. For more clarification, employed crossovers are shown in Figs. 4 and 5 for a problem with 8 flights.

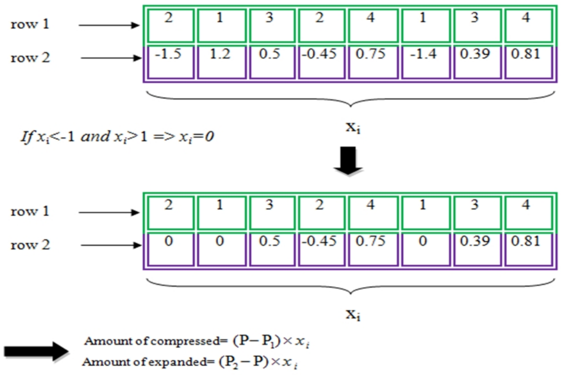


Fig. 3. The structure of chromosome

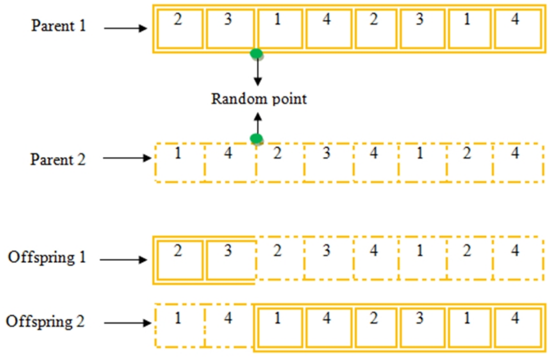


Fig. 4. Single-point crossover

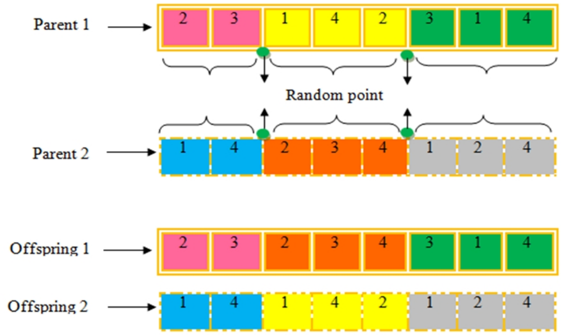


Fig. 5. Two-point crossover

As mentioned before, the second row represents the amount of normal processing time that must have compressed/expanded. The crossover operator is executed as follows,

Offspring$_1$=round (*Parent*$_1$×R+*Parent*$_2$× (1-R)),
Offspring$_2$=round (*Parent*$_2$×R+*Parent*$_1$× (1-R)),

Where child 1 (*Offspring*$_1$) and child 2 (*Offspring*$_2$) are generatedfromparent 1 (*Parent*$_1$) and parent 2 (*Parent*$_2$) using an arbitrary number R (continuous) between(0, 1).Fig. 6 demonstrates the sample.

The detailed descriptions of discrete mutation are given as follows:

1-*Swap mutation*: For using this strategy, firstly two flights in the permutation are randomly selected and then the positions of the selected flights are exchanged.

2-*Reversion mutation*: In this case, the flight in the second position is located immediate after the flight in the first location and the other flights are shifted right hand side accordingly (See Fig. 7).

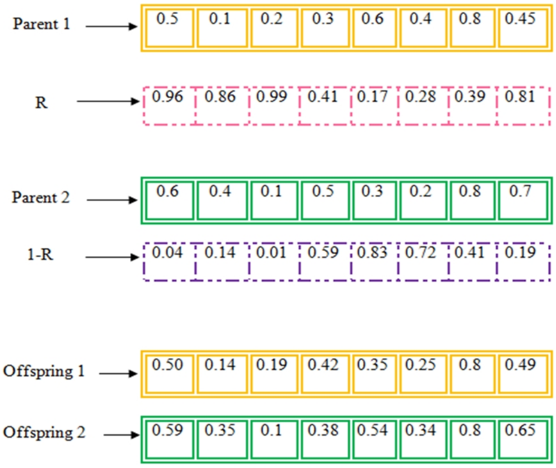


Fig. 6. The structure of continuous crossover

Fig. 7. The structure of discrete mutation, swap and reversion

In addition, for continuous mutation first of all one flight in the permutation is randomly, and selected then a random number between (0, 1) is added to it (See Fig. 8).

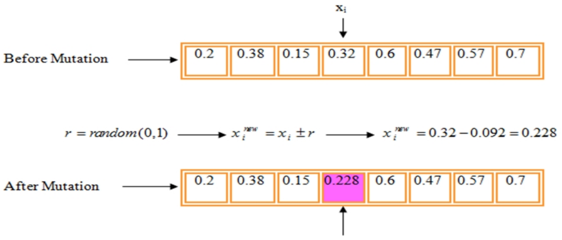


Fig. 8. The structure of continuous mutation

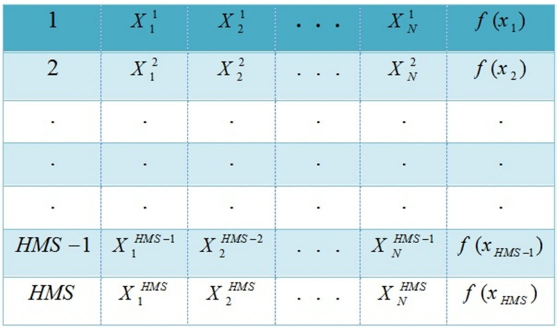
The steps in the procedure of the NSGA-II are as follows:
Step1. Initialize Population
Step2. Check feasibility and perform fast non-dominated sorting approach.
Step3. Generate child population (select population for crossover and mutation operators based on tournament binary selection).
Step4. Merge old and new population.
Step5. Check the feasibility and implement fast non-dominated sorting approach.
Step6. Repeat Step 3, 4 and 5 until maximum iteration.

4.2. Harmony Search Algorithm

Harmony Search (HS) is an emerging global optimization algorithm which was initially proposed by Geem in (2001). The algorithm has been successfully applied in a wide range of optimization problems.

HS algorithm generally consists of a number of optimization operators, such as the harmony memory (HM), the harmony memory size (HMS means number of solution vectors in harmony memory), the harmony memory considering rate (HMCR), and the pitch adjusting rate (PAR) and band width (BW). In the HS algorithm, the feasible vectors are stored in harmony memory (HM).

For an N-dimension problem, the HM with the size of HMS can be represented as Fig. 9, where $x_{i,j}$ is the j-th element of i-th harmony memory vector, and $f(x_i)$ is the fitness function.



1	X_1^1	X_2^1	. . .	X_N^1	$f(x_1)$
2	X_1^2	X_2^2	. . .	X_N^2	$f(x_2)$
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
$HMS-1$	X_1^{HMS-1}	X_2^{HMS-2}	. . .	X_N^{HMS-1}	$f(x_{HMS-1})$
HMS	X_1^{HMS}	X_2^{HMS}	. . .	X_N^{HMS}	$f(x_{HMS})$

Fig. 9. The structure of harmony memory

The solution representation is exactly the same as explained in NSGA-II. Instead of chromosome, we use harmony's concept.

After initialization, the search procedure is started which is called improvisation. The harmony memory consideration rate (HMCR), which is between 0 and 1, controls the balance between exploration and exploitation during improvisation. A random number is generated and compared with HMCR during search process for each decision variable. If it is smaller than HMCR, the memory vector in HM is taken into consideration for generating the new value; otherwise a value is randomly selected from the possible ranges of the decision variable. Each decision variable of the new solution vector obtained from HM is examined to determine whether it should be pitch adjusted. The pitch adjusting rate (PAR) decides the ratio of pitch-adjustment. Another random number between 0 and 1 generated and the pitch-adjustmentoperating as Equation (1) is executed if it is not bigger than PAR (Wang et al. 2011).

$$x_i^{new} = \{(x_i^{new} + r_1.BW) \qquad (1)$$

In continuous space

$$x_i^{new} = \{round(LB_i + r_2.(UB_i - LB_i)) \qquad (2)$$

In discrete space

Here x_i^{new} is the i-th element of new harmony solution vector; r_1 and r_2 are the random number; BW is an arbitrary distance band width; LB_iandUB_i are lower bound and upper bound of variable x_i^{new}.

If the fitness value of new harmony is better than the ex-solution vector, it will be included in the HM and the existing harmony solution vector is excluded from HM. This process runs iteratively till the maximum iteration. The steps in the procedure of the multi-objective harmony search algorithm (MOHSA) are as follows:

Step1. Initialize the problem and algorithm parameters.
Step2. Initialize the harmony memory (HM).
Step3. Improvise a new harmony from HM,
Step4. Update the HM.
Step5. Merge old and new harmony
Step6. Check the feasibility and perform fast non-dominated sorting approach.
Step7. Repeat Step 3 till 6 until maximum iteration.

5. Computational Result

To examine the effectiveness of the proposed algorithms, 9 test problems are considered. The real data obtained from Mehrabad International Airport are applied for testing the algorithms. Before solving the problems, the parameters of the algorithms must be configured. This will be explained in the next section.

5.1. Parameter configuration

For considering the effect of parameters on performance of the algorithms quickly, Taguchi method is used. Before calibration of the NSGA-II and MOHSA, some preliminary tests for finding appropriate parameters levels are run. For obtaining more accurate results for these algorithms, the following parameters are configured: Popsize, MaxIte, Pc, Pm for NSGA-II and HMS, HMCR, PAR, MaxIte for MOHSA.

These parameters and their levels are given in Table1 and Table 2.

Calculating all the test results using Taguchi method, mean rate of S/N for both algorithms have been obtained in Fig. 10 and Fig. 11. As can be seen in both Figure optimal levels for NSGA-II are A(3), B(1), C(3), D(3) and for MOHSA are A(3), B(1), C(3), D(3). These mean that the larger value for S/N is better. The final configuration of parameters is given in Table 4 and Table 5.

Table 1
Parameters and their levels

NSGA-II- Parameters

	A	B	C	D
Level	Popsize	MaxIte	Pc	Pm
1	10	20	0.8	0.01
2	20	50	0.9	0.05
3	50	100	0.95	0.1

Table 2
Parameters and their levels

MOHSA- Parameters

	A	B	C	D
Level	HMS	MaxIte	HMCR	PAR
1	10	50	0.1	0.1
2	50	100	0.5	0.5
3	100	200	0.9	0.9

Table 3
The orthogonal array L_9

Experiment	A	B	C	D
1	A(1)	B(1)	C(1)	D(2)
2	A(1)	B(2)	C(3)	D(3)
3	A(1)	B(3)	C(2)	D(3)
4	A(2)	B(1)	C(3)	D(3)
5	A(2)	B(2)	C(2)	D(1)
6	A(2)	B(3)	C(1)	D(2)
7	A(3)	B(1)	C(2)	D(2)
8	A(3)	B(2)	C(1)	D(3)
9	A(3)	B(3)	C(3)	D(1)

Table 4
Tuned value of the NSGA-II parameters

Parameters NSGA-II	Tuned value
Popsize	50
MaxIte	100
Pc	0.95
Pm	0.1

Table 5
Tuned value of the MOHSA parameters

Parameters MOHSA	Tuned value
HMS	100
MaxIte	50
HMCR	0.9
PAR	0.9

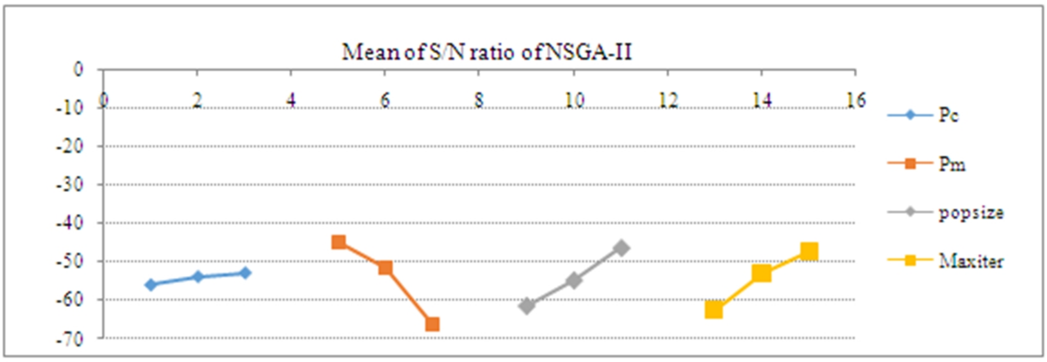


Fig. 10. Diagram on mean effect of the SN ratio of NSGA-II

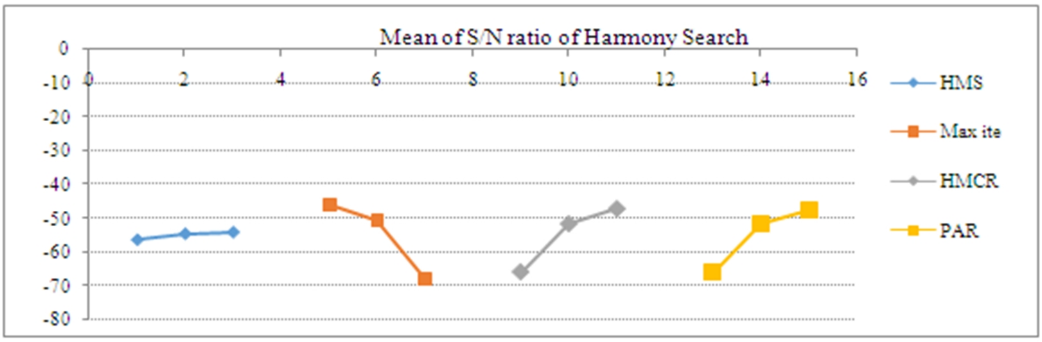


Fig. 11. Diagram on mean effect of the SN ratio of MOHSA

5.2. Performance criteria

For assessing and comparing the performance of the multi-objective algorithms we apply the following criteria:

1. Number of Pareto Solutions (NPS): This performance criterion is calculated by counting the number of non-dominated solutions obtained from each algorithm. The larger this number is, the better the performance of the algorithm will be.
2. Diversification Matrix (DM): This performance criterion, which is calculated in Equation 1, gives an indication on the diversity of solutions obtained from a given algorithm. The larger value of DM is,the better performance of the algorithm will be.

$$DM = \sqrt{(\max f_{1i} - \min f_{1i})^2 + (\max f_{2i} - \min f_{2i})^2} \quad (1)$$

3. Mean Ideal Distance (MID): The nearness or closeness between Pareto solutions and the ideal point is measured by this criterion, whose value is calculated

through the following equation 2. Where n is the number of non-dominated solutions and$c_i = \sqrt{f_{1i}^2 + f_{2i}^2}$.

4. The lower value of MID is, better performance of the algorithm will be.

$$MID = \frac{\sum_{i=1}^{n} c_i}{n} \qquad (2)$$

5. Spacing (SP): This measure was proposed (Schott 1995) as a way of measuring the distance variance of neighboring vectors in Pareto front known. This measure is defined as Equation (3):

$$SP = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n} (\bar{d} - d_i)^2}\ , \qquad (3)$$

$$d_i = \min_{i,j \neq} \left(\sum_{k=1}^{K} \left| f_i^k - f_j^k \right| \right)$$

Where *n* is the number of vector is the Pareto front found by the algorithm being evaluated, *K* is the number of objectives, $\bar{d}$ is the mean of all d_i. *SP*=0 indicates that all the foundnon-dominated solutions found are equidistantly spaced. The lower value of SP is, better performance of the algorithm will be.

5. Time (T): The total time for running the algorithm is used as the last measures. The shorter running time indicates the better performance of the algorithm.

5.3. Analysis of algorithm

In this section, the performance of algorithms NSGA-II and MOHSA have been evaluated. Both algorithms were executed using MATLAB R2012a software. They were executed on computer with 2.53 GHz processor and 4GB of RAM memory.

The algorithms were tested with real life data which was provided by Mehrabad International Airport for medium size problems. In all test problems, the average number of flights and gates respectively 89and 10 are considered.

The performance of these algorithms was assessed by solving 9 randomly generated problems. Table 6, 7 gives a comparison of performances of these two algorithms. The comparison which was made using five performance criteria as explained above are shown in Fig. 12 till 16.Fig. 12 showed Spacing criteria. As explained before, the lower value of SP is better performance of the algorithm. MOHSA in most cases generated the lower value of SP in compare to NSGA-II, therefore, as it is shown; the orange line is at the bottom of blue line. As second criteria, MOHSA algorithm generated lower value of MID. In most cases, the criteria of two algorithms are approximately same except one which was shown in Fig. 13. The larger value of DM criteria in MOHSA algorithm indicated the diversity of Pareto solution of the algorithm. The orange line in Fig. 14 is at the above of blue line in the most of cases. The counting number of Pareto solution of MOHSA in most cases is larger than NSGA-II algorithm. Hence, as it can be seen in Fig. 15, the orange line is at the above the blue line. Finally, in Fig. 16 the running time of NSGA-II is lower than MOHSA (the blue line is at the bottom of the orange line), therefore, NSGA-II algorithm indicates the better performance in compare to MOHSA.

Analysis of the results shows that MOHSA is superior to NSGA-II in most of four performance criteria, i.e., Spacing, MID, DM, and NPS (except Time). But, experimental results reveal that NSGA-II has better convergence near the true Pareto-optimal front as compared to MOHSA because the Pareto solution which was generated by NSGA-II is dominated the Pareto solution by MOHSA in 91% cases. Two random test problems are shown the Pareto solutionsofthe algorithms in Fig. 17, 18.It is shown that the Pareto solutions which were achieved from NSGA-II are lower than MOHSA. As we are going to minimize the objective functions, the minimum solutions are acceptable. Therefore, as can be seen in Figs,the blue dotsareat the bottom of the orange one. This indicates better convergence of NSGA-II near the true Pareto-optimal front as compared to MOHSA.

Table 6
Results of the qualitative performance measures for 9 test problems

Test Problem Number	Spacing		MID		DM	
	NSGA-II	MOHSA	NSGA-II	MOHSA	NSGA-II	MOHSA
1	1.0838	**1.0297**	1.0196	**1.0165**	83186	**108350**
2	**0.5783**	0.9237	1.0128	**1.0120**	50307	**128205**
3	1.1151	**0.350**	1.0121	**1.0102**	**42870**	30542
4	**0**	**0**	1.0145	**1.0143**	28533	**7378.3**
5	1.90	**0.746**	1.14	**1.0118**	12171	**102850**
6	1.15	**0.947**	1.0129	**1.0111**	96030	**143610**
7	1.128	**0**	1.0154	**1.0135**	62757	**76210**
8	1.90	**0.47**	1.0116	**1.0089**	132260	**137780**
9	1.0916	**0.833**	1.0102	**1.0102**	126230	**163670**

Table 7
Results of the qualitative performance measures for 9 test problems

Test Problem Number	NPS		Time (sec.)	
	NSGA-II	MOHSA	NSGA-II	MOHSA
1	8	**10**	1096.81	**1059.52**
2	4	**5**	1216.02	**1192.2**
3	**4**	3	1393.17	**1174.55**
4	**2**	**2**	1470	**1410.12**
5	3	**5**	**1423.61**	1503.82
6	3	**6**	**1513**	1551
7	**5**	2	**1460.47**	1519.98
8	4	**8**	**1614.67**	1693.90
9	**6**	**6**	**1718**	1974.83

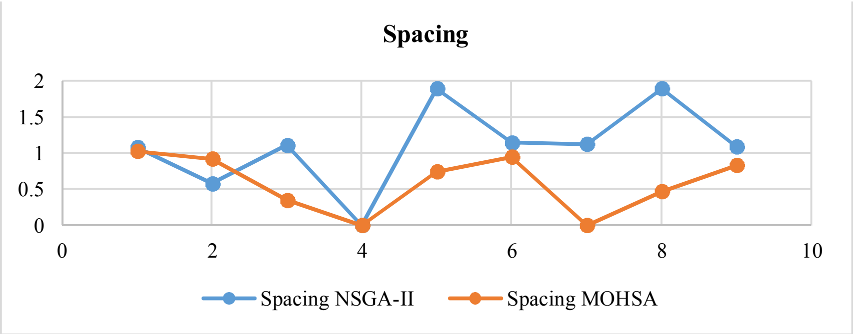


Fig.12. Comparison between Spacing performance criteria of NSGA-II and MOHSA

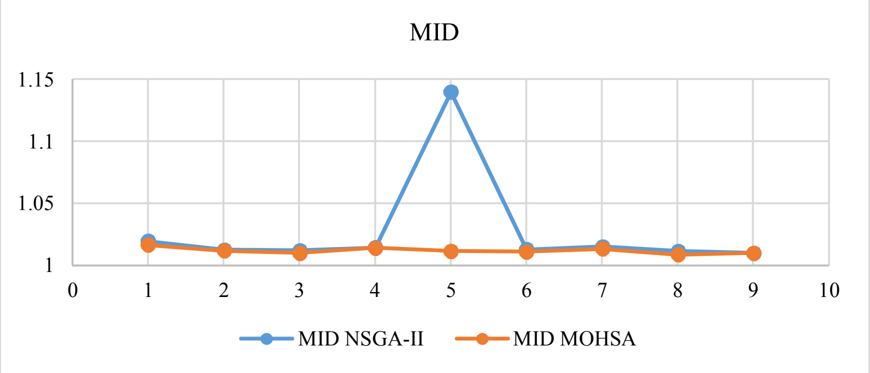


Fig. 13. Comparison between MID performance criteria of NSGA-II and MOHSA

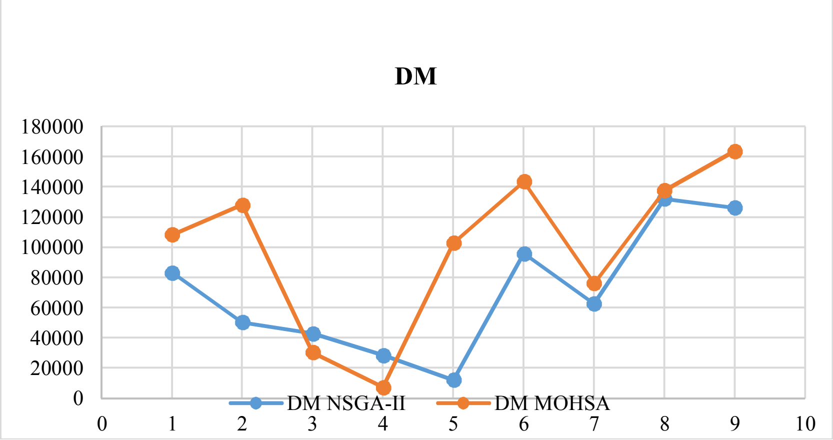


Fig. 14. Comparison between DM performance criteria of NSGA-II and MOHSA

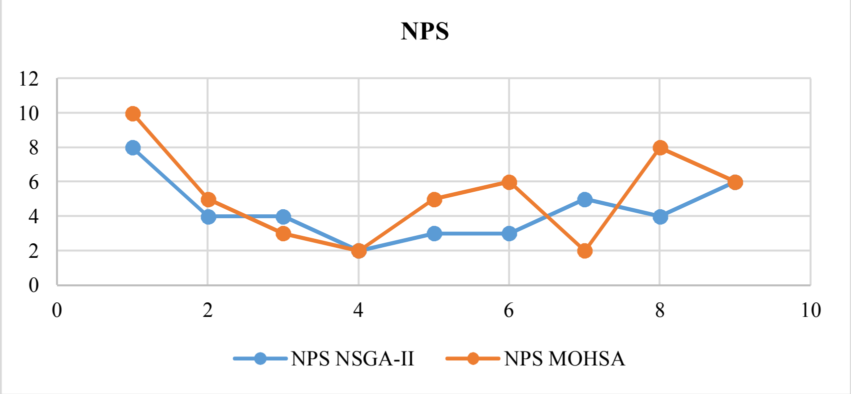


Fig. 15. Comparison between NPS performance criteria of NSGA-II and MOHSA

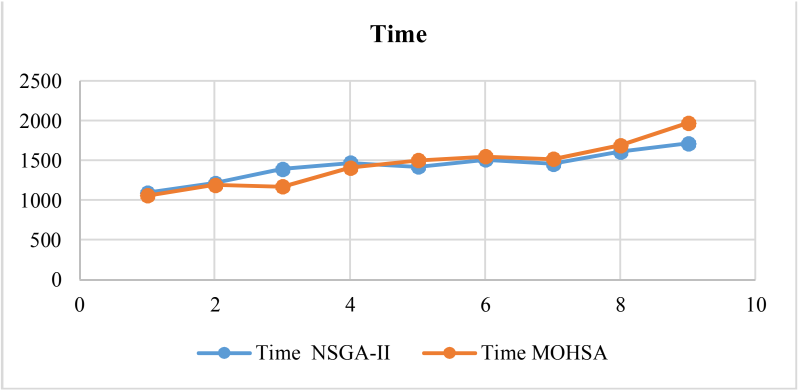


Fig 16. Comparison between Time performance criteria of NSGA-II and MOHSA

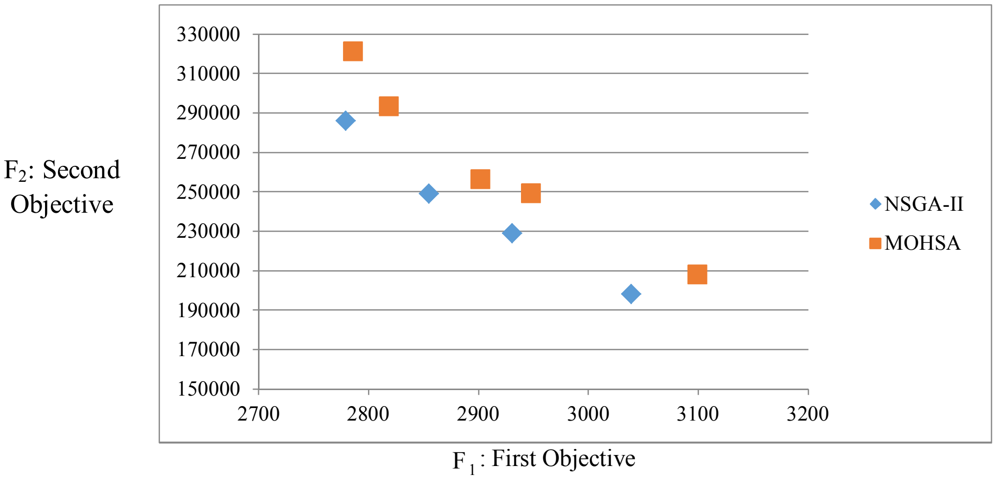


Fig. 17.Pareto solutions of NSGA-II and MOHSA test number 2

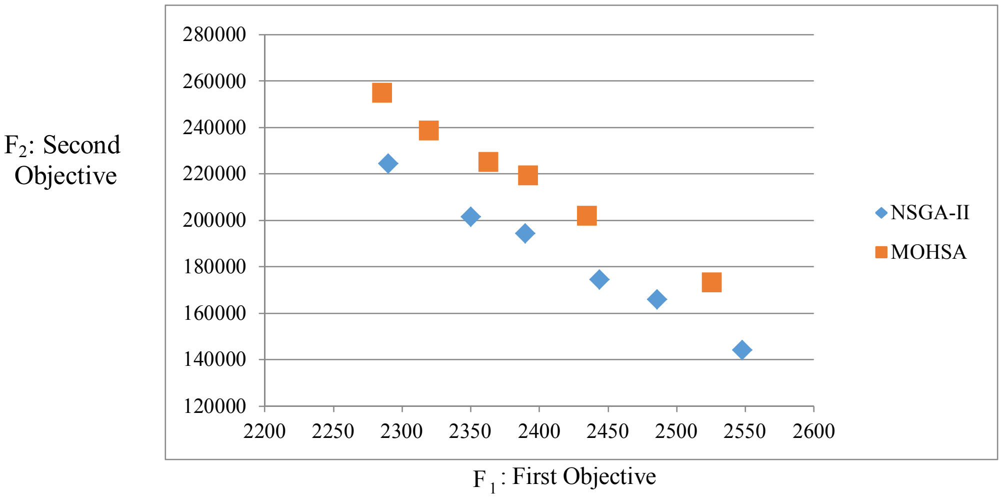


Fig. 18.Pareto solutions of NSGA-II and MOHSA test number 9

6. Conclusion and Future Research

In this research, we have developed a bi-objective decision making model for airport gate scheduling problem. The study contributes to the literature by considering controllable processing time concept for the scheduling model.The problem has been formulated as a Mixed Integer Programming (MIP) and Two important objectives were considered. The first objective function, which were consisted of four criteria, minimized the total cost of tardiness, earliness, delay and compression as well as expansion costs of job processing time. The second one minimized the passengers overcrowding on gate. By using airport gate processing time's concept, more flights could be serviced; it also could decrease crowd congestion in terminal's hall building and could increase passengers' satisfaction.

In addition, the gate assignment is NP-Hard problem; therefore, in order to solve the model, we applied two meta-heuristic methods, i.e.,NSGA-II and MOHSA,with real life data fromMehrabad International Airport for differentmedium size problems.

In order to assess the performance of these two algorithms, 9 different problems were solved. The performance of the proposed algorithms was studied in term of five performance measures, called Spacing, MID, DM, NPS and T. The computational results showed that in four out of five metrics (i.e. Spacing, MID, DMand NPS) MOHSA generated better Pareto solution compared to NSGA-II. However, theresults showed that NSGA-II had better convergence near the true Pareto-optimal front as compared to MOHSA.But in order to suggest one algorithm, we have to consider some priorities.

The first priority from airport authority's point of view is time. Because the problem is dynamic and it needs to modify the schedule during a day several times. Therefore, a method which is able to assign flights to gates in a short time is very valuable. Since the run time of both algorithms is similar; thus both algorithms have same priority. The second fact is the objectives. It means that which schedules (solutions) fulfill the goals (i.e., reduce the cost of earliness, tardiness, and so forth). But the results showed that NSGA-II generated better Pareto solution in compare to MOHSA; therefore, NSGA-II is a practical algorithm that can be chosen for this problem.

As research limitation, this study considered the only situation that all parameters are deterministic, however, as a direction for further research, it is recommended to consider uncertainty concept in parameter of the mathematical model as well as fuzzy logic to tackle the problem. Also, considering some realistic constraints and other objective functions, are worthy of study.

Furthermore, real world gate scheduling problems are usually very different from the mathematical models which are proposed in academia. Because it is not easy to list all differences between the real gate assignments problems at the airports and the theoretical models, as every real problem has its own particular idiosyncrasies.

For example, in our model we assumed that the number of gates and flights are exactly fix and static, however, in real problem at airports these numbers during operational days may change (due to flight cancelation, emergency flights, gate breakdown, and etc.). Thus, it needs to reassign the schedule several times. So considering this problem as on-line scheduling in future research is valuable.

Moreover, the model can be applied to cross-docking optimization in areas other than airports such as freight terminals, where material arrival times (via trucks, ships) can fluctuate.

Finally, applying the other efficient meta-heuristic algorithms will be suggested as future research. The results also can be compared with the ones of this study.

References

Babic O, Teodorovic D. & Tosic V. (1984). Aircraft stand assignment to minimize walking, Journal of Transportation Engineering, 110, 55–66.

Bouras, A., Ghaleb, M., Suryahatmaja, U.& Salem, A. (2014). The airport gate assignment problem: A survey, The scientific world journal, 2014, 1-27.

Das, N. (2009). The airport gate assignment problem with some practical constraints, Applied Management Science, 1, (.3),.315-323.

Diepen, G., Pieters, B., van den Akker, J.&Hoogeveen, J.(2012). “Finding a robust assignment of flights to gates at Amsterdam Airport Schiphol”, Journal of Scheduling, 15(6), 703-715.

Deb, K., Agrawal, S., Pratap, A.&Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, Parallel problem solving from nature PPSN VI, 1917, 849-858.

Ding H., Lim A., Rodrigues B.& Zhu Y. (2004). Aircraft and gate scheduling optimization at airports, In: Proceedings of the 37th Hawaii International Conference on System Sciences, Big Island, Hawaii, USA.

Ding H., Lim A., Rodrigues, B.& Zhu, Y. (2005).The over-constrained airport gate assignment problem, Computer & Operation Research, 32, 1867-1880.

Dorndorf, U. (2002). Project scheduling with time windows: from theory to applications, PhysicaVerlag.

Dorndorf, U., Drexl, A., Nikulin, Y. &Pesch, E., (2007), Flight gate scheduling state-of-the-art and recent development, The International Journal of Management Science,.35,326-334.

Dorndorf, U., Jaehn, F., Lin, C., Ma, H.&Pesch, E. (2007).Disruption management in flight gate scheduling, StatisticaNeerlandica, 61, 92–114.

Dorndorf, U., Jaehn, F.&Pesch, E. (2008).Modeling robust flight-gate scheduling as a clique partitioning problem. Transportation Science, 42, 292–301.

Dorndorf, U., Jaehn, F.&Pesch, E. (2012). Flight gate scheduling with respect to a reference schedule, Annals of Operations Research Vol. 194, PP.177–187.

Drexl, A., Nikulin, Y., (2008), Multicriteria airport gate assignment and pareto simulated annealing, IIE Transactions,.40, 385-397.

Geem, Z., Kim, J.&Loganathan, J. (2001).A new heuristic optimization algorithm: harmony search. J. Simulations 76, 60–68.

Guepet, J., Acuna-Agost, R., Briant, O., Gayon, J.P. (2015), European Journal of Operation Research, 246, 597-608.

Gurel S. &Akturk M.S. (2007).Scheduling parallel CNC machines with time/cost trade-off considerations. Computer Operation Research, 34, 2774–2789.

Haghani A. & Chen MC.(1998), Optimizing gate assignments at airport terminals, Transportation Research A, 32, 437–54.

Hidayatno, A., Moeis, A.&AryaSatyaDharam, G. (2015). Designing Gate Assignment Model to Find the Optimum Airport Gate Assignment Order, JurnalKeilmunandanAplikasiTeknikIndustri,.17, (1), DOI.10.9744/jti.17.1.1-6.

Jaehn, F. (2010). “Solving the flight gate assignment problem using dynamic programming”, ZeitschriftfürBetriebswirtschaft, 80, (10), 1027-1039.

Li K., Shi Y., Yang S.& Cheng BY. (2011). Parallel machine scheduling problem to minimize the makespan with resource dependent processing times, Application Software Computer, 11, 5551–5557.

Lim A., Rodrigues, B.& Zhu, Y.(2005). Airport gate scheduling with time windows, Artificial intelligence Review, 24,5-31.

Mahdavi, I., Kayvanfar, V.& Komaki, G.M.(2010). Minimizing total tardiness and earliness problem with controllable processing times using an effective heuristic, 40th International Conference on Computer and Industrial Engineering.

Mangoubi DFX. &Mathaisel RS. (1985). Optimizing gate assignments at airport terminals, Transportation Science,19, 173–88.

Marinelli, M., Dell’Orco, D. &Sassanelli, M.(2015). A metaheuristic approach to solve the flight gat assignment problem, Transportation Research Procedia, 5, 211-220.

Nikulin, Y.&Drexl, A. (2010). Theoretical aspects of multicriteria flight gate scheduling deterministic and fuzzy models.Journal of Scheduling, 13, 261–280.

Narciso, M.&Piera, M. (2015). Robust gate assignment procedures from an airport management perspective, Omega, 50, 82-95.

Palmisano, G., Marinelli, M., Dell’Orco, M.&Ottomanelli, M.. (2015),.Fusion of Two metaheuristic approaches to solve the flight gate assignment problem, Transportation Research Procedia, 10, 920-930.

Pintea C., Pop, P., Chira, C.&Dumitrescu, D. (2008).A hybrid ant-based system for gate Assignment Problem, Springer-Verlag Berlin Heidelberg,273-280.

Vickson,R.G. (1980). Two single machine sequencing problems involving controllable job processing times, AIIE Transactions, 12, 258–262.

Seshadri A.(2006). Multi-objective optimization using evolutionary algorithms (MOEA).http://www.mathworks.com/matlabcentral/fileexchange.

Schott, J. (1995). Fault Tolerant Design Using single and Multicriteria Genetic Algorithm Optimization.Master thesis, Department of Aeronautics and Astronautics, Institute of Technology, Cambridge, Massachusetts.

Shabtay D. & Steiner G. (2007).A survey of scheduling with controllable processing times.DiscretAppl Math, Vol.155, PP.1643–1666.

Shmoys, D.B. &Tardos, E. (1993).Scheduling unrelated machines with costs, In: Proceedings of the 4th Annual ACM-SIAM Symposium, Austin, TX, PP. 448–454.

Sørensen, J.F., Kragh, K.M., Sibbesen, O., Delcour, J., Goesaert, H., Svensson, B.& et al.(2004). Potential role of glycosidase inhibitors in industrial biotechnological applications.BiochimicaetBiophysicaActa (BBA)-Proteins & amp; Proteomics 1696(2),275–287.

Wang A, Huang Y, Taunk P, Magnin D.R., Ghosh K,& Robertson JG. (2003). Application of robotics to steady state enzyme kinetics: analysis of tight-binding inhibitors of dipeptidyl peptidase IV. AnalBiochem, 321, 157–166.

Wang, L., Mao, Y., Niu, Q.&Fei, M. (2011). A multi-objective binary harmony search algorithm, Advances in Swarm Intelligence, 6729,.74-81.

Xu J. & Bailey G. (2001).The airport gate assignment problem: Mathematical model and a tabu search algorithm. In: Proceedings of the 34th Hawaii International Conference on System Sciences, Island of Maui, Hawaii, USA.

Yan S. & Chang C-M. (1998). A network model for gate assignment, Journal of Advanced Transportation, 32(2), 176–89.

Yan S. &Huo C-M. (2001). Optimization of multiple objective gate assignments, Transportation Research Part A, Policy and Practice, 35, 413–32.

Zarandi, M., H.&Kayvanfar, V.(2015). A bi-objective identical parallel machine scheduling problem with controllable processing times: a just-in-time approach, 77, 545-563.

This article can be cited: khakzar Bafruei, M., khatibi, S. & Rahmani, M. (2018). A bi-objective airport gate scheduling with controllable processing times using Harmony Search and NSGA-II algorithmsJournal of Optimization in Industrial Engineering.11 (1), 77-90.

URL: http://www.qjie.ir/article_234.html
DOI: 10.22094/JOIE.2018.234