A NOVEL SCHEDULING APPROACH FOR PILGRIM FLIGHTS OPTIMIZATION PROBLEM

Mohd Khaled Shambour[1] & Esam A. Khan[2]*

[1,2]The Custodian of the Two Holy Mosques Institute for Hajj and Umrah Research, Umm Al-Qura University, Makkah, Saudi Arabia

Email: myshambour@uqu.edu.sa[1*] (corresponding author), eakhan@uqu.edu.sa[2]

DOI: https://doi.org/10.22452/mjcs.vol35no4.1

ABSTRACT

The main goal of airport administrations around the world is to facilitate the conduct of passenger services and reduce waiting time as much as possible. This can be achieved by regulating the flow of passengers at the various stages of the airport, including arrival and departure halls, passport checkpoints, luggage handling, and customs. This study focuses on improving the flow of passengers in the Hajj terminal at King Abdulaziz International Airport (KAIA) in the Kingdom of Saudi Arabia, as it is one of the most welcoming stations for travelers during the Hajj season and is the fourth largest passenger terminal in the world. Three different optimization algorithms are applied to improve the scheduling process of assigning the arrival flights to available airport gates, as well as the stages inside the various airport lounges and areas. These algorithms are genetic algorithm (GA), harmony search algorithm (HSA), and differential evolution algorithm (DEA). The results give a prior knowledge of how the whole passengers' arrival process and show the stages that are prone to congestion and cause process delay. Experimental performance results in terms of fitness value and convergence rate show that GA outperforms HSA and DEA when the population size is equal to 5, whereas DEA provides better performance compared to other algorithms when the population size is equal to 20 and 50. Moreover, the results show that the largest waiting time for passengers was in the arrival gate lounges due to the lack of allocated spaces in the passport areas, followed by the luggage area, then the passport control and customs areas, respectively.

***Keywords:* Flight scheduling, King Abdulaziz International Airport, Differential Evolution, Genetic algorithm, Harmony search algorithm**

1.0 INTRODUCTION

One of the major issues for airport agencies is the development of operating procedures for airport passenger terminals [1,2]. This results in improved passenger flow, movement, and mobility from one stage to another, increased passenger satisfaction and experience, and increased safety and security [1].

In recent years (2017-2019), before the Covid-19 pandemic, more than one and a half million travelers have been coming to the Kingdom of Saudi Arabia through airports to perform the rituals of Hajj, the fifth pillar of Islam. They start arriving about a month before the start of Hajj rituals. Two airports are dedicated to the reception of pilgrims: King Abdulaziz International Airport (KAIA) in Jeddah and Prince Muhammad Bin Abdulaziz International Airport (PMIA) in Medina [3].

In KIAI, there exist four passenger terminals: the new terminal, the northern terminal, the southern terminal, and Hajj terminal. The Hajj terminal is the fourth largest passenger terminal in the world [4]. It is the main welcoming port for travelers who come to perform Hajj and Umrah rituals. Hajj terminal contains ten gates on the first floor that serve aircraft passengers using airline bridges and four other gates on the ground floor for passengers arriving by buses whose planes were parked in a remote area [1].

According to Idris [5] & Ghamdi [6], Hajj terminal can accommodate 50,000 arriving passengers for up to 18 hours and 80,000 departing passengers for up to 36 hours. Also, the official release by the general authority of statistics [7] stated that the capacity of halls in Hajj terminal for arrivals and departures are 3,800 and 3,500 passengers per hour, respectively. Furthermore, the Saudi vision 2030 aims at increasing Umrah performers from 8 million to 30 million by 2030 [8].

Therefore, a challenging goal is to increase the capacity of Hajj terminal to the maximum possible. One of the most important issues to achieve this goal is the process of distributing aircraft to airport gates, known as the Gate Assignment Problem (GAP). This process needs to take into consideration all airport stages, including arrivals to gate lounges, passing through passport control, luggage collection, and customs check.

In this paper, an intelligent flight scheduling technique capable of improving passenger flow in Hajj terminal through efficient use of available resources is developed. To achieve this, three algorithms have been developed and adapted to optimize the flow of passengers at various airport stages: genetic algorithm (GA), harmony search algorithm (HSA), and differential evolution algorithm (DEA). These algorithms are among the popular optimization algorithms that have exceptional performance on a variety of optimization problems [9,10]. The study identified ten constraints, equally divided between hard and soft constraints; assuming that all airport gates can receive aircraft of any size and that the walking time for passengers is included in the waiting time at all airport stages.

The performances of the applied algorithms in terms of fitness value and convergence rate were compared to observe the best algorithm performance. The results showed that GA has better performance with a population size of 5, while DEA is better for population sizes of 20 and 50. Based on the results of this study, a number of recommendations are provided that can be practically applied by airport management in order to improve the efficiency of airport operations which will result in more passenger satisfaction.

This paper is organized as follows. Section 2 provides a literature review of related work. In Section 3, the flight scheduling problem is described and formulated. The proposed approach and experimental results are presented in Section 4 and Section 5, respectively. Finally, the conclusion of the paper is given in Section 6.

2.0 RELATED WORK

One of the most important issues in managing airport operations is the process of distributing aircraft to the airport gates, since faults in this process may cause aircraft delays or some accidents. The distribution process is one of the most important issues that airport operation managers face on a daily basis, as those in charge of airport management operations must find the best gate at which a plane can stop, taking into account some limitations such as reducing the distance between gates for connecting flights, minimizing the total walking distance for passengers, and others. Several researchers have studied the issue of determining parking spots for incoming aircraft at airport gates, known as the Gate Assignment Problem (GAP). Numerous heuristic techniques have been successfully used in literature to address a variety of challenging optimization problems [11,12,13,14].

Marinelli et al. [15] proposed a genetic approach called Bee Colony Optimization based on biogeography to find the best gateway for an aircraft to stop. This approach was obtained through a combination of biogeography-based and bee colony optimization algorithms, and the researchers depended on building a practical solution called (a feasible solution) and then improving the quality of the obtained solution through the work of the search algorithm.

Xu and Bailey [16] worked to reduce passengers' walking distances while considering connected flights using the Tabu search algorithm. To solve the problem, the researchers used a two-stage algorithm. In the first stage, the Greedy strategy was used to reduce the number of unscheduled flights at the gates to a minimum. In the second phase, an improved search algorithm was developed to reduce the total flight link time.

Ding et al. [17] and Al-Sultan et al. [18] conducted studies entitled “Aircraft and gate scheduling optimization at airports” with the aim of reducing the number of unscheduled flights at the gates as well as reducing the total travel distances within the airport. The researchers followed the same approach used in the study of Xu and Bailey [16], where the Greedy algorithm was used to build a practical solution called (a feasible solution). The optimized search algorithm (Tabu search) was used to search for better improvements to the current solution using a function called (Interval Exchange Move), which allows flexibility in searching for good solutions, especially if the flight schedules are dense.

Al-Sultan [19] proposed a simulation method to find the optimum number of gates required for a specified percentage of the total number of flights for aircraft not assigned to a specific gate. In addition, the proposed method predicts the arrival rate of flights and then performs simulations to schedule flights for one week.

In their research, Hu and Paolo [20] used Genetic Algorithm to solve the problem of assigning aircraft to airport gates. Instead of representing the chromosomes by the actual locations of planes on gates, the relative locations were used. The researchers used a uniform crossover method to obtain a good balance between diversity and convergence in the evolutionary process.

Hidayatno et al. [21] applied the simulated annealing algorithm to find the optimum allocation of aircraft to gates and to reduce the number of aircraft not assigned to gates in Soekarno-Hatta International Airport in Indonesia.

Cheng and Ho [22] used the Tabu Search algorithm with path relinking to address the problem of mapping aircraft to gates. The researchers used data from real flights from Incheon International Airport (ICN) to test the proposed algorithm.

In another paper, Drexl and Nikulin [23] used the simulated annealing algorithm to reduce the number of flights not assigned to gates and to minimize the total walking distances or inter-flight interconnection times, in addition, to maximize the use of all airport gates.

Lim and Wang [24] used stochastic programming and converted it to binary programming in order to reduce the number of aircraft not assigned to gates and to publish the time schedule of gates as soon as possible. Bouras [25] conducted a survey study for airport gate assignment problem that included both theoretical and practical sides, with a description of the mathematical formulas and search methods used, such as the exact algorithm, heuristic algorithms and metaheuristic algorithms.

Ghazouani et al. [26] used the genetic algorithm (GA) for the scheduling process, where the researchers used integers in the gene coding process. The index of each gene includes the flight number, and its value represents the gate number to which the flight will be allocated. The "Roulette Wheel" method was used in the process of selecting the next generation of chromosomes. For comparing the performance of Tabu Search and Simulated Annealing algorithms, Aktel et al. [27] applied both algorithms to solve the airport gate mapping problem. The objective of their work was to reduce the number of unassigned flights and the total travel distances of passengers. The results showed that the simulated annealing algorithm achieved the best performance on average, whereas the proposed Tabu Search algorithm achieved better performance for large-size problems.

Elm Company [28] conducted a study to evaluate the guests’ journey from the ports to the residence during the Hajj season of 2017. The study included an assessment of the current situation at that time and presented the problems in the arrival and departure periods at Hajj Terminal of King Abdulaziz International Airport. The study also included developing solutions to be applied in the 2018 season.

Previous studies show the importance of finding practical and applicable solutions to assign incoming aircraft to the available gates in the best possible way in order to provide maximum comfort to traveling passengers. Considering the increasing number of pilgrims, which is one of the main objectives of Saudi vision 2030, it is required to increase the capacity of airports, and hence comes the motivation of this study. To the best of our knowledge, this study is the first to discuss how to improve the scheduling of pilgrim flights to the terminal gates of King Abdulaziz Airport in Jeddah.

3.0 PROBLEM DESCRIPTION AND FORMULATION

Optimizing airport resource scheduling, respecting several constraints, is the primary process for developing operating procedures for airport passenger terminals. The following subsections present detailed descriptions of the stages of Hajj terminal at King Abdulaziz Airport in Jeddah, in addition to the problem attributes, and mathematical formulation of the problem.

3.1 Stages of Hajj Terminal at KAIA

Incoming passengers must go through five main stages in the Hajj terminal, as shown in Figure 1. There are four stages inside the arrival terminal (indoor), including the gate lounges, passport control, luggage handling, and customs checkpoints. The fifth stage is located outdoor the terminal in an open area called the Plaza. More details on the stages are given below.

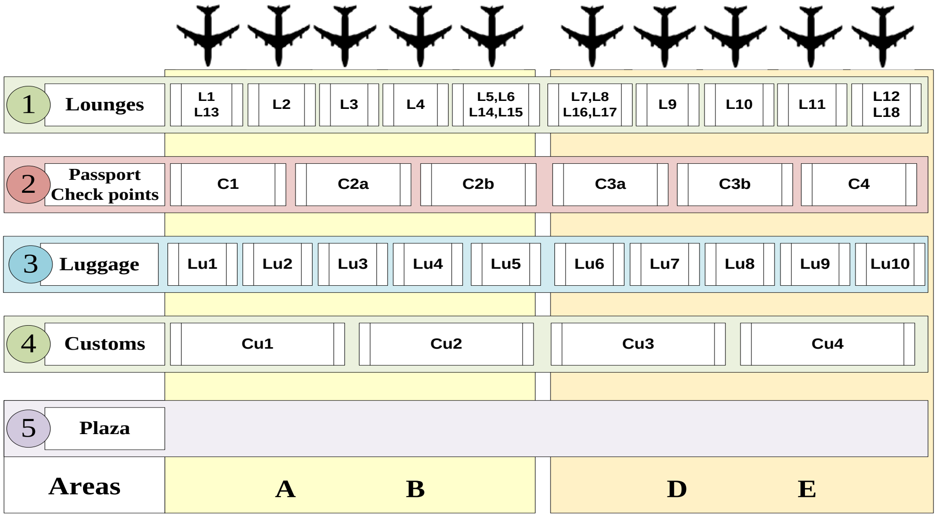

Fig. 1: Main stages of the Hajj Terminal

3.1.1 Gate lounges

Gate lounges are the first stage in which aircraft passengers arrive at the terminal. According to KAIA [29] and GACA [4], the total number of lounge gates is eighteen gates divided over two floors. Twelve lounge gates are located on the first floor to serve passengers who arrive via jetways, and another six lounges are on the ground floor for passengers arriving via buses. At this stage, a medical checkup of pilgrims is performed, and medical vaccinations are provided if necessary [30]. Table 1 displays the assumed area size and maximum passenger seat capacity per lounge, noting that capacity is calculated by giving $1.5m^2$ of waiting area for each passenger.

Table 1: Characteristics of the arrival lounges

	Lounge ID	Space (m2)	Capacity (passenger seats)	Area
First Floor	L1	800	533	A/B
	L2	500	333	A/B
	L3	500	333	A/B
	L4	500	333	A/B
	L5	500	333	A/B
	L6	600	400	A/B
	L7	600	400	D/E
	L8	500	333	D/E
	L9	500	333	D/E
	L10	500	333	D/E
	L11	500	333	D/E
	L12	800	533	D/E
Ground Floor	L13	800	533	A/B
	L14	500	333	A/B
	L15	500	333	A/B
	L16	500	333	D/E
	L17	500	333	D/E
	L18	800	533	D/E
	Total	10400	6928	

3.1.2 Passport checkpoints

In this stage, passport personnel verify passport data and take vital identifiers (fingerprints and iris scans). There are 132 checking counters divided into six sections [1,30]. Table 2 shows the characteristics of the passport stage.

Table 1: Characteristics of the passport check points

Counter Section ID	Number of Counters	Area
C1	18	A/B
C2a	32	A/B
C2b	16	A/B
C3a	32	D/E
C3b	16	D/E
C4	18	D/E
Total	**132**	

3.1.3 Luggage handling

In this phase, luggage is collected from two sections located in A/B and D/E areas, each containing five conveyor belts [6,30]. Table 3 shows the main specifications of luggage halls.

Table 3: Characteristics of the luggage halls

Luggage Section ID	Conveyor belts	Area
Lu1	5	A/B
Lu2	5	D/E
Total	**10**	

3.1.4 Custom checkpoints

At this stage, luggage is checked by x-ray machines and passengers have to disclose whether they are carrying cash or valuables. There are 16 checking points divided into 4 sections, each containing 4 x-ray checking points [1, 30]. Table 4 provides the main features of custom sections.

Table 2: Characteristics of the customs areas

Custom Section ID	x-ray devices	Area
Cu1	4	A
Cu2	4	B
Cu3	4	D
Cu4	4	E
Total	**16**	

3.1.5 Plaza area

The Plaza is the outer area of the Hajj terminal, covered with tent-shaped roofs, with a total area of 140 thousand square meters [4]. It includes a variety of services such as banks, shops, restaurants, waiting areas, and bus stations. At this point, passengers prepare to travel to the holy city of Makkah by bus to begin their ritual journey.

3.2 Problem Formulation

The problem constraints are mathematically formulated in this section, including ten constraints equally divided between hard and soft constraints. Hard constraints must be satisfied to guarantee the feasibility solution, whereas satisfying more soft constraints could improve the solution quality [31]. The problem attributes and their abbreviations are defined as shown in Table 5.

Table 3: Problem Attributes

Attribute	Description	Abbreviation
Att.1	Flight of passengers	F
Att.2	Stage	S
Att.3	Gate	G
Att.4	Passport counter section	P
Att.5	Luggage	L
Att.6	Customs section	C
Att.7	Plaza	Z
Att.8	Time	T
Att.9	Number of passengers	n
Att.10	Gate capacity	y
Att.11	Minimum Processing time	$minPT$
Att.12	Minimum number of workers	$minWR$
Att.13	Minimum number of passport counters	$minPC$
Att.14	Minimum number of customs inspection devices	$minCD$

Following is a mathematical representation of the problem:
Assignment (A) is a function of resources (S, F), where the constraints of the problem can be mathematically represented as follows:

H1: All flights must be assigned to the airport gates.

$$A_F^{S_G} = A_{F_i}^{S_{G_j}} \qquad \forall j \in G, \forall i \in F$$

H2: Each flight is assigned to only one airport gate.

$$A_{F_i}^{S_{G_j}} \neq A_{F_i}^{S_{G_k}} \qquad \forall j,k \in G, \forall i \in F$$

H3: A particular airport gate cannot handle more than one flight at the same time.

$$A_{F_{i1}}^{S_{G_j^{t1}}} \neq A_{F_{i2}}^{S_{G_j^{t1}}} \qquad \forall t1 \in T, \forall j \in G, i1 \neq i2, \forall\, i1, i2 \in F$$

H4: A flight must be assigned to a gate with a capacity that can accommodate the number of passengers of that flight.

$$F_{i_n} \leq \quad G_{j_y} \qquad \forall\, A_{F_i}^{S_{G_j}}, \forall i \in F, \forall j \in G$$

H5: Avoid clashes between stages that could occur during passenger transportation.

$$A_{F_{i1}}^{S_{X_m}^{t1}} \neq A_{F_{i2}}^{S_{X_m}^{t1}} \quad where\ X_m \in \{G_{1,2,..,18}\} or\ \{P_{1,2,..,6}\} or\ \{L_{1,2,..,10}\} or\ \{C_{1,2,..,4}\}, t1 \in T,$$
$$i1 \neq i2, \forall\, i1, i2 \in F$$

S1: Eliminate the waiting time for passengers during transportation between different stages.

$$WT(F_i, S_X) \cong 0 \qquad \forall\, A_{F_i}^{S_X}, \forall i \in F, where\ \ X \in G\ or\ P\ or\ L\ or\ C,$$
$$WT(S_X, F_i)\ is\ the\ waiting\ time\ of\ Flight\ i\ at\ stage\ X$$

S2: Minimize the processing time to complete medical examination and passport control procedures as much as possible.

$$PT\left(S_{Y_{F_i}}\right) \cong minPT_Y \qquad \forall\, A_{F_i}^{S_Y}, \forall i \in F, where\ \ Y \in G\ or\ P\ ,$$
$$PT(S_{Y_{F_i}})\ is\ the\ processing\ time\ of\ Flight\ i\ at\ stage\ Y$$

S3: Minimize the number of medical examination workers as much as possible.

$$WR\left(S_{G_j}\right) \cong minWR, \forall\ j\ \in G,$$
$$where\ WR\left(S_{G_j}\right)\ \ is\ the\ number\ of\ workers\ at\ Gate\ j$$

S4: Minimize the number of passport counters as much as possible.

$$PC\left(S_{P_k}\right) \cong minPC, \quad \forall\, k \in P, where$$

$$PC\left(S_{P_k}\right)\ is\ the\ number\ of\ passport\ counters\ at\ passport\ section\ k$$

S5: Minimize the number of customs inspection devices as much as possible.

$$CD\left(S_{C_v}\right) \cong minCD, \qquad \forall\, v \in C, where$$

$$CD\left(S_{C_v}\right)\ is\ \ the\ number\ of customs\ inspection\ devices\ at\ custom\ section\ v$$

Different weights were applied to the constraints violations before identifying the optimal, depending on the type of constraints and their importance in meeting the produced solution [32]. Therefore, each violation of a hard constraint is assigned a value of 1,000, while each violation of a soft constraint is assigned a value of 100, with the exception of the first two constraints, S1 and S2, which were assigned values of 1 and 10 based on our empirical experience. Table 6 provides the violation cost (weight) for each constraint.

Table 4: Weights of constraint violations

Constraint	H1	H2	H3	H4	H5	S1	S2	S3	S4	S5
weight	1,000	1,000	1,000	1,000	1,000	1	10	100	100	100

The problem of developing operating procedures for airport passenger terminals is a minimization problem, as the goal is to obtain an optimal solution through the efficient utilization of available resources while respecting several hard and soft constraints. Therefore, the objective function is designed to minimize the costs resulting from constraint violations of the final solution. The declaration of the objective function can be formulated as follows:

$$Min(\sum_{n=1}^{5}\left(n_{v_{h_n}} \times 1000\right) + n_{v_{s_1}} + n_{v_{s_2}} * 10 + \sum_{m=3}^{5} n_{v_{s_m}} * 100\ )$$

Where n_{v_h} and n_{v_s} refer to the number of violations of hard and soft constraints in the produced solution, respectively.

4.0 PROPOSED APPROACH

4.1 Scheduling Algorithm Design

To design an effective method that optimizes the scheduling process for all phases of an airport, it requires a deep understanding of all procedure parameters during the stages that passengers pass through, from when the plane stops until they leave the airport. Therefore, the main process that must be ensured in scheduling is that flight passengers at a particular airport stage can only move to the next stage after making sure that the next stage can receive them.

The following are the main airport stages for the scheduling problem:

Stage 1. The stage of passenger transportation from the airplane to the gate lounge and the start of health procedures.

Stage 2. The stage of passenger transportation from the gate lounge to the passport counters.

Stage 3. The stage of passenger transportation from passport counters to the luggage handling area.

Stage 4. The stage of passenger transportation from the luggage handling area to the customs area.

Stage 5. The stage of passenger transportation from the customs area to the Plaza area.

Algorithm 1 displays the main steps used in the scheduling process, while Algorithms 2 through 6 provide more details for each stage step mentioned above.

Algorithm 1: The main scheduling steps of the airport scheduling problem

FlightsSchdAlg(PT_{S_G}, $WR_{S_{g_1}}$, $WR_{S_{g_2}}$, $WR_{S_{g_3}}$, $WR_{S_{g_4}}$, $WR_{S_{g_5}}$, $WR_{S_{g_6}}$, $WR_{S_{g_7}}$,
$WR_{S_{g_8}}$, $WR_{S_{g_9}}$, $WR_{S_{g_{10}}}$, $WR_{S_{g_{11}}}$,
$WR_{S_{g_{12}}}$, $WR_{S_{g_{13}}}$, $WR_{S_{g_{14}}}$, $WR_{S_{g_{15}}}$, $WR_{S_{g_{16}}}$, $WR_{S_{g_{17}}}$, $WR_{S_{g_{18}}}$, PT_{S_P}, $P_{S_{p_1}}$, $P_{S_{p_2}}$, $P_{S_{p_3}}$, $P_{S_{p_4}}$, $P_{S_{p_5}}$, $P_{S_{p_6}}$, CD_{S_C})
Begin
[F]= Ø // *initialized list of flights*
[L]= {*1, 2,…, no of gates*} // *initialized list of 18 gate lounges*
[P] = Ø // *initialized list of Passport Counters * initialized list of 6 sections*
[L] = Ø // *initialized list of Luggage area * initialized list of 10 conveyor belts*
[C] = Ø // *initialized list of Customs * initialized list of 4 sections*
[Z] = Ø // *initialized list of Plaza area*
For ts=1 to 288 // *A day is divided to 288 timeslots (ts), each represents 5 minutes*
/*update Gate Lounges data at timeslot ts*/
$\boldsymbol{AircraftToGateLounges}(F, G, ts, PT_{S_G}, WR_{S_{g_1}}, WR_{S_{g_2}}, WR_{S_{g_3}}, WR_{S_{g_4}}, WR_{S_{g_5}}, WR_{S_{g_6}}, WR_{S_{g_7}},$
$WR_{S_{g_8}}, WR_{S_{g_9}}, WR_{S_{g_{10}}}, WR_{S_{g_{11}}}, WR_{S_{g_{12}}}, WR_{S_{g_{13}}}, WR_{S_{g_{14}}}, WR_{S_{g_{15}}}, WR_{S_{g_{16}}}, WR_{S_{g_{17}}}, WR_{S_{g_{18}}});$
/*update Gate Lounge and Passport Counters data at timeslot ts*/
$\boldsymbol{GateLoungeToPass}(G, P, ts, PT_{S_P}, P_{S_{p_1}}, P_{S_{p_2}}, P_{S_{p_3}}, P_{S_{p_4}}, P_{S_{p_5}}, P_{S_{p_6}});$
/*update Passport Counters and Luggage data at timeslot ts*/
$\boldsymbol{PassToLuggage}(P, L, ts);$
/*update Luggage and Customs data at timeslot ts*/
$\boldsymbol{LuggageToCustom}(L, C, ts, CD_{S_{c_1}}, CD_{S_{c_2}}, CD_{S_{c_3}}, CD_{S_{c_4}});$
/* update Customs and Plaza data at timeslot ts*/
$\boldsymbol{CustomsToPlaza}(C, Z, ts);$
End For// *end for ts*
End

Algorithm 2: AircraftToLounges Step Algorithm

$\boldsymbol{AircraftToGateLounges}(F, G, ts, PT_{S_G}, WR_{S_{g_1}}, WR_{S_{g_2}}, WR_{S_{g_3}}, WR_{S_{g_4}}, WR_{S_{g_5}}, WR_{S_{g_6}}, WR_{S_{g_7}},$
$WR_{S_{g_8}}, WR_{S_{g_9}}, WR_{S_{g_{10}}}, WR_{S_{g_{11}}}, WR_{S_{g_{12}}}, WR_{S_{g_{13}}}, WR_{S_{g_{14}}}, WR_{S_{g_{15}}}, WR_{S_{g_{16}}}, WR_{S_{g_{17}}}, WR_{S_{g_{18}}})$
Begin
/* *return number of flights at time ts*/*
NoF = ReturnNumOfFlights (ts)
IF NoF > 0
/* *return number of passengers of all flights at time ts*/*
[$N_{F_f^{ts}}$]= *ReturnNumOfpassengers* (F_f^{ts}) // *f*ϵ *{F}*/*
[$N_{F_f^{ts}}$]=*Sort* ([$N_{F_f^{ts}}$]) //*sort flights Dec according to no of passengers N*
For *i*=1 to NoF
/**assign a flight f to best fit Gate Lounge l*/*
G_g^{ts}=*AssignG*(G^{ts}, $N_{F_i^{ts}}$, PT_{S_G}, $WR_{S_{g_1}}$, $WR_{S_{g_2}}$, $WR_{S_{g_3}}$, $WR_{S_{g_4}}$, $WR_{S_{g_5}}$, $WR_{S_{g_6}}$, $WR_{S_{g_7}}$, $WR_{S_{g_8}}$,
$WR_{S_{g_9}}$, $WR_{S_{g_{10}}}$, $WR_{S_{g_{11}}}$, $WR_{S_{g_{12}}}$, $WR_{S_{g_{13}}}$, $WR_{S_{g_{14}}}$, $WR_{S_{g_{15}}}$, $WR_{S_{g_{16}}}$, $WR_{S_{g_{17}}}$, $WR_{S_{g_{18}}}$)
[F]= *UpdateAssignedFlights* (F_i^{ts}) //*update the Flight contents*
[G]= *UpdateAssignedLounge* (G_g^{ts}) /* *update the GateLounge contents according to Expected Time to leave lounge= Time In(ts) + no of passengers * Time to deal for each passenger (PT_{S_G}) / number of workers in current lounge (G_g)*/*
End For // *end for i*
End IF // *end if NoF>0*
End

Algorithm 3: LoungeToPass Step Algorithm

$\boldsymbol{GateLoungeToPass}(G, P, ts, PT_{S_P}, P_{S_{p_1}}, P_{S_{p_2}}, P_{S_{p_3}}, P_{S_{p_4}}, P_{S_{p_5}}, P_{S_{p_6}})$
Begin
 / return number of available lounges at time ts*/*
 NoG= ReturnNumOfGateLounges (ts)
 IF NoG > 0
  $[G_g^{ts}]$=*Sort*($[G_g^{ts}]$) *// sort available GateLounges Dec according to their size g*ϵ *(1,* NoG*)*
  For *i*=1 to NoG
   $P_p^{ts} = AssignP(G_i^{ts}, p_1, p_2, p_3, p_4, p_5, p_6, PT_{S_P})$ *//assign a lounge l to best fit PassCounter p,p*ϵ*{P}*
   [G]= *UpdateAssignedLounges*(G_i^{ts}) *// update the GateLounges contents*
   [P]= *UpdateAssignedPassCounter*(P_p^{ts}) */* update the Counters contents according to Expected Time to leave Counters=Time In(ts) + no of passengers * PT_{S_P}/ no. of Counters */*
  End For // *end for i*
 End IF // *end if NoG>0*
End

Algorithm 4: PassToLuggage Step Algorithm

$\boldsymbol{PassToLuggage}(P, L, ts, LTD)$ *<u>// LTD</u> Is the luggage time duration*
Begin
 / return number of available Passport Counters at time ts*/*
 NoP= ReturnNumOfPassCounters(ts)
 IF NoP > 0
  $[P_p^{ts}]$=*Sort*($[P_p^{ts}]$) *//sort Passport Counters Dec according to their number of Passport counters*
  For *i*=1 to NoP
   $L_l^{ts} = AssignL(P_i^{ts}, l_1, l_2, l_3, l_4, l_5, l_6, l_7, l_8, l_9, l_{10})$ */*assign a passport counter i to best fit conveyor belt in Luggage l, l*ϵ *{L}*/*
   [P]= *UpdateAssignedCounters*(P_i^{ts}) *// update Passport Counters contents*
   [L]= *UpdateAssignedLuggage*(L_l^{ts}) */* update the Luggage contents according to Expected Time to leave L= Time In(ts) + LTD */*
  End For // *end for i*
 End IF // *end if NoP>0*
End

Algorithm 5: LuggageToCustoms Step Algorithm

$\boldsymbol{LuggageToCustoms}(L, C, ts)$
Begin
 / return number of available conveyor belts in Luggage area at time ts*/*
 NoL= ReturnNumOfConveyorBelts (ts)
 IF NoL > 0
  For *i*=1 to NoL
   $C_c^{ts} = AssigC(L_i^{ts}, c_1, c_2, c_3, c_4)$ *//assign a Luggage* $\mathrm{L_i}$ *to best fit Customs C, c*ϵ *{C}*
   [L]= *UpdateAssignedLuggage*(L_i^{ts}) *// update Luggage contents*
   [C]= *UpdateAssignedCustoms* (C_c^{ts}) */*update the Customs contents according to Expected Time to leave Customs area= Time In(ts) + no of passengers * 15 seconds/NoOfXRayMachine;*/*
  End For // *end for i*
 End IF // *end if NoL>0*
End

Algorithm 6: CustomsToPlaza Step Algorithm

```
CustomsToPlaza(C, Z, ts)
Begin
    /* return number of available customs devices at time ts*/
    NoC= ReturnNumOf Customs (ts)  //return number of Customs areas
    IF NoC > 0
        For i=1 to NoC
            Z_z^ts=AssignZ(C_i^ts, z) //assign a Custom_i to Plaza z, zϵ { Z }
            [C]= UpdateAssignedCustoms(C_i^ts) // update Customs contents
            [Z]= UpdateAssignedPlaza(Z_z^ts) // update the Plaza contents
        End For // end for i
     End IF // end if NoC>0
End
```

4.2 Optimization of problem parameters using optimization algorithms

Three algorithms were adapted to solve the addressed scheduling problem, including genetic algorithm (GA), harmony search algorithm (HSA), and differential evolution algorithm (DEA). These algorithms are types of evolutionary algorithm (EA) proposed by Holland [33], Geem et al. [34], and Storn and Price [35], respectively. EA relies on its work on a biological evolutionary mechanism inspired by nature, where the mechanism of work depends mainly on maintaining a set of solutions during the search process, so that these solutions participate in the production of one or more new solutions in each search iteration. After that, the new solutions are replaced by the existing ones according to the quality of the solutions produced. However, EA's search mechanisms generally follow these key steps:

- Selection step: Generation members (solutions) are selected for the process of forming a new generation.
- Recombination step: Formation of new members based on the individuals selected.
- Mutation step: Introduce random changes in the structures of the new generation.
- Evaluation step: Members of the new generation are given values that reflect their competence.
- Updating step: The best performance among existing and new members, in terms of evaluation cost, is kept for further rounds of evolution.

The presented methods are utilized to improve the problem's scheduling processes by locating parameter values that result in the lowest solution cost in terms of the objective function. Table 7 lists the problem parameters and their range values. The parameters are determined based on the current situation of the airport stages described in Section 3. For parameter selection, all algorithms employed a random selection method in their search procedure.

Table 5: The problem parameters and their domain values

	Parameter Name	Number of parameters	Abbr.	Range
1	Processing time to complete medical examination procedures	1	PT_{S_G}	[5,20]
2	Processing time to complete passport control procedures	1	PT_{S_P}	[90,300]
3	Medical examination workers in every gate lounge	18	$NR_{S_{g1-g18}}$	[1,50]
4	Number of passport counters -section 1	1	$P_{S_{p_1}}$	[22, 50]
5	Number of passport counters -section 2	1	$P_{S_{p_2}}$	[32, 50]
6	Number of passport counters -section 3	1	$P_{S_{p_3}}$	[16, 50]
7	Number of passport counters -section 4	1	$P_{S_{p_4}}$	[32, 50]
8	Number of passport counters -section 5	1	$P_{S_{p_5}}$	[16, 50]
9	Number of passport counters -section 6	1	$P_{S_{p_6}}$	[18, 50]
10	Number of custom inspection devices in every section	4	$CD_{S_{c1-c4}}$	[4,10]

The solution representation for the airport scheduling problem is shown in Equation 1. The solution representation is divided into airport stages $(1,..,i)$ and flights $(1,..,j)$, with *i* and *j* denoting the number of airport stages and flights, respectively. The population memory of all algorithms (e.g. Harmony memory) is a matrix of solutions with *n* population sizes, as stated in Equation 2, where $f(Sol^n)$ is the objective function value of solution Sol^n. It is worth mentioning that all algorithms use the greedy algorithm to build their initial solutions.

$$Sol = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,j} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ x_{i,1} & x_{i,2} & \cdots & x_{i,j} \end{bmatrix} \quad \text{Equation (1)}$$

$$\text{Pop} = \left[\begin{array}{c|c} Sol^1 & f(Sol^1) \\ Sol^2 & f(Sol^2) \\ \vdots & \vdots \\ Sol^n & f(Sol^n) \end{array} \right] \quad \text{Equation (2)}$$

The algorithms employ their mutation operators to generate neighbor solutions $(Sol^{\prime})$: the mutation operator for GA (Equation 3), the pitch adjustment operator for HSA (Equation 4), and the DE/rand/1 operator for DEA (Equation 5). Algorithms 7-9 demonstrate the proposed GA, HSA, and DEA pseudocodes including the major steps.

$$Sol_p^{\acute{a}} = rand(MinMaxParameters\ (min_p\ , max_p)) \quad \text{Equation (3)}$$

where $MinMaxParameters$ denotes the domain values of the problem parameter *p*.

$$Sol_p^{\acute{a}} = Sol_p^{a} \pm rand(0,1).\left(Sol_p^{b} - Sol_p^{c}\right) \quad \text{Equation (4)}$$

$$Sol_p^{\acute{a}} = Sol_p^{a} + F.\left(Sol_p^{v} - Sol_p^{w}\right) \quad \text{Equation (5)}$$

Where $b, c, v, w \in (1, HMS\ )$, b $\neq$ c, $v \neq w$; F is a scaling factor used to control the mutation process.

Algorithm 7: Pseudocode of the proposed GA

Step 1: Set the algorithm and problem parameters.
Including: number of population (*nPop*), $Probability\ of\ crossover(P_c)$, $Probability\ of\ mutation(P_m)$, $Number\ of\ Parameters(NP)$, $Number\ \ of\ \ Iterations(NI)$,
$MinMaxParameters = [(min_{PT_{S_G}}, max_{PT_{S_G}}), (min_{PT_{S_p}}, max_{PT_{S_p}}), (min_{WR_{S_{g1-g18}}}, max_{WR_{S_{g1-g18}}}),$
$(min_{P_{S_{P_1-P_6}}}, max_{P_{S_{P_1-P_6}}}), (min_{CD_{S_{c1-c4}}}, max_{CD_{S_{c1-c4}}})]$, *Pop*=[ ], *NewPop*=[ ]
Step 2: Generate n number of initial solutions
For *i=1 to* nPop
Sol^i=*FlightsSchdAlg(*$PT^i_{S_G}, WR^i_{S_{g_1}}, WR^i_{S_{g_2}}, WR^i_{S_{g_3}}, WR^i_{S_{g_4}}, WR^i_{S_{g_5}}, WR^i_{S_{g_6}}, WR^i_{S_{g_7}}, WR^i_{S_{g_8}}, WR^i_{S_{g_9}}, WR^i_{S_{g_{10}}},$
$WR^i_{S_{g_{11}}}, WR^i_{S_{g_{12}}}, WR^i_{S_{g_{13}}}, WR^i_{S_{g_{14}}}, WR^i_{S_{g_{15}}}, WR^i_{S_{g_{16}}}, WR^i_{S_{g_{17}}}, WR^i_{S_{g_{18}}} PT^i_{S_P}, P^i_{S_{p_1}}, P^i_{S_2}, P^i_{S_3}, P^i_{S_4},$
$P^i_{S_{p_5}}, P^i_{S_{p_6}}, CD^i_{S_C}$)
Evaluation(Sol^i)
$Pop = Pop\ \cup\ Sol^i$ *// adding the initial solution to Pop*
End For
Step 3: Evolution Process
$\boldsymbol{Mpool = RWSelection(Pop, n)} /*\ select\ n\ random\ solutions\ from\ Pop, n \in (1, nPop)\ using$
$RouletteWheelSelection\ method\ to\ build\ mating\ pool\ (Mpool)\ /*\ Selection\ */$
For n=1 to NI
$[Sol^x, Sol^y] = RndSelection(Mpool\ )\ //\ randomly\ select\ Sol^x, Sol^y from\ Mpool$
For i=1 to NP
$\boldsymbol{IF}\ (rnd(0,1) \leq p_c)\ /*\ Recombination/Crossover\ */$
$Sol^{\acute{x},i} = Sol^{y,i} \quad Sol^{\acute{y},i} = Sol^{x,i}$

Algorithm 7: Continued.

Else
$Sol^{\acute{x},i} = Sol^{x,i}$
$Sol^{\acute{y},i} = Sol^{y,i}$
End IF
IF $(rnd(0,1) \leq p_m)$ /* *Mutation* */
$Sol^{\acute{x},i} = rnd(MinMaxParameters\ (min_i\ , max_i))//\ rnd\ is\ a\ random\ generator$
Else
$Sol^{\acute{x},i} = Sol^{x,i}$
End IF
End For
Evaluation$(Sol^{\acute{x}}, Sol^{\acute{y}})$
$NewPop = NewPop\ \cup\ Sol^{\acute{x}}\ \cup\ Sol^{\acute{y}}$
End For
Step 4: Elective Process /* keep best solutions in Pop */
$Pop = Bestof(Pop, NewPop)$/* *return best solutions of both Pop and NewPop based on fitness value* */
Step 5: Stopping improvisation if the termination criterion is met; otherwise go to Step 3.

Algorithm 8: Pseudocode of the proposed HSA

Step 1: Set the algorithm and problem parameters.
Including: *Harmony Memory* (*HM*), *Harmony Memory Size*(*HMS*), *Harmony Memory Consideration Rate* (*HMCR*), *PitchAdjusment Rate*(*PAR*), *Number of Parameters*(*NP*), $MinMaxParameters(min_{PT_{S_G}}, max_{PT_{S_G}}), (min_{PT_{S_p}}, max_{PT_{S_p}}), (min_{WR_{S_{g1-g18}}}, max_{WR_{S_{g1-g18}}}),$
$(min_{P_{S_{P_1-P_6}}}, max_{P_{S_{P_1-P_6}}}), (min_{CD_{S_{c1-c4}}}, max_{CD_{S_{c1-c4}}})], Number\ of\ Iterations(NI), HM = [\ ]$
Step 2: Generate *n* number of initial solutions (Harmonies)
For i=1 to *HMS*
H^i=*FlightsSchdAlg*$(PT^i_{S_G}, WR^i_{S_{g_1}}, WR^i_{S_{g_2}}, WR^i_{S_{g_3}}, WR^i_{S_{g_4}}, WR^i_{S_{g_5}}, WR^i_{S_{g_6}}, WR^i_{S_{g_7}}, WR^i_{S_{g_8}}, WR^i_{S_{g_9}}, WR^i_{S_{g_{10}}},$
$WR^i_{S_{g_{11}}}, WR^i_{S_{g_{12}}}, WR^i_{S_{g_{13}}}, WR^i_{S_{g_{14}}}, WR^i_{S_{g_{15}}}, WR^i_{S_{g_{16}}}, WR^i_{S_{g_{17}}}, WR^i_{S_{g_{18}}} PT^i_{S_P}, P^i_{S_{p_1}}, P^i_{S_2}, P^i_{S_3},$
$P^i_{S_4}, P^i_{S_{p_5}}, P^i_{S_{p_6}}, CD^i_{S_C})$
Evaluation(H^i)
$HM = HM\ \cup\ H^i$*// adding the initial solution to HM*
End For
Step 3: Improvisation Process
For n=1 to NI
For i=1 to NP
IF $(rnd(0,1) \leq HMCR)$ /* *Memory Consideration* */
$H^x = RndSelection(HM)//randomly\ selects\ a\ harmony\ solution\ from\ HM, x \in (1, HMS)$
$H'^{,i} = H^{x,i}$
IF $(rnd(0,1) \leq PAR)$/* *Pitch Adjustment* */
Generate two random integers $y, z \in (1, HMS), y \neq z$
$H'^{,i} = H'^{,i} \pm rnd(0,1).(H^{y,i} - H^{z,i})$
End IF
Else /* *Random Consideration* */
$H'^{,i} = rnd\big(MinMaxParameters\ (min_i)\ , MinMaxParameters\ (max_i)\big)$
End IF
End For
Evaluation(H')
End For
Step 4: Elective Process /* keep best solutions in HM */
$UpdateHM(H', worst(H))$/* *update HM with the best of new harmony*(H') *and worst harmony in HM based on their fitness values*/*
Step 5: Stopping improvisation if the termination criterion is met; otherwise go to Step 3.

Algorithm 9: Pseudocode of the proposed DEA

Step 1: Set the algorithm and problem parameters.
Including $nPop$, P_c, F , $Number\ of\ Parameters(NP), MinMaxParameters = [(min_{PT_{S_G}}, max_{PT_{S_G}}),$
$(min_{PT_{S_p}}, max_{PT_{S_p}}), (min_{WR_{S_{g1-g18}}}, max_{WR_{S_{g1-g18}}}), (min_{P_{S_{P_1-P_6}}}, max_{P_{S_{P_1-P_6}}}), (min_{CD_{S_{c1-c4}}}, max_{CD_{S_{c1-c4}}})]$

Step 2: Generate n number of initial solutions
For i=1 to *nPop*
Sol^i=*FlightsSchdAlg(*$PT^i_{S_G}, WR^i_{S_{g_1}}, WR^i_{S_{g_2}}, WR^i_{S_{g_3}}, WR^i_{S_{g_4}}, WR^i_{S_{g_5}}, WR^i_{S_{g_6}}, WR^i_{S_{g_7}}, WR^i_{S_{g_8}}, WR^i_{S_{g_9}},$
$WR^i_{S_{g_{10}}}, WR^i_{S_{g_{11}}}, WR^i_{S_{g_{12}}}, WR^i_{S_{g_{13}}}, WR^i_{S_{g_{14}}}, WR^i_{S_{g_{15}}}, WR^i_{S_{g_{16}}}, WR^i_{S_{g_{17}}}, WR^i_{S_{g_{18}}}, PT^i_{S_P}, P^i_{S_{p_1}}, P^i_{S_2}, P^i_{S_3}, P^i_{S_4},$
$P^i_{S_{p_5}}, P^i_{S_{p_6}}, CD^i_{S_C}$)
Evaluation(Sol^i)
End For

Step 3: Evolution Process
$Selection()/*\ select\ three\ random\ numbers\ a, b, c \in (1, nPop)\ , a \neq b \neq c$/*Selection*/
For *i=1 to NP*
$\boldsymbol{IF}\ (rand(0,1)\ \leq p_c)\ /*\ Recombination/Crossover\ */$
$Sol^{á,i} = Sol^{a,i} + F.(Sol^{b,i} - Sol^{c,i})\quad /*\ Mutation\ */$
$\boldsymbol{Else}$
$Sol^{á,i} = Sol^{a,i}$
$\boldsymbol{End\ IF}$
$\boldsymbol{End\ For}$

Step 4: Update the best solution
$\boldsymbol{IF}\ \ f(Sol^{á})\ better\ than\ \ f\ (Sol^{a})$
$Sol^a = Sol^{á}$
End IF

Step 5: Stopping improvisation if the termination criterion is met; otherwise go to Step 3.

5.0 EXPERIMENTS AND RESULTS

This section presents an evaluation of the performance of GA, HSA, and DEA in solving the airport scheduling problem. Two performance criteria were used to compare the performance of these algorithms, namely fitness values and convergence rates.

5.1 Experimental design

The paper assumes that the total number of passengers on the peak day was 61,788, arriving in 260 flights, at an average of 10.8 flights per hour. The number of passengers per flight is randomly generated according to [1] and ranges from 100 to 500 passengers with an average of 237.6 passengers per flight as shown in Figure 2. Moreover, the time required, in minutes, to collect luggage for every flight is randomly generated between 9 to 31 according to Aljamal et al. [36].

All algorithms applied the same problem parameters and were evaluated on the same performance measures. Table 8 presents the experimental settings used by the comparison algorithms. Eighteen experiments, each running 30 times, were performed in each algorithm. Different population sizes of 5, 20, and 50 were examined to observe their effect on the algorithms' performances as previously used in [37, 38]. The maximum number of cost evaluations was set to 5,000 in all experiments. It should be noted that the number of cost evaluations used by GA is greater than that of HSA and DEA in every search iteration. This is because GA generates a number of solutions in every search iteration according to the size of the mating pool, which is not used in both HSA and DEA as only one solution is generated per iteration. The conducted experiments were coded using Matlab 2020b on Windows 10 64-bit on Intel 3.4GHz processor with 32 GB of RAM.

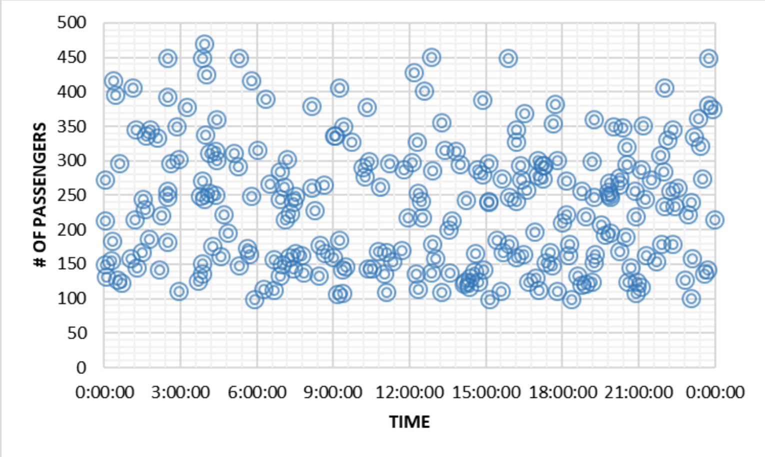

Fig 2: Distribution of arrival flights for passengers throughout the day

Table 6: Experimental Settings

Settings	GA		HSA		DEA	
	Pc	Pm	HMCR	PAR	Pc	F
S1	0.5	0.3	0.5	0.3	0.5	0.3
S2	0.5	0.5	0.5	0.5	0.5	0.5
S3	0.7	0.3	0.7	0.3	0.7	0.3
S4	0.7	0.5	0.7	0.5	0.7	0.5
S5	0.9	0.3	0.9	0.3	0.9	0.3
S6	0.9	0.5	0.9	0.5	0.9	0.5

5.2 Experimental results

This section presents the performance comparisons between the studied algorithms according to the different values of population size (i.e. number of population (nPop) and Harmony Memory Size (HMS)).

5.2.1 Experimental results with a population size of 5

Table 9 displays the statistical results for fitness values of the comparison algorithms, including mean, standard deviation, best, and worst**.** The fitness value results show that GA achieved superior performance compared to HSA and DEA in all experiments, and the best GA results were observed through the use of the S3 parameter settings. DEA came second in achieving the best mean fitness results, and HSA came last. The best convergence results were also obtained by GA in 83.3% of the total experiments. Figures 3-8 show the best convergence performance achieved by the comparison algorithms in different parameter settings (i.e. S1, S2, …, S6).

Table 7: Statistical fitness value results of 30 experimental trials with a population size of 5

		S1	S2	S3	S4	S5	S6
GA	**Mean**	**25950.3**	**27823.4**	**25916.1**	**27765.0**	**25952.2**	**27717.9**
	Std.	486.9	406.8	447.7	501.9	398.1	376.5
	Best	25323	26797	25161	26586	25374	26983
	Worst	26997	28555	26932	28829	26992	28424
HSA	**Mean**	39449.1	38907.8	38828.3	38868.7	38430.4	38422.7
	Std.	1007.8	1088.7	1371.4	1268.6	1653.3	1129.9
	Best	37419	37239	36247	35352	34874	36488
	Worst	41482	41583	41394	41730	40798	41084
DEA	**Mean**	36069.5	31516.3	38909.6	34955.4	42304.2	38844.7
	Std.	2993.7	2753.1	3110.0	3326.6	3295.6	2906.9
	Best	30445	26478	32589	29169	36709	33739
	Worst	41558	37239	44545	42441	51460	44195

**Note: The best mean results are highlighted by bold font*

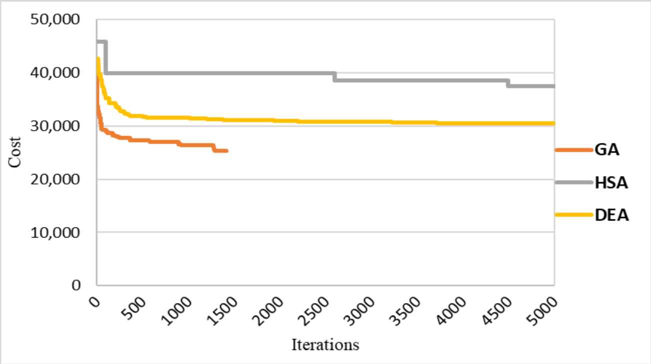

Fig 3: Convergence rates of the compared algorithms for S1 parameter settings (population size = 5)

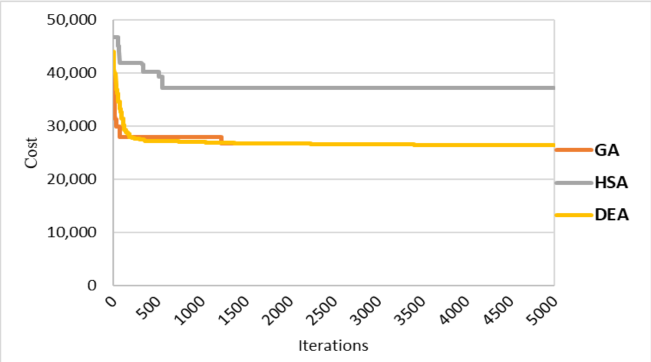

Fig 4: Convergence rates of the compared algorithms for S2 parameter settings (population size = 5)

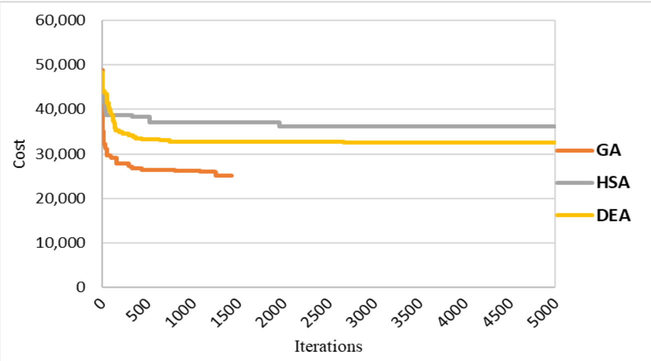

Fig 5: Convergence rates of the compared algorithms for S3 parameter settings (population size = 5)

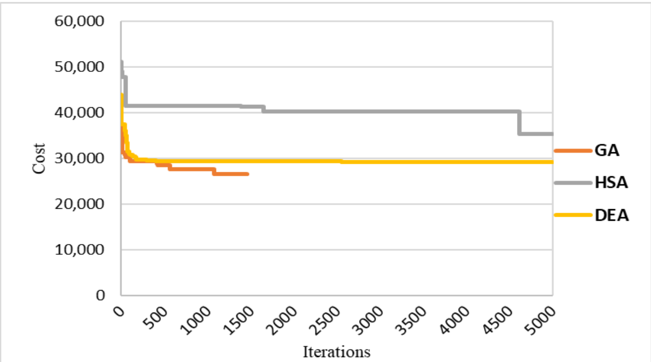

Fig 6: Convergence rates of the compared algorithms for S4 parameter settings (population size = 5)

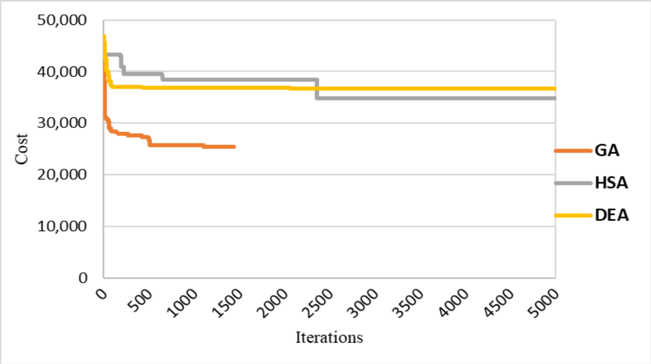

Figure 7: Convergence rates of the compared algorithms for S5 parameter settings (population size = 5)

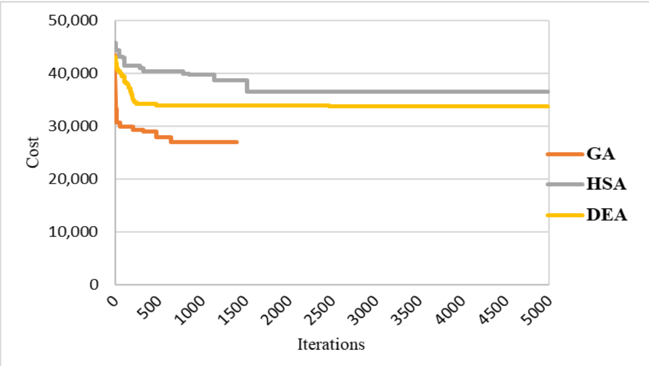

Figure 8: Convergence rates of the compared algorithms for S6 parameter settings (population size = 5)

Moreover, an ANOVA statistical method is used to verify whether there is a significant difference between the results achieved by the compared algorithms or not. The null hypothesis (h_0) states that there is no difference between the mean fitness values of compared algorithms, whereas the alternative hypothesis (h_1) rejects the null one, such that:

- h_0: μ1 = μ2 = μ3, where μ is the mean.
- h_1: At least one of the means is different.

The ANOVA results, provided in Table 10, show that there are significant differences between the mean fitness values as the p-value is lower than the significance level of 0.05 and F-values are always greater than the F-critical value. Thus, there is sufficient evidence to reject the null hypothesis h_0 and accept the alternative hypothesis h_1, which concludes that not all mean fitness values are equal to each other and a significant difference between the fitness mean results are found.

Table 8: ANOVA descriptive statistics with population size of 5

Settings	F-critical	F-value	*P*-value
S1	3.10	434.7	5.14E-46
S2	3.10	321.04	6.89E-41
S3	3.10	428.2	9.28E-46
S4	3.10	220.8	8.16E-35
S5	3.10	477.8	1.20E-47
S6	3.10	362.6	6.26E-43

5.2.2 Experimental results with a population size of 20

Table 11 displays the statistical results for fitness values of the compared algorithms. Superior performance results were obtained by DEA compared to GA and HSA for both the best mean and best convergence results in all experiments followed by GA and then HSA. Also, results show that the best performance of DEA was observed through the use of the S2 parameter settings. Figures 9-14 show the best convergence performance achieved by the comparison algorithms in different parameter settings.

Table 9: Statistical fitness value results of 30 experimental trials with a population size of 20

		S1	S2	S3	S4	S5	S6
GA	**Mean**	26196.5	28092.3	26154.1	27870.0	26106.1	27902.1
	Std.	353.3	406.1	369.1	463.9	452.2	360.9
	Best	25493	26968	25213	26649	25478	26698
	Worst	26807	28845	26784	28639	27099	28346
HSA	**Mean**	39609.1	39152.2	39737.7	39635.7	39731.8	39570.0
	Std.	1044.6	884.5	1362.9	1106.7	1133.8	1060.2
	Best	37740	37563	34933	36835	37707	37429
	Worst	41597	40608	41710	42292	41845	41315
DEA	**Mean**	**22469.9**	**22406.6**	**22684.4**	**22532.8**	**22628.3**	**22690.5**
	Std.	335.9	63.9	354.1	134.3	370.5	574.6
	Best	22179	22291	22366	22319	22397	22370
	Worst	24158	22546	23834	23027	24382	24449

** Note: The best mean results are highlighted by bold font*

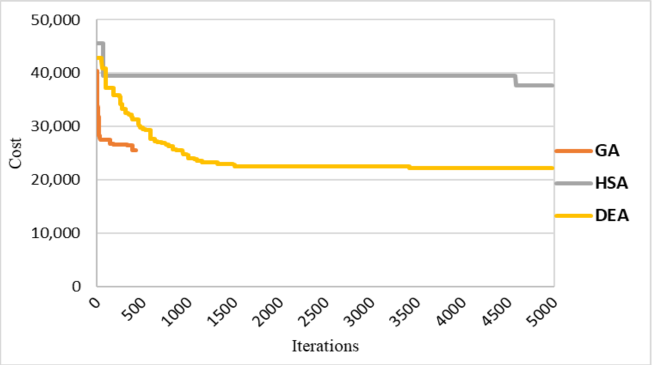

Fig 9: Convergence rates of the compared algorithms for S1 parameter settings

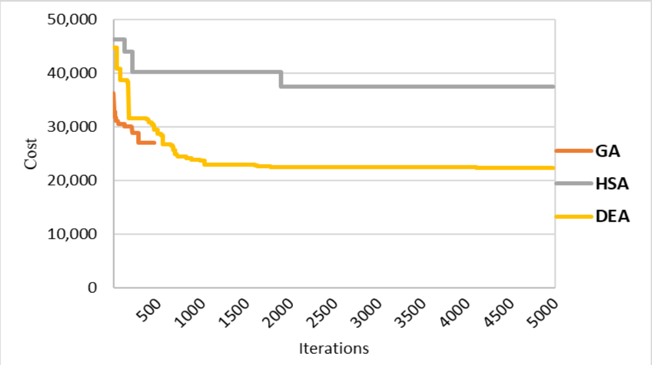

Fig 10: Convergence rates of the compared algorithms for S2 parameter settings

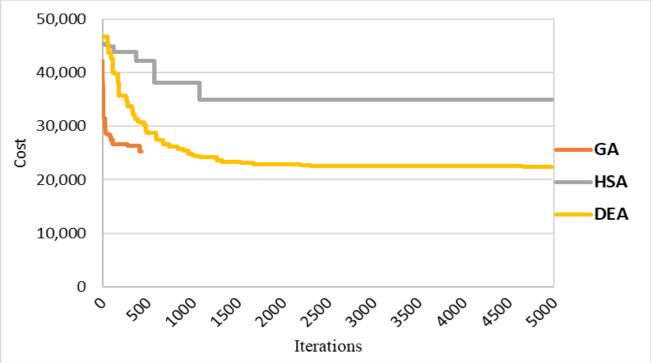

Fig 11: Convergence rates of the compared algorithms for S3 parameter settings

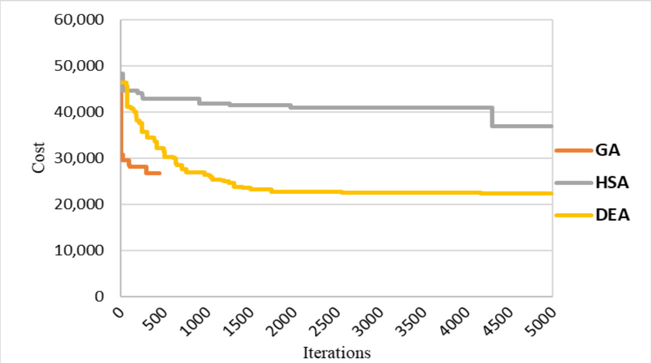

Fig 12: Convergence rates of the compared algorithms for S4 parameter settings

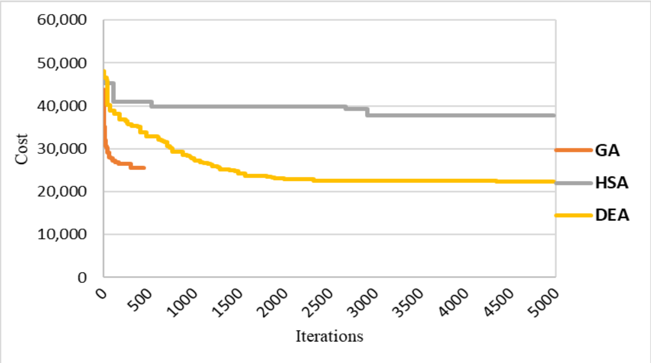

Fig 13: Convergence rates of the compared algorithms for S5 parameter settings

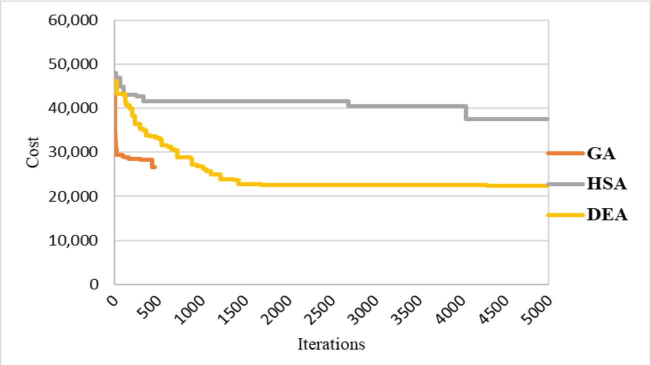

Fig 14: Convergence rates of the compared algorithms for S6 parameter settings

Furthermore, an ANOVA statistical method is used to verify whether there is a significant difference between the results achieved by the compared algorithm or not, such that the null hypothesis (h_0) and the alternative hypothesis (h_1) are given as follows:

- h_0: μ1 = μ2 = μ3, where μ is the mean.
- h_1: At least one of the means is different.

The ANOVA results, provided in Table 12, show that there are significant differences between the mean fitness values as the p-value is lower than the significance level of 0.05 and F-values are always greater than the F-critical value. Thus, there is sufficient evidence to reject the null hypothesis h_0 and accept the alternative hypothesis h_1, which concludes that not all mean fitness values are equal to each other and a significant difference between the mean fitness results is found.

Table 10: ANOVA descriptive statistics with population size of 20

Settings	F-critical	F-value	*P*-value
S1	3.10	5502.9	2.57E-92
S2	3.10	6859.4	1.88E-96
S3	3.10	3449.9	1.38E-83
S4	3.10	4726.5	1.81E-89
S5	3.10	2105.6	2.09E-74
S6	3.10	1379.8	1.27E-66

5.2.3 Experimental results with a population size of 50

Table 13 displays the statistical results for fitness values of the comparison algorithms. Again, DEA shows superior performance compared to GA and HSA for both the best mean and best convergence results in all experiments followed by GA and HSA, respectively. Besides, results show that the best performance of DEA was observed through the use of the S1 parameter settings. Figures 15-20 show the best convergence performance achieved by the comparison algorithms in different parameter settings.

Table 11: Statistical fitness value results of 30 experimental trials with a population size of 50

		S1	S2	S3	S4	S5	S6
GA	Mean	26772.4	28407.1	27086.2	28707.3	26997.2	28750.3
	Std.	363.8	433.9	519.0	428.6	366.4	432.3
	Best	25526	27420	25991	27795	26204	27603
	Worst	27261	29564	27978	29357	27641	29333
HSA	Mean	39983.4	39395.4	39836.9	39953.1	39524.7	39312.4
	Std.	999.7	1439.7	822.3	918.3	1188.2	1405.4
	Best	37368	34977	38096	37075	37345	34851
	Worst	41952	42001	41213	41564	41294	41023
DEA	Mean	**22507.3**	**22615.4**	**22617.4**	**22866.1**	**22554.1**	**22778.7**
	Std.	76.3	84.4	99.9	181.5	87.6	173.1
	Best	22307	22381	22452	22618	22380	22494
	Worst	22645	22741	22867	23369	22728	23275

** Note: The best mean results are highlighted by bold font*

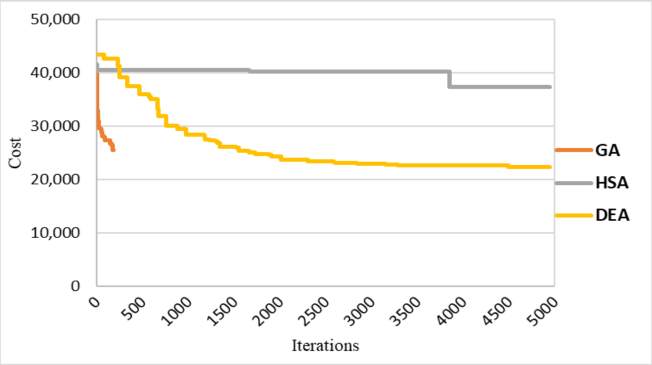

Fig 15: Convergence rates of the compared algorithms for S1 parameter settings

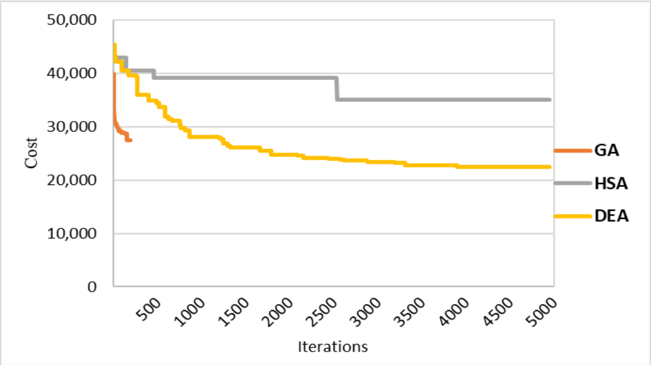

Fig 16: Convergence rates of the compared algorithms for S2 parameter settings

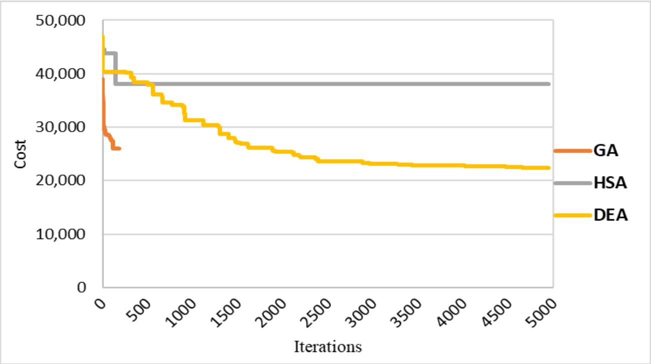

Fig 17: Convergence rates of the compared algorithms for S3 parameter settings

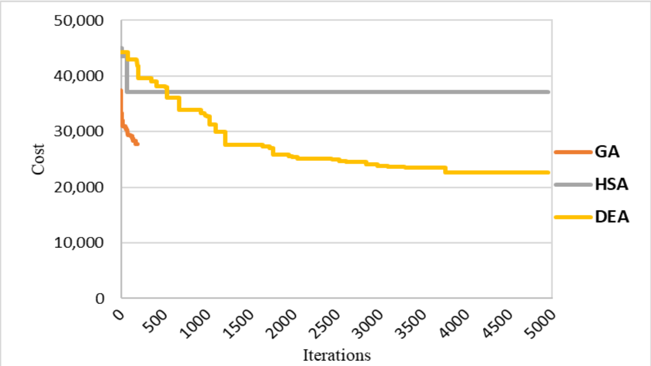

Fig 18: Convergence rates of the compared algorithms for S4 parameter settings

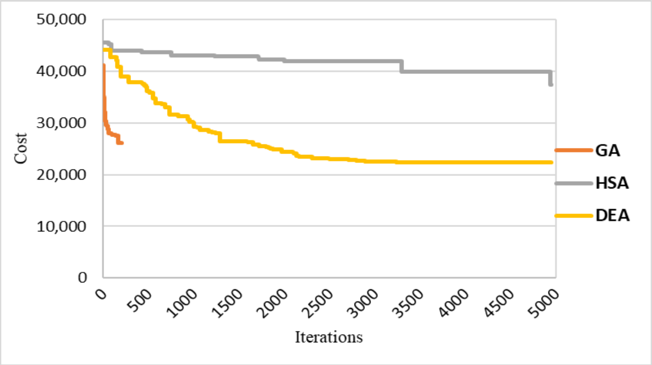

Fig 19: Convergence rates of the compared algorithms for S5 parameter settings

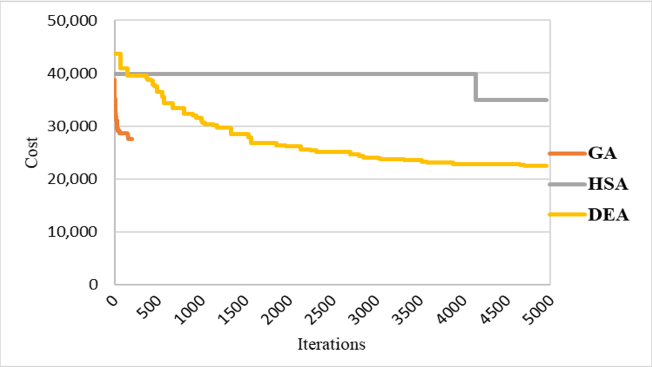

Fig 20: Convergence rates of the compared algorithms for S6 parameter settings

Furthermore, an ANOVA statistical method is used to verify whether there is a significant difference between the results achieved by the compared algorithms or not, such that the null hypothesis (h_0) and the alternative hypothesis (h_1) are given as follows:

- h_0: μ1 = μ2 = μ3, where μ is the mean.
- h_1: At least one of the means is different.

The ANOVA results, provided in Table 14, show that there are significant differences between the mean fitness values as the p-value is lower than the significance level of 0.05 and F-values are always greater than the F-critical value. Thus, there is sufficient evidence to reject the null hypothesis h_0 and accept the alternative hypothesis h_1, which concludes that not all mean fitness values are equal to each other and a significant difference between the mean fitness results is found.

Table 12: ANOVA descriptive statistics with a population size of 50

Settings	F-critical	F-value	*P*-value
S1	3.10	6568.4	1.23E-95
S2	3.10	2882.5	3.09E-80
S3	3.10	7519.7	3.55E-98
S4	3.10	6404.9	3.66E-95
S5	3.10	4486.4	1.71E-88
S6	3.10	2878.1	3.30E-80

5.3 Discussion

The previous section shows that GA is doing better compared to HSA and DEA in most experimental cases when the population size is equal to 5. On the other hand, DEA provides better performance compared to other algorithms when the population size is equal to 20 and 50, in terms of both fitness values and convergence rate. Also, the results reveal that HSA did not provide any superior results compared to GA and DEA in all experiments performed.

Moreover, simulation experiments show worse performance results for both GA and HSA as population size increases in nearly all experiments performed. Also, the best-achieved performance result was observed by DEA when the population size is equal to 20, with a mean fitness value is equal to 22406.6 using the parameter settings Pc=0.5 and F=0.5.

Table 15 provides the best problem parameters achieved by each algorithm for the airport scheduling problem parameters, while Figure 21 shows the waiting time for passengers of all flights at each stage of the airport. The results show that the DEA has the lowest passenger waiting time compared to the waiting times achieved by the other algorithms. Also, the results show that the largest waiting time for passengers was in the arrival gate lounges due to the lack of allocated spaces in the passport control area, followed by the baggage area, then the passport control and customs areas, and that the minimum waiting time was before entering the gate lounges.

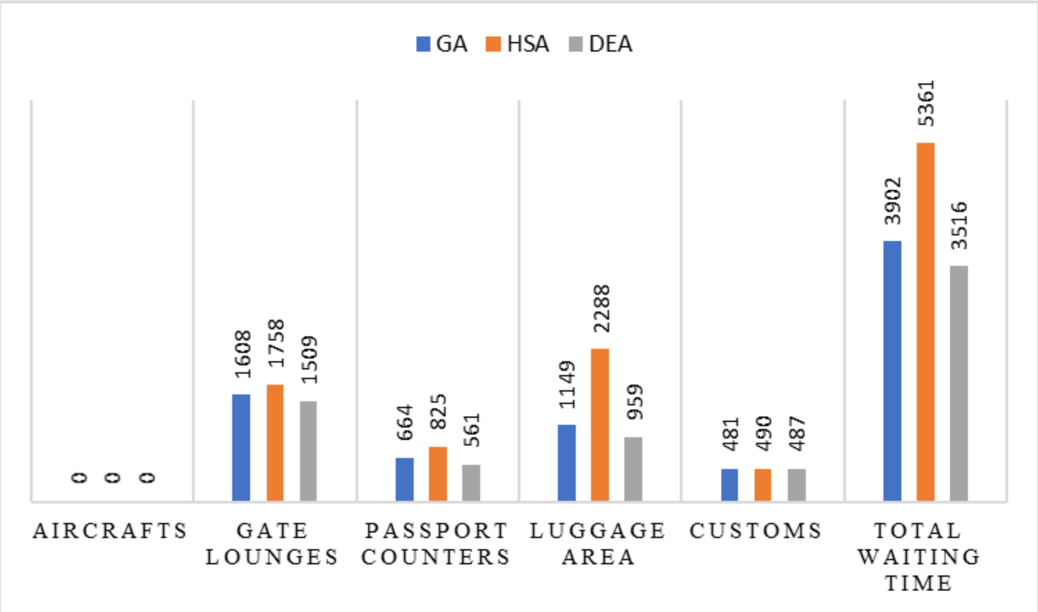

Fig 21: Waiting time (in minutes) for passengers in every airport's stage

Table 13: Best solution representation for airport's flight scheduling problem

Parameter Name	GA	HSA	DEA
Processing time to complete medical examination procedures	13	12	5
Processing time to complete passport control procedures	91	96	90
Medical examination workers in gate lounge no.1	3	4	1
Medical examination workers in gate lounge no.2	1	1	1
Medical examination workers in gate lounge no.3	2	6	1
Medical examination workers in gate lounge no.4	4	3	1
Medical examination workers in gate lounge no.5	1	4	1

Table 14: Continued.

Parameter Name	**GA**	**HSA**	**DEA**
Medical examination workers in gate lounge no.6	1	1	1
Medical examination workers in gate lounge no.7	3	1	1
Medical examination workers in gate lounge no.8	1	11	1
Medical examination workers in gate lounge no.9	3	5	1
Medical examination workers in gate lounge no.10	2	3	1
Medical examination workers in gate lounge no.11	3	3	1
Medical examination workers in gate lounge no.12	2	3	1
Medical examination workers in gate lounge no.13	1	3	1
Medical examination workers in gate lounge no.14	2	19	1
Medical examination workers in gate lounge no.15	1	1	1
Medical examination workers in gate lounge no.16	1	6	1
Medical examination workers in gate lounge no.17	1	4	1
Medical examination workers in gate lounge no.18	1	6	1
Number of passport counters -section 1	25	32	22
Number of passport counters -section 2	32	39	32
Number of passport counters -section 3	17	24	16
Number of passport counters -section 4	32	42	32
Number of passport counters -section 5	17	27	16
Number of passport counters -section 6	23	21	18
Number of custom inspection devices in every section	7	5	7
Best fitness value cost	**25161**	**34851**	**22179**

6.0 CONCLUSION AND FUTURE WORK

The purpose of this study is to improve the flight scheduling procedure at the King Abdulaziz International Airport in Jeddah's Hajj terminal. Therefore, all airport stages through which passengers pass are investigated. Moreover, a mathematical model of the optimization problem was formulated, as well as several hard and soft constraints were developed. Three optimization algorithms were developed and adapted to handle the airport optimization problem: the genetic algorithm (GA), the harmony search algorithm (HSA), and the differential evolution algorithm (DEA). The developed algorithms provide advanced knowledge on how to organize passenger group arrivals and show which stages are congested and may cause process delays. The proposed DEA obtained superior results compared to GA and HSA in terms of fitness value and convergence performance. The results revealed that due to work pressure in the passport area, passengers spent the greatest time waiting in the arrival gate lounges, followed by the luggage areas, and finally the customs areas. Future research will focus on improving the performance of DEA by hybridizing it with other heuristic algorithms. In addition, conducting several comparisons with various state-of-the-art algorithms on the same data set.

7.0 ACKNOWLEDGMENTS

The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: (22UQU4361183DSR02).

REFERENCES

[1] Gronfula, M. G. *Intelligent optimisation system for airport operation: Hajj Terminal in Saudi Arabia*, 2014, (Doctoral dissertation, Brunel University London).

[2] Correia, A.R., Wirasinghe, S.C., de Barros, A.G., *A global index for level of service evaluation at airport passenger terminals.* Transportation Research Part E: Logistics and Transportation Review, 2008, 44, p.607–620. https://doi.org/10.1016/j.tre.2007.05.009

[3] Shambour Mohd Khaled & Khan Esam A., 2022, Using Artificial Intelligence Algorithms to Improve the Scheduling Mechanism of Pilgrims' Flights. In: 21th Scientific Symposium for Hajj, Umrah & Madinah Visit, Madinah, Saudi Arabia, pp. 208–217.

[4] GACA, *General Authority of Civil Aviation, https://gaca.gov.sa/*, last accessed Oct.2022.

[5] Idris, J. *Spiritual motivation for religious tourism destinations*. Spiritual and Religious Tourism: Motivations and Management, 2019, p. 48-59.

[6] Ghamdi Z., Hajj terminal project, http://saudiprojects.net/, last accessed Oct.2022.

[7] GAS, *General Authority for Statistics, https:// stats.gov.sa*/, last accessed Oct.2022.

[8] Vision, *Saudi vision Programs, https://vision2030.gov.sa*/, last accessed Oct.2022.

[9] Handibag, S. and Sutkar, P.S., 2021, *Optimization algorithms and their applications*. Malaya Journal of Matematik, 9(1),pp.1006-1014. https://doi.org/10.26637/MJM0901/0177

[10] Pirozmand, P., Alrezaamiri, H., Ebrahimnejad, A. and Motameni, H., 2021. *A New Model Of Parallel Particle Swarm Optimization Algorithm For Solving Numerical Problems*. Malaysian Journal of Computer Science, 34(4), pp.389-407. https://doi.org/10.22452/mjcs.vol34no4.5

[11] Shambour Mohd Khaled, 2022, Analyzing perceptions of a global event using CNN-LSTM deep learning approach: the case of Hajj 1442 (2021), PeerJ Computer Science, 8:e1087 https://doi.org/10.7717/peerj-cs.1087

[12] Shehab, M., Abu-Hashem, M.A., Shambour, M.K.Y., Alsalibi, A.I., Alomari, O.A., Gupta, J.N., Alsoud, A.R., Abuhaija, B. and Abualigah, L., 2022. A Comprehensive Review of Bat Inspired Algorithm: Variants, Applications, and Hybridization. Archives of Computational Methods in Engineering, pp.1-33.

[13] Shambour, Mohd Khaled & Gutub, Adnan, 2022, Progress of IoT Research Technologies and Applications Serving Hajj and Umrah. Arab J Sci Eng 47, 1253–1273. https://doi.org/10.1007/s13369-021-05838-7

[14] Shambour Mohd Khaled Y. & Khan Esam A., A Heuristic Approach for Distributing Pilgrims over Mina Tents, Journal of King Abdulaziz University: Engineering Sciences, 30 (2), (2019), 11-23. DOI:10.4197/Eng.30-2.2

[15] Marinelli, M., Dell'Orco, M., & Sassanelli, D. *A metaheuristic approach to solve the flight gate assignment problem.* Transportation Research Procedia, 2015,5, p. 211-220. https://doi.org/10.1016/j.trpro.2015.01.013

[16] Xu, J., & Bailey, G., *The airport gate assignment problem: mathematical model and a tabu search algorithm*, In Proceedings of the 34th annual Hawaii international conference on system sciences, 2001, p.1-10. IEEE. DOI: 10.1109/HICSS.2001.926327

[17] Ding, H., Lim, A., Rodrigues, B., & Zhu, Y. *Aircraft and gate scheduling optimization at airports*. In 37th Annual Hawaii International Conference on System Sciences, 2004. p. 1-8. IEEE. DOI: 10.1109/HICSS.2004.1265219

[18] Al-Sultan, A. T., Ishioka, F., & Kurihara, K., *Aircraft and Gate scheduling optimization at Airports*. In Proceedings of the symposium of Japanese Society of Computational Statistics, 2009, 23, p.191-194. Japanese Society of Computational Statistics.

[19] AlSultan, A. T. *The airport gate assignment problem: scheduling algorithms and simulation approach,* (Doctoral dissertation, Okayama University), March. 2012.

[20] Hu, XB., Di Paolo, E. An Efficient Genetic Algorithm with Uniform Crossover for the Multi-Objective Airport Gate Assignment Problem. In: Goh, CK., Ong, YS., Tan, K.C. (eds) Multi-Objective Memetic Algorithms. Studies in Computational Intelligence, 2009, vol 171. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-88051-6_4

[21] Hidayatno, A., Moeis, A. O., & Dharma, G. A. S. *Designing gate assignment model to find the optimum airport gate assignment order*. Jurnal Teknik Industri, 2015, 17(1), p.1-6.

[22] Cheng, C. H., Ho, S. C., & Kwan, C. L. *The use of meta-heuristics for airport gate assignment.* Expert systems with applications, 2012, 39(16), p.12430-12437. https://doi.org/10.1016/j.eswa.2012.04.071

[23] Drexl, A., & Nikulin, Y. *Multicriteria airport gate assignment and Pareto simulated annealing*. IEEE Transactions, 2008, 40(4), p.385-397. https://doi.org/10.1080/07408170701416673

[24] Lim, A., & Wang, F. *Robust airport gate assignment*. In 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05), 2005, p. 1-8. IEEE. doi: 10.1109/ICTAI.2005.110.

[25] Bouras, A., Ghaleb, M. A., Suryahatmaja, U. S., & Salem, A. M. *The airport gate assignment problem: a survey.* The scientific world journal, 2014, p.1-27. https://doi.org/10.1155/2014/923859

[26] Ghazouani, H., Hammami, M., & Korbaa, O. *Solving airport gate assignment problem using Genetic Algorithms approach*. In 2015 4th International Conference on Advanced Logistics and Transport (ICALT), 2015, p. 175-180. IEEE. DOI: 10.1109/ICAdLT.2015.7136615

[27] Aktel, A., Yagmahan, B., Özcan, T., Yenisey, M. M., & Sansarcı, E. *The comparison of the metaheuristic algorithms performances on airport gate assignment problem*. Transportation research procedia, 2017, 22, p.469-478. https://doi.org/10.1016/j.trpro.2017.03.061

[28] Elm, *Consulting study project for the guest's journey from the ports to the residence,* Al-Elm Information Security Company, 2017, Techniqal report.

[29] KAIA: King Abdulaziz International Airport, https://kaia.sa/, last accessed Oct.2022.

[30] Alrabghi, A., *Modelling Passengers Flow at Hajj Terminal in Jeddah*. International Journal of Simulation: Systems, Science & Technology,2019, 20(6):p.1-7. DOI: 10.5013/IJSSST.a.20.06.04

[31] Shambour, Mohd Khaled, Khader, Ahmad Kheiri, and Özcan. A Two Stage Approach for High School Timetabling. In: Lee, M., Hirose, A., Hou, ZG., Kil, R.M. (eds) Neural Information Processing. ICONIP 2013. Lecture Notes in Computer Science, vol 8226. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-42054-2_9

[32] Shambour, Mohd Khaled and Khan, Esam, 2022. A Late Acceptance Hyper-Heuristic Approach for the Optimization Problem of Distributing Pilgrims over Mina Tents. Journal of Universal Computer Science, 28(4), pp.396-413.

[33] Holland, J. H. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*, 1975. Cambridge, MA: MIT Press. University of Michigan Press.

[34] Geem, Z. W., Kim, J. H., & Loganathan, G. V. *A new heuristic optimization algorithm: harmony search*. simulation, 2001, 76(2), p.60-68. https://doi.org/10.1177/003754970107600201

[35] Storn, R., & Price, K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization, 1997, 11(4), p.341-359. https://doi.org/10.1023/A:1008202821328

[36] Aljamal, M., Yasin, M., and Almawrei, Rami, 2019. *Measuring Time indicators for the quality of airports for pilgrims*, The custodian of the two holy mosques institute for Hajj and Umrah, p.1-45.

[37] Mostefai, A., Berrah, S., & Abid, H., 2021. Modeling and Simulation of MOSFET (High-k Dielectric) Using Genetic Algorithms. Journal of Nano-and Electronic Physics, 13(6). Journal of Nano-and Electronic Physics, 13(6). DOI:10.21272/jnep.13(6).06004

[38] Shambour, Mohd Khaled Y., 2018. Vibrant Search Mechanism for Numerical Optimization Functions. Journal of Information and Communication Technology, 17(4), 679–702. https://doi.org/10.32890/jict2018.17.4.8276