Neural Computing and Applications (2020) 32:11745–11765
https://doi.org/10.1007/s00521-019-04659-y

ORIGINAL ARTICLE

# ISA: a hybridization between iterated local search and simulated annealing for multiple-runway aircraft landing problem

Abdelaziz I. Hammouri[1] · Malik Sh. Braik[2] · Mohammed Azmi Al-Betar[3] · Mohammed A. Awadallah[4]

Received: 24 April 2019 / Accepted: 3 December 2019 / Published online: 13 December 2019

**Abstract**
This paper presents an efficient method for aircraft landing problem (ALP) based on a mechanism that hybridizes the iterated local search (ILS) and simulated annealing (SA) algorithms. ALP is handled by scheduling each incoming aircraft to land on a runway in accordance with a predefined landing time frame. The main objective to address is to find a feasible aircraft scheduling solution within the range of target time. The proposed hybridization method complements the advantages of both ILS and SA in a single optimization framework, referred to as iterated simulated annealing (ISA). The optimization framework of ISA has two main loops: an inner loop and an outer loop. In the inner loop, SA is utilized through a cooling schedule and a relaxing acceptance strategy to allow ISA to escape the local optima. In the outer loop, the restart mechanism and perturbation operation of the standard ILS are used to empower ISA to broadly navigate different search space regions. Extensive evaluation experiments were conducted on thirteen small- and large-sized ALP instances to assess the effectiveness and solution quality of ISA. The obtained results confirm that the proposed ISA method considerably performs better than other state-of-the-art methods in which ISA is capable of reaching new best results in 4 out of 24 large-sized problem instances as well as obtaining the best results in all small-sized instances. Evaluation on large-sized instances confirms a high degree of performance. As a new line of research, ISA is an effective method for ALP which can be further investigated for other combinatorial optimization problems.

**Keywords** Aircraft landing problem · Iterated local search · Simulated annealing · Iterated simulated annealing · Scheduling optimization problems

## 1 Introduction

Air transport has definitely established itself as one of the most important means of transportation. The last two decades have experienced an immense evolution of air transport [1] for both passengers and cargo. All types of aircraft must pass prior to landing at an approaching stage under the guidance of the air traffic control (ATC) tower, which has a responsibility for securing the flights. This is a complex daily task of the ATC towers. The presence of a limited number of runways at airports is a major obstacle for the ATC tower, particularly during the rapid increase in air traffic and rush times of landing and takeoff operations.

✉ Abdelaziz I. Hammouri
Aziz@bau.edu.jo

Malik Sh. Braik
mbraik@bau.edu.jo

Mohammed Azmi Al-Betar
mohbetar@bau.edu.jo

Mohammed A. Awadallah
ma.awadallah@alaqsa.edu.ps

[1] Department of Computer Information Systems, Al-Balqa Applied University, Al-Salt 19117, Jordan

[2] Department of Computer Science, Al-Balqa Applied University, Al-Salt 19117, Jordan

[3] Department of Information Technology, Al-Huson University College, Al-Balqa Applied University, P.O. Box 50, Al-Huson, Irbid, Jordan

[4] Department of Computer Science, Al-Aqsa University, P.O. Box 4051, Gaza, Palestine

The high demand for more flights beyond the airport's capacity leads to congestion at the airport, caused by its inability to accommodate all progressing flights, and therefore, airports will not be able to meet future needs.

In optimization context, there are two different optimization problems of aircraft sequencing and scheduling in the literature: (1) aircraft scheduling problem (ASP) and (2) aircraft landing problem (ALP) [2]. Both problems are different in terms of the objective function and also in terms of the complexity involved in solving the problem. Furthermore, the optimization methods proposed in the literature for both are also significantly different. To elaborate, ASP is a problem concerned with landing sequences and functional landing times for arrivals aircraft at the airport. The main constraints of this problem are to: (1) ensure the safe separation between aircraft arrival, (2) utilize the obtainable assimilation at the airport and (3) reduce the airborne delays. Indeed, ASP is a significant issue in the daily functioning of airports.

ALP, the main concern of this paper, is a combinatorial optimization problem (COP) which belongs to NP-hard category in almost all of its variation [3]. ALP is tackled by assigning a set of aircrafts arriving to runways and landing times in accordance with a set of hard and soft constraints. The hard constraints must be compulsorily met to arrive at a feasible solution for aircraft scheduling. An example of hard constraint in ALP is time window whereby each aircraft has earliest and latest time to land. On the other hand, soft constraint satisfaction is preferred, but not mandatory. However, the more the soft constraints are met, the better the quality of the ALP solution will be. An example of soft constraint is that each aircraft will record its preferred (target) landing time. Basically, the overall objective is to build a feasible ALP solution with a high quality [4]. As the ALP is a combinatorial optimization problem, most of researchers build their methods by working on feasible search space regions; therefore, the solution(s) feasibility will be maintained during the search. In case of violations occurring during the search, the repair process has to be invoked. Furthermore, the initial solution(s) must also be feasible; therefore, a suitable heuristic technique such as first-come first-serve is used to ensure the feasibility [5].

Several methods have been introduced for ALP. The earliest were exact (or calculus-based or deterministic) methods. This type of method is problem-dependent whereby any optimal solution is reached by constructing it event by event. However, this type of algorithm is not efficient for large-scale ALP that has large number of aircraft, airways and number/kind of constraints to be met. The large-scale ALP problems can be categorized into NP-hard class, which cannot be resolved in a polynomial time. Although these methods cannot effectively find a high-quality solution for large-scale ALP, they are nowadays used to handle some real-world problems with small size or find initial feasible solution for more advanced methods. The most popular exact methods which resolved ALP were linear programming [2, 6], dynamic programming [7, 8], mixed-integer goal programming [9], mixed-integer programming [4, 10], linear programming-based tree search [11], exact polynomial algorithm [12] and two-stage algorithm with heuristics [13]. To emphasize, deterministic methods are very powerful in finding exact solution for small instances. They cannot solve the large-scale instances in polynomial time. Therefore, approximation methods are devoted.

Recently, the emergence of metaheuristic-based algorithms for ALP has captured the attention of the scheduling research community [14]. Metaheuristic-based algorithms are general optimization templates that can be efficiently used for several optimization problems. They search for the optimal solution using their own operators controlled by some parameters to explore new regions in the problem search space and exploit the accumulative search [15]. Several studies have categorized these methods into either natural-based or non-natural-based, population-based or single-point search, dynamic or static objective function, memory or memory-less methods, etc. [16]. The common features of the metaheuristic-based algorithms are the concepts of exploration and exploitation which are reached differently from one algorithm to another. Exploration refers to the ability of the algorithm to explore the unvisited search space regions, whereas exploitation identifies the ability of the algorithm to deeply exploit the visited search space areas to reach the local optima on that area. These two concepts are contradictory and the balance between them is essential during the search [17].

There are specifically two familiar types of metaheuristic-based algorithms that have addressed APL as optimization problems as outlined below: (1) evolutionary-based algorithms such as scatter search [5], genetic algorithm [9, 18, 19] and population heuristic [20] and (2) swarm-based optimizers such as particle swarm optimization [3], ant colony optimization [21, 22], artificial bee colony [23] and gravitational search algorithm [24]. Local search-based algorithms which are the main concern of the work presented in this paper begin with a single provisional solution. At each iteration, that solution undergoes changes using efficient neighborhood moves until a locally optimized solution, in the same search space region of the initial one, is reached. Although trajectory-based algorithms are very powerful in finding the local optimal solution to which they converge, they track through a trajectory in the search space without scanning wider regions; thus, they are easily stuck in local optima. It has to be noted that the search space regions of a scheduling problem like ALP are very rugged and require a very powerful method in deepening search rather than shallow search. Therefore, the research community of the scheduling domain has recently turned its attention to trajectory-based approaches rather than other approaches [25].

It is worth mentioning that the attention of the current research on ALP was turned toward adapting local search-based methods for several reasons: Local search-based algorithms are much faster than other methods. They are also very powerful in deepening search which is very useful for the scheduling problem. Simulated annealing (SA) is a very powerful local search-based method can escape from the trap of local optima using particular acceptance criterion [26]. SA is successfully adapted for ALP in [9]. Furthermore, iterated local search (ILS) is another powerful local search-based method that can also escape the dilemma of local optima by means of using multi-start mechanism to try several search space regions [27]. As traditionally known, ALP is a COP or, in more specific terms, is a new hybrid local search method that complements the powerful features of two or more local search-based algorithms to enable deep search capability as well as the escape from the local optima. Recently, ILS and SA have been hybridized in different ways to cope with the search space of complex optimization problems [28, 29]. For example, Rajalakshmi et al. [29] integrated SA within the iterated local search algorithm to solve switching cells to switch in cellular mobile network. Rajalakshmi et al. used SA to locate the optimal solutions generated by the iterated local search algorithm. Experimentally, it was found that the performance of the hybrid algorithm [29] is better than that of SA and ILS algorithms when implemented individually.

Due to the fact that there is no free lunch (NFL) in optimization [30], we propose in this paper a hybridization between the successful features of SA and ILS for ALP which is called iterated simulated annealing (ISA). The optimization framework of ISA has two main loops: inner and outer loops. In the inner loop, SA is utilized by a cooling schedule and relaxing acceptance strategy to allow ISA to escape the local optima. In the outer loop, the restart mechanism and perturbation operation of the original ILS are utilized to empower ISA to navigate different search space regions.

To evaluate the performance of the ISA method, we conducted a test on a dataset consisting of 13 datasets with 49 benchmark instances of different sizes and complexities. The parameter sensitivity analysis for the proposed ISA is studied to check their effect on the convergence behavior. For comparative evaluation, the results obtained by ISA are compared with six well-regarded methods that produced the state-of-the-art results. It is worth mentioning that the results yielded by the proposed method excel those produced by other comparative methods in 4 out of 49 problem instances. Also, ISA is able to produce the best-known results in 32 problem instances as achieved by others. Indeed, the proposed method can be considered a very efficient addition to the ALP domain, which can be easily tailored to resolve other similar problems.

The rest of the paper is organized as follows: Sect. 2 provides a detailed description of the mathematical formula of the ALP. An elaborated illustration of the proposed approach is introduced in Sect. 3. Section 4 then provides a brief summary of both ILS and SA algorithms. Section 5 describes and analyzes the computational results and time efforts of the proposed method along with other state-of-the-art methods that have addressed ALP, with conclusions and future work in Sect. 6.

## 2 Aircraft landing problem: definition and formulation

To implement a feasible approach for the ALP, all aircraft arrival times should be known beforehand along with all other information about each one. Each aircraft has a preferred landing time, a predefined landing time window and a set of runways. The main goal is to schedule each runway to the suitable landing time for each arriving aircraft with a minimal overall deviation from the preferred landing time. A penalty cost is assigned to the scheduling solution subject to the landing time of each aircraft that lands before or after its preferred landing time. The necessary mathematical notations for the ALP are provided in Table 1. The mathematical formulation of the ALP model is introduced in the next constraint formulation based on the formulation provided in [11].

### 2.1 Constraints of the ALP model

In order to preserve the feasibility of the ALP solution, three hard constraints must be fulfilled: time window, separation time and multiple runways. These constraints are formalized as follows:

*Constraint 1* **Time window** For aircraft $i$, the landing time $t_i$ is scheduled to be within its predefined landing time window, $[E_i, L_i]$:

$$E_i \leq t_i \leq L_i, \qquad \forall i \in \{1, 2, \ldots, N\}. \tag{1}$$

*Constraint 2* **Separation time** Separation time constraints are conditioned by the landing order of aircraft and their assigned runways. The landing order of aircraft signals that either aircraft $i$ lands before aircraft $j$ which denotes $x_{ij} = 1$ or aircraft $j$ lands before aircraft $i$ which denotes $x_{ji} = 1$. This constraint is defined as shown in Eq. (2).

Table 1 Notations used to formulate the ALP

| Symbol | Description |
|---|---|
| $N$ | Number of aircraft waiting to land |
| $R$ | Number of available runways |
| $E_i$ | Earliest possible landing time for aircraft $i$ when it lands down before its target time |
| $T_i$ | Targeted (preferred) landing time for aircraft $i$ |
| $L_i$ | Latest possible landing time for aircraft $i$ when it lands down after its target time |
| $S_{ij}$ | The required separation time between landing aircraft $i$ and $j$, $(S_{ij} > 0, j = 1, 2, \ldots, N; i \neq j)$, where aircraft $i$ lands before aircraft $j$ on the same runway |
| $t_{ij}$ | The required separation time between aircraft $i$ and aircraft $j$, $(t_{ij} > 0, j = 1, 2, \ldots, N; i \neq j)$, where aircraft $i$ lands before aircraft $j$ on different runways |
| $P_{1i}$ | The incurred penalty cost per unit of time for aircraft $i$ when landing before its target time |
| $P_{2i}$ | The incurred penalty cost per unit of time for aircraft $i$ when landing after its target time |
| $t_i$ | The scheduled landing time of aircraft $i$, $t_i \geq 0$ |
| $\alpha_i$ | Tardiness of landing when aircraft $i$ is scheduled to land after its target time $T_i, \alpha_i = max(0, t_i - T_i), \alpha_i \geq 0$ |
| $\beta_i$ | Earliness of landing when aircraft $i$ is scheduled to land before its target time $T_i, \beta_i = max(0, T_i - t_i), \beta_i \geq 0$ |

The decision variables $x_{ij}$, $y_{ir}$ and $z_{ij}$ for aircraft $i$, where $i \in \{1, 2, \ldots, N\}$ and $j$, where $j \in \{1, 2, \ldots, N\}$ on runway $r$, where $r \in \{1, 2, \ldots, R\}$, where $i \neq j$, are defined as follows:

$$x_{ij} = \begin{cases} 1 & \text{if aircraft } i \text{ is assigned to land before aircraft } j \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ir} = \begin{cases} 1 & \text{if aircraft } i \text{ is scheduled to land on runway } r \\ 0 & \text{otherwise} \end{cases}$$

$$z_{ij} = \begin{cases} 1 & \text{if aircrafts } i \text{ and } j \text{ are scheduled to land on the same runway.} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ij} + x_{ji} = 1 \qquad \forall i,j \in \{1, 2, \ldots, N\} \wedge j > i \wedge x_{ij} \in (0, 1). \tag{2}$$

In some cases, for a particular pair ($i$, $j$) of aircraft, it can be immediately decided whether $x_{ij} = 1$ or $x_{ij} = 0$; for example, if $L_i < E_j$ then $x_{ij} = 1$ and $x_{ji} = 0$. For any pair of successive landings on the same runway ($i$, $j$) of aircraft, there is a requirement constraint that must be respected as shown in Eq. (3):

$$x_{ij} \times t_j \geq x_{ij} \times t_i + S_{ij} \times z_{ij} + t_{ij}(1 - z_{ij}) \quad \forall i,j \in \{1, 2, \ldots, N\} \wedge j > i. \tag{3}$$

Let ($i$, $j$) be a pair of aircraft such that $i < j$ and assume that aircraft $i$ and $j$ land on the same runway, i.e., $z_{ij} = 1$ and $1 - z_{ij} = 0$. If aircraft $i$ lands before aircraft $j$ then $x_{ij} = 1$, then the constraint in Eq. (4) becomes:

$$t_j \geq t_i + S_{ij} \qquad \forall i,j \in \{1, 2, \ldots, N\}. \tag{4}$$

Equation (4) asserts that the separation criterion between aircraft $i$ and $j$ is conformed. This implies that the time $S_{ij}$ must elapse after the landing of aircraft $i$ at $t_i$ before $j$ can land at $t_j$.

*Constraint 3* **Multiple runways** The following constraint is used to assure that aircraft $i$ must be assigned to only one runway:

$$\sum_{r \in R} y_{ir} = 1 \qquad \forall i \in \{1, 2, \ldots, N\}. \tag{5}$$

There are some constraints formulated in Eqs. (6, 7) that are needed to guarantee that the runways assigned to aircraft $i$ and $j$ are indistinguishable when they are set to land on the same runway. The constraint in Eq. (6) shows that the matrix $z_{ij}$ is symmetric:

$$z_{ij} = z_{ji} \qquad \forall i,j \in \{1, 2, \ldots, N\} \wedge j > i. \tag{6}$$

The decision variables $y_{ir}, y_{jr}$ and $z_{ij}$ are linked together using Eq. (7).

$$z_{ij} \geq y_{ir} + y_{jr} - 1 \quad \forall i,j = (1, 2, \ldots, N) \wedge i \neq j \quad \forall r = (1, 2, \ldots, R). \tag{7}$$

However, if one aircraft $i$ or $j$ lands on runway $r$ and the other does not; then 0 must be assigned to $z_{ij}$.

## 2.2 Objective function of the ALP model

The objective function, $f$, considered here for the ALP is formulated as shown in Eq. (8), subject to the three constraints identified before (i.e., time window, separation time and multiple runways).

$$\min f(X) = \sum_{i=1}^{N} \alpha_i \times P_{1i} + \beta_i \times P_{2i}. \quad (8)$$

The expression in the parentheses of Eq. (8) defines total penalty corresponding to aircraft $i$. If aircraft $i$ lands at its target landing time, then both $\alpha_i$ and $\beta_i$ are equal to zero and the cost incurred by its landing is zero. On the other hand, if aircraft $i$ does not land at $T_i$, then either $\alpha_i$ or $\beta_i$ is nonzero and there is a strictly positive cost incurred. In practical terms, this objective function is valuable, which has the ability to reflect prevailing conditions at an airport. Note that both $\alpha_i$ and $\beta_i$ are constant and their values are extracted from the dataset used. The values of both $\alpha_i$ and $\beta_i$ are connected with the decision variables in the solution $X$ as in Eqs. 9–13.

$$x_i = T_i - \alpha_i + \beta_i \qquad \forall i \in \{1, 2, \ldots, P\} \quad (9)$$

$$0 \leq \alpha_i \leq T_i - E_i \qquad \forall i \in \{1, 2, \ldots, P\} \quad (10)$$

$$\alpha_i \geq T_i - x_i \qquad \forall i \in \{1, 2, \ldots, P\} \quad (11)$$

$$0 \leq \alpha_i \leq L_i - T_i \qquad \forall i \in \{1, 2, \ldots, P\} \quad (12)$$

$$\alpha_i \geq x_i - T_i \qquad \forall i \in \{1, 2, \ldots, P\}. \quad (13)$$

## 3 Research background

In order to build a self-exploratory paper, this section provides the elementary knowledge about the methods hybridized here. In the next subsection, basic fundamentals to the iterated local search (ILS) are illustrated as established in the original version [31]. Thereafter, a discussion about the basic concepts of simulated annealing is provided in Sect. 3.2.

### 3.1 Fundamentals of the iterated local search algorithm

ILS is an advanced extension method for basic local search method. It attempts to escape the local optima by means of restarting the local search technique several times starting with a different initial point at the search space. Normally, ILS has two important stages: improvement stage and perturbation stage. In the improvement stage, the simple local search mechanism is used to find a local optimal solution $X^{\text{opt}}$ which is passed to the perturbation stage in order to shuffle its elements for starting a new perturbed solution $X'^{\text{opt}}$. Finally, a greedy selection between $X^{\text{opt}}$ and $X'^{\text{opt}}$ is used. This process is repeated several times until a "good enough" local optima is obtained.

The perturbation is usually non-deterministic in order to avoid being cyclic. The perturbation stage is important to transform the solution from the current local optima to another point in the search region. The strength of perturbation has a great effect on the efficiency of ILS which takes control on the force of perturbation. Too small perturbation may lead the local search process to return to previous local optima already found, leading to misuse of computational resources. Too large perturbation may dominate the local search method to work as a local search method with a multi-random initial point. In other words, large perturbation in the early stages manages the local search procedure to explore new areas of the search space. The strength of perturbation is either static or dynamic. Indeed, static perturbation shuffles the local optima during the search and remains static from initial search until the final course of run. In contrast, dynamic perturbation strength progressively decreases with the advance of the local search algorithm to focus on exploitation of new areas of the search space.

The acceptance criterion is the key feature of metaheuristic algorithms because it allows for a more comprehensive search for optimal solutions. The pseudo-code of ILS is described in Algorithm 1.

**Algorithm 1** A pseudo-code of the iterated local search method.

```
1: $MaxIter \leftarrow$ maximum number of iterations.
2: $Iter \leftarrow$ current iteration
3: $X \leftarrow$ new random solution
4: $X^{opt} \leftarrow$ X
5: $H \leftarrow X^{opt}$ {The current "home base" local optimum.}
6: repeat
7:     $Iter \leftarrow$ random value between 1 and $MaxIter$
8:     $MaxIter \leftarrow MaxIter$ - $Iter$
9:     repeat
10:        $X' \leftarrow$ Tweak($X$)
11:        if ($f(X') < f(X)$) then
12:            $X \leftarrow X'$
13:        end if
14:        Decrease $Iter$
15:    until ($X$ is the ideal solution or $Iter = 0$)
16:    if ($f(X) < f(X^{opt})$) then
17:        $X^{opt} \leftarrow X$
18:    end if
19:    if ($f(X) \leq f(H)$) then
20:        $H \leftarrow X$
21:    end if
22:    $X \leftarrow$ Perturb($H$)
23: until ($X^{opt}$ is the ideal solution or $MaxIter = 0$)
24: return $X^{opt}$
```

In Algorithm 1, ILS is initiated with a provisional solution $X$. The historical point ($H$) is also initially defined and assigned by $X$. ILS consists of two nested loops: *improvement* which is the inner loop (i.e., lines 9–15) and *perturbation* which is the outer loop. In the improvement loop, a simple local search strategy is employed. The *Tweak*(...) function is used as a neighborhood move with a one-step random improvement method. The output of improvement stage is a local optima $X'$ which is used in the perturbation stage. In the perturbation stage, $X'$ replaces $X$, if better and the historical point ($H$) is updated. Finally, *Perturb*(...) function is used to shuffle $H$ to be used as an initial point to the improvement loop. This process is repeated several times until the ideal solution (i.e., $f(X) = 0$) *or* the maximum number of iterations is reached. Note that the ideal solution is the one with zero cost. This solution is the exact one.

## 3.2 Simulating annealing

Simulated annealing is another extension to the simple local search method that simulates a bridge between the physical process of the metal annealing and the optimization process [32]. Similar to any local search method, SA is initiated with a random solution. Iteratively, this solution will be updated using neighborhood move associated with the problem to be solved. The solution update is then accepted based on two acceptance criteria:

(i) Greedy acceptance rule: in which the updated solution replaces the current one, if better in terms of solution quality. Formally, let the current solution be $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$ and the updated solution be $\boldsymbol{X}' = (X'_1, X'_2, \ldots, X'_n)$. The greedy selection rule accepts the updated solution when $f(\boldsymbol{X}) < f(\boldsymbol{X}')$, where $f(.)$ is the quality function.

(ii) Relaxed acceptance rule: In case that the Greedy acceptance rule does not recognize any improvement in the updated solution, the relaxed acceptance rule is checked. The relaxed acceptance rule can accept the worst updated solution $\boldsymbol{X}'$ than the current one $\boldsymbol{X}$ if the following condition becomes true:

$$\sim U(0,1) < \exp^{\frac{-(f(\boldsymbol{X}')-f(\boldsymbol{X}))}{T}},$$

where $\sim U(0,1)$ is a uniform distributed function that generates a random digit between 0 and 1.

It is worth noting that the temperature $T$ is exponentially decreased by a fraction $\alpha$, where $T(t+1) = \alpha \times T(t)$. In SA, this is called the *cooling schedule* process in which the SA begins with a high temperature value and it gradually decreases during the search until it almost reaches 0. Note that, when the value of T is high, the updated solution replaces the current one, if worse in large margin. This acceptance mechanism is degraded until a stagnation situation is reached where only greedy acceptance rule is used. Conventionally, this is in line with the main principle of optimization where the exploration is very useful in the initial search, but not important in the last course of runs.

The procedure of SA is pseudo-coded in Algorithm 2. As a matter of fact, the performance of SA depends on two factors: the initial temperature value $T$ and the cooling schedule. High value of $T$ leads to a heavily use of the relaxed acceptance rule in the initial search, and the SA might be required large time to converge. Here, SA behaves as a random search. In contrast, the lower the value of $T$ is, the faster the convergence of SA will be. Here, SA behaves as a simple local search method. In cooling scheduling, a large value of $\alpha$ may lead to slow convergence whereby SA will accept the worst solution in the initial search for large iterations.

**Algorithm 2** A pseudo-code of the simulated annealing algorithm.

```
1: $T$ ←temperature (initially a high number)
2: $\alpha$ ← Value very close to 1 (but less than 1)
3: $MaxIter$ ← maximum number of iterations.
4: $X$ ← new random solution
5: $Best$ ← X
6: repeat
7:     $X' \leftarrow$ Tweak($X$)
8:     if $f(X') < f(Best) || U(0,1) \leq \exp^{\frac{f(X')-f(X)}{T}}$ then
9:         $X \leftarrow X'$
10:    end if
11:    $T = \alpha \times T$
12:    Decrease $MaxIter$
13:    if $(f(X) \leq f(Best))$ then
14:        $Best \leftarrow X$
15:    end if
16: until ($Best$ is the ideal solution || $MaxIter = 0$ || T$\leq 0$ )
17: return $Best$
```

## 4 Iterated simulating annealing (ISA) algorithm for ALP

In this section, the proposed methodology which hybridizes the ILS with SA is adapted for ALP. Initially, the ALP solution is represented as a vector $X = (X_1, X_2, \ldots, X_N)$ of length $N$ aircraft. Each decision variable $X_i$ in the ALP solution can be assigned by a tuple of values of aircraft index, landing runway and landing time. Table 2 provides an example of ALP solution representation. In the example, $X_1$(aircraft Index) = 2 means the first aircraft to land is aircraft #2. $X_1$(Landing Runway) = 3 means the aircraft number #2 will land in the airway #3 and the $X_1$(Landing Time) = 98 means the aircraft number #2 will land at a time unit #98 and so on. Each solution is evaluated using the objective function formulated in Eq. (1).

The main role of ILS in the proposed method is not only to avoid local minima since SA has a cooling strategy to handle this, but also to be used as multi-restart algorithm that empower the proposed method to try different search space regions at each iteration and keep their local minima with assist of SA.

In the proposed method, the relaxed acceptance criterion of SA is incorporated with ILS optimization framework to improve its local exploitation. Additionally, by means of adding the relaxed acceptance criterion in the ISA, another way of avoiding the local optima is provided to empower the performance. The cooling schedule of ISA is based on the same theory of SA where it is initiated with high value of temperature. It cools down using a structured cooling schedule based on a predetermined cooling rate until an equilibrium state (or steady state) is reached.

The feasibility of the ALP solution is maintained during the search in which the hard constraints (i.e., time window, separation time and multiple runways) must be respected. It is worth mentioning that the proposed method only considers solutions in the feasible search space region. Therefore, any ALP solution with violation of hard constraints must be repaired. If the separation time constraint is violated between two aircraft (say, aircraft $A$ lands before $B$), the proposed approach earlier changes the landing time of aircraft $A$ with respect to its landing time window. If such changes keep the violation remaining, the proposed approach will restart the operation all over again.

The proposed ISA method complements the strength of both ILS and SA in single optimization algorithm. ISA has five main steps which will be discussed below. The flowchart of ISA is provided in Fig. 1, and its pseudo-code is provided in Algorithm 3.

**Algorithm 3** A pseudo-code of the proposed ISA algorithm.

```
1: $MaxIter \leftarrow$ Maximum number of iterations.
2: $\alpha \leftarrow$ Value very close to 1 (but less than 1)
3: $X \leftarrow$ New random solution
4: $H \leftarrow X$
5: $Best \leftarrow$ X
6: repeat
7:     $T \quad \leftarrow$ Temperature (initially a high number)
8:     $Iter \leftarrow \sim U(1 \ldots MaxIter)$
9:     $MaxIter = MaxIter - Iter$
10:    repeat
11:        $X' \leftarrow$ Tweak($X$)
12:        if $f(X') < f(Best) \; || \; U(0,1) \leq \exp^{\frac{f(X')-f(X)}{T}}$ then
13:            $X \leftarrow \boldsymbol{X'}$
14:        end if
15:        $T = \alpha \times T$
16:        Decrease $Iter$
17:        if $(f(X) \leq f(Best))$ then
18:            $Best \leftarrow X$
19:        end if
20:    until ($Best$ is the ideal solution || $Iter = 0$)
21:    if $(f(X) \leq f(H))$ then
22:        $H \leftarrow X$
23:    end if
24:    $X \leftarrow$ Perturb($H$)
25: until ($Best$ is the ideal solution || $MaxIter = 0$)
26: return $Best$
```

*Step I* Initialize the parameters of both ALP and ISA: Initially, the parameters of ALP are extracted from the dataset to suitable data structure. These parameters are the number of arrival aircraft $N$, number of available runways $R$, earliest possible landing $E_i$, targeted landing time $T_i$, latest possible landing time $L_i$, etc. as mathematically formulated in Table 1. Furthermore, the parameter of ISA required for tackling ALP is also initialized in this step. These parameters are temperature ($T$), cooling rate $\alpha$ and number of iteration ($MaxIter$).

*Step II* Generation of initial solution: In ISA, the second step randomly generates the initial solution $\boldsymbol{X}$. As aforementioned, each element in the solution $X_i$ in initialized by a tuple of values: aircraft index, landing runway and landing time. In order to construct the initial solution, all hard constraints discussed in Sect. 2 must be satisfied in the initial solution to ensure its feasibility. Indeed, ensuring the feasibility of the initial solution of ALP is not a trivial task. In the literature, some works proposed heuristic methods based on greedy rules to generate the initial solution [27]. Others proposed heuristic methods based on the first-come first-serve technique to build the initial solution [6].

**Algorithm 4** A pseudo-code describing how to generate a feasible initial solution.

```
1: $X_i \leftarrow$ Aircraft number $i$.
2: $t_i \leftarrow$ Landing time for aircraft $i$.
3: $r_i \leftarrow$ Landing runway for aircraft $i$.
4: $R \leftarrow$ Number of available runways.
5: $S_{ij} \leftarrow$ Separation time between landing aircraft $i$ and $j$.
6: repeat
7:     for $i \in \{1, \ldots, N\}$ do
8:         $X_i = i$
9:         $t_i = \sim$ U($E_i$ , $L_i$)
10:        $r_i = \sim$ U(1 , $R$)
11:    end for
12:    Sort $\boldsymbol{X}$ based on landing times $(t_1, t_2, \ldots, t_N)$
13:    $Stop = 1$
14:    for $j \in \{2, \ldots, N\}$ do
15:        $i = j - 1$
16:        $diff = t_i - t_j$
17:        if $(diff < S_{ij} \; \& \; r_i = r_j)$ then
18:            $Stop = 0$
19:        end if
20:    end for
21: until $Stop$
22: Return $X$
```

In ISA, in order to ensure the feasibility of the initial solution, a heuristic method based on time window constraint is proposed. This heuristic method is pseudo-coded in Algorithm 4. In the algorithm, $\sim U(E_i, L_i)$ generates a random number between $E_i$ and $L_i$ and $\sim$ U(1 , $R$) is a function generating a random number between 1 and $R$. In the initial loop, each decision variable is randomly assigned by a group of values. Thereafter, the whole solution is sorted based on the aircraft landing time. For example, if the random solution generated after sorting process becomes $X = ((2,3,98),(3,2,106),\ldots,(1,2,258))$, then the aircraft #2 has a sequence of 1 which will be landed at time 98s in the landing airway #3. In lines 9–14, the separation time constraint is checked. In case it is violated, the heuristic method will restart the entire generating process from scratch.

**Table 2** Example of ALP solution representation

| | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | $X_6$ | $X_7$ | $X_8$ | $X_9$ | $X_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| Aircraft index | 2 | 3 | 4 | 6 | 5 | 7 | 8 | 0 | 9 | 1 |
| Landing runway | 3 | 1 | 0 | 1 | 0 | 3 | 0 | 2 | 1 | 2 |
| Landing time | 98 | 106 | 118 | 126 | 134 | 142 | 150 | 165 | 180 | 258 |

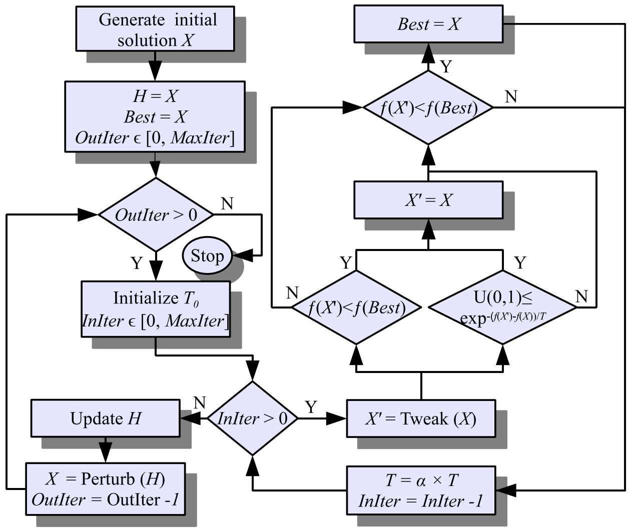

**Fig. 1** ISA flowchart

*Step III* Improvement loop In the improvement loop, the initial solution $\boldsymbol{X}$ is iteratively tweaked using *Tweak*() function (see Algorithm 3) to yield $\boldsymbol{X}' = (X_1', X_2', \ldots, X_N')$. If the objective function value of $\boldsymbol{X}'$ turns out to be better than that of $\boldsymbol{X}$ (i.e., $f(\boldsymbol{X}') < f(\boldsymbol{X})$), the current solution $\boldsymbol{X}$ will be replaced by the tweaked solution $\boldsymbol{X}'$. If not, the relaxed acceptance criteria will be triggered. The relaxed acceptance criteria might replace the current solution by the tweaked one, if worse. This process is done based on the value of temperature (*T*). When the value of *T* is large in the initial search, the percentage of accepting the worst solution is high. This percentage will be degraded by decreasing the value of *T* using the equation in Algorithm 3, line 14, until no further worst solution is accepted. In $Tweak(\boldsymbol{X})$ function, four neighborhood search methods are proposed for ALP. These neighborhood search methods are found to be very efficient in the navigation of the ALP search space and also maintain the feasibility of $\boldsymbol{X}$ during the search. The new solution $\boldsymbol{X}'$ is adjusted based on the switching mechanism shown in Eq. (14). Based upon Eq. (14), $r$ is randomly generated in the range of $r \in [0, 1]$. Apparently, the selection probability of each neighborhood search method is in the same context of 1:2:3 ratio. Each activated neighborhood method only performs a single random move without perform checking whether this move leads to a better solution.

$$\boldsymbol{X}' = \begin{cases} NB1(\boldsymbol{X}) & r \leq 0.25. \\ B2(\boldsymbol{X}) & 0.25 < r \leq 0.50 \\ NB3(\boldsymbol{X}) & 0.5 < r \leq 0.75 \\ NB4(\boldsymbol{X}) & \text{otherwise.} \end{cases} \tag{14}$$

The new ALP solution ($\boldsymbol{X}'$) is randomly adjusted in ISA by one of the following neighborhood methods:

- *NB1* select aircraft, $X_i$, randomly and then randomly change its landing time $t_i \in [E_i, L_i]$, where the index $i$ denotes to aircraft $i$. The landing sequence is updated based on landing times. The runway will not be changed.
- *NB2* select aircraft, $X_i$, at random and then randomly change its landing time $t_i \in [E_i, L_i]$ and its landing runway $r_i \in [1, R]$.
- *NB3* select randomly two aircraft, $X_i$ and $X_j$, from the same runway. Then set randomly the landing times $t_i$ and $t_j$ for $X_i$ and $X_j$, respectively, where $i$ and $j$ represent aircraft $i$ and $j$, respectively. Swap the landing times $t_i$ and $t_j$ for $X_i$ and $X_j$. The landing times $t_i$ and $t_j$ are restricted within $[E_i, L_i]$ and $[E_j, L_j]$, respectively. The feasibility of $\boldsymbol{X}'$ must be maintained.
- *NB4* randomly select aircraft $X_i$ and $X_j$ from two different runways $r_i$ and $r_j$, where $i \neq j$. Swap landing times $t_i$ and $t_j$ for aircraft $i$ and $j$. The utility of $\boldsymbol{X}'$ must be conserved.

As it can be noticed from the objective function formulated in Eq. (1), separation time is not directly computed and is related to the aircraft sequence. Therefore, if the sequence is updated in any movement process, consequently, the separation time will also be affected. Apparently, the four movement processes update the sequence in different ways with respect to the separation time constraints.

At each iteration of ISA, each new solution of $\boldsymbol{X}'$ replaces the current one of the $\boldsymbol{X}$ using the acceptance rule of SA: greedy and relaxed. The relaxed acceptance rule depends on the value of $T$ which is iteratively updated using cooling rate $\alpha$.

At the end of the improvement loop, the resulting solution (i.e., *best*) is the local optimal solution and that will be passed on to the perturbation step to reshuffle its component randomly.

It should be stressed that a heuristic method for generating the initial feasible solution is used in two places in the algorithm: to build initial feasible solution and to preserve the generated solution during the improvement loop. This method randomly chooses the landing time within the time window for each aircraft. It then sorts the aircraft based on their landing time and then generates the landing sequence. This method is successfully applied on all instances without having problems in generating the initial feasible solution. Some works in the literature use the first-come first-serve (FCFS) method for generating the initial solution [27]. ISA specifically uses FCFS method with some random components to diversify the initial feasible solution. The same heuristic method is also used to maintain the feasibility of the generated solutions during the search.

| | |
|---|---|
| Step IV | Perturbation: This is an important step in ISA to diversify the new resulting solution (i.e., *best*) from the improvement loop. The diversification process is done by means of using the same neighborhood methods discussed in the improvement loop, but without guidance from objective function. It is responsible for moving the improved solution (obtained by SA) to a new area in the search space. The perturbation process ensures that the solution is moved to a new region in the search space. The perturbed solution is considered the initial solution for the upcoming improvement loop which is repeated again. |
| Step V | Termination criterion: As aforementioned in Algorithm 3, there are two main loops: the inner loop which is the improvement loop based on SA and the outer loop is repeated as many times as the SA is repeated based on ILS concepts and conditions. The ISA is terminated if the *best* becomes ideal solution or the maximum number of iterations is reached. |

## 5 Experimental results and discussion

The proposed ISA is programmed in Java under Microsoft Windows 10 platform. All experiments are implemented on Intel Machine with Core *i*5 CPU with 2.5 GHz and 6 GB of RAM. The experiments are designed to evaluate the effect of the hybridization concepts between ILS and SA. Each experiment is repeated 21 times. The results are recorded in terms of the percentage gap $\Delta(\%)$, which refers to the difference between the best-known values (BKVs) in the literature and the results obtained by the proposed ISA method. $\Delta(\%)$ criterion measure is defined as shown below [27]:

$$\Delta(\%) = \frac{B - \text{BKV}}{\text{BKV}} \times 100\%, \qquad (15)$$

where $B$ is the best result returned by an algorithm evaluated on a set of runs and the BKV results are taken from [5]. Note that when the value of $\Delta(\%) = 0$, this means that the proposed algorithm is able to obtained the BKV results. When the value of $\Delta(\%) > 0$, this means that the proposed algorithm is obtained a result worse than the BKV results. When the value of $\Delta(\%) < 0$, this means that the proposed algorithm reaches a result that is superior to the BKV result.

In this section, the experimental results obtained by the proposed ISA are summarized in terms of the difference in BKV as formulated in Eq. (15). The parameter settings used to run the proposed method are configured as given in Table 3. These parameters are set after conducting a set of comprehensive experiments, and the best parameter values are defined.

Initially, the datasets under consideration are described in Sect. 5.1. The effect of parameter $\alpha$ on the convergence behavior of ISA is presented in Sect. 5.2. Thereafter, comparison between the proposed methods including ISA, ILS and SA is made in Sect. 5.3. Comparison with other state-of-the-art methods is provided in Sect. 5.4. To show the significance test among comparative methods, Friedman's test and post hoc Holm method are utilized as demonstrated in Sect. 5.5. Finally, the computational time for each comparative method is presented and elucidated in Sect. 5.6.

### 5.1 Dataset used

The performance of the ISA was evaluated using thirteen ALP groups of datasets, which has 49 different problem instances. These datasets are introduced in [11] and are publicly available in OR-library [33].[1]

The characteristics of datasets used are summarized in Table 4. These datasets are varied in terms of number of aircraft and runways. Each dataset has a different number of runways with the same number of aircraft. The first column shows the dataset name, while the second column shows the number of problem instances in each dataset. The column $R$ in Table 4 refers to the series of runways used to differentiate between the dataset instances. The final column refers to the number of aircraft in each dataset $N$. These problem instances can be grouped into small datasets (1 to 25 instances in 1–8 datasets) and large datasets (26 to 49 instances in 9–13 datasets). The problem instances are categorized into small and large based on the number of aircraft $N$ in which the $N \leq 50$ is considered as small, while the $N \geq 100$ is considered as large problem instances [27].

### 5.2 Effect of $\alpha$ on the convergence rate of ISA

To demonstrate the influence of cooling rate (i.e., $\alpha$) on the convergence characteristic behavior of ISA, a series of experiments with different values of $\alpha$ are conducted.

[1] http://people.brunel.ac.uk/~mastjjb/jeb/orlib/airlandinfo.html.

**Table 3** Parameter setting of ISA

| Parameter name | Parameter value |
|---|---|
| Temperature (T) | 100 |
| *Cooling rate* ($\alpha$) | 0.999 |
| *Maximum iteration for small-size instances* (*Maxiter*) | 100,000 |
| *Maximum iteration for large-size instances* (*Maxiter*) | 10,000,000 |

These values are specifically: $\alpha = 0.5, \alpha = 0.90, \alpha = 0.97$ and $\alpha = 0.999$. Recall that cooling rate is an important parameter in ISA and is responsible for managing the speed of convergence. $\alpha$ with large value close to 1 refers to the slow decrease in the temperature $T$ from its initial value to its final value. This means that smooth changes in the value of $T$ enable ISA to escape the local minimum and therefore achieve better exploration features. $\alpha$ with small value close to 0 ISA is returned to a steady-state stage very quickly, and exploitation behavior is focused. The results of various values of $\alpha$ are recorded in Table 5. The statistical results of ALP solution are summarized in terms of the best (Best), average (Avg.), worst solutions (Worst) and the standard deviation (STD) among the best results obtained over ten replicated runs. Best results obtained among various values of $\alpha$ are emboldened.

It appears that small dataset instances (from instance 1 to 25), ISA with $\alpha = 0.999$, ISA with $\alpha = 0.97$ and $\alpha = 0.90$ are able to produce almost the same best results equal to the BKV. However, ISA with $\alpha = 0.5$ reveals different convergence behavior where there are different results values in favor of the large $\alpha$ values. In conclusion, large value of cooling rate $\alpha$ close to one is preferable to gain better outcomes in small instances.

For large dataset instances (from instance 26 to 49), there are noticeable differences among almost all values of $\alpha$. The closer the value of $\alpha$ is to 1, the better the obtained results are. This can be borne out by the results recorded in Table 5. In conclusion, larger value of cooling rate $\alpha$ is much useful for ISA-based ALP method whereby the search space of ALP can be more touchable and the navigation of ALP search space will be more efficient. Thus, the value of $\alpha = 0.999$ will be used in the upcoming experiments.

## 5.3 Comparison with the proposed local search methods

To study the effect of combining the different local search methods proposed in this paper (i.e., SA, ILS and ISA), we present here the results of SA, ILS and a combination of both SA and ILS, or shortened as ISA. The results of each proposed method are exhibited in Table 6. The results are a summary of 10 replicated runs and are recorded in terms of Best, Avg., Worst and STD. The best results are highlighted in bold. The starting search points of all used algorithms are stabilized to ensure fair comparisons for all replicated runs.

Notably, for small dataset instances of ALP (from instance 1 to 25), the results of ISA and ILS are almost the same, while there is a clear difference in comparison with SA alone. For large instances (from instance 26 to 49), ISA is able to obtain the best results, followed by SA, followed by ILS. In conclusion, both local search methods used in this paper have almost the same performance and share their powerful features in ISA yield powerful algorithm. Some of the comparative experimental cases are visualized as boxplot representation as illustrated in Fig. 2. These cases are selected according to the relevancy of the information. The main aim of using boxplot representation is to demonstrate the robustness of the proposed ISA algorithm. In boxplot representation, the lower line in the box represents the first quartile (Q1), and the line inside the box represents the median (M), while the upper line is the third quartile (Q3). The lower wisher is the best solution obtained, while the upper wisher is the worst solution. The "+" sign indicates the exceptional values. It is clearly observed that the results obtained by ISA algorithm are visualized by a compact box, which means that it is a robust algorithm. This is borne out by the boxes drawn where a very narrow space between Q1, M and Q3. Furthermore, it can be observed that the behavior of the ISA algorithm is better than those of both SA and ILS by obtaining almost the same best results for 10 times of runs.

Figure 3 shows the convergence behavior of ISA, SA and ILS algorithms against 1500–5000 iterations for four instances (i.e., 14, 23, 35 and 40). These instances are selected randomly to cover different sizes and complexities of the instances. The number of iterations is selected to clearly visualize the differences between the presented algorithms. In these convergence plots, $x$-axis represents the number of iterations and the $y$-axis denotes the values of the objective function. Figure 3 shows that the slop of the ISA algorithm is better than the slope of the other two algorithms in all instances. It is observed that the performance of SA is the worst compared to ILS and ISA algorithms on instance 14, while the performance of ILS is the worst in comparison with the other algorithms on the remaining instances. This proves that the integration between SA and ILS is able to balance exploration and exploitation and thus achieve reasonable optimization results.

Table 4 Characteristics of ALP datasets

| Dataset name | No. of runways | Instance no. | *R* | *N* | Category |
|---|---|---|---|---|---|
| Airland1 | 3 | 1–3 | (1, 2, 3) | 10 | Small |
| Airland2 | 3 | 4–6 | (1, 2, 3) | 15 | |
| Airland3 | 3 | 7–9 | (1, 2, 3) | 20 | |
| Airland4 | 4 | 10–13 | (1, 2, 3, 4) | 20 | |
| Airland5 | 4 | 14–17 | (1, 2, 3, 4) | 20 | |
| Airland6 | 3 | 18–20 | (1, 2, 3) | 30 | |
| Airland7 | 2 | 21–22 | (1, 2) | 44 | |
| Airland8 | 3 | 23–25 | (1, 2, 3) | 50 | |
| Airland9 | 4 | 26–29 | (1, 2, 3, 4) | 100 | Large |
| Airland10 | 5 | 30–34 | (1, 2, 3, 4, 5) | 150 | |
| Airland11 | 5 | 35–39 | (1, 2, 3, 4, 5) | 200 | |
| Airland12 | 5 | 40–44 | (1, 2, 3, 4, 5) | 250 | |
| Airland13 | 5 | 45–49 | (1, 2, 3, 4, 5) | 500 | |

## 5.4 Comparison with previous methods

For comparative evaluation, Table 7 summarizes parameter setting and abbreviations of the comparative methods along with the parameter settings used. These comparative methods are able to produce high-quality results for ALP using the same problem instances under study. As other comparative methods, the results of ISA are reported in regard to $\Delta(\%)$ over 21 replicated runs.

The comparative results are summarized in Tables 8 and 9. Results reported in Table 8 show small-sized instances as indicated by datasets 1–8, and Table 9 shows large-sized instances as indicated by datasets 9–13. Note that the best-known value (BKV) is given for each problem instance in those tables. Other results are those obtained by HPSO, ILS, SS, BA, SA1 and SA2 methods in terms of the percentage gap of BKV. The best results are highlighted in bold font (lowest is best).

As shown in Table 8, the proposed ISA was able to obtain the best results for 25 out of 25 small-sized problem instances as achieved by some comparative methods which are HPSO, ILS, SS and BA.

The results summarized in Table 9 show that the proposed ISA method was able to outperform the other comparative methods in 4 out of the 24 (large-sized) instances as achieved in [3]. Furthermore, ISA was able to achieve results similar to BKV for 37, 42, 43 and 49 problem instances. In some cases, the ISA results were not be able to outperform those produced by HPSO, ILS, SA1 and SA2 in some instances with single runway (i.e., 26, 30, 35 and 40 problem instances). However, the difference margin to the corresponding instances of the BKV results is relatively small. These problem instances are highly complex with a very rugged search space. Since ISA adopts the random improvement acceptance strategy in the neighborhood search, improving the solution in the last course of run will be very rare. In [27], the best improvement acceptance strategy in the neighborhood search is adopted which is very efficient for complex and rugged search space.

From a different perspective, the summarized results recorded in Tables 8 and 9 prove that the results obtained by ISA method are considerably better than those produced by SS, BA, SA1 and SA2 methods. In particular, the method contributed in [27] has previously gotten best achievements using the same dataset with some new BKV. Our ISA was able to achieve excellent results in almost all problem instances except those produced in HPSO. Note that the method proposed in [27] is ILS and our ISA and can be considered as an extended improvement to ILS.

## 5.5 Statistical analysis

To validate the general efficiency of the proposed ISA method more strictly, a statistical analysis test using Friedman's statistical test [34] with a significance level of $\alpha = 5\%$ was performed. This is to rank the ISA method long with the competitor methods in accordance with the obtained results on large-sized instances. Table 10 summarizes the ranking of algorithms that were obtained by Friedman's test. From Table 10, HPSO was ranked first (with an average ranking 2.06), ILS ranked second (with an average ranking 2.79) and ISA ranked third (with an average ranking 3.00) followed in order by SA1, SA2, SS and BA in the last rank.

The *p*-value computed by Friedman's test was 0.036E-11, which is below the significance level $\alpha = 5\%$. This means that there is a significant difference in the results of the evaluated algorithms. Thus, we performed a post hoc procedure (Holm's method) to confirm that there are significant differences in the performance of the control algorithm (the best performing one—HPSO) and the remaining methods in the comparison, as well as reject the

**Table 5** Sensitivity analysis for alpha parameter in ISA

| Dataset name | Inst. no. | No. of runways | BKV | α = 0.5 | | | α = 0.9 | | | α = 0.97 | | | α = 0.999 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Best | Avg. | STD | Best | Avg. | STD | Best | Avg. | STD | Best | Avg. | STD |
| Airland1 | 1 | 1 | 700 | **700.00** | 717 | 33.02 | **700.00** | 718 | 32.93 | **700.00** | 706 | 5.16 | **700.00** | 700 | 0 |
| | 2 | 2 | 90 | **90.00** | 90 | 0 | **90.00** | 90 | 0 | **90.00** | 90 | 0 | **90.00** | 90 | 0 |
| | 3 | 3 | 0 | **0.00** | 0 | 0 | **0.00** | 0 | 0 | **0.00** | 0 | 0 | **0.00** | 0 | 0 |
| Airland2 | 4 | 1 | 1480 | **1480.00** | 1501 | 20.79 | **1480.00** | 1497 | 14.94 | **1480.00** | 1501 | 14.49 | **1480.00** | 1480 | 0 |
| | 5 | 2 | 210 | **210.00** | 246 | 77.2 | **210.00** | 255 | 81.55 | **210.00** | 240 | 69.28 | **210.00** | 210 | 0 |
| | 6 | 3 | 0 | **0.00** | 36 | 86.95 | **0.00** | 0 | 0 | **0.00** | 45 | 142.3 | **0.00** | 0 | 0 |
| Airland3 | 7 | 1 | 820 | **820.00** | 882 | 61.43 | **820.00** | 834 | 37.77 | **820.00** | 836 | 44.02 | **820.00** | 820 | 0 |
| | 8 | 2 | 60 | **60.00** | 68 | 16.87 | **60.00** | 83 | 20.03 | **60.00** | 75 | 19.58 | **60.00** | 60 | 0 |
| | 9 | 3 | 0 | **0.00** | 0 | 0 | **0.00** | 0 | 0 | **0.00** | 6 | 18.97 | **0.00** | 0 | 0 |
| Airland4 | 10 | 1 | 2520 | 2530 | 2541 | 24.7 | 2530 | 2551 | 42.02 | **2520.00** | 2581 | 72.03 | **2520.00** | 2520 | 0 |
| | 11 | 2 | 640 | **640.00** | 678 | 50.73 | **640.00** | 680 | 88.94 | **640.00** | 654 | 17.13 | **640.00** | 640 | 0 |
| | 12 | 3 | 130 | **130.00** | 163 | 29.83 | **130.00** | 154 | 27.57 | **130.00** | 168 | 81.89 | **130.00** | 130 | 0 |
| | 13 | 4 | 0 | **0.00** | 24 | 41.95 | **0.00** | 18 | 28.98 | **0.00** | 24 | 30.98 | **0.00** | 0 | 0 |
| Airland5 | 14 | 1 | 3100 | **3100.00** | 3152 | 90.9 | **3100.00** | 3152 | 164.44 | **3100.00** | 3249 | 223.88 | **3100.00** | 3100 | 0 |
| | 15 | 2 | 650 | **650.00** | 739 | 63.67 | 680 | 738 | 44.92 | **650.00** | 804 | 176.9 | **650.00** | 650 | 0 |
| | 16 | 3 | 170 | **170.00** | 222 | 30.11 | **170.00** | 221 | 40.12 | **170.00** | 215 | 33.42 | **170.00** | 170 | 0 |
| | 17 | 4 | 0 | **0.00** | 0 | 0 | **0.00** | 0 | 0 | **0.00** | 20 | 53.54 | **0.00** | 0 | 0 |
| Airland6 | 18 | 1 | 24,442 | **24,442.00** | 24,442 | 0 | **24,442.00** | 24,442 | 0 | **24,442.00** | 24,442 | 0 | **24,442.00** | 24,442 | 0 |
| | 19 | 2 | 554 | **554.00** | 554 | 0 | **554.00** | 554 | 0 | **554.00** | 554 | 0 | **554.00** | 554 | 0 |
| | 20 | 3 | 0 | **0.00** | 0.8 | 2.53 | **0.00** | 0 | 0 | **0.00** | 0.8 | 2.53 | **0.00** | 0 | 0 |
| Airland7 | 21 | 1 | 1550 | **1550.00** | 1701 | 176.39 | **1550.00** | 1665.7 | 166.52 | **1550.00** | 1736.3 | 178.13 | **1550.00** | 1550 | 0 |
| | 22 | 2 | 0 | **0.00** | 0 | 0 | **0.00** | 0 | 0 | **0.00** | 0 | 0 | **0.00** | 0 | 0 |
| Airland8 | 23 | 1 | 1950 | 1960 | 2462 | 368.07 | 1965 | 2151.5 | 209.43 | 1990 | 2312 | 244.38 | **1950.00** | 1982.5 | 19.36 |
| | 24 | 2 | 135 | **135.00** | 258 | 113.68 | 155 | 281 | 130.74 | **135.00** | 231 | 77.31 | **135.00** | 135 | 0 |
| | 25 | 3 | 0 | **0.00** | 40.5 | 49.02 | **0.00** | 77 | 74.43 | **0.00** | 51.5 | 52.65 | **0.00** | 0 | 0 |
| Airland9 | 26 | 1 | 5611.7 | 5906.78 | 6164.19 | 168.84 | 5847.54 | 6073.27 | 189.91 | 5875.32 | 6101.38 | 170.89 | **5685.86** | 5711.6 | 20.33 |
| | 27 | 2 | 452.92 | 456.11 | 514.41 | 61.97 | 452.49 | 532.59 | 113.78 | 468.21 | 604.93 | 92.35 | **449.11** | 451.54 | 1.18 |
| | 28 | 3 | 75.75 | **75.75** | 98.85 | 16.83 | **75.75** | 82.16 | 9.4 | **75.75** | 84.65 | 9.4 | **75.75** | 75.75 | 0 |
| | 29 | 4 | 0 | **0.00** | 2.09 | 5.03 | **0.00** | 0 | 0 | **0.00** | 5.85 | 12.33 | **0.00** | 0 | 0 |
| Airland10 | 30 | 1 | 12,329.31 | 15,393.95 | 16,626.51 | 845.94 | 14,346.24 | 16,648.2 | 1880.41 | 15,072.67 | 16,532.45 | 1136.91 | **12,527.95** | 12,627.24 | 75.19 |
| | 31 | 2 | 1288.73 | 1249.77 | 1406.67 | 173.43 | 1271.93 | 1377.52 | 94.29 | 1259.51 | 1378.84 | 81.97 | **1154.10** | 1170.97 | 10.22 |
| | 32 | 3 | 220.79 | 209.05 | 242.28 | 56.85 | 213.52 | 231.16 | 19.87 | 214.16 | 236.34 | 20.37 | **206.56** | 210.4 | 3.1 |
| | 33 | 4 | 34.22 | **34.22** | 52.64 | 28.84 | **34.22** | 53.73 | 27.36 | **34.22** | 39.01 | 11.5 | **34.22** | 34.22 | 0 |
| | 34 | 5 | 0 | **0.00** | 0.11 | 0.36 | **0.00** | 3.47 | 10.57 | **0.00** | 0 | 0 | **0.00** | 0 | 0 |

**Table 5** (continued)

| Dataset name | Inst. no. | No. of runways | BKV | $\alpha = 0.5$ | | | $\alpha = 0.9$ | | | $\alpha = 0.97$ | | | $\alpha = 0.999$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Best | Avg. | STD | Best | Avg. | STD | Best | Avg. | STD | Best | Avg. | STD |
| Airland11 | 35 | 1 | 12,418.32 | 13,457.01 | 14,024.98 | 531.78 | 13,419.12 | 14,087.82 | 519.48 | 13,300.75 | 14,196.07 | 541.04 | **12,870.47** | 13,038.43 | 60.52 |
| | 36 | 2 | 1540.84 | 1483.11 | 1576.52 | 83 | 1507.43 | 1565.4 | 68.71 | 1506.42 | 1566.21 | 60.94 | **1344.96** | 1360.25 | 6.74 |
| | 37 | 3 | 280.82 | 257.44 | 282.83 | 16.66 | 254.67 | 289.33 | 34.06 | 268.21 | 327.21 | 41.4 | **253.07** | 255.73 | 1.53 |
| | 38 | 4 | 54.53 | **54.53** | 55.96 | 4.52 | **54.53** | 54.53 | 0 | **54.53** | 67.99 | 42.55 | **54.53** | 54.53 | 0 |
| | 39 | 5 | 0 | **0.00** | 6.47 | 20.46 | **0.00** | 0.17 | 0.55 | **0.00** | 0 | 0 | **0.00** | 0 | 0 |
| Airland12 | 40 | 1 | 16,209.78 | 17,457.72 | 18,008.09 | 391.82 | 17,237.28 | 18,470.88 | 1074.82 | 17,442.41 | 17,993.38 | 548.49 | **16,675.64** | 16,744.78 | 49.23 |
| | 41 | 2 | 1961.39 | 1952.5 | 1990.05 | 42.94 | 1945.45 | 2023.12 | 107.55 | 1944.58 | 2032.08 | 115.06 | **1723.29** | 1737.97 | 8.77 |
| | 42 | 3 | 290.04 | 259.36 | 312.71 | 64.65 | 261.81 | 298.66 | 35.12 | 232.50 | 302.58 | 51.06 | **221.97** | 226.87 | 2.98 |
| | 43 | 4 | 3.49 | **2.44** | 26.09 | 57.52 | **2.44** | 11.13 | 24.13 | **2.44** | 21.54 | 34.61 | **2.44** | 2.44 | 0 |
| | 44 | 5 | 0 | **0.00** | 0.24 | 0.77 | **0.00** | 0 | 0 | **0.00** | 3.28 | 10.36 | **0.00** | 0 | 0 |
| Airland13 | 45 | 1 | 44,832.38 | 44,763.62 | 44,898.06 | 319.47 | 44,647.9 | 44,768.59 | 70.69 | 44,516.32 | 44,755.37 | 89.78 | **39,419.19** | 40,049.32 | 464.68 |
| | 46 | 2 | 5501.96 | 5302.33 | 5453.82 | 59.45 | 5379.43 | 5457.06 | 42.16 | 5366.78 | 5457.11 | 50.11 | **4053.27** | 4077.37 | 19.28 |
| | 47 | 3 | 1108.51 | 1068.58 | 1094.63 | 14.08 | 1060.65 | 1086.83 | 17.07 | 1036.19 | 1084.51 | 24.04 | **707.97** | 725.24 | 14.08 |
| | 48 | 4 | 188.46 | 150.87 | 180.08 | 18.53 | 149.77 | 169.17 | 11.58 | 136.36 | 170.56 | 16.46 | **90.56** | 94.09 | 3.21 |
| | 49 | 5 | 7.35 | **0.00** | 16.93 | 37.22 | **0.00** | 7.4 | 19.4 | **0.00** | 1.89 | 2.62 | **0.00** | 0 | 0 |

Best results are indicated in bold to denote their significance and to ease comparison of results

Table 6 A comparison between ISA, ILS and SA computational results

| Dataset name | Inst. no. | No. of runways | BKV | ISA | | | ILS | | | SA | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Best | Avg. | STD | Best | Avg. | STD | Best | Avg. | STD |
| Airland1 | 1 | 1 | 700 | **700.00** | 700 | 0 | **700.00** | 702 | 4.22 | 710 | 748 | 120.17 |
| | 2 | 2 | 90 | **90.00** | 90 | 0 | **90.00** | 90 | 0 | **90.00** | 99 | 28.46 |
| | 3 | 3 | 0 | **0.00** | 0 | 0 | **0.00** | 0 | 0 | **0.00** | 0 | 0.00 |
| Airland2 | 4 | 1 | 1480 | **1480.00** | 1480 | 0 | **1480.00** | 1501 | 16.63 | **1480.00** | 1514 | 17.76 |
| | 5 | 2 | 210 | **210.00** | 210 | 0 | **210.00** | 216 | 12.65 | **210.00** | 341 | 174.58 |
| | 6 | 3 | 0 | **0.00** | 0 | 0 | **0.00** | 0 | 0 | **0.00** | 45 | 142.30 |
| Airland3 | 7 | 1 | 820 | **820.00** | 820 | 0 | **820.00** | 822 | 6.32 | **820.00** | 882 | 65.63 |
| | 8 | 2 | 60 | **60.00** | 60 | 0 | **60.00** | 64 | 12.65 | **60.00** | 85 | 27.99 |
| | 9 | 3 | 0 | **0.00** | 0 | 0 | **0.00** | 0 | 0 | **0.00** | 6 | 18.97 |
| Airland4 | 10 | 1 | 2520 | **2520.00** | 2520 | 0 | 2530 | 2557 | 46.2 | 2530 | 2625 | 128.43 |
| | 11 | 2 | 640 | **640.00** | 640 | 0 | **640.00** | 651 | 15.24 | **640.00** | 722 | 81.35 |
| | 12 | 3 | 130 | **130.00** | 130 | 0 | **130.00** | 162 | 64.08 | **130.00** | 151 | 28.46 |
| | 13 | 4 | 0 | **0.00** | 0 | 0 | **0.00** | 6 | 18.97 | **0.00** | 12 | 25.30 |
| Airland5 | 14 | 1 | 3100 | **3100.00** | 3100 | 0 | **3100.00** | 3139 | 87.74 | **3100.00** | 3210 | 227.94 |
| | 15 | 2 | 650 | **650.00** | 650 | 0 | 680 | 745 | 71.38 | 690 | 839 | 125.30 |
| | 16 | 3 | 170 | **170.00** | 170 | 0 | **170.00** | 222 | 30.11 | 180 | 249 | 30.71 |
| | 17 | 4 | 0 | **0.00** | 0 | 0 | **0.00** | 3 | 9.49 | **0.00** | 18 | 37.95 |
| Airland6 | 18 | 1 | 24,442 | **24,442.00** | 24,442 | 0 | **24,442.00** | 24,442 | 0 | **24,442.00** | 24,442 | 0.00 |
| | 19 | 2 | 554 | **554.00** | 554 | 0 | **554.00** | 554 | 0 | **554.00** | 554 | 0.00 |
| | 20 | 3 | 0 | **0.00** | 0 | 0 | **0.00** | 0 | 0 | **0.00** | 0 | 0.00 |
| Airland7 | 21 | 1 | 1550 | **1550.00** | 1550 | 0 | **1550.00** | 1816.7 | 141.49 | 1903 | 1903 | 0.00 |
| | 22 | 2 | 0 | **0.00** | 0 | 0 | **0.00** | 0 | 0 | **0.00** | 0 | 0.00 |
| Airland8 | 23 | 1 | 1950 | **1950.00** | 1982.5 | 19.36 | 1968 | 2045 | 120.35 | 1990 | 2593.5 | 861.95 |
| | 24 | 2 | 135 | **135.00** | 135 | 0 | 150 | 232.5 | 92.14 | 160 | 367.5 | 219.83 |
| | 25 | 3 | 0 | **0.00** | 0 | 0 | **0.00** | 20.5 | 28.91 | **0.00** | 72.5 | 67.67 |
| Airland9 | 26 | 1 | 5611.7 | **5685.86** | 5711.6 | 20.33 | 5819.08 | 5999.89 | 137.75 | 5901.22 | 6363.9 | 410.29 |
| | 27 | 2 | 452.92 | **449.11** | 451.54 | 1.18 | 450.09 | 507.4 | 52.96 | 488.58 | 601.79 | 64.36 |
| | 28 | 3 | 75.75 | **75.75** | 75.75 | 0 | 75.79 | 86.65 | 12.96 | **75.75** | 94.47 | 17.00 |
| | 29 | 4 | 0 | **0** | 0 | 0 | **0** | 0.42 | 1.32 | **0** | 4.78 | 13.35 |
| Airland10 | 30 | 1 | 12,329.31 | **12,527.95** | 12,627.24 | 75.19 | 14,654.16 | 15,895.81 | 723.64 | 13,608.06 | 16,257.52 | 1343.83 |
| | 31 | 2 | 1288.73 | **1154.10** | 1170.97 | 10.22 | 1280.74 | 1407.5 | 114.51 | 1244.26 | 1450.31 | 157.85 |
| | 32 | 3 | 220.79 | **206.56** | 210.4 | 3.1 | 211.97 | 271.02 | 50.01 | 217.17 | 286.33 | 73.32 |
| | 33 | 4 | 34.22 | **34.22** | 34.22 | 0 | **34.22** | 40.17 | 11.93 | **34.22** | 55.51 | 29.00 |
| | 34 | 5 | 0 | **0.00** | 0 | 0 | **0.00** | 0 | 0 | **0.00** | 0 | 0.00 |
| Airland11 | 35 | 1 | 12,418.32 | **12,870.47** | 13,038.43 | 60.52 | 13,550.24 | 13,992.44 | 261.27 | 13,148.27 | 14,015.27 | 838.87 |
| | 36 | 2 | 15,40.84 | **1344.96** | 1360.25 | 6.74 | 1508.17 | 1634.5 | 107.11 | 1470.21 | 1648.68 | 173.46 |
| | 37 | 3 | 280.82 | **253.07** | 255.73 | 1.53 | 269.51 | 291.31 | 28.18 | 253.59 | 308.93 | 48.99 |
| | 38 | 4 | 54.53 | **54.53** | 54.53 | 0 | **54.53** | 62.48 | 17.89 | **54.53** | 62.71 | 15.44 |
| | 39 | 5 | 0 | **0.00** | 0 | 0 | **0.00** | 0 | 0 | **0.00** | 0 | 0.00 |
| Airland12 | 40 | 1 | 16,209.78 | **16,675.64** | 16,744.78 | 49.23 | 17,796.31 | 18,077.77 | 324.74 | 17,332.99 | 18,107.26 | 643.01 |
| | 41 | 2 | 1961.39 | **1723.29** | 1737.97 | 8.77 | 1901.5 | 2030.39 | 118.64 | 1905.6 | 2024.4 | 96.30 |
| | 42 | 3 | 290.04 | **221.97** | 226.87 | 2.98 | 233.72 | 309.73 | 67.65 | 264.21 | 337.49 | 87.22 |
| | 43 | 4 | 3.49 | **2.44** | 2.44 | 0 | **2.44** | 12.33 | 16.38 | **2.44** | 6.55 | 9.44 |
| | 44 | 5 | 0 | **0.00** | 0 | 0 | **0.00** | 0 | 0 | **0.00** | 0 | 0.00 |

**Table 6** (continued)

| Dataset name | Inst. no. | No. of runways | BKV | ISA | | | ILS | | | SA | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Best | Avg. | STD | Best | Avg. | STD | Best | Avg. | STD |
| Airland13 | 45 | 1 | 44,832.38 | **39,419.19** | 40,049.32 | 464.68 | 44,679.7 | 44,796.07 | 64.85 | 44,703.6 | 44,777.06 | 50.51 |
| | 46 | 2 | 5501.96 | **4053.27** | 4077.37 | 19.28 | 5284.68 | 5440.5 | 67.65 | 5424.31 | 5474.28 | 29.69 |
| | 47 | 3 | 1108.51 | **707.97** | 725.24 | 14.08 | 1043.68 | 1088.36 | 20.11 | 993.69 | 1082.68 | 33.43 |
| | 48 | 4 | 188.46 | **90.56** | 94.09 | 3.21 | 124.3 | 185.07 | 51.58 | 117.55 | 164.07 | 19.50 |
| | 49 | 5 | 7.35 | **0.00** | 0 | 0 | **0.00** | 0.84 | 1.78 | **0.00** | 0.9 | 2.04 |

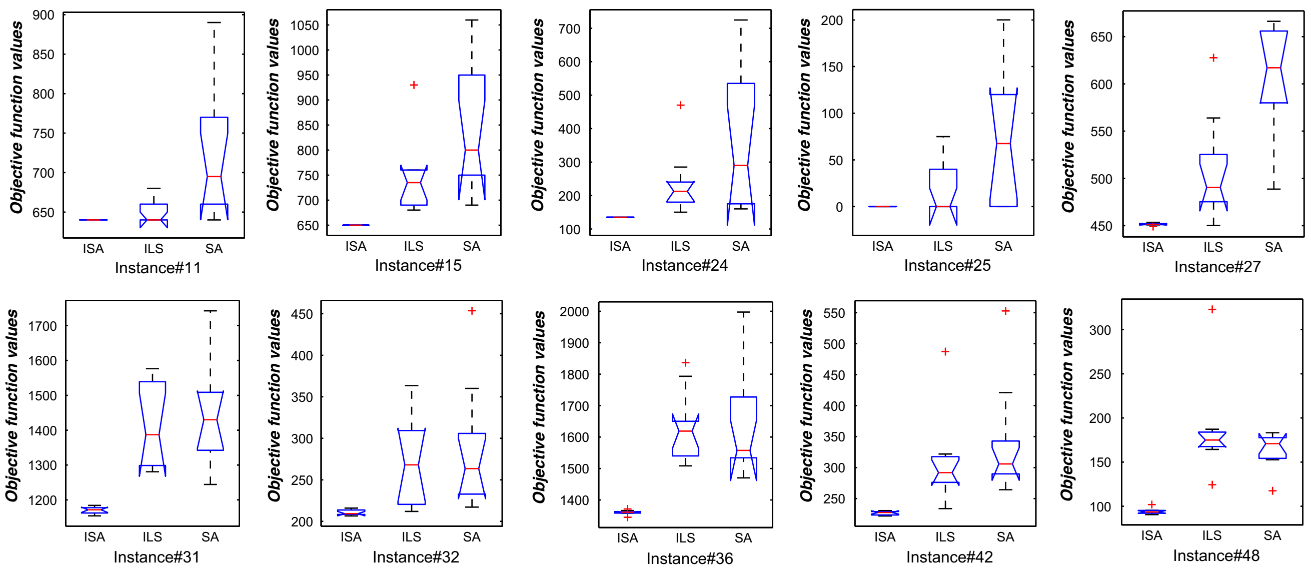

**Fig. 2** Distribution of the best results over 10 runs performed using ISA, ILS and SA algorithms

null hypothesis of equivalent performance between the evaluated methods.

As shown in Table 11, Holm's procedure reveals that the control method (HPSO) is statistically better than SA1, SA2, SS and BA. On the other hand, there is no significant difference between the performance of HPSO and both of ISA and ILS methods. Based on the results obtained from Holm's method, we can conclude that the performance of the proposed ISA method is highly comparable to other competitors in the literature, and can be considered as a viable alternative in tackling the ALP.

The results show that ISA presented a high level of performance as the instance size increased. This reveals that the ISA is well behaved on large instances. As a result, these findings confirm that the integration of SA with ILS contributes effectively to attaining optimal solutions on both small- and large-sized instances. In short, the results achieved by ISA assess the benefits of the hybridization of ILS and SA algorithms to accommodate a high degree of efficiency in addressing the ALP. This confirms that the aircraft landing scheduling approach described here is representative of various lines of research, favorable in terms of reported performance, and manages a good statement of progress for intelligent approaches to be clearly used in processing the ALP.

## 5.6 Computational time

It is not practical to compare the computational resources of ISA with other ALP-based methods as no standard tests exist and it is difficult to quantitatively compare time efforts among different scheduling approaches that use different machine specifications. In addition, a direct comparison is problematic because specific information related to software requirements is often not given. Moreover, details such as operating system, programming language, professionalism and the programming skills of the programmer, the compiler and number of iterations are often not specified. However, the computational time required to execute each problem instance using ISA is reported in Table 12 in comparison with that obtained by comparative methods available to the authors.

Table 12 shows that the mean computational time of ISA is almost better than that required for other methods.

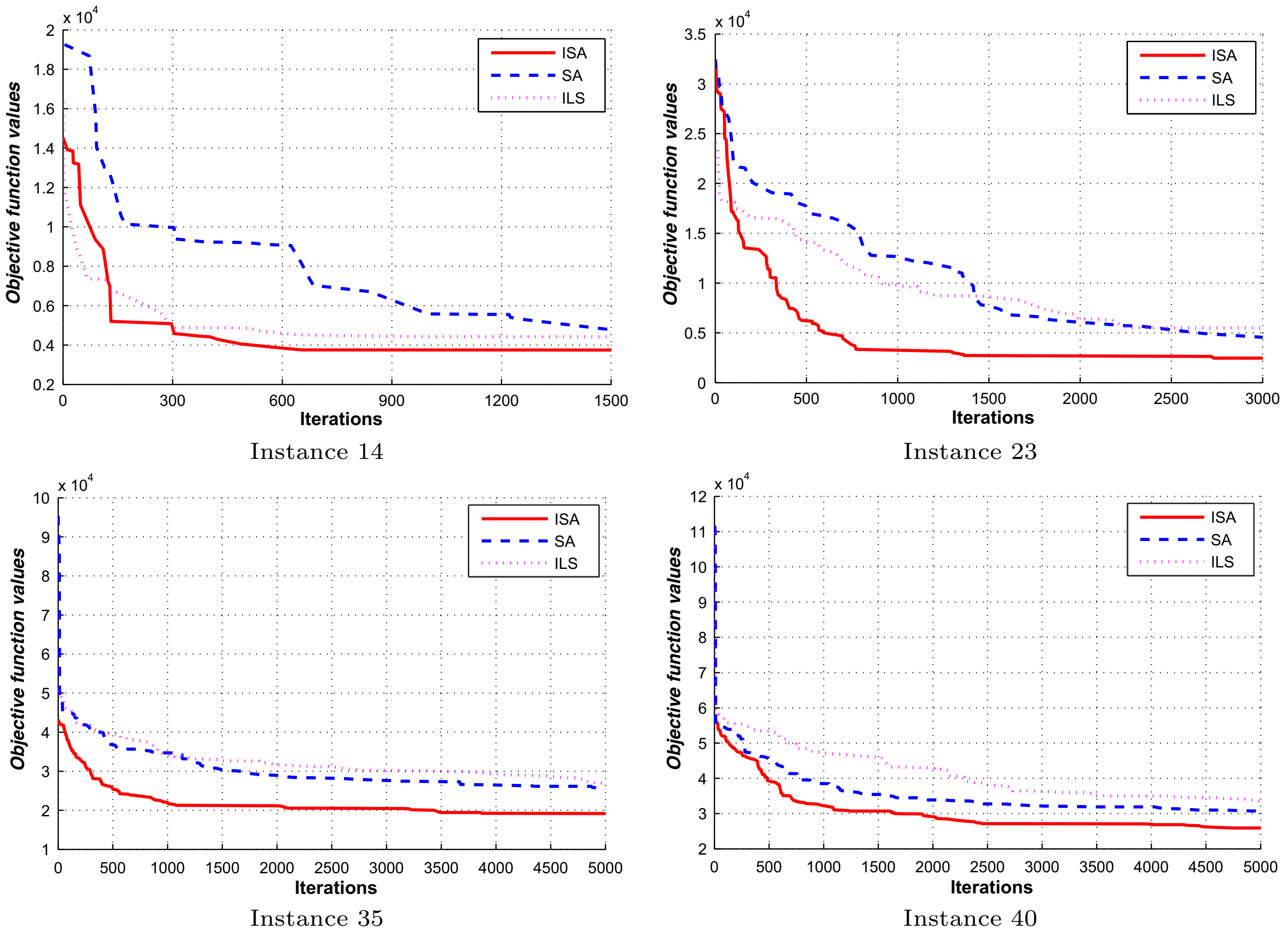

**Fig. 3** Convergence behavior of ISA, ILS and SA

**Table 7** Parameter values of the competitor algorithms

| Abbreviations | Method name | Parameter setting |
|---|---|---|
| ILS | Iterated local search method [27] | MaxIter = 150, minimum time varying = 0.1, maximum time varying = 0.9, the size of RS in GR = 10% of aircraft |
| HPSO | Hybrid particle swarm optimization with rolling horizon framework [3] | $W = 30, R = 10, \alpha = 1.6$, swarm_size = 16, num_iter = 40 |
| SS | Scatter search method [5] | Not available |
| BA | Bionomic method [5] | Not available |
| SA1 | A hybrid simulated annealing and variable neighborhood descent method [9] | $t_0$ = 50, $c = 0.01$, maximum iterations = $\infty$ |
| SA2 | A hybrid simulated annealing and variable neighborhood search method [9] | $t_0$ = 50, $c = 0.01$, maximum iterations = $\infty$ |

This concur that the computational effort of ISA is convinced within the range of the computational efforts of comparative methods. This emphasizes that ISA is a computationally efficient method for processing ALP which produces high-quality solutions in a reasonable computational time. In a nutshell, Table 12 affirms that the ISA method is capable of handling large-sized instances at very high speed on a computer with modest specifications such as Intel Core I5 processor running at Lenovo laptop with 6 GB RAM, running Windows operating system, on Java platform.

**Table 8** A comparison between the computational results of the ISA method with the results of the BKV and the best-known results of the best existing performing methods for ALP on small-sized instances

| Dataset name | Inst. no. | No. of runways | BKV | ISA$\Delta$(%) | HPSO $\Delta$(%) | ILS $\Delta$(%) | SS $\Delta$(%) | BA $\Delta$(%) | SA1 $\Delta$(%) | SA2 $\Delta$(%) |
|---|---|---|---|---|---|---|---|---|---|---|
| Airland1 | 1 | 1 | 700 | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| | 2 | 2 | 90 | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| | 3 | 3 | 0 | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| Airland2 | 4 | 1 | 1480 | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| | 5 | 2 | 210 | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| | 6 | 3 | 0 | **0** | **0** | **0** | **0** | **0** | 100 | 100 |
| Airland3 | 7 | 1 | 820 | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| | 8 | 2 | 60 | **0** | **0** | **0** | **0** | **0** | 16.66 | 16.66 |
| | 9 | 3 | 0 | **0** | **0** | **0** | **0** | **0** | 100 | 100 |
| Airland4 | 10 | 1 | 2520 | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| | 11 | 2 | 640 | **0** | **0** | **0** | **0** | **0** | 3.12 | 3.12 |
| | 12 | 3 | 130 | **0** | **0** | **0** | **0** | **0** | 23.07 | 27.07 |
| | 13 | 4 | 0 | **0** | **0** | **0** | **0** | **0** | 100 | 100 |
| Airland5 | 14 | 1 | 3100 | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| | 15 | 2 | 650 | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| | 16 | 3 | 170 | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| | 17 | 4 | 0 | **0** | **0** | **0** | **0** | **0** | 100 | 100 |
| Airland6 | 18 | 1 | 24,442 | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| | 19 | 2 | 554 | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| | 20 | 3 | 0 | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| Airland7 | 21 | 1 | 1550 | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| | 22 | 2 | 0 | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| Airland8 | 23 | 1 | 1950 | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| | 24 | 2 | 135 | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| | 25 | 3 | 0 | **0** | **0** | **0** | **0** | **0** | 100 | 100 |

## 6 Conclusion and future work

This work has evinced the use of a hybridization of iterated local search (ILS) and simulated annealing (SA) in a single framework to develop a rapid and adequate optimization method to solve the aircraft landing problem (ALP) for 49 benchmark problem instances. This single optimization framework is referred to as iterated simulated annealing (ISA) and has been adopted to solve the ALP, which aims to schedule each incoming aircraft subjected to land on a runway within a predetermined time frame for landing. The key aims of ILS and SA in ISA are to navigate large search space and circumvent local optima in order to improve the ultimate outcomes of ALP. The flexibility of this ISA scheme in reliably addressing ALP is shown for benchmark problem instances with problems of various sizes and number of aircraft and runways in each problem. Specifically, the standard problem instances under study were spanned over thirteen datasets split into small-sized and large-sized groups. De facto, this scheduling problem could be considered as a combinatorial optimization problem that cannot be easily tackled by basic algorithms.

The statistical results (best, average and standard deviation) compared to the best-known values showed a high degree of performance and reliability. Convergence curves were presented to demonstrate the capacity of the proposed ISA algorithm to provide sensible solution for the ALP. Experimental results show that the ISA approach positively outperforms persuasive state-of-the-art methods on large-sized instances. Astonishingly, it arrived at new best results for 4 large-sized instances out of 24 instances and reached the best-known results in all small-sized instances using little computational efforts.

There are several trends for further work that could be considered:

- The assessment of adaptability to various datasets with high levels of complexity and for datasets with very large-sized problem instances.
- The evaluation of adaptability to address a dynamic case of ALP with instances of various structures that

Table 9 A comparison between the computational results of the ISA with the BKV results and the results of the best performing methods for ALP in the literature on large-sized instances

| Dataset name | Inst. no. | No. of runways | BKV | ISA $\Delta$(%) | HPSO $\Delta$(%) | ILS $\Delta$(%) | SS $\Delta$(%) | BA $\Delta$(%) | SA1 $\Delta$(%) | SA2 $\Delta$(%) |
|---|---|---|---|---|---|---|---|---|---|---|
| Airland9 | 26 | 1 | 5611.7 | 1.15 | **0** | **0** | 30.06 | 14.51 | 8.55 | 8.55 |
| | 27 | 2 | 452.92 | − 0.84 | − **1.95** | − 1.74 | 5.67 | 54.73 | − 0.58 | 0 |
| | 28 | 3 | 75.75 | 0 | 0 | − **2.31** | 0 | 87.46 | 0 | 0 |
| | 29 | 4 | 0 | **0** | **0** | **0** | **0** | – | **0** | **0** |
| Airland10 | 30 | 1 | 12,329.3 | 1.61 | − **0.3** | − 0.06 | 44.96 | 33.9 | 0 | 0 |
| | 31 | 2 | 1288.73 | − 10.45 | − **11.25** | − 1.37 | 7.87 | 25.95 | − 5.39 | 0 |
| | 32 | 3 | 220.79 | − 6.45 | − 7.06 | − **9.41** | 8.88 | 195.88 | − 6.49 | 0 |
| | 33 | 4 | 34.22 | 0 | 0 | − **6.16** | 16.74 | 292.4 | 3.09 | 3.09 |
| | 34 | 5 | 0 | **0** | **0** | **0** | **0** | – | 100 | 100 |
| Airland11 | 35 | 1 | 12,418.3 | 3.6 | 0 | − **0.05** | 17.95 | 16.67 | 0 | 0 |
| | 36 | 2 | 1540.84 | − 12.71 | − **13.62** | − 8.49 | 9.19 | 38.54 | − 8.04 | 0 |
| | 37 | 3 | 280.82 | − **9.88** | − **9.88** | − 3.46 | 21.59 | 290.09 | − 2.81 | 0 |
| | 38 | 4 | 54.53 | 0 | 0 | − **6.47** | 2.77 | 474.74 | 0 | 0 |
| | 39 | 5 | 0 | **0** | **0** | **0** | **0** | – | **0** | **0** |
| Airland12 | 40 | 1 | 16,209.8 | 2.66 | − **0.54** | 0 | 22.15 | 23.58 | 0 | 0 |
| | 41 | 2 | 1961.39 | − 12.14 | −**13.55** | 0 | 18.8 | 50.18 | 0 | 0 |
| | 42 | 3 | 290.04 | − **23.47** | − **23.47** | − 6.21 | 17.48 | 198.01 | − 3.56 | 0 |
| | 43 | 4 | 3.49 | − **30.09** | − **30.09** | − 2.57 | 271.63 | 13,216.91 | 0 | 0 |
| | 44 | 5 | 0 | **0** | **0** | **0** | **0** | – | **0** | **0** |
| Airland13 | 45 | 1 | 44,832.4 | − 12.07 | −**17.33** | − 7.7 | 3.24 | 1.03 | − 7.54 | 0 |
| | 46 | 2 | 5501.96 | − 26.66 | −**28.75** | − 0.79 | 3.72 | 37.47 | − 0.47 | 0 |
| | 47 | 3 | 1108.51 | − 37.14 | −**39.21** | 0 | 1.98 | 182.69 | − 32.79 | 0 |
| | 48 | 4 | 188.46 | − 51.95 | −**52.27** | − 50.71 | 22.98 | 1188.81 | − 46.62 | 0 |
| | 49 | 5 | 7.35 | − **100** | − **100** | − 59.18 | 0 | 22,308.44 | − 48.16 | 0 |

Table 10 Average ranking of the comparative methods calculated by Friedman's test

| Algorithm | Ranking |
|---|---|
| HPSO | 2.06 |
| ILS | 2.79 |
| ISA | 3.00 |
| SA1 | 3.98 |
| SA2 | 4.56 |
| SS | 5.77 |
| BA | 5.83 |

possess additional features such as a combination of takeoff and landing on the same or on different runways

- Further work could be investigated to assess the suitability of the ISA method to schedule other problems in a range of scalability and complexity such as course time-tabling [35], examination time-tabling [36], Nurse restoring [37], multiple-reservoir scheduling [38], patient admission scheduling problems [39], economic load dispatch [40] and traveling salesman problem [41].

Table 11 Adjusted $p$ value of the Holm procedure table for $\alpha = 0.05$ (Friedman)

| $i$ | Algorithm | Adjusted $p$-value of Holm's procedure | $\alpha \div i$ | Null hypothesis |
|---|---|---|---|---|
| 6 | BA | 1.478E−9 | 0.008 | Rejected |
| 5 | SS | 2.738E−9 | 0.010 | Rejected |
| 4 | SA2 | 6.100E−5 | 0.013 | Rejected |
| 3 | SA1 | 0.002 | 0.017 | Rejected |
| 2 | ISA | 0.132 | 0.025 | Not rejected |
| 1 | ILS | 0.242 | 0.050 | Not rejected |

**Table 12** Computational time (in seconds) of ISA and the presented state-of-the-art methods that addressed ALP over all instances

| Dataset name | Inst. no. | No. of runways | HPSO | ISA | ILS | SS | BA | SA1 | SA2 |
|---|---|---|---|---|---|---|---|---|---|
| Airland1 | 1 | 1 | 0.01 | 0.002 | 0 | 4 | 60 | 0 | 0 |
| | 2 | 2 | 0.01 | 0 | 0 | 24 | 45 | 0 | 0 |
| | 3 | 3 | 0.01 | 0 | 0 | 39 | 34 | 0 | 0 |
| Airland2 | 4 | 1 | 0.03 | 0.007 | 0 | 45 | 49 | 1.66 | 1.65 |
| | 5 | 2 | 0.03 | 0 | 0 | 45 | 49 | 1.66 | 1.65 |
| | 6 | 3 | 0.03 | 0 | 0 | 46 | 43 | 1.98 | 1.91 |
| Airland3 | 7 | 1 | 0.07 | 0.001 | 0 | 8 | 99 | 1.78 | 1.73 |
| | 8 | 2 | 0.06 | 0 | 0.8 | 48 | 58 | 3.12 | 4.22 |
| | 9 | 3 | 0.06 | 0 | 0.1 | 62 | 63 | 3.29 | 5.11 |
| Airland4 | 10 | 1 | 0.1 | 0.009 | 1.7 | 8 | 95 | 1,98 | 2.85 |
| | 11 | 2 | 0.1 | 0.001 | 1.9 | 52 | 55 | 3.56 | 3.94 |
| | 12 | 3 | 0.1 | 0 | 2 | 46 | 57 | 3.74 | 5.05 |
| | 13 | 4 | 0.1 | 0 | 2.3 | 56 | 52 | 4.06 | 7.15 |
| Airland5 | 14 | 1 | 0.13 | 0.005 | 1.3 | 9 | 100 | 1.85 | 1.89 |
| | 15 | 2 | 0.22 | 0.008 | 2.4 | 50 | 61 | 3.04 | 4.84 |
| | 16 | 3 | 0.41 | 0.001 | 3.7 | 54 | 43 | 4.11 | 4.92 |
| | 17 | 4 | 0.11 | 0 | 3.1 | 56 | 68 | 4.35 | 3.04 |
| Airland6 | 18 | 1 | 0.22 | 0 | 1.7 | 158 | 274 | 2.12 | 2.14 |
| | 19 | 2 | 0.24 | 0.001 | 2.6 | 70 | 101 | 3.98 | 4.01 |
| | 20 | 3 | 0.13 | 0.0 | 2.5 | 54 | 87 | 4.41 | 5.91 |
| Airland7 | 21 | 1 | 1.0 | 0.013 | 1.8 | 195 | 79 | 2.68 | 2.65 |
| | 22 | 2 | 0.18 | 0.0 | 1.6 | 118 | 124 | 2.83 | 2.37 |
| Airland8 | 23 | 1 | 0.76 | 0.148 | 4.8 | 42 | 287 | 7.1 | 7.31 |
| | 24 | 2 | 0.72 | 0.208 | 6.2 | 121 | 196 | 10.73 | 9.85 |
| | 25 | 3 | 0.62 | 0.002 | 9.5 | 139 | 181 | 14.11 | 17.39 |
| Airland9 | 26 | 1 | 5.8 | 25.032 | 7.6 | 119 | 554 | 11.59 | 10.12 |
| | 27 | 2 | 4.1 | 20.641 | 11.4 | 342 | 487 | 13.78 | 13.64 |
| | 28 | 3 | 75.75 | 14.094 | 10.9 | 390 | 466 | 17.95 | 18.46 |
| | 29 | 4 | 0.0 | 8.907 | 13.7 | 336 | 439 | 19.69 | 21.18 |
| Airland10 | 30 | 1 | 18.8 | 44.407 | 14.33 | 227 | 925 | 20.12 | 20.75 |
| | 31 | 2 | 13.7 | 28.297 | 15.6 | 608 | 845 | 21.33 | 22.04 |
| | 32 | 3 | 7.0 | 26.641 | 17.3 | 668 | 803 | 27.62 | 25.19 |
| | 33 | 4 | 5.3 | 26.25 | 22.7 | 647 | 788 | 30.12 | 41.28 |
| | 34 | 5 | 5.5 | 20.688 | 34.3 | 607 | 762 | 39.85 | 40.15 |
| Airland11 | 35 | 1 | 15.6 | 58.657 | 18.4 | 256 | 1417 | 24.17 | 33.84 |
| | 36 | 2 | 11.5 | 41.797 | 21.7 | 959 | 1287 | 29.09 | 33.99 |
| | 37 | 3 | 7.3 | 36.922 | 34.2 | 1021 | 1203 | 41.22 | 37.19 |
| | 38 | 4 | 7.5 | 37 | 37.1 | 993 | 1168 | 42.4 | 45.96 |
| | 39 | 5 | 8.2 | 26.985 | 54.8 | 956 | 1158 | 66.23 | 61.05 |
| Airland12 | 40 | 1 | 35.1 | 73.828 | 197.7 | 381 | 2011 | 219.03 | 198.85 |
| | 41 | 2 | 21.7 | 55.141 | 310.4 | 1266 | 1835 | 362.6 | 313.46 |
| | 42 | 3 | 15.7 | 49.828 | 401.5 | 1454 | 1710 | 412.73 | 379.91 |
| | 43 | 4 | 10.9 | 50.454 | 398.1 | 1445 | 1688 | 410.22 | 401.04 |
| | 44 | 5 | 11.8 | 49.86 | 357.6 | 1386 | 1662 | 394.6 | 386.16 |
| Airland13 | 45 | 1 | 115.1 | 168.97 | 486.4 | 1237 | 5852 | 566.82 | 528.84 |
| | 46 | 2 | 111.7 | 117.329 | 1011.2 | 3836 | 5379 | 1047.93 | 1294.23 |
| | 47 | 3 | 48.3 | 117.141 | 1123.4 | 4560 | 5158 | 1241 | 1334.33 |
| | 48 | 4 | 35.7 | 117.236 | 1181.2 | 4413 | 4977 | 1201.8 | 1197.48 |
| | 49 | 5 | 37.6 | 125.516 | 1152.4 | 4421 | 4887 | 1203.93 | 1185.46 |

- The proposed ISA could be further improved by hybridizing it with other effective features that could derived from promising local search algorithms such as tabu list from tabu search [42], GRASP operators from GRASP [43] and $\beta$ operator from $\beta$-hill climbing [44].

## Compliance with ethical standards

**Conflict of interest** The authors declare that they have no conflict of interest.

## References

1. Bennell JA, Mesgarpour M, Potts CN (2011) Airport runway scheduling. 4OR 9(2):115
2. Faye A (2015) Solving the aircraft landing problem with time discretization approach. Eur J Oper Res 242(3):1028–1038
3. Girish BS (2016) An efficient hybrid particle swarm optimization algorithm in a rolling horizon framework for the aircraft landing problem. Appl Soft Comput 44:200–221
4. Briskorn D, Stolletz R (2014) Aircraft landing problems with aircraft classes. J Sched 17(1):31–45
5. Pinol H, Beasley JE (2006) Scatter search and bionomic algorithms for the aircraft landing problem. Eur J Oper Res 171(2):439–462
6. Beasley JE, Krishnamoorthy M, Sharaiha YM, Abramson D (2004) Displacement problem and dynamically scheduling aircraft landings. J Oper Res Soc 55(1):54–64
7. Lieder A, Briskorn D, Stolletz R (2015) A dynamic programming approach for the aircraft landing problem with aircraft classes. Eur J Oper Res 243(1):61–69
8. Lieder A, Stolletz R (2016) Scheduling aircraft take-offs and landings on interdependent and heterogeneous runways. Transp Res Part E Logist Transp Rev 88(Supplement C):167–188
9. Salehipour A, Modarres M, Naeni LM (2013) An efficient hybrid meta-heuristic for aircraft landing problem. Comput Oper Res 40(1):207–213
10. Furini F, Kidd MP, Persiani CA, Toth P (2015) Improved rolling horizon approaches to the aircraft sequencing problem. J Sched 18(5):435–447
11. Beasley JE, Krishnamoorthy M, Sharaiha YM, Abramson D (2000) Scheduling aircraft landings—the static case. Transp Sci 34(2):180–197
12. Awasthi A, Kramer O, Lassig J (2013) Aircraft landing problem: an efficient algorithm for a given landing sequence. In: 2013 IEEE 16th international conference on computational science and engineering, pp 20–27
13. DÁpice C, De Nicola C, Manzo R, Moccia V (2014) Optimal scheduling for aircraft departures. J Ambient Intell Humani Comput 5(6):799–807
14. Farhadi F (2016) Heuristics and meta-heuristics for runway scheduling problems. In: Rabadi G (ed) Heuristics, metaheuristics and approximate methods in planning and scheduling. Springer International Publishing, Switzerland, pp 141–163
15. Osman IH, Laporte G (1996) Metaheuristics: a bibliography. Ann Oper Res 63(5):511–623
16. Blum C, Roli A (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. ACM Comput Surv: CSUR 35(3):268–308
17. Črepinšek M, Liu S-H, Mernik M (2013) Exploration and exploitation in evolutionary algorithms: a survey. ACM Comput Surv 45(3):35:1–35:33
18. Capri S, Ignaccolo M (2004) Genetic algorithms for solving the aircraft-sequencing problem: the introduction of departures into the dynamic model. J Air Transp Manag 10(5):345–351
19. Xiao-Bing H, Chen W-H (2005) Genetic algorithm based on receding horizon control for arrival sequencing and scheduling. Eng Appl Artif Intell 18(5):633–642
20. Beasley JE, Sonander J, Havelock P (2001) Scheduling aircraft landings at London Heathrow using a population heuristic. J Oper Res Soc 52:483–493
21. Farah I, Kansou A, Yassine A, Galinho T (2011) Ant colony optimization for aircraft landings. In: 2011 4th international conference on logistics, pp 235–240
22. Ma W, Bo X, Liu M, Huang H (2014) An efficient approximation algorithm for aircraft arrival sequencing and scheduling problem. Math Probl Eng 2014:1–8
23. Ng KKH, Lee CKM, Chan FTS, Qin Y (2017) Robust aircraft sequencing and scheduling problem with arrival/departure delay using the min–max regret approach. Transp Res Part E Logist Transp Rev 106:115–136
24. Dastgerdi K, Mehrshad N, Farshad M (2016) A new intelligent approach for air traffic control using gravitational search algorithm. Sadhana 41(2):183–191
25. Al-Betar MA, Khader AT, Doush IA (2014) Memetic techniques for examination timetabling. Ann OR 218(1):23–50
26. Van Laarhoven PJM, Aarts EHL (eds) (1987) Simulated annealing. In: Simulated annealing: theory and applications. Springer, Netherlands pp 7–15
27. Sabar NR, Kendall G (2015) An iterated local search with multiple perturbation operators and time varying perturbation strength for the aircraft landing problem. Omega 56:88–98
28. Martin OC, Otto SW (1996) Combining simulated annealing with local search heuristics. Ann Oper Res 63(1):57–75
29. Rajalakshmi K, Kumar P, Bindu HM (2010) Hybridizing iterative local search algorithm for assigning cells to switch in cellular mobile network. Int J Soft Comput 5(1):7–12
30. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82
31. Stútzle T (2006) Iterated local search for the quadratic assignment problem. Eur J Oper Res 174(3):1519–1539
32. Kirkpatrick S, Gelatt CD, Vecchi MP et al (1983) Optimization by simulated annealing. Science 220(4598):671–680
33. Beasley JE (1990) Or-library: distributing test problems by electronic mail. J Oper Res Soc 41(11):1069–1072
34. García S, Fernández A, Luengo J, Herrera F (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. Inf Sci 180(10):2044–2064
35. Al-Betar MA, Khader AT (2012) A harmony search algorithm for university course timetabling. Ann Oper Res 194(1):3–31
36. Al-Betar MA, Khader AT, Doush IA (2014) Memetic techniques for examination timetabling. Ann Oper Res 218(1):23–50
37. Awadallah MA, Bolaji AL, Al-Betar MA (2015) A hybrid artificial bee colony for a nurse rostering problem. Appl Soft Comput 35:726–739
38. Alsukni E, Arabeyyat OS, Awadallah MA, Alsamarraie L, Abu-Doush I, Al-Betar MA (2019) Multiple-reservoir scheduling using $\beta$-hill climbing algorithm. J Intell Syst 28(4):559–570
39. Hammouri AI, Alrifai B (2014) Investigating biogeography-based optimisation for patient admission scheduling problems. J Theor Appl Inf Technol 70(3):413–421
40. Sheta A, Faris H, Braik M, Mirjalili S (2020) Nature-inspired metaheuristics search algorithms for solving the economic load dispatch problem of power system: a comparison study. In: Dey

N, Ashour AS, Bhattacharyya S (eds) Applied nature-inspired computing: algorithms and case studies. Springer, Singapore, pp 199–230

41. Hammouri AI, Samra ETA, Al-Betar MA, Khalil RM, Alasmer Z, Kanan M (2018) A dragonfly algorithm for solving traveling salesman problem. In: 2018 8th IEEE international conference on control system, computing and engineering (ICCSCE). IEEE, pp 136–141
42. Glover F (1990) Tabu search: a tutorial. Interfaces 20(4):74–94
43. Resende MGC, Velarde JLG (2003) Grasp: Greedy randomized adaptive search procedures. Intel Artif Rev Iberoam Intel Artif 19(1):61–76
44. Al-Betar MA (2017) $\beta$-hill climbing: an exploratory local search. Neural Comput Appl 28(1):153–168