Omega 56 (2015) 88–98

Contents lists available at ScienceDirect

Omega

journal homepage: www.elsevier.com/locate/omega

An iterated local search with multiple perturbation operators and time varying perturbation strength for the aircraft landing problem ☆

Nasser R. Sabar [a,*], Graham Kendall [a,b]

[a] ASAP Research Group, The University of Nottingham Malaysia Campus, Jalan Broga, Semenyih 43500, Selangor, Malaysia.
[b] ASAP Research Group, The University of Nottingham, Nottingham, UK

ARTICLE INFO

Article history:
Received 5 March 2014
Accepted 18 March 2015
Available online 28 March 2015

Keywords:
Aircraft landing problem
Iterated local search
Meta-heuristics
Airport operation

ABSTRACT

Landing aircraft safely is an important operation that air traffic controllers have to deal with on a daily basis. For each arriving aircraft a runway and a landing time must be allocated. If these allocations can be done in an efficient way, it could give the airport a competitive advantage. The Aircraft Landing Problem (ALP) aims to minimize the deviation from a preferred target time of each aircraft. It is an NP-hard problem, meaning that we may have to resort to heuristic methods as exact methods may not be suitable, especially as the problem size increases. This paper proposes an iterated local search (ILS) algorithm for the ALP. ILS is a single solution based search methodology that successively invokes a local search procedure to find a local optimum solution. A perturbation operator is used to modify the current solution in order to escape from the local optimum and to provide a new solution for the local search procedure. As different problems and/or instances have different characteristics, the success of the ILS is highly dependent on the local search, the perturbation operator(s) and the perturbation strength. To address these issues, we utilize four perturbation operators and a time varying perturbation strength which changes as the algorithm progresses. A variable neighborhood descent algorithm is used as our local search. The proposed ILS generates high quality solutions for the ALP benchmark instances taken from the scientific literature, demonstrating its efficiency in terms of both solution quality and computational time. Moreover, the proposed ILS produces new best results for some instances.

1. Introduction

The last decade has seen a growing demand for air transportation [1,2]. This continual increase stretches airport capacity, ultimately leading to airports being unable to cope with future demand. Given the lead time in building a new airport, the controversy that often surrounds new airports (or even new runways), the costs involved and, perhaps, the lack of space to extend means that it may not be possible to simply build our way out of trouble [1,2]. Therefore, the industry has to find other ways to meet the ever increasing demand for air travel. The Aircraft Landing Problem (ALP) is an important element in airport operations [1–4]. Efficient solution methodologies for tackling the ALP is important, both from an economical and environmental perspective [1–4].

The ALP was introduced by Beasley et al. [3]. It assigns each arriving aircraft a runway and a landing time. Each plane is required to land within a given time window. The goal is to minimize the sum of the penalties incurred when an aircraft lands before or after its preferred target time. ALP can be considered as a combination of two sub-problems [5]: a sequencing problem (which decides the sequence of aircraft landings) and a scheduling problem (which decides the landing times for each aircraft in the generated sequence). In addition, Beasley et al. [3] also highlighted that the practical complexities of ALP are mainly due to the inclusion of additional constraints and considerations such as the control, separation times, latest times, runway allocation and additional terms in the objective function. The ALP can be formulated as a machine job scheduling problem with sequence-dependent processing times, where penalties are added for earliness and tardiness violations [5], representing the earliest and latest landing times. ALP is an NP-hard optimization problem, making it more and more difficult to solve to optimality as the problem size increases [2,3]. It may therefore be necessary to utilize heuristic or meta-heuristic approaches [6].

Beasley et al. [3] presented eight small-sized ALP instances that include single and multiple runways and proposed a mixed-integer program to solve them. They also applied a first-come-first-serve (FCFS) rule and obtained the optimal solutions for instances containing up to 50 aircraft. Pinol et al. [6] presented two meta-heuristic algorithms for ALP, scatter search and a bionomic algorithm, as well as large-scaled ALP instances that involve up to 500 aircraft and five runways. Both algorithms were tested on small and large-sized instances and they

☆ This manuscript was processed by Associate Editor Pesch.

* Corresponding author.
E-mail addresses: Nasser.Sabar@nottingham.edu.my (N.R. Sabar), Graham.Kendall@nottingham.edu.my (G. Kendall).

http://dx.doi.org/10.1016/j.omega.2015.03.007
0305-0483/

managed to find the optimal solutions for instances with up to 50 aircraft. However, the performance markedly decreased as the problem size increased. Salehipour et al. [7] presented a mixed-integer goal programming approach and a hybrid meta-heuristic that combined simulated annealing with variable neighborhood descent and variable neighborhood search for the ALP. The proposed mixed-integer goal programming approach was able to provide optimal solutions for instances with up to 100 aircraft. For larger instances, with up to 500 aircraft, the proposed hybrid meta-heuristic produced good results when compared to existing methodologies. Comprehensive reviews on recent methods in airport runway optimization can be found in [1,2].

Mixed-integer programming formulations and meta-heuristic approaches have been able to find good quality solutions for the ALP, even obtaining optimal solutions for small instances. However, performance deteriorates on larger instances. This suggests that further research on ALP might be useful and may improve on the best known results. In this work, we propose an Iterated Local Search (ILS) algorithm for the ALP. ILS is an iterative, single solution based meta-heuristic [8,9]. ILS extends classical local search algorithms by including a diversification mechanism, the idea being to perform a randomized walk in the neighborhood of the current local optimum to generate a new starting solution instead of generating a new solution from scratch [8]. ILS iteratively invokes a local search procedure (to reach a local optimum solution) and a perturbation operator (to modify the current solution in order to escape from the local optimum) for a predefined number of iterations. Like many other meta-heuristic methodologies [19-22] ILS is based on a simple framework, yet it has been shown to be an effective and efficient solution methodology for many real world problems [8,9].

We were also motivated by the fact that meta-heuristic frameworks are very adaptable, enabling other meta-heuristic algorithms such as variable neighborhood search to be easily hybridized with ILS [10]. In addition, the ILS designer has several degrees of freedom to select the appropriate ILS components such as the local search procedure and the perturbation operator. Blum and Roli [10] pointed out that the perturbation operator has a large influence on the performance of ILS and controlling the perturbation strength is quite important. A small perturbation strength may lead the local search to return to previously visited solutions. If the perturbation strength is too large, this may lead the algorithm to behave as a random restart method, which typically leads to low quality solutions. Lourenço et al. [8] suggested that "A good perturbation transforms one excellent solution into an excellent starting point for a local search". Consequently, various ILS variants that use different perturbation operators have been proposed. For example, Katayama and Narihisa [11] used 4-opt with a greedy algorithm as a perturbation mechanism, Thierens [12] proposed a perturbation mechanism that utilizes a population of solutions, Katayama and Narihisa [13] used a crossover operator as a perturbation mechanism for ILS and Zhang et al. [14] used the guided mutation operator as perturbation operator in ILS. More details about ILS variants can be found in [8,9,23,24].

Despite ILS producing very good results for various optimization problems, most existing ILS's use a single perturbation operator and the perturbation strength remains the same during the optimization process. In addition, the success of ILS is highly dependent on the employed local search and the type of perturbation operator. This is because different problems and/or instances possess different characteristics, and therefore require different ILS parameters/configurations. To address these issues, in this work, we propose an ILS with the following components:

i) *Local search phase*: the proposed ILS uses variable neighborhood decent (VND) as a local search. VND escapes from the current local optimum by using a set of neighborhood structures that are applied in a systematic way. The idea is that different neighborhood structures generate different search trajectories.
ii) *Perturbation phase*: in this work, we employ multiple perturbation operators and a time varying perturbation strength. We utilize four different perturbation operators, where each one modifies the current local optimum solution. The time varying perturbation strength changes as the algorithm progresses. The idea is to assign a larger perturbation strength in the early stages of the search, in order to focus on exploring the search space. The perturbation strength is gradually decreased so that we gradually focus more on exploitation.

The 13 small and large ALP benchmark instances introduced in [3] are used to demonstrate the effectiveness of our proposed algorithm. An experimental comparison is conducted to evaluate ILS with, and without, the additional components. Our results demonstrate that ILS, with the additional components, produces very good results across all ALP instances. In addition, the proposed ILS finds new best solutions for some ALP instances when compared to the best known results in the scientific literature.

2. Problem description

ALP is a combinatorial optimization problem and can be defined as follows: given a set of arrival aircraft, each one associated with a target landing time, a predefined time window for landing, and a set of runways, the goal is to assign a runway and a landing time for each aircraft with a minimum total cost deviation from the target landing times, while respecting the following constraints:

- Each aircraft is assigned to only one runway.
- A maximum of one aircraft is assigned to a runway at a specific landing time.
- The landing time of each aircraft should be within the aircraft's landing time window.
- The separation time between two aircraft landing on the same runway should be respected.

A penalty is incurred if the aircraft is scheduled to land before or after its target time. The objective is to minimize the overall penalty by generating the best landing sequence and landing time for the given set of aircraft. The following formulation presents the model more formally (adopted from [3]).

Notation:

- n: the number of the arrival aircraft.
- m: the number of runways.
- s_{ij}: the separation time ($s_{ij} > 0$) between aircraft i and j when they are assigned to the same runway.
- t_{ij}: the separation time between aircraft i and j when they are assigned to different runways.
- T_i: the preferred landing time (target time) of aircraft i.
- E_i: the earliest landing time of aircraft i.
- L_i: the latest landing time of aircraft i.
- $C1_i$: the incurred penalty for late lading of aircraft i.
- $C2_i$: the incurred penalty for early landing of aircraft i.

Decision variables

- x_i: the assigned landing time of aircraft $i (i = 1, 2, \ldots, n)$.
- y_{ij}: equal to 1 if aircraft i is assigned to land before aircraft j. Otherwise it takes 0.
- y_{ir}: equal to 1 if aircraft i is scheduled to land on a runway $r (r = 1, 2, \ldots, m)$. Otherwise, it takes 0.
- δ_{ij}: equal to 1 if aircraft i and j are scheduled to land on the same runway. Otherwise, it takes 0.