INTRODUCTION

Prior to landing, an aircraft must go through an approaching stage directed by the air traffic control (ATC) tower. The ATC gives instructions to the aircraft regarding to the runway, speed and altitude of the aircraft in order to align it with the allocated runway and maintain the safety distance with its preceding aircraft. During peak hours, controllers must handle safely and effectively landings of a continuous flow of aircraft entering the radar range onto

the assigned runway(s). The capacity of runways is highly constrained and this makes the scheduling of landings a difficult task to perform effectively. Increasing the capacity of an airport by building new runways is an expensive and difficult affair. Hence, the air traffic controllers face the problem of allocating a landing sequence and landing times to the aircraft in the radar range. Additionally, in case of airports with multiple runways, they have to take a decision on the runway allotment too, *i.e.* which aircraft lands on which runway. These decisions are made with the availability of certain information about the aircraft in the radar range [[1](#_bookmark15), [2](#_bookmark16), [3](#_bookmark17)]. A target landing time is defined as the time at which an aircraft can land if it flies straight to the runway at its cruise speed (most economical). This target landing time is bounded by an earliest landing time and a latest landing time known as the time window. The earliest landing time is determined as the time at which an aircraft can land if it flies straight to the runway at its fastest speed with no holding, and the latest landing time is determined as the time at which an aircraft can land after it is held for its maximal allowable time before landing. All the aircraft have to land within their time window and there are asymmetric penalties associated with each aircraft for landing earlier or later than its target landing time. Besides, there is another constraint of the safety distance that has to be maintained by any aircraft with its preceding aircraft. This separation is necessary as every aircraft creates a wake vortices at its rear and can cause a serious aerodynamic instability to a closely following aircraft. There are several types of planes which land on a runway hence the safety distance between any two aircraft depends on their types. This safety distance between any two aircraft can be easily converted to a safety time by considering the required separation and their relative speeds. If several runways are available for landing, the application of this constraint for aircraft landing on different runways usually depends upon the relative positions of the runways [[1](#_bookmark15), [2](#_bookmark16), [3](#_bookmark17)]. A formal definition of the ALP is given in Section [3](#_bookmark0).

The objective of the ALP is to minimize the total penalty incurred due to the deviation of the scheduled landing times of all the aircraft with their target landing times. Hence, the air traffic controllers not only have to find suitable landing times for all the aircraft but also a landing sequence so as to reduce the total penalty. This work considers the static case of the aircraft landing problem where the set of aircraft that are waiting to land is already known. For a special but practically very common case of the safety constraint, we present a polynomially bound exact algorithm for optimizing any given feasible landing sequence for the single runway case and an effective strategy for the multiple runway case. In the later part of the paper we present our results for all the benchmark instances provided by Beasley [[4](#_bookmark18)] and compare the results with previous work on this problem.

# 

Related Work

The aircraft landing problem described in this paper was first introduced and studied by

Beasley in the mid-nineties [5]. Since then, it has been studied by several researchers using

different metaheuristics, hybrid metaheuristics, linear programming, variants of exact branch

and bound algorithms etc., for both the static and dynamic cases of the problem. In 1995,

Beasley et al. presented a mixed-integer zero-one formulation of the problem for the single

runway case and later extended it to the multiple runway case [5]. The ALP was studied

for up to 50 aircraft with multiple runways using linear programming based tree search and

an effective heuristic algorithm for the problem. Again in 1995, Abela et al. [6] proposed

2

a genetic algorithm and a branch and bound algorithm to solve the problem of scheduling

aircraft landings. Ernst et al. presented a simplex algorithm which evaluated the landing

times based on some partial ordering information. This method was used in a problem

space search heuristic as well as a branch-and-boundmethod for both, the single and multiple

runway case, for again up to 50 aircraft [2]. Beasley et al. adopted similar methodologies and

presented extended results [1]. In 1998, Ciesielski et al. developed a real time algorithm for

the aircraft landings using a genetic algorithm and performed experiments on landing data

for the Sydney airport on the busiest day of the year [7]. In 2001, Beasley et al. developed

a population heuristic and implemented it on actual operational data related to aircraft

landings at the London Heathrow airport [8]. The dynamic case of the ALP was studied

again by Beasley et al. by expressing it as a displacement problem and using heuristics

and linear programming [9]. In 2006, Pinol and Beasley presented two heuristic techniques,

Scatter Search and the Bionomic Algorithm and published results for the available test

problems involving up to 500 aircraft and 5 runways [3]. The dynamic case of the problem

for the single-runway case was again studied by Moser et al. in 2007 [10]. They used extremal

optimization along with a deterministic algorithm to optimize a landing sequence. In 2008

Tang et al. implemented a multi-objective evolutionary approach to simultaneouslyminimize

the total scheduled time of arrival and the total cost incurred [11]. In 2009, Bencheikh et

al. approached the ALP using hybrid methods combining genetic algorithms and ant colony

optimization by formulating the problem as a job shop scheduling problem [12]. The same

authors presented an ant colony algorithm along with a new heuristic to adjust the landing

times of the aircraft in a given landing sequence in order to reduce the total penalty cost,

in 2011 [13]. In 2012, a hybrid meta-heuristic algorithm was suggested using simulated

# annealing with variable neighbourhood search and variable neighbourhood descent [14].

# Problem Formulation

In this section we give the mathematical formulation of the static aircraft landing problem based on [[3](#_bookmark17)]. We also define some new parameters which are later used in the presented algorithm in the next sections.

Let,

*N* = the number of aircraft,

*Ei* = the earliest landing time for aircraft *i*, *i* = 1*,* 2*, . . . , N* , *Li* = the latest landing time for aircraft *i*, *i* = 1*,* 2*, . . . , N* , *Ti* = the target landing time for aircraft *i*, *i* = 1*,* 2*, . . . , N* , *STi* = the scheduled landing time for aircraft *i*,

*Si,j* = the required separation time between planes *i* and *j*, where plane *i* lands before plane *j* on the same runway, *i* ƒ= *j*,

*si,j* = the required separation time between planes *i* and *j*, where plane *i* lands before plane

*j* on different runways, *i* ƒ= *j*,

*gi* = the penalty cost per time unit associated with plane *i* for landing before *Ti*, *hi* = the penalty cost per time unit associated with plane *i* for landing after *Ti*, *αi* = earliness (time) of plane *i* from *Ti*, *αi* = max{0*, Ti* − *STi*}, *i* = 1*,* 2*, . . . , N* , *βi* = tardiness (time) of plane *i* from *Ti*, *βi* = max{0*, STi* − *Ti*}, *i* = 1*,* 2*, . . . , N* .

The total penalty corresponding to any aircraft *i* is then expressed as *αigi* + *βihi*. If aircraft *i* lands at its target landing time then both *αi* and *βi* are equal to zero and the cost incurred by its landing is equal to zero. However, if aircraft *i* does not land at *Ti*, either *αi* or *βi* is non-zero and there is a strictly positive cost incurred. The objective function of the problem can now be defined as

*N*

min (*αigi* + *βihi*) *.* (1)

Σ

*i*=1

The Solution to single runway problem:

---------------------------------------------------

1. Assign a start time (s)
2. The assumption is the first plane will land at that the start time (s) and it takes a second for landing.

The situation can be relatively changed when being in applied in a real life scenario.

1. Each plane will have a landing window gap and a minimum collision prevention gap which can be modified.

Earliest Landing Time (Ei ) . This will be provided to each plane depending upon starting time and the assumption is that no plane can reach before this time. After the first plane lands the earliest landing time for other planes will be given while takin in the consideration the minimum collision gap.

Latest Landing Time( Li). This Time will depend on Ei and cetail value will be assigned depending on the landing window gap

Collision Prevention Gap: C

1. An Optimal time for each plane to be landed is the mid time between Ei and Li This time is given by O(t).
2. The cost associated with a plane landing before *Ti*, is given by gi and the cost associated with the plane landing after O(t) is given by hi .
3. Obviously in real life scenerios chances are that the plane will not land at the optimal time O(t). The time at which it will be landing is the *STi* = the scheduled landing time.

This time will be taken as input for each plane and depending upon this input new landing windows , minimum collision gaps and Optimal time for next planes will be created.

1. There are also chances that a plane might miss it’s landing window. The plane will then be provided a new scheduled time inbetween the the landing window gaps of next flights.
2. If a plane lands before/after its target time and and no previous planes are in the air then nothing changes and the minimum collision gap is still matained.
3. If a plane lands before/after its target time and some previous plane is in the air then it will be landed and it will be made sure due to this landing the minimum collision gap is still mantained as for example the plane which was in the air is landed just at latest time (Li) of the previously landed plane than the minimum landing window gap and the Earliest Landing time (Ei) of the next plane is also affected and that must not happen.

The Solution to single runway problem:

---------------------------------------------------

Time complexity Analysis:

**For the single runway case:**

**Best case:** This is the case when all the planes arrive on Target time. No plane gets delayed, comes in early or misses its landing window. The time complexity will be Ω(n).

**Average Case:** This is the case when not all but some of the planes arrive on Target time. Some planes may get delayed, came in early or missed their landing window. The time complexity will be (nlogn).

**Worst Case:** This is the case when mostly all of the planes do not arrive on Target time. Planes may get delayed, came in early or missed their landing window. The time complexity will be (n^2). This happens because the schedule rechanged for all of the planes and also new scheduled times have to be given to those planes that missed their landing window. Some if/else checks have to be performed.

**For the multiple runway case:**

It is difficult to determine the complexity in this case as multiple flights will be landing simultaneously with just collision gap between them so as to avoid collision during simultaneous landing. Overall the complexity will shoot up to (n^3) since now we also have take in consideration all the planes landing on all of the runways and also along with all of that of single runway. One more inner for loop will be added. The complexity will fluctuate between O(n) and O(n^3) with a closed bound.