International Journal of Latest Engineering Research and Applications (IJLERA) ISSN: 2455-7137

Volume – 03, Issue – 03, March 2018, PP – 09-12

Hybrid Population Heuristic Method used for solving Aircraft Landing Problem (ALP)

Saïd BOURAZZA

(Mathematics Department, Faculty of Science/University of Jazan, KSA)

Abstract: The Airline Scheduling Problem (ALP) is well known problem in the literature. Many research has been done concerning and several heuristic methods used to solve it with a good results. In this paper, we combine three methods; Genetic Algorithm (GA), Scatter Search (SS) and Bionomic Algorithm (BA) for resolving ALP problem is presented. This new method gives us best results.

Keywords: Airline Scheduling Problem, Bionomic Algorithm, Genetic Algorithm, Scatter Search

I. INTRODUCTION

The Given a set of planes (I = {1, 2, ... , P}) with target landing times Ti and time windows for landings [Ei , Li] and runways (J = {1, 2, ...,R}), the objective of the ALP is to minimize the total (weighted) deviation from the target landing time for each plane. There are costs associated with landing either earlier Av_i or later Ap_i than a target landing time for each plane i ∈ I. Each plane has to land on one of the runways within its predetermined time windows such that separation criteria S_{ij}) between all pairs (i, j) of plane are satisfied. This type of problem is a large-scale optimization problem, which occurs at busy airports where making optimal use of the bottleneck resource (the runways) is crucial to keep the airport operating smoothly. Upon entering within the radar range (or horizon) of an air traffic control (ATC) at an airport, a plane requires ATC to assign a landing time and also a runway if more than one runways are in use. The landing time must lie within a predetermined time window, bounded by an earliest landing time and a latest landing time. The time windows are different for different planes. The earliest time represents the time required if a plane flies at its maximum airspeed. The latest time corresponds to the landing time of a plane flying at its most fuel efficient airspeed while holding (circling) for the maximum allowable time ([4]).

In the second section, we give the mathematic formulation of the Airline scheduling problem. In the third section, we present the three methods; Bionomic Algorithm (BA), Genetic Algorithm(GA), and Scatter Search (SS) used to obtain our algorithm in the fourth section. In the last section, we present the results obtained by our new algorithm applied on benchmarks of Beasly [3].

II. A MATHEMATICAL MODEL OF THE ALP

This section presents a mixed integer formulation of the static multiple runway aircraft landing problem based on the formulation presented in Beasley [2]. Given a set of planes I , each plane i has a predetermined landing time windows [Ei , Li], and also, a target time T_i ($E_i \leq T_i \leq L_i$) at which time the plane is landed with cost 0. S_{ij} is the required separation time between plane I and j (where i lands before j) for landing these on the same runway:

$$t_i + S_{ij} \leq t_j \qquad (1.1)$$

As customary in the multiple runway case, we assume that the separation time between two planes on different runways is 0. Av_i and Ap_i denote the unit costs for plane i landing earlier and later than the target time respectively. Furthermore, we use the decision variables: $x_{i:}$ the landing time for plane I (i ∈ I);

$$y_{ir} = \begin{cases} 1 & \text{if plane i lands on runway r}(j \in I;\ r \in J); \\ 0 & \text{otherwise.} \end{cases}$$

The objective function is:

$$\min F_X = \min \sum_{r\in J} \sum_{i\in I} y_{ir} (Av_i \times \max\{0, T_i - t_i\} + Ap_i \times \max\{0, t_i - T_i\}) \qquad (1.2)$$

III. POPULATION HEURISTIC METHODS

The Population heuristics are based on Darwin's theory of evolution, where selection and mutation are two key concepts. In such kinds of algorithms individuals that represent solutions to the problem are manipulated. Each individual is encoded using a set of chromosomes that define the parameters relevant in the considered problem. A better solution gives a high fitness value that makes it more likely to be passed onto later generations. The most widely known method is the Genetic Algorithm (GA).

3.1. The Bionomic Algorithm

The Bionomic Algorithm (BA) is less well-known than SS. It used a linear combination to build new individuals. Features specific for the BA are: a maturation step to improve individuals; structured construction of parent sets based on a graph which represents the population structure; parent selection based on fitness and distance between individuals; a generational approach to replace the population. The general framework for the BA is:

Generate an initial population
Improve each individual in the initial population
Repeat
    Build a graph that represents the population structure
    Compute parent sets from this graph
    Create new individuals as linear combinations for each parent set
    Improve each new individual
    Update the population with some of the best new individuals
Until termination, whereupon report the best solution encountered.

3.2. Genetic Algorithm

This method was developed by by Ciesielski and Scerri [1]. Applying Genetic algorithm to the aircraft landing scheduling problem gives good results. Each time an aircraft lands or enters the landing area a new calculation must be started. There are two different initialization procedures that have been used namely random initialization and an approach in where the initial population is taken to be the final population in the followed methods.

3.2.1. Encoding :

One solution (chromosome) is a table with three rows and P columns. Each gene consists of one column and three rows. The first row denotes the number of aircraft, the second a random float number yi and the last bit denotes the runway. This together with a table containing information about the optimal landing times for each runway and the size of the aircraft the encoding is complete.

9	3	1	4	6	7	2	5	8
y9	y3	y1	y4	y6	y7	y2	y5	y8
r9	r3	r1	r4	r6	r7	r2	r5	r8

Table 1: Chromosome encoding

The proportion of time is defined as $x_i = E_i + y_i \times (L_i - E_i)$ with $y_i \in [0, 1]$ (1.3)
Where x_i is the scheduled landing time, E_i is the earliest landing time and L_i is the latest landing time.

3.2.2. Evaluation:

The above described encoding scheme can result in invalid solutions and in solutions where aircrafts can be scheduled to land at the same time. This has to be dealt with using a fitness function that punishes bad solutions and rigorously punishes invalid ones. Researchers used a fitness function dealing with all constraints described in section 1 where a high fitness was given to solutions that were invalid, solutions with overlapping landing times, solutions with aircraft landing with a short time gap between them and solutions that proposed aircraft landing on crossing runways. There was also a small fitness given to aircrafts landing too early and too late. It is obvious that a low fitness is desirable with this fitness function. The elements in the fitness measure where punished with different weights in order to make, for example, invalid solutions less likely to be passed on to the next generation.

3.2.3. Genetic operators

Both standard operators and operators with incorporated problem specific knowledge were used for crossover and mutation. It is not desirable for an aircraft scheduled to land relatively soon to get a large change in its landing time. Therefore the modified mutation and crossover operator makes it less likely for aircrafts encoded early in the chromosome to be mutated since aircrafts about to land are placed in the beginning of the chromosome. Newly arrived aircrafts are placed in the end of the chromosome. The operators leave the individuals that have been in the system for some time while focusing on the newly arrived aircrafts.

3.3. Scatter Search

In contrast to a regular GA the Scatter Search (SS) is not driven by randomization. Instead SS uses a deterministic approach and it also has properties making it well suited for combinatorial optimization problems, such as the aircraft landing problem. Problem specific knowledge is implemented in the algorithm. The general framework for (SS) can be expressed as follows:

Generate the initial population called the reference set
Improve each individual in the reference set
Repeat
    Select a subset of the reference set
    Create a new individual as linear combination of the subset
    Improve the new individual
    Update the reference set
Until termination, whereupon report the best solution encountered

IV. IMPLEMENTATION

The population is composed of three subpopulations. On each one, we applied one of three methods. At the end of each iteration, we copied the best individual in the place of the weakest individuals on three subpopulations. The size of each population is 100 individuals. We used simultaneously three methods with each one is applied on 100 individuals. The program is written in C language, running on PC machine of CPU 1.99 Ghz using SUSE Linux operating system. The linear function uses a slightly different approach. It is formulated as a cost function where aircrafts arrived at scheduled time gets zero cost and aircrafts arriving before or after its preferred landing time gets a cost linear to the time deviation. The cost for landing before scheduled time is relatively smaller than the cost for landing after the scheduled time. Individuals are all evaluated under the constraints of earliest and latest landing times, landing separation times and multiple runways. The remaining times are randomly generated. Parent selection for SS relies on simple tournament selection.

V. RESULTS AND CONCLUSION

Pinol and Beasly [2] are notice that for large problems both SS and the BA perform relatively slow but generates high quality solutions. SS outperformed the BA for the linear objective. These results are valid for problems where only one runway is present. When the algorithm was presented with runway dependent optimal landing times as in the case with the GA the results however was good. Both algorithms performed better than those earlier presented in literature. The data used is publicly available test data. Ciesielski and Scerri [1] determine that the supposed performance-improving modified genetic operators didn't outperform the standard operators. The results differed very little and no conclusions can be drawn from them. It is crucial that the algorithm can produce a valid schedule at every given time and the analysis of the results were therefore focused on the number of valid solutions in the population as well as the fitness and the variation with different parameters in the algorithm. We find the following results while we applied our new method on Beasley benchmarks [3]:

Instances	P	Best Known solution	Our Method
Airland1	10	700	700
Airland2	15	1720	1700
Airland3	20	1040	1020
Airland4	20	4480	4480
Airland5	20	4800	4800
Airland6	30	7194	7194
Airland7	44	1550	1500
Airland8	50	3050	3050
Airland9	100	9520	9654
Airland10	150	24415	24415
Airland11	200	22011	21671
Airland12	250	28946	27983

Table 2: Numerical results on one ranway.

In this table, the values of best known solutions obtained for each instances are shown in the third column. The results show clearly that our procedure gives a much higher quality of the solutions in 11 instances from 12. In 5 instances out of 11, we obtain the best results than the best known results. This new combined method is very promising method.

REFERENCES

[1]. V. Ciesielski and P. Scerri. Real time genetic scheduling of aircraft landing times. New York, USA, 1998.

[2]. H. Pinol and J.E. Beasly. Scatter Search and Bionomic Algorithms for the Aircraft Landing Problem. Tanaka Business School,
Imperial College, London, SW7 2AZ, UK, 2004

[3]. http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/J.Abela