Real-time Construction of Aircraft Landing Schedules Using an Evolutionary Algorithm

Neil Urquhart
Center for Emergent Computing
Napier University
10 Colinton Rd
Edinburgh, Scotland
n.urquhart@napier.ac.uk

Track: Real-World Applications

ABSTRACT

This poster investigates the use of Evolutionary Algorithms (EAs) to optimise solutions to the Dynamic Aircraft Landing Problem (DALP). The approach adopted here uses an EA to evolve improvements to the solution so as to move towards an optimum solution. By using the EA to evolve gradual improvements to an existing feasible solution, a feasible solution is always maintained, an important consideration in a safety critical area such as air traffic control.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*scheduling*; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Plan execution, formation, and generation*

General Terms

Algorithms

1. INTRODUCTION

The Dynamic Aircraft Landing Problem (DALP) involves the optimisation of aircraft landing times, the aim being to build solutions that allow aircraft to land within cost and safety constraints. Within the DALP aircraft appear and reveal their requirements during the problem solving process. Decisions must made quickly in response to the arrival of aircraft. The benchmark DALP problems examined in this paper may be solved feasibly by landing the aircraft in the order that they are received (First Come First Served - FCFS) but such solutions will tend to be expensive. Thus the challenge is not in finding feasibility, but in optimising the FCFS solution in order to lessen the cost of that solution.

DALP instances are currently available from OR-Library[1], these have previously been used in [beasley04]. They are suitable for use as instances of the static or dynamic ALP.

[1]OR-library is currently located on-line at http://people.brunel.ac.uk/ mastjjb/jeb/info.html

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.
ACM 1-59593-186-4/06/0007.

The DALP may be solved for 1 or more runways, in multi-runway DALPs the aircraft may land on differing runways concurrently. Each of the datasets obtained contains details of a number of aircraft values are provided for earliest suitable landing time, the target landing time, and the latest possible landing time. In addition also specified are, penalty values for being before or after the target time, a freeze time and an aircraft separation matrix. The separation matrix specifies the minimum time space that must exist between any two aircraft when the land consecutively on the same runway. A full definition of the problem may be found int [beasley04].

A dynamic environment such as this has not been viewed traditionally as suitable for an Evolutionary Algorithm, the time taken to evolve a solution being in many cases greater then the decision time available. Traditional research into such problems has used overall CPU times for comparisons. This author argues that for dynamic problems it is not the overall time to solve the problem that is relevant, but the maximum time taken to produce an updated solution after a change in the dynamic environment.

2. BUILDING LANDING SCHEDULES USING AN EA

Within the problem instances contained within OR-LIB a feasible solution may be constructed by ordering the aircraft in FCFS order. This property ensures that it is always possible to maintain a feasible solution, we shall term the *master solution.* It is this solution that contains the decisions, which are communicated to, and actioned upon, by aircraft. When placing a new flight within the solution a greedy placement heuristic is utilised which initially attempts to place at the target time, if the flight can't be placed each time slot until the end of the window is tried. If the flight is still not placed, slots from the target time back to the start time are tried.

The approach utilised here, known as Land-EA, seeks to take a feasible master solution, and evolve some minor changes to it in an attempt to discover another feasible but less costly solution. This approach uses the EA to evolve strings of commands, which are interpreted by a schedule builder and applied to a copy of the current solution to assess their impact . Ultimately if Land-EA fails to find an improved solution the origional master solution is retained. The role of the EA is not to modify the master solution directly, but to discover changes that allow further optimisation. The commands that may be used to modify the solution are:

Change runway <flight><runway> Remove <flight> from it's current runway and find it a slot on <runway>

Reset <flight> Use the greedy heuristic to place the current flight on its runway

Up <flight> Swap order with the flight scheduled to land directly before

Down <flight> Swap order with the flight scheduled to land directly behind

Move <flight><pos> Allocate a specific landing time to <flight> based on <pos>, pos being a value 0..1 representing the time window

Aircraft are added to the solution at their appearance time, they are initially added to the first available runway in default FCFS order. After an aircraft is added the EA is executed for a short period to produce an updated solution. Any planes within the updated solution whose landing time is now equal to current time + the frozen period are determined to be frozen and are removed from the solution.

As each chromosome is therefore a string of commands; to evaluate the chromosome the schedule builder applies the commands to a copy of the current solution and measures the difference in solution cost after applying each command. The cost of the modified solution is noted after each individual command is applied, where the command results in a feasible solution. After a number of generations if the best individual represents modification which results in a reduction in solution cost then it is permanently applied to the solution. The EA may then either be executed again, or new aircraft added/deleted from the solution. Thus each time the EA is invoked it makes a number of runs, each run consisting of a number of generations. Because this is a real-time problem, the EA is only allowed to execute for a maximum of 120 seconds. Elapsed time is monitored at the end of each generation, should the allowed time be exceeded the EA halts and returns the best solution it has found.

3. EXPERIMENTAL SETUP AND RESULTS

Land-EA was tested over the problem instances obtained from OR-LIB for 1,2 and 3 runways. Each result is based the average of 20 runs, values are presented for the highest, lowest and average results for each dataset. Results for the smaller datasets are presented in table 1. Within each run upon the presentation of an new aircraft a maximum of 11 cycles of 7 generations is undertaken. The maximum chromosome length being set to 7 instructions. Land-EA was tested with a recombination rate of 0.5 and of 0.

For comparison results from [beasley04] are shown in the column labelled DALP-OPT. Further research is required to investigate and clarify why Land-EA appears to outperform DAL-OPT on these instances. The discrepancies may be due to differing interpretations of the model or due to DAL-OPT not being allowed to run to completion. Land-EA appears to perform best on the multiple-runway problems in these cases it can equal the results obtained using DALP-OPT.

Overall the best results are achieved using smaller datasets, with performance degrading as the datasets become bigger.

Table 1: Results for Land-EA.

Problem	Avg	max	min	DALP-OPT
airland1-1	956	1150	810	740
airland1-2	109.5	120	90	90
airland1-3	0	0	0	0
airland2-1	1706.5	1720	1630	1730
airland2-2	210	210	210	210
airland2-3	0	0	0	0
airland3-1	880	880	880	940
airland3-2	60	60	60	60
airland3-3	0	0	0	0
airland4-1	3630.5	3730	2880	2700
airland4-2	686	740	680	680
airland4-3	132.5	160	130	130
airland5-1	4193	4430	3810	3810
airland5-2	714.5	760	680	680
airland5-3	237	280	190	240
airland6-1	24442	24442	24442	24442
airland6-2	1394.85	1557	1283	809
airland6-3	1083.1	1183	1015	0
airland7-1	3974	3974	3974	3974
airland7-2	0	0	0	0
airland7-3	0	0	0	n/a
airland8-1	2235.5	2510	2085	2000
airland8-2	160	160	160	135
airland8-3	0	0	0	0

This may be due to the time limitation placed on the EA of 120 seconds not being enough for larger problem instances or due to parameters requiring adaptation for differing types of problem. This second theory would be supported by the experiences of adjusting the recombination rate discussed above.

4. CONCLUSIONS AND FUTURE WORK

The approach taken here seeks to use an EA to make incremental changes to the overall problem solution. As there exists a default feasible solution (the FCFS solution) this may always be used in the first instance. The EA is then used to optimise this solution, thus if the EA is unable to evolve a beneficial change to the current solution, then the default solution may be used to ensure feasibility is maintained. Best results are obtained when applied to the smaller problems in that it either equals or betters existing solutions.

5. REFERENCES

Beasley JE, Sharaiha YM and Abranson, D. (2004) "Displacement problem and dynamically scheduling aircraft landings," Journal of the Operational Research Society, Vol 55 (pp54-64).

Atkin J, Burke E, Greenwood J, Reeson D. (2005) "A Meta-Heuristic Approach to Aircraft Departure Scheduling at London Heathrow Airport," proceedings of 9th International Conference on Computer-Aided Scheduling of Public Transport (CASPT). Springer Lecture Notes in Economics and Mathematical Systems .