Journal of the Operational Research Society (2004) 55, 54–64

www.palgrave-journals.com/jors

Displacement problem and dynamically scheduling aircraft landings

JE Beasley[1]*, M Krishnamoorthy[2], YM Sharaiha[1] and D Abramson[3]

[1]*Imperial College, London, UK;* [2]*CSIRO Mathematical and Information Sciences, Clayton South MDC, Victoria, Australia; and* [3]*Monash University, Clayton, Victoria, Australia*

In this paper we define a generic decision problem — the displacement problem. The displacement problem arises when we have to make a sequence of decisions and each new decision that must be made has an explicit link back to the previous decision that was made. This link is quantified by means of the displacement function. One situation where the displacement problem arises is that of dynamically scheduling aircraft landings at an airport. Here decisions about the landing times for aircraft (and the runways they land on) must be taken in a dynamic fashion as time passes and the operational environment changes. We illustrate the application of the displacement problem to the dynamic aircraft landing problem. Computational results are presented for a number of publicly available test problems involving up to 500 aircraft and five runways.

Journal of the Operational Research Society (2004) **55,** 54–64. doi:10.1057/palgrave.jors.2601650

Keywords: displacement problem; air traffic control; runway operations; scheduling

Displacement problem

In many decision problems we assume a static operational environment. However, often in applications the operational environment changes, typically as *time passes* new information makes it necessary to revise the previous decisions that have been made. Such situations are *dynamic* in the sense that planned decisions are continually having to be revisited as the operational environment changes.

However it is clear that, from an operational viewpoint, it may often be undesirable to perturb (change, displace) the previous decision 'too much' in making a new decision. Typically therefore in making a new decision we have to consider *explicitly* the previous decision that was made as we have some degree of commitment to it and are reluctant to change it too much. Hence we have a generic decision problem, the *displacement problem*, which arises when we have to make a sequence of decisions and each successive decision that is made involves an explicit link back to the previous decision.

Let the original (static) problem (SP) consist of N decision variables x_i, $i=1,\ldots,N$ (where we use $\underline{x}$ to denote the vector of decision variables). Then a general mathematical description of SP is given by *minimise* $Z(\underline{x})$ *subject to* $C(\underline{x})$, where $Z(\underline{x})$ is the objective function and $C(\underline{x})$ represents the constraints of the problem. Z and C can be linear or nonlinear functions, and $\underline{x}$ can be a mix of both integer and continuous variables.

*Correspondence: JE Beasley, The Management School, Imperial College, London SW7 2AZ, UK.
E-mail: j.beasley@imperial.ac.uk

Suppose that we solve SP and let X_i, $i=1,\ldots,N$ be a feasible solution (as found by some algorithm) to SP. We use $\underline{X}$ to denote this vector. Note here that $\underline{X}$ need not be the optimal solution to SP, for example, $\underline{X}$ may be obtained via a heuristic solution algorithm. Consider now that *something* in the original problem changes, for example: a piece of data (a number, involved in Z or C) changes; a new variable (decision) appears; a new constraint is added. Such changes will occur because when we action the decisions $\underline{X}$ derived from solving SP we will find that something in the operational environment is different from what we had previously assumed in arriving at those decisions.

As a result of this changing operational environment, we need to *resolve* the original problem, incorporating any changes, but with some *additional restriction* (*link*) back to the original (previous) solution $\underline{X}$. In our approach, we define a *displacement function* $D(\underline{X}, \underline{x})$ that quantifies the effect of displacing each decision variable i from its previous (known) solution value X_i to its new (currently unknown) value x_i.

The exact mathematical form of the displacement function $D(\underline{X}, \underline{x})$ is our choice, and it can be a linear or nonlinear function. It is convenient, however, both for conceptual and for modelling reasons, to impose three restrictions upon the displacement function $D(\underline{X}, \underline{x})$:

(1) $D(\underline{X}, \underline{x}) \geqslant 0 \; \forall \underline{X}, \underline{x}$, so that displacement is non-negative.
(2) $D(\underline{X}, \underline{x}) = 0$ if $x_i = X_i$, $i=1,\ldots,N$, so that displacement takes the value zero if the new decisions $[x_i]$ are the same as the old decisions $[X_i]$.

(3) $D(\underline{X}, \underline{x})$ is a separable function into a summation of terms involving only X_i and x_i, so that the contribution to total displacement of a change in decision variable i, from X_i to x_i, can be identified separately from the effect of all other changes in decision variables.

Let:

$D_i(\underline{X}, \underline{x})$ represent the contribution to total displacement of a change in decision variable i from X_i to x_i (eg if $D(\underline{X}, \underline{x}) = \sum_{i=1}^{N} (X_i - x_i)^2$ then $D_i(\underline{X}, \underline{x}) = (X_i - x_i)^2$)

p_i be the objective function weighting ($\geqslant 0$) per unit of displacement in variable i ($i = 1, \ldots, N$)

Δ_i be the maximum displacement ($\geqslant 0$) that we are prepared to accept in variable i ($i = 1, \ldots, N$)

Then the general form of the *displacement problem* is:

minimise

$$\lambda_{\text{cost}} Z^*(\underline{x}) + \lambda_{\text{disp}} \sum_{i=1}^{N} p_i D_i(\underline{X}, \underline{x}) + \lambda_{\max} D_{\max} \quad (1)$$

$$\text{subject to} \quad C^*(\underline{x}) \quad (2)$$

$$D_i(\underline{X}, \underline{x}) \leqslant \Delta_i, \quad i = 1, \ldots, N \quad (3)$$

$$D_{\max} = \max[D_i(\underline{X}, \underline{x}) | i = 1, \ldots, N] \quad (4)$$

Here, Z^* and C^* represent the original objective and constraints (Z and C) but amended to reflect any changes that have occurred (eg the addition of new decision variables). In the objective function, Equation (1), λ_{cost}, λ_{disp} and $\lambda_{\max}$ ($\geqslant 0$) are the weights (respectively) attached to: total cost of solution $Z^*(\underline{x})$; total cost of displacement $\sum_{i=1}^{N} p_i D_i(\underline{X}, \underline{x})$; and maximum displacement $D_{\max}$. Equation (3) limits the displacement for any variable, while Equation (4) defines the maximum displacement $D_{\max}$.

Aircraft landing

In this section, we first briefly review the aircraft landing problem (henceforth ALP) and then go on to define an example displacement function for the dynamic ALP.

Aircraft landing problem

The ALP is the problem of deciding a landing time on an appropriate runway for each aircraft in a given set of aircraft such that each aircraft lands within a predetermined time window; and separation criteria between the landing of an aircraft, and the landing of all successive aircraft, are respected. In order to formulate the problem define:

P the number of aircraft

E_i the earliest landing time for aircraft i

L_i the latest landing time for aircraft i

T_i the target (preferred) landing time for aircraft i

g_i the penalty cost ($\geqslant 0$) per unit of time for landing before target T_i for aircraft i

h_i the penalty cost ($\geqslant 0$) per unit of time for landing after target T_i for aircraft i

S_{ij} the required separation time ($\geqslant 0$) between aircraft i landing and aircraft j landing (where aircraft i lands before aircraft j)

x_i the landing time ($\geqslant 0$) for aircraft i, a decision variable

δ_{ij} $= 1$ if aircraft i lands before aircraft j
$= 0$ otherwise

then the (single runway) static ALP has

$$Z(\underline{x}): \quad \sum_{i=1}^{P} (g_i \max[0, T_i - x_i] + h_i \max[0, x_i - T_i]) \quad (5)$$

$$C(\underline{x}): \quad \delta_{ij} + \delta_{ji} = 1, \quad i = 1, \ldots, P; \quad j = 1, \ldots, P; \; j > i \quad (6)$$

$$x_j \geqslant x_i + S_{ij}\delta_{ij} - (L_i - E_j)\delta_{ji}, \quad i = 1, \ldots, P; \; j = 1, \ldots, P; \; i \neq j \quad (7)$$

$$E_i \leqslant x_i \leqslant L_i, \quad i = 1, \ldots, P \quad (8)$$

Equation (6) ensures that for each pair of aircraft one lands before the other, Equation (7) enforces separation between aircraft and Equation (8) ensures that each aircraft lands within its time window. The first/second maximisation terms in Equation (5) account for aircraft that land before/after target. This cost function is illustrated diagrammatically in Figure 1. Colloquially g_i and h_i are the slope of the cost function before and after the target time T_i respectively. Extending the above single runway formulation to the multiple runway case is easily done and, for reasons of space, will not be presented here. Note here that given a fixed landing sequence (equivalently values for the zero–one variables δ_{ij} above), an effective approach to deciding optimal landing times that minimise cost with respect to that given sequence is to solve the LP that results from Equations (5)–(8) when the zero–one variables are eliminated. For convenience, we refer to this LP as ALP$^{\text{F}}$ indicating that it is the ALP with a fixed landing sequence.

Beasley *et al*[1] first formulated the static ALP as a mixed-integer zero–one linear program and solved it numerically for a number of test problems involving up to 50 aircraft and four runways. Beasley *et al*[2] studied the single runway static

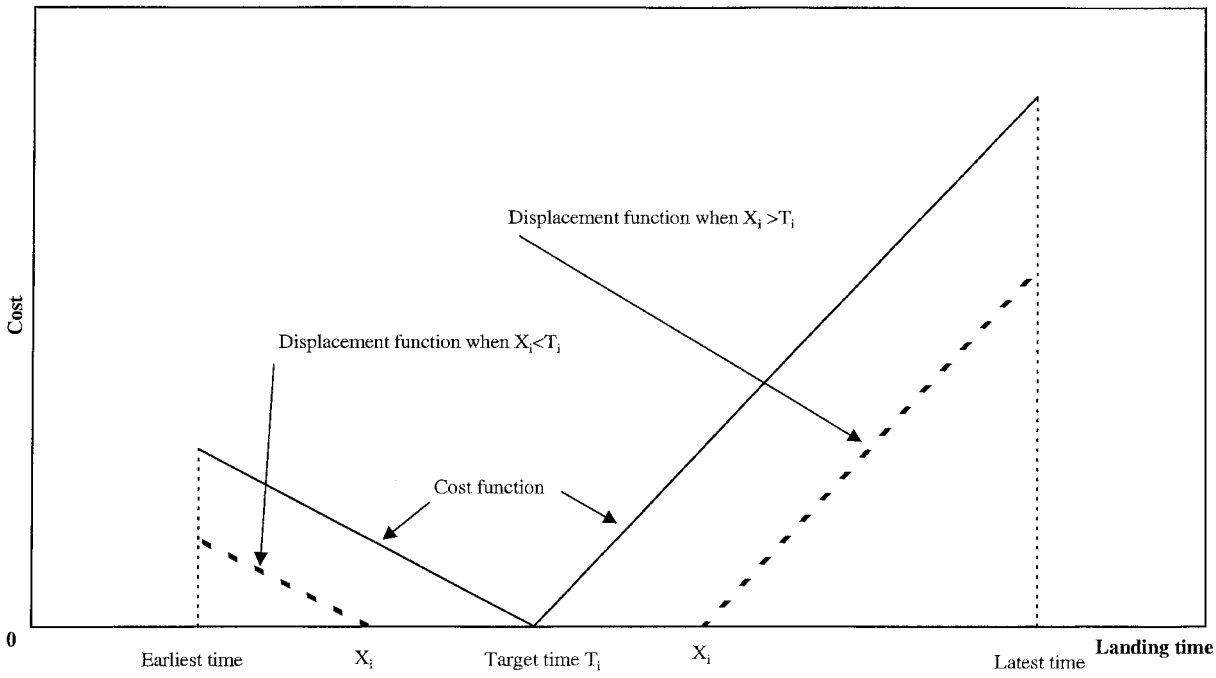

Figure 1 Cost function and displacement function.

ALP and presented a population heuristic (genetic algorithm) for the problem. Other work relating to the static ALP is described in Beasley *et al*[1] and, for reasons of space, that description will not be repeated here.

In this paper, we deal with the dynamic, or on-line, ALP, where decisions about the landing times (and runways) for aircraft must be made as time passes and as the operational environment changes (aircraft land, new aircraft appear, etc). Current air traffic control practice[3–5] for dealing with this problem is to schedule aircraft to land in a first-come, first-served (FCFS) manner.

Previous work

There appears to have been relatively little work published with regard to the dynamic ALP. Andreussi *et al*[6] referred to the problem as the *aircraft sequencing* problem and presented a paper concerned with developing a discrete-event simulation model to evaluate different sequencing strategies. Computational results were presented for a number of simulated scenarios involving three runways. Dear and Sherif[7,8] discussed both the static and dynamic ALP and presented a heuristic algorithm for the single runway dynamic ALP based upon *constrained position shifting*. This involves finding, for a small set of aircraft, the best possible positions for them in the landing queue subject to the constraint that no aircraft can be moved more than a pre-specified number of positions away from the position it had in the landing queue based on FCFS, see also Dear.[9] Computational results were presented for three simulated scenarios involving 500 aircraft and one runway.

Brinton[10] presented a tree search approach for the ALP. In his approach, the tree represents the sequence in which aircraft should be landed. The dynamic ALP is dealt with via freezing the position of an aircraft in the landing sequence and via costs associated with changes in the scheduled landing time. No detailed computational results were presented however. Venkatakrishnan *et al*[11] observed separation times adopted on landing at Logan Airport Boston. Using these observed separation times they applied the work of Psaraftis,[12,13] which they modified in a heuristic manner to take account of aircraft time windows, to see the improvement that could result from better sequencing. They presented two approaches to the dynamic ALP. In both approaches aircraft are frozen in the landing sequence once they are near to landing. The difference between their approaches is that one leaves the aircraft time window unchanged, while the other gradually reduces the size of the time window as an aircraft approaches landing. Computational results were presented for six data sets involving up to 92 aircraft and two runways.

Ciesielski and Scerri[14,15] presented a genetic algorithm for the problem. In their approach, landing times are allocated by specifying a 30-s time slot. Their approach consists of finding the best solution they can within 3 min of elapsed time, updating the situation with respect to aircraft that have landed/appeared and resolving a new problem. Unlike the work presented in this paper, they include no link between the previous set of landing time decisions and the new set.

Computational results were presented for two data sets involving 28 and 29 aircraft and two runways. Milan[5] considered the problem of assigning priorities for landing to aircraft in arrival batches (a batch comprising aircraft due to arrive at approximately the same time). Priorities were based upon factors such as number of passengers, cost of passenger delays and proportion of transfer passengers. Once priorities had been assigned the aircraft in a batch were landed in priority order. Computational results were presented for one example with 30 aircraft and one runway.

Carr *et al*[16–18] presented papers concerned with modifying the standard FCFS approach to allow individual airlines to express priorities with regard to the landing of their aircraft. Computational results were presented for a number of simulated scenarios involving three runways. Bolender and Slater[19] approached the dynamic ALP via queueing theory and discrete-event simulation. They assumed that aircraft appear according to a Poisson process so that aircraft interarrival times follow a negative exponential distribution. Their analysis focuses on differing ways of allocating a newly appeared aircraft to a runway for landing (when multiple runways are present). Once allocated to a runway aircraft land in FCFS order. Computational results were presented for a number of problems involving up to three runways. Wong[20] discusses the algorithms underlying the CTAS (Center TRACON (terminal radar approach control) Automation System) system developed at the NASA Ames Research Center. For the dynamic ALP, an FCFS approach is used with aircraft being frozen once they are close to landing.

Displacement function

In order to deal with the dynamic ALP as a displacement problem, we need to define an appropriate displacement function. Considering Figure 1 suppose that aircraft i has been assigned a landing time X_i from the solution to the original static ALP and further suppose that this time is later than its desired target time (ie $X_i > T_i$). When we come to solve the displacement problem the effect of the cost component ($g_i \max[0,\ T_i - x_i] + h_i \max[0,\ x_i - T_i]$) associated with aircraft i in the displacement problem objective function will be to try and move the new landing time for aircraft i (x_i) closer to the desired target time T_i (ie to have $x_i < X_i$). This will be a desirable displacement from the current landing time X_i.

It may be however that other factors (eg newly appeared aircraft that must be scheduled for landing) will mean that aircraft i has its landing time further increased (ie $x_i > X_i$). This will not be a desirable displacement and so should incur an *extra cost*, such as shown by the dotted displacement function line in Figure 1. For the sake of illustration we shall assume, as Figure 1, that this extra cost is also h_i for each unit of time the aircraft is displaced later than X_i, that is, that the displacement function is $h_i \max[0,\ x_i - X_i]$ if $X_i > T_i$.

Similarly, if the assigned landing time X_i is less than the target time T_i, then when the displacement problem is solved, aircraft i will not wish to move further away from its desired target time T_i and any such movement should be penalised by an additional cost. Again for the sake of illustration we shall assume, as in Figure 1, that this extra cost is g_i for each unit of time the aircraft is displaced earlier than X_i, that is, that the displacement function is $g_i \max[0,\ X_i - x_i]$ if $X_i < T_i$. Hence our displacement function is:

$$D_i(\underline{X}, \underline{x}) = \begin{cases} g_i \max[0, X_i - x_i] & \text{if } X_i < T_i \\ h_i \max[0, x_i - X_i] & \text{if } X_i > T_i \\ g_i \max[0, X_i - x_i] + h_i \max[0, x_i - X_i] & \text{if } X_i = T_i \end{cases} \tag{9}$$

where for the case $X_i = T_i$ we ensure that deviations from target (in either direction) are further penalised. This displacement function satisfies the restrictions ($D(\underline{X}, \underline{x})$ non-negative, zero if $x_i = X_i$ and a separable function) mentioned previously.

Solving the displacement problem

In order to solve the displacement problem we adapted three solution approaches, one optimal[1] and two heuristic,[1,2] given previously in the literature for the static ALP. We believe it to be a natural state of affairs that solution approaches (both heuristic and optimal) developed for the static ALP can with suitable amendment be applied directly to the dynamic ALP. One should reasonably expect that solution approaches developed for the original static decision problem have an important role to play in solving the displacement problem defined from the original static decision problem. The adaptations we made were as briefly outlined below.

Beasley *et al*[1] presented an optimal solution algorithm based upon linear programming (LP)-based tree search for the static ALP. Although the displacement function (Equation (9)) defined above for the dynamic ALP is nonlinear, by utilising the definition of the displacement problem (Equations (1)–(4)) together with the definition of the static ALP (Equations (5)–(8)), it is possible to show that the dynamic ALP as formulated above can be transformed into a mixed-integer zero–one linear program. Hence, LP-based tree search can be directly applied in order to find the optimal solution to each successive displacement problem. We refer to this algorithm as DALP-OPT.

The heuristic solution algorithm presented in Beasley *et al*[1] for the static ALP used two steps: firstly a simple constructive step to decide the sequence in which aircraft are to land (and on which runway), based upon sorting aircraft into target time order; and secondly solving ALP^{F} to decide the landing times for each aircraft. With respect to adapting this heuristic for the solution of the displacement

problem we found that, computationally, directly applying this heuristic to the displacement problem could lead to infeasibility, that is, we could end up with a displacement problem for which the heuristic could not find a feasible solution. We found that, for the particular test problems we considered, this issue of infeasibility did not arise if we also generated a landing sequence based upon first sorting aircraft using their landing time as decided at the solution to the previous displacement problem and then including any newly appeared aircraft in target time order. Again given a landing sequence we solved ALP^F to decide landing times. In the computational results presented below we applied both of these heuristics and took the best solution found. We refer to this algorithm as DALP-H1.

In order to adapt the population heuristic (genetic algorithm) presented by Beasley *et al*[2] for the single runway static ALP to the multiple runway dynamic ALP, we:

- extended the representation, together with the crossover and mutation operators, to include runway choice;
- incorporated maximum displacement by time window modification;
- seeded the initial population with suitable individuals based upon the landing sequences given by sorting aircraft into earliest/target/latest time order;
- applied the population heuristic after first using DALP-H1 and seeding the initial population with the best solution found by DALP-H1
- took the best solution as found by the population heuristic and solved ALP^F to decide landing times.

We refer to this algorithm as DALP-H2.

Computational results

The algorithms DALP-H1, DALP-H2 and DALP-OPT for the dynamic ALP outlined in this paper were programmed in FORTRAN and run on a Silicon Graphics Indigo workstation (R4000, 100 MHz, 64 MB main memory) for a number of test problems involving up to 500 aircraft and five runways. This machine is approximately 25 times slower than a 2.5 GHz Pentium pc. In order to solve the mixed-integer zero–one formulation of the displacement problem to optimality using LP-based tree search, and also to solve ALP^F, we used the CPLEX (version 6.5) software package.[21]

Methodology

The methodology we adopted was as follows:

(1) Each aircraft i ($i=1,\ldots,P$) had an appearance time A_i, the time at which it was first available to be assigned a landing time. We also defined a freeze time t^* such that any aircraft assigned a landing time within t^* of the current time had its landing time (and runway) frozen. Set the landing times $X_i=\infty$, $i=1,\ldots,P$. Let:

$F_0(t)$ represent the set of aircraft that have not yet appeared by time t, that is, $F_0(t)=[i|A_i>t, i=1,\ldots,P]$

$F_1(t)$ represent the set of aircraft that have appeared by time t, but have not yet landed (or had their landing times frozen), that is, $F_1(t)=[i|A_i\leqslant t$ and $X_i>t+t^*, i=1,\ldots,P]$

$F_2(t)$ represent the set of aircraft that have appeared by time t and have landed (or have had their landing times frozen), that is, $F_2(t)=[i|A_i\leqslant t$ and $X_i\leqslant t+t^*, i=1,\ldots,P]$

γ be the iteration counter

Z_{disp} be the accumulated displacement cost

(2) Set $\gamma=0$ and $Z_{\text{disp}}=0$. Set the current time $t_0=\min[A_i|i=1,\ldots,P]$ and solve the original static ALP involving just those aircraft in $F_1(t_0)$. The solution to this static ALP gives the initial landing times X_i $\forall i\in F_1(t_0)$.

(3) If aircraft are still to appear ($|F_0(t_\gamma)|\geqslant 1$) then go to step (4), otherwise ($|F_0(t_\gamma)|=0$) all aircraft have appeared in which case go to step (5).

(4) Set $\gamma=\gamma+1$. Advance the time to $t_\gamma=\min[A_i|i\in F_0(t_{\gamma-1})]$ and solve the displacement problem involving just those aircraft in $F_1(t_\gamma)\cup F_2(t_\gamma)$, where the aircraft in $F_2(t_\gamma)$ are constrained to land at the landing times X_i ($\forall i\in F_2(t_\gamma)$), and on the appropriate runways, as were obtained from the previous displacement solution. Add the displacement cost component $(\lambda_{\text{disp}}\sum_{i\in F_1(t_\gamma)\cap F_1(t_{\gamma-1})} p_i D_i(\underline{X},\underline{x}) + \lambda_{\max} D_{\max})$ of this solution to Z_{disp} and go to step (3).

(5) All aircraft in $F_1(t_\gamma)$ are now deemed to land at the landing times (and on the appropriate runways) as were obtained from the last displacement solution. Compute $Z_{\text{sol}}=\sum_{i=1}^{P}(g_i\max[0, T_i-X_i]+h_i\max[0, X_i-T_i])$ which is the cost of the final solution in terms of the cost function associated with the original (static) ALP.

Results

We considered two sets of test problems. The first set, involving up to 50 aircraft, were those previously considered in Beasley *et al*[1] for which optimal static solutions are known. The second set we considered were larger, involving between 100 and 500 aircraft, and were generated in the following manner:

- aircraft appeared according to a negative exponential distribution with a mean interarrival distance of 6.5 nautical miles, converted into an earliest time by assuming a speed of 210 knots (nautical miles per hour)

- the appearance time for each aircraft was 10 min before its earliest time; the target time was randomly generated between 1 and 10 min after the earliest time; the latest time was 30 min after the earliest time
- costs for appearing before/after target were real numbers randomly generated from the interval [1,2] and a freeze time of 12 min was adopted
- aircraft were classified by four types: heavy, upper-medium, lower-medium and small, with the type being randomly generated with probabilities of 0.4, 0.3, 0.2 and 0.1 respectively. Separation distances/times on landing were calculated as in[2].

For the largest problem solved this corresponds to scheduling the landing of 500 aircraft over a 15-h time period. All of the test problems solved in this paper are publicly available from OR-Library,[22,23] see http://mscmga.ms.ic.ac.uk/jeb/orlib/airlandinfo.html.

Table 1 gives the results for the smaller problems with $\lambda_{cost}=\lambda_{disp}=1$, $\lambda_{max}=0$ and $p_i=1$, $\Delta_i=\infty$ $\forall i$. In that table we, in order to provide some insight into the quality of our results, have given the solution value associated with the optimal static approach (from Beasley *et al*[1]). This value provides a benchmark since (in terms of the original cost, Equation (5)) the best possible sequence of decisions (landing times and runways) we can make, over a succession of displacement problem solutions, correspond to the decisions (landing times and runways) arrived at by solving, just once, the static ALP involving all aircraft. We also show in Table 1 the solution value found by taking the best solution from two FCFS approaches: schedule each aircraft as early as possible; and schedule each aircraft at its target time (if that is feasible) but as early as possible if the target time is not feasible. Summarising then we have that Table 1 shows, for each problem: the number of aircraft; the number of runways; the optimal static solution value (Z_{static}); the FCFS solution value (Z_{FCFS}); the solution values (Z_{sol}, Z_{disp}) and the total time taken when each successive displacement problem is solved heuristically/optimally.

Table 2 shows similar information as Table 1 for the larger problems. Table 3 also shows results for these problems but where, compared to Table 2, we have increased λ_{disp} to 2, λ_{max} to 5, and have set $\Delta_i=10$ $\forall i$. This corresponds to a scenario in which we are trying to further discourage displacement and also explicitly limit displacement. We only show in Table 3 those test problems that exhibited some displacement in Table 2. Examining Tables 1–3 we would make the following observations:

- Excluding from Table 1 those problems for which $Z_{static}=0$, the average percentage cost increase over and above the static solution is 591.1% for FCFS ($100(Z_{FCFS}-Z_{static})/Z_{static}$) 55.1% for DALP-H1 ($100(Z_{sol}+Z_{disp}-Z_{static})/Z_{static}$), 50.9% for DALP-H2, but only 36.4% for DALP-OPT.
- Of the 39 larger problems shown in Tables 2 and 3, we have that the best solution value ($Z_{sol}+Z_{disp}$) is given by DALP-OPT 37 times. This compares with 15 times for DALP-H2 and eight times for DALP-H1 and indicates the value of resolving each successive displacement problem optimally rather than heuristically. Note here that normally one expects an optimal algorithm (by its very nature) to always produce a solution superior (or equal) to that produced by a heuristic algorithm. Here however for two of these 39 problems (problem 12 with three runways in Tables 2 and 3), our heuristic DALP-H2 produces a better solution than DALP-OPT. The reason for this is a generic one in that we are solving a succession of displacement problems as time passes (aircraft land, new aircraft appear) and the values shown in Tables 2 and 3 are summary statistics of the overall effect of these solutions in cost terms. It can happen, as here, that solving a displacement problem heuristically leads to decisions that are better (in terms of aircraft yet to appear — which are unknown) than the decisions made by solving the same displacement problem optimally. As such, the overall solution produced by successive applications of a heuristic algorithm can be superior to the overall solution produced by successive applications of an optimal algorithm.
- It is clear that the benefit gained by DALP-H2, compared to DALP-H1, is more marked for larger problems. DALP-H2 improves upon the DALP-H1 solution ($Z_{sol}+Z_{disp}$) for only three of the 25 problems in Table 1, but for 31 of the 39 problems in Tables 2 and 3.

In order to investigate problems where DALP-OPT becomes computationally ineffective we show in Table 4 the results for the same problems as in Table 2 but with the freeze time reduced to zero. Reducing the freeze time increases the number of aircraft that are available to have their landing times rescheduled and hence increases the size of the displacement problem that has to be solved optimally by DALP-OPT. It is clear from Table 4 that while for some problems DALP-OPT is still computationally effective, there are a number of problems (in particular those that terminated without a solution due to time limit considerations) for which DALP-OPT is computationally ineffective. Note here, however, that both DALP-H1 and DALP-H2 are computationally effective for all of the problems shown in Table 4.

Table 1 Computational results: small problems

Problem number	*Number of aircraft*	*Number of runways*	*Optimal static solution* Z_{static}	*FCFS solution* Z_{FCFS}	*Heuristic solution to each successive displacement problem*						*Optimal solution to each successive displacement problem DALP-OPT*		
					DALP-H1			*DALP-H2*					
					Z_{sol}	Z_{disp}	*Total time (min)*	Z_{sol}	Z_{disp}	*Total time (min)*	Z_{sol}	Z_{disp}	*Total time (min)*
1	10	1	700	1790	740	260	0.1	Same		0.1	740	260	0.1
		2	90	120	120	0	0.1	Same		0.1	90	30	0.1
		3	0	0	0	0	0.1	Same		0.1	0	0	0.1
2	15	1	1480	2610	1870	110	0.1	Same		0.2	1730	250	0.5
		2	210	210	210	0	0.2	Same		0.2	210	0	0.3
		3	0	0	0	0	0.1	Same		0.1	0	0	0.1
3	20	1	820	2930	1440	290	0.2	Same		0.4	940	230	0.5
		2	60	60	60	0	0.2	Same		0.3	60	0	0.3
		3	0	0	0	0	0.1	Same		0.1	0	0	0.1
4	20	1	2520	6290	2670	960	0.2	Same		0.3	2700	420	0.9
		2	640	1560	680	170	0.2	Same		0.3	680	80	2.0
		3	130	330	130	10	0.2	Same		0.3	130	10	0.6
		4	0	60	0	0	0.1	Same		0.1	0	0	0.1
5	20	1	3100	8370	6130	490	0.3	6130	250	0.4	3810	630	1.3
		2	650	1440	1070	60	0.2	1050	120	0.3	680	170	4.0
		3	170	240	240	0	0.2	Same		0.3	240	0	0.9
		4	0	0	0	0	0.1	Same		0.1	0	0	0.1
6	30	1	24 442	24 442	24 442	0	0.4	Same		0.6	24 442	0	0.6
		2	554	882	882	0	0.4	Same		0.4	809	98	0.5
		3	0	0	0	0	0.2	Same		0.2	0	0	0.2
7	44	1	1550	1550	3974	0	1.0	Same		1.4	3974	0	1.5
		2	0	0	0	0	0.6	Same		0.6	0	0	0.6
8	50	1	1950	26835	2915	735	1.3	2710	430	2.0	2000	455	3.1
		2	135	10140	255	15	1.1	135	75	1.2	135	75	1.2
		3	0	4825	0	0	0.7	Same		0.7	0	0	0.7

Note: 'Same' means that the solution values for both Z_{sol} and Z_{disp} for DALP-H2 were identical to those for DALP-H1.

Table 2 Computational results: large problems

Problem number	*Number of aircraft*	*Number of runways*	*FCFS solution* Z_{FCFS}	*Heuristic solution to each successive displacement problem*						*Optimal solution to each successive displacement problem DALP-OPT*		
				DALP-H1			*DALP-H2*					
				Z_{sol}	Z_{disp}	*Total time (min)*	Z_{sol}	Z_{disp}	*Total time (min)*	Z_{sol}	Z_{disp}	*Total time (min)*
9	100	1	36 839.36	13 554.77	1213.12	3.7	12 553.87	1041.40	4.7	7848.42	1171.12	5.7
		2	10 661.92	578.71	57.94	3.0	577.57	33.12	2.8	573.25	33.12	3.4
		3	4142.07	88.72	33.28	2.6	Same		2.6	88.72	33.28	2.6
		4	1518.04	0	0	2.3	Same		2.3	0	0	2.3
10	150	1	54 113.54	31 945.03	1602.79	7.6	31 034.04	1220.52	9.7	17 726.06	2164.97	13.5
		2	13 318.09	1903.00	428.36	6.9	1417.82	307.82	6.8	1372.21	53.70	8.7
		3	5290.35	219.44	34.68	5.7	216.25	34.68	5.4	246.15	1.14	6.0
		4	3203.35	47.20	0	5.6	34.22	0	5.1	34.22	0	5.1
		5	770.90	0	0	4.9	Same		4.9	0	0	4.9
11	200	1	66 427.28	27 417.23	2714.15	16.6	23 963.48	2150.87	18.5	19 327.45	1954.41	22.6
		2	17 381.29	1671.13	455.31	12.3	1692.97	245.84	12.1	1683.75	91.32	15.9
		3	5901.27	347.37	0	10.2	333.53	0	9.4	333.53	0	13.2
		4	2040.41	69.66	0	8.9	Same		8.9	69.66	0	9.3
		5	516.84	0	0	8.6	Same		8.6	0	0	9.1
12	250	1	81 916.40	34 246.39	2599.64	22.3	31 439.77	1792.73	28.2	25 049.24	1941.88	43.2
		2	22 790.99	2848.87	718.05	18.6	2497.06	545.68	19.3	2204.96	155.31	43.2
		3	8883.11	265.24	44.10	15.8	252.52	44.10	15.7	430.50	37.74	17.1
		4	3261.29	2.86	0	14.1	Same		14.1	2.86	0	14.1
		5	1053.63	0	0	13.5	Same		13.5	0	0	13.5
13	500	1	17 8725.16	80 551.35	6001.64	90.7	78 008.48	4320.72	113.3	58 392.69	5107.40	249.5
		2	52 714.51	24 275.34	1359.87	67.1	6006.65	588.42	84.6	4897.92	312.28	105.5
		3	19 871.45	1098.25	195.75	65.8	921.76	138.87	72.2	821.82	30.62	84.5
		4	8132.80	152.50	49.01	74.2	134.37	0	58.4	123.30	4.62	57.5
		5	2942.23	0	0	56.8	Same		56.8	0	0	56.8

Note: 'Same' means that the solution values for both Z_{sol} and Z_{disp} for DALP-H2 were identical to those for DALP-H1.

Table 3 Computational results: large problems with displacement limit

Problem number	*Number of aircraft*	*Number of runways*	*Heuristic solution to each successive displacement problem*						*Optimal solution to each successive displacement problem DALP-OPT*		
			DALP-H1			*DALP-H2*					
			Z_{sol}	Z_{disp}	*Total time (min)*	Z_{sol}	Z_{disp}	*Total time (min)*	Z_{sol}	Z_{disp}	*Total time (min)*
9	100	1	25 807.86	128.88	3.6	19 606.45	0	5.2	9971.66	325.86	5.7
		2	940.60	63.34	3.2	599.60	40.98	3.1	599.60	40.98	3.6
		3	160.32	0	2.4	126.88	0	2.6	126.88	0	2.7
10	150	1	39 746.07	0	8.1	38 939.21	0	10.0	33 777.61	0	12.6
		2	6661.18	26.82	7.4	1708.01	34.76	7.1	1471.39	34.76	8.5
		3	329.05	7.28	6.4	325.86	7.28	5.8	246.15	7.28	6.2
11	200	1	42 516.85	0	14.3	35 976.61	0	17.5	31 450.44	52.30	22.1
		2	3766.69	38.76	17.0	2421.64	93.60	13.1	1910.73	52.10	15.3
12	250	1	48 514.77	0	24.6	42 652.68	19.38	28.7	32 744.58	76.92	37.1
		2	4186.92	64.86	20.5	3536.68	0	20.6	2391.19	34.94	24.1
		3	711.79	0	24.0	339.77	0	16.9	579.91	0	19.7
13	500	1	135 592.03	145.42	91.4	107 899.41	145.82	118.6	80 714.10	257.70	160.8
		2	28 616.54	39.82	85.8	8064.53	41.18	90.1	6167.20	63.04	105.1
		3	1587.48	183.50	72.6	1001.94	156.24	78.4	894.75	81.88	79.0
		4	230.74	0	70.7	128.58	0	63.0	128.58	0	63.0

Table 4 Computational results: large problems, zero freeze time

Problem number	*Number of aircraft*	*Number of runways*	*Heuristic solution to each successive displacement problem*						*Optimal solution to each successive displacement problem DALP-OPT*		
			DALP-H1			*DALP-H2*					
			Z_{sol}	Z_{disp}	*Total time (min)*	Z_{sol}	Z_{disp}	*Total time (min)*	Z_{sol}	Z_{disp}	*Total time (min)*
9	100	1	8185.25	3762.06	4.1	7800.93	2947.55	5.5	6219.54	1371.35	77.1
		2	510.40	205.44	3.8	480.00	117.12	4.6	480.58	83.40	9.7
		3	88.72	33.28	3.2	Same		3.2	88.72	33.28	4.0
		4	0	0	2.3	Same		2.3	0	0	2.4
10	150	1	23 569.43	4105.49	9.0	23 137.89	2885.86	12.0			(781.2:59)
		2	1361.51	418.55	7.7	1385.02	373.05	8.9	1317.69	81.00	119.3
		3	218.29	65.97	6.1	216.25	34.68	6.6	246.15	1.14	24.7
		4	47.20	0	5.1	34.22	0	5.1	34.22	0	6.0
		5	0	0	4.9	Same		4.9	0	0	5.0
11	200	1	16 474.79	5064.40	16.9	16 865.47	4843.90	21.5	14 152.40	2204.73	134.3
		2	1655.53	493.03	14.4	1654.30	484.10	17.1	1587.39	109.06	177.5
		3	347.37	0	10.5	333.53	0	11.0	333.53	0	485.6
		4	69.66	0	9.3	Same		9.5	69.66	0	31.4
		5	0	0	8.8	Same		8.8	0	0	8.9
12	250	1	25 547.94	4842.58	27.2	25 750.13	3922.92	39.9			(631.5:38)
		2	1934.68	778.28	21.8	1921.38	790.78	26.0			(833.4:37)
		3	265.24	44.10	15.2	Same		15.6	252.52	44.10	632.2
		4	2.86	0	14.0	Same		14.0	2.86	0	15.6
		5	0	0	13.7	Same		13.7	0	0	13.9
13	500	1	55 360.02	15 668.19	102.9	54 268.76	13 940.70	133.0			(923.6:279)
		2	29 683.09	2647.33	75.6	5440.55	855.13	101.3	4505.96	442.69	1290.9
		3	938.38	202.74	68.0	904.84	202.74	72.9	821.82	30.62	531.6
		4	152.50	49.01	57.2	120.65	17.16	57.6	122.49	3.81	228.7
		5	0	0	67.6	Same		67.6	0	0	67.6

Note: 'Same' means that the solution values for both Z_{sol} and Z_{disp} for DALP-H2 were identical to those for DALP-H1.
For DALP-OPT, a time limit of 10 h for the solution of any displacement problem was imposed, figures in brackets indicate the total time taken: the number of aircraft that had landed, when this time limit was reached.

Acknowledgements—JE Beasley acknowledges the financial support of the CSIRO Australia. We also acknowledge comments made on earlier versions of this paper by referees.

References

1 Beasley JE, Krishnamoorthy M, Sharaiha YM and Abramson D (2000). Scheduling aircraft landings — the static case. *Transport Sci* **34**: 180–197.

2 Beasley JE, Sonander J and Havelock P (2001). Scheduling aircraft landings at London Heathrow using a population heuristic. *J Opl Res Soc* **52**: 483–493.

3 Odoni AR, Rousseau J-M and Wilson NHM (1994). Models in urban and air transportation. In: Pollock SM, Rothkopf MH and Barnett A (eds) *Operations Research and the Public Sector: Handbooks in Operations Research and Management Science*, Vol 6. North-Holland: Amsterdam, The Netherlands, pp 107–150.

4 Erzberger H (1995). Design principles and algorithms for automated air traffic management: In: *Knowledge-based Functions in Aerospace Systems*. AGARD Lecture Series no. 200. NATO Neuilly-Sur-Seine, France, 7:1–7:31.

5 Milan J (1997). The flow management problem in air traffic control: a model of assigning priorities for landings at a congested airport. *Transport Plann Technol* **20**: 131–162.

6 Andreussi A, Bianco L and Ricciardelli S (1981). A simulation model for aircraft sequencing in the near terminal area. *Eur J Opl Res* **8**: 345–354.

7 Dear RG and Sherif YS (1989). The dynamic scheduling of aircraft in high density terminal areas. *Microelectron Reliab* **29**: 743–749.

8 Dear RG and Sherif YS (1991). An algorithm for computer assisted sequencing and scheduling of terminal area operations. *Transport Res A* **25A**: 129–139.

9 Dear RG (1976). *The dynamic scheduling of aircraft in the near terminal area*, Report R76-9 Flight Transportation Laboratory, MIT, Cambridge, MA, USA.

10 Brinton CR (1992). An implicit enumeration algorithm for arrival aircraft scheduling. In: *Proceedings of the 11th IEEE/AIAA Digital Avionics Systems Conference Seattle, WA*. IEEE, NY, USA, pp 268–274.

11 Venkatakrishnan CS, Barnett A and Odoni AR (1993). Landings at Logan Airport: describing and increasing airport capacity. *Transport Sci* **27**: 211–227.

12 Psaraftis HN (1978). *A dynamic programming approach to the aircraft sequencing problem*, Report R78-4 Flight Transportation Laboratory, MIT, Cambridge MA, USA.

13 Psaraftis HN (1980). A dynamic programming approach for sequencing groups of identical jobs. *Opns Res* **28**: 1347–1359.

14 Ciesielski V and Scerri P (1997). An anytime algorithm for scheduling of aircraft landing times using genetic algorithms. *Aust J Intell Inf Process Systems* **4**: 206–213.

15 Ciesielski V and Scerri P (1998). Real time genetic scheduling of aircraft landing times. In: Fogel D (ed) *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation (ICEC98)*. IEEE, NY, USA, pp 360–364.

16 Carr GC, Erzberger H and Neuman F (1998). *Airline arrival prioritization in sequencing and scheduling*. Paper presented at the second USA/Europe Air Traffic Management R&D Seminar, Orlando. Available from. http://www.ctas.arc.nasa.gov/publications.

17 Carr GC, Erzberger H and Neuman F (1999). Delay exchanges in arrival sequencing and scheduling. *J Aircraft* **36**: 785–791.

18 Carr GC, Erzberger H and Neuman F (2000). Fast-time study of airline-influenced arrival sequencing and scheduling. *J Guidance Control Dyn* **23**: 526–531.

19 Bolender MA and Slater GL (2000). Evaluation of scheduling methods for multiple runways. *J Aircraft* **37**: 410–416.

20 Wong GL (2000). *The dynamic planner: the sequencer, scheduler, and runway allocator for air traffic control automation*, Report NASA/TM-2000-209586, NASA Ames Research Center, Moffett Field, CA, USA, Available from. http://www.ctas.arc.nasa.gov/publications.

21 ILOG Inc (1999). *ILOG CPLEX 6.5 User's Manual*. ILOG Inc.: Mountain View, CA, USA.

22 Beasley JE (1990). OR-Library: distributing test problems by electronic mail. *J Opl Res Soc* **41**: 1069–1072.

23 Beasley JE (1996). Obtaining test problems via Internet. *J Global Optim* **8**: 429–433.

Received January 2003;
accepted October 2003 after two revisions