Journal of the Operational Research Society (2001) 52, 483–493 ©2001 Operational Research Society Ltd. All rights reserved. 0160-5682/01 $15.00

www.palgrave.com/jors

Scheduling aircraft landings at London Heathrow using a population heuristic

JE Beasley[1]*, J Sonander[2] and P Havelock[3]

[1]*Imperial College of Science, Technology and Medicine, London, UK;* [2]*National Air Traffic Services, London, UK; and* [3]*Civil Aviation Authority, London, UK*

With increasing levels of air traffic, making effective use of limited airport capacity is obviously important. This paper reports on an investigation undertaken by National Air Traffic Services in the UK into improving runway utilisation at London Heathrow. This investigation centred on developing an algorithm for improving the scheduling of aircraft waiting to land. The heuristic algorithm developed (a population heuristic) is discussed and results presented using actual operational data relating to aircraft landings at London Heathrow. This data indicates that our algorithm could have improved on air traffic control decisions in such cases by between 2–5% in terms of reducing the timespan required to land all of the aircraft considered.

Keywords: air traffic control; aircraft landing; population heuristic

Introduction

It is well known that throughout the world air traffic is increasing. With these increases in traffic there is an increasing demand for decision support tools to make effective use of the limited capacity (airspace, runways, etc) available. The UK is no exception to this world growth in air traffic, and within the UK, National Air Traffic Services (NATS) are responsible for effective management of a number of aspects of the UK air traffic control system. This paper discusses how a modern heuristic technique (a population heuristic) can be used in a decision support tool to enable more effective decisions to be made with respect to one aspect of the air traffic control system, the scheduling of aircraft waiting to land.

The UK's busiest airport is London Heathrow. Most, if not all, of the readers of this paper will have flown in or out of Heathrow. At Heathrow one runway is typically used solely for takeoffs and the other runway solely for landings. This mode of operation is known as segregated mode and has obvious advantages in terms of controlling the movement of aircraft. Mixed-mode operation (where the same runway is used both for takeoffs and landings) is rarely used at Heathrow.

This paper reports on an investigation undertaken by NATS into improving runway utilisation at London Heathrow. This investigation centred on developing an algorithm for improving the scheduling of aircraft waiting to land. In the short term, improved scheduling would mean less delays for passengers as aircraft would land quicker than they otherwise would have done. In the long term, improved scheduling would give potential for increasing the number of flights scheduled in/out of Heathrow. The algorithm developed, a population heuristic, is described in this paper and computational results presented for operational data relating to landings at London Heathrow.

Aircraft landing

The decision problem

At the simplest conceptual level one can imagine a set of aircraft circling Heathrow waiting to land and a single air traffic controller. This air traffic controller has a simple decision problem to solve—*which aircraft should land next?* This decision problem is one that has to be solved repeatedly over the course of a working day. If a decision support tool could be developed to assist (not replace) the controller in making this decision then perhaps more effective use of runway capacity could be made.

A controller actually has to do more than decide which aircraft lands next. They have to think ahead and (implicitly or explicitly) form the set of aircraft waiting to land into a landing *sequence*, namely decide the order in which the aircraft will land. But deciding the sequence in which aircraft will land is not sufficient. A controller must go

*Correspondence: *JE Beasley, The Management School, Imperial College of Science, Technology and Medicine, London SW7 2AZ, UK.*
E-mail: j.beasley@ic.ac.uk

further and (implicitly or explicitly) form a *schedule*, namely decide a landing time for each aircraft. This time schedule is necessary because the controller has to ensure that:

(a) an aircraft has time to safely fly from its current position to the runway so as to land at the appropriate position in the sequence; and
(b) aircraft do not run low on fuel whilst waiting to land; and
(c) aircraft do not land too close together.

These first two conditions (a) and (b) imply that for each aircraft there is a window of time within which it must land, whilst the third condition (c) means that a reasonable amount of time must elapse between successive landings.

Note here that throughout this paper we use the phrase 'scheduled landing time' to mean the time at which an aircraft is scheduled to land by a controller, not the time at which an airline advertised a flight as being scheduled to land in its timetable.

Constraints

In the aircraft landing problem the constraints of the problem are that each aircraft must land within a predetermined time window; and that the time between a particular aircraft landing, and the landing of any successive aircraft, must be greater than a specified minimum (the *separation time*) which is dependent upon the aircraft involved.

Separation times are bounded below by aerodynamic considerations. Aircraft in flight generate *wake vortices* as an unwanted product of lift. Lift is generated by the pressure difference between the upper and lower surfaces of the wing, the pressure at the lower surface being greater than the pressure at the upper surface. At the wing tips this difference in pressure between the upper and lower wing surfaces generates a rotating mass of air—an effect known as wake vortex. The strength of the wake vortices caused by an aircraft is proportional to its mass. Wake vortices can cause turbulent conditions for an aircraft following too close, with associated passenger discomfort and possible damage to the following aircraft. Indeed a number of aircraft accidents are believed to have been caused by this phenomena.[1]

For safety reasons therefore landing a large aircraft necessitates a (relatively) large time delay before other aircraft can land. A light aircraft, by contrast, generates little air turbulence and therefore landing such an aircraft necessitates only a (relatively) small time delay before other aircraft can land.

Within the UK aircraft are classified into five types: Heavy, Upper-Medium, Lower-Medium, Small and Light for separation purposes at London Heathrow. These types are essentially related to aircraft sizes (as measured by maximum takeoff weight). Separation distances on landing are defined to be:

Leading aircraft	*Following aircraft*	*Separation distance (nautical miles)*
Heavy	Heavy	4
	Medium (Upper or Lower)	5
	Small	6
	Light	7
Upper-Medium	Lower-Medium or Small	4
	Light	6
Lower-Medium	Light	5

All separation distances not explicitly shown above (eg leading aircraft Lower-Medium, following aircraft Heavy) are 3 nautical miles.

Separation is defined here in terms of distance. The usual approach for converting these separation distances into separation times is to assume a standard landing speed for each class of aircraft. For the purposes of the results reported in this paper this standard landing speed has been taken as 160 knots (nautical miles per h) for all aircraft types. This implies that the time between the successive landing of two Heavies, for example, must be at least $3600(4/160)=90$ s. We note in passing here that there were no Light aircraft in any of the Heathrow data sets we considered.

The separation distances given above are *crucial factors* in terms of runway (and hence airport) capacity:

(a) The minimum separation distance is 3 nautical miles, which implies a separation time of $3600(3/160)=67.5$ s. This implies that, *at most*, $3600/67.5=53.3$ aircraft can be landed per h per runway. Irrespective of airspace restrictions, runway capacity is a severe bottleneck upon the amount of traffic that an airport can deal with.
(b) As the separation distances vary between classes there is scope, given a set of aircraft of different classes, for more effective decisions.

With regard to this second point suppose that there is a set of three aircraft, two Heavies (Hs) and an Upper-Medium (UM), waiting to land, There are three possible distinct landing sequences: H–H–UM, H–UM–H and UM–H–H. The separation distances involved are $4+5=9$, $5+3=8$ and $3+4=7$, respectively. It is clear that the last of these sequences (UM–H–H) involves the least separation distance and hence the least time, ie deciding to adopt this sequence would land the entire set of aircraft in the shortest possible time.

Whilst for a three-aircraft example all possible landing sequences can be easily enumerated, in general the number of distinct landing sequences grows combinatorially with the number of aircraft waiting to land. For this reason an enumeration approach would not be practicable for larger problems.

Objective

In our three-aircraft example above we chose a sequence so as to land the entire set of aircraft in the shortest possible time. Previous work on the aircraft landing problem[2–4] has taken into consideration a target landing time for each aircraft and considered deviations from this target time as important. All of that work considered *linear* (or linearisable) objective functions.

Considering deviations from target landing time was also the approach adopted in this investigation. Computational experience though indicated that, in the view of NATS personnel, a *nonlinear* objective would in many cases be necessary in order to achieve acceptable results (landing sequences with desirable characteristics).

In order to try and arrive at an appropriate objective function we considered two questions:

(1) which factors should be included in the objective function; and
(2) for the factors that we do decide to include, how should they be combined together in a mathematical formula for the objective function.

With regard to the first of these questions consideration was given to factors such as deviation from target time (as mentioned above) and the origin of the flight (European or non-European). It might seem natural to also consider the number of passengers on each aircraft, so that for example we expedite the landing of an aircraft with many passengers, whilst delaying the landing of an aircraft with few passengers. There are a number of problems with including this factor in the objective function:

(a) The data relating to separation distances on landing is such that, from the viewpoint of runway utilisation, the 'natural order' is to land smaller aircraft first (eg in the example given above the sequence that landed all aircraft as quickly as possible was UM–H–H). As such we may have to be prepared to delay larger aircraft (irrespective of their passenger load) in order to gain better runway utilisation.
(b) Airlines regard passenger numbers as commercially sensitive data and there would be difficulties in obtaining such data from them: indeed, data on aircraft capacity (maximum number of passengers) would probably have to be used instead.
(c) If we include passenger numbers in deciding the scheduling of aircraft landings then we will probably be biasing the solution in favour of airlines operating larger aircraft, which may not be desirable in the interests of competition.

With regard to the mathematical formula for the objective function we have, as will become apparent below, adopted an objective involving squared deviations from target time. Other formulae, eg adopting higher powers, utilising an exponential function, weighting together normalised factors, are all possible. We would make two points:

(a) One of the advantages of the solution methodology adopted (a population heuristic) is that the nature of the objective function (linear or nonlinear) is irrelevant in terms of the solvability of the problem (although naturally different objective functions lead to different results). Moreover, varying the objective function can be accomplished relatively easily. This ability to adopt differing objective functions was, in the view of NATS personnel, a desirable feature of the algorithm developed.
(b) We believe that the practical air traffic control benefits of adopting an algorithmic approach (such as given in this paper) to the problem of scheduling aircraft landings will be due to a consistent application of a systematic approach, rather than due to the precise details of the objective function formula adopted.

Control

The population heuristic presented in this paper solves a *decision problem*. That is, it solves the problem of deciding landing times which lie in aircraft time windows and which satisfy the separation criteria, whilst optimising an appropriate objective. This decision problem leaves to one side the associated *control problem*, namely can the aircraft be flown (controlled) in such a manner as to enable the solution to the decision problem to be safely implemented (eg separation must be maintained not only at landing but throughout aircraft descent onto the runway).

Plainly the decision problem and the control problem are linked. However, as in many Operations Research models, we believe that there are benefits to be gained in separating out these two aspects of the problem. We would also comment here that, across the world, the common[3,5,6] practice of scheduling aircraft to land in a first-come, first-served, manner also effectively utilises a (trivial) decision problem, whilst leaving the control problem for later resolution.

Population heuristics

Population heuristics started with the development of genetic algorithms by Holland.[7] A population heuristic (PH) can be described as an 'intelligent' probabilistic search algorithm and is based on the evolutionary process of biological organisms in nature. During the course of evolution, natural populations evolve according to the principles of natural selection and 'survival of the fittest'. Individuals who are more successful in adapting to their environment will have a better chance of surviving and reproducing, whilst individuals who are less fit will be eliminated. This means that the *genes* from highly fit individuals will spread to an increasing number of individuals in each successive generation. The combination of good characteristics from highly adapted parents may produce even more fit offspring. In this way, species evolve to become increasingly better adapted to their environment.

A PH simulates these processes in a computer by taking an initial population of individuals and applying genetic operators in each reproduction. In optimisation terms, each individual in the population is encoded into a string or *chromosome* which represents a possible *solution* to a given problem. The fitness of an individual is evaluated with respect to a given objective function. Highly fit individuals or *solutions* are given opportunities to reproduce by exchanging pieces of their genetic information, in a *crossover* procedure, with other highly fit individuals. This produces new 'offspring' solutions (ie children), who share some characteristics taken from both parents. Mutation is often applied after crossover by altering some genes in the strings. The offspring can either replace the whole population (*generational* approach) or replace less fit individuals (*steady-state* approach). This evaluation–selection–reproduction cycle is repeated until a satisfactory solution is found. The basic steps of a simple PH are:

```
Generate an initial population
Evaluate fitness of individuals in the population
repeat
–  Select individuals from the population to be parents
–  Combine (mate) parents to produce children
–  Mutate the children
–  Evaluate fitness of the children
–  Replace some or all of the population by the children
until
–  you decide to stop whereupon report the best solution
   encountered
```

Readers wishing to learn more about population heuristics are referred to references 8–12.

For readers who have some familiarity with PHs we would comment here that one of the difficulties of working with PHs is that there are *many* schemes for doing the basic operations that you need:

- parent selection: binary tournament, K tournament, ranking, fitness proportionate, etc.;
- having children: uniform, one-point, restricted one-point, fusion, two-point, etc.;
- mutation: constant, adaptive;
- population replacement: steady-state, generational (with/without elitism), island models.

In order to avoid exhaustive examination of a large number of combinations of possible operations we have limited ourselves to those operations outlined in the following sections. Those choices have been made in the light of our experience[13–19] in developing computationally successful PHs and limited computational experimentation with different choices for various operations.

A population heuristic for aircraft landing

In this section we will outline the PH that we have developed for the aircraft landing problem. For reasons relating to commercial confidentiality however, we have had to omit from our discussion a number of the computational devices we adopted to speed the convergence of our population heuristic.

Notation

In order to explain our PH we introduce the following notation. Let:

P be the number of aircraft
E_i be the earliest landing time for aircraft i ($i = 1, \ldots, P$)
L_i be the latest landing time for aircraft i ($i = 1, \ldots, P$)
T_i be the target (preferred) landing time for aircraft i ($i = 1, \ldots, P$)
S_{ij} be the required separation time ($\geqslant 0$) between aircraft i landing and aircraft j landing (where aircraft i lands before aircraft j), $i = 1, \ldots, P$; $j = 1, \ldots, P$; $i \neq j$.

All of the above have known values in any given situation and typically all times are expressed in seconds. Our decision variables (what we are trying to decide) are:

x_i the scheduled landing time for aircraft i

The problem therefore is to decide values for the x_i which lie in the time windows [E_i, L_j] and satisfy the separation criteria, whilst attempting to ensure that aircraft land at (or before) their target time. Below we discuss the elements of our PH relating to:

- representation, how we represent a solution to the problem in our PH;
- fitness, assessing the value of a solution;
- parent selection, choosing who shall have a child;
- crossover, having a child from parents;
- dealing with the separation constraints;
- population replacement, placing the child in the population.

Each of these elements is discussed separately below.

Representation

The first decision in any PH is how to represent a solution to the problem. For the problem of scheduling aircraft landings we used a real-numbered representation comprising of P real numbers, which we will call y_i, satisfying $0 \leqslant y_i \leqslant 1$. The value of y_i for aircraft i represents the proportion of the interval [E_i, L_i] that elapses before aircraft i lands, ie the scheduled landing time x_i is given by

$x_i = E_i + y_i(L_i - E_i)$. For example, if $[E_i, L_i]$ is [100, 300] (expressed in terms of seconds from a given base time) and $y_i = 0.45$ then $x_i = 100 + 0.45(300 - 100) = 190$.

Hence each individual in our PH world for scheduling aircraft landings has the representation:

$$[y_1, y_2, \ldots, y_P]$$

Such a representation has the property that we automatically ensure that the computed scheduled landing time x_i lies within the required time window $[E_i, L_i]$.

For example, suppose that $P = 3$ with time windows [100, 300], [200, 1700] and [50, 200] for aircraft 1, 2 and 3, respectively. If a PH individual is [0.45, 0.21, 0.71] then the computed landing times are $x_1 = 100 + 0.45(300 - 100) = 190$, $x_2 = 200 + 0.21(1700 - 200) = 515$ and $x_3 = 50 + 0.71(200 - 50) = 156.5$ (rounded to 157).

Fitness

Fitness in PHs relates to assessing the value (worth) of an individual. This issue relates to the overall objective, namely what are we trying to achieve when we schedule a set of aircraft to land. After discussion and consideration of initial results with NATS personnel a number of different fitness functions were identified. All of these functions were based on the difference between the scheduled landing time and the target landing time. For the purposes of brevity in this paper, and in order to preserve commercial confidentiality, we shall only present detailed discussion relating to some of these functions.

Let $D_i = (x_i - T_i)$ be the difference (deviation) between the scheduled landing time x_i and the target (preferred) landing time T_i for aircraft i. Note in particular that, if $D_i < 0$, then an aircraft lands before its target (preferred) time and if $D_i > 0$ then an aircraft lands after its target (preferred) time, ie it is delayed. Then our fitness function is

$$\text{fitness} = \sum_{i=1}^{P} z_i \quad \text{where } z_i = \begin{cases} -(D_i)^2 & \text{if } D_i \geqslant 0 \\ +(D_i)^2 & \text{otherwise} \end{cases}$$

This fitness function is nonlinear: large deviations receive a disproportionately larger weighting. Since we seek to maximise fitness this function implies that we would prefer to land aircraft before their target time and dislike landing aircraft after their target time. As mentioned previously computational experience indicated that, in the view of NATS personnel, a nonlinear objective such as the one shown here would in many cases be necessary in order to achieve acceptable results (landing sequences with desirable characteristics).

Parent selection

In our PH we have two parents coming together to have a single child. In order to choose which individuals can become parents we used binary tournament selection. This works as follows:

(a) from the population select two individuals at random. Of the two that have been selected the one with the best (maximum) fitness will be the first parent.
(b) repeat the process described in (a) above to obtain a second parent.

This procedure can result in the two parents being the same individual, but this occurs infrequently if the population is large.

Crossover

We used uniform crossover. In uniform crossover two parents have a single child. In our PH each of the P values in the child are taken from one or other parent chosen at random. For example, assuming $P = 3$, if the parents are

$$[0.45, 0.21, 0.71]$$

and

$$[0.93, 0.31, 0.75]$$

and our random choice of parents for the 3 values (aircraft) is

aircraft 1–parent 1
aircraft 2–parent 2
aircraft 3–parent 2

then the child is

$$[0.45, 0.31, 0.75]$$

Although we experimented with mutating each child our experience was that this did not have a significant impact upon the results obtained.

Dealing with the separation constraints

Probably the most fundamental difficulty in applying PHs to the problem of scheduling aircraft landings is to try and ensure that the PH generates solutions that are feasible (satisfy the constraints). As discussed above the representation we have adopted automatically means that the time window constraints are satisfied. This therefore leaves the separation constraints. In our PH for scheduling aircraft landings we dealt with the problem of attempting to ensure that the separation constraints are satisfied by separating the evaluation of fitness and infeasibility. We associate with each individual in our PH for scheduling aircraft landings two numbers, (unfitness, fitness), where fitness is our adopted objective function value (which we are trying to maximise) and unfitness is a measure of constraint violation.

In scheduling aircraft landings the separation constraints may be conveniently expressed mathematically as

$$x_j - x_i \geqslant S_{ij} \quad \text{if } x_i \leqslant x_j, \quad i = 1, \ldots, P; \; j = 1, \ldots, P; \; j \neq i$$

ie that the difference in the scheduled landing times is at least S_{ij} if i lands before $j(x_i \leqslant x_j)$. Given a PH solution $[y_1, y_2, \ldots, y_P]$ we can compute the scheduled landing times $x_i (i = 1, \ldots, P)$ as indicated previously. These may, or may not, satisfy the separation constraints. The quantity

$$\sum_{i=1}^{P} \sum_{\substack{j=1 \\ j \neq i,\, x_i \leqslant x_j}}^{P} \max[0, S_{ij} - (x_j - x_i)]$$

will be zero if the scheduled landing times satisfy the separation constraints and nonzero (strictly positive) if the scheduled landing times do not satisfy the separation constraints. This quantity is the *unfitness* associated with any PH solution and is a measure of constraint infeasibility. Informally the higher the unfitness value the more infeasible a solution.

For example, suppose $P = 3$ and a PH solution has associated with it scheduled landing times $x_1 = 10$, $x_2 = 200$ and $x_3 = 250$. If the required separation times are $S_{12} = 180$, $S_{13} = 200$, and $S_{23} = 80$ then plainly:

(a) the time that elapses between the landing of the first and second aircraft $(x_2 - x_1 = 190)$ satisfies the required separation $(S_{12} = 180)$; and
(b) the time that elapses between the landing of the first and third aircraft $(x_3 - x_1 = 240)$ satisfies the required separation $(S_{13} = 200)$; but
(c) the time that elapses between the landing of the second and third aircraft $(x_3 - x_2 = 50)$ does not satisfy the required separation $(S_{23} = 80)$.

For this PH solution the unfitness is

$$\sum_{i=1}^{P} \sum_{\substack{j=1 \\ j \neq i,\, x_i \leqslant x_j}}^{P} \max[0, S_{ij} - (x_j - x_i)]$$

$$\begin{aligned} &= \max[0,\ S_{12} - (x_2 - x_1)] + \max[0,\ S_{13} - (x_3 - x_1)] \\ &\quad + \max[0,\ S_{23} - (x_3 - x_2)] \\ &= \max[0,\ 180 - (200 - 10)] + \max[0,\ 200 - (250 - 10)] \\ &\quad + \max[0,\ 80 - (250 - 200)] \\ &= \max[0,\ -10] + \max[0,\ -40] + \max[0,\ 30] \\ &= 0 + 0 + 30 = 30 \end{aligned}$$

where the first two zeros in this sum indicate that separation was satisfied both for aircraft 1 and 2 and for aircraft 1 and 3, but the 30 indicates that separation was violated (by 30 time units) for aircraft 2 and 3. It is clear from the above that any PH solution with unfitness zero is feasible.

Informally this use of (unfitness, fitness) allows us to design our PH to evolve feasible solutions (ie solutions with unfitness zero) through an appropriate population replacement scheme. Note here, however, that we did not consider unfitness in parent selection. We have found computationally that it is sufficient to consider just fitness in parent selection, provided that the population replacement scheme adopted considers both fitness and unfitness in the manner detailed below.

Population replacement

We used a steady-state population replacement scheme that makes use of the two numbers (unfitness, fitness) associated with each PH solution. Suppose that we have produced a child in our PH. This child will have certain (unfitness, fitness) values. For the purposes of illustration we shall assume that this child has unfitness > 0, ie is not feasible.

In order to explain our population replacement scheme suppose we draw a graph with fitness as the vertical axis and unfitness as the horizontal axis, as in Figure 1. Now plot on this graph:

(a) the PH child; and
(b) each member of the population, as each population member will also have (unfitness, fitness) values. For the purposes of illustration these population members are shown as X's in Figure 1, scattered around the child.

Considering Figure 1 it is clear that the presence of the child has naturally divided the population into four groups corresponding to each of the four quadrants created by drawing horizonal and vertical lines through the child. These four groups are labelled G1–G4 in Figure 1. Informally it is clear that:

(a) Our child is superior to any member of the population that is in group G1, as all members of that group are more infeasible (have a higher unfitness) and are worse in objective function terms (have a lower fitness).

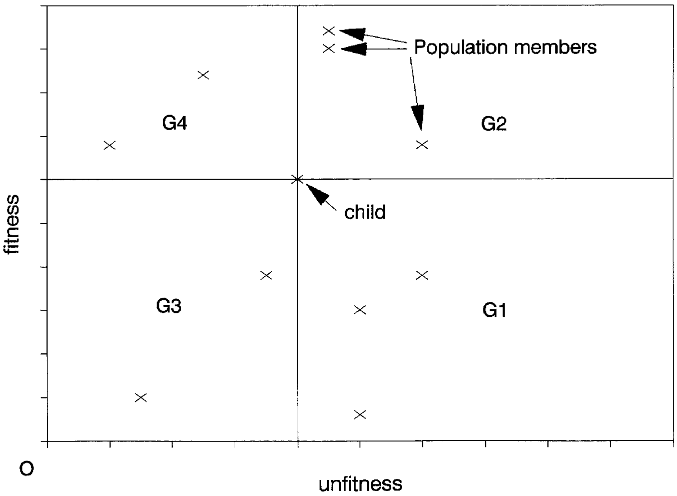

Figure 1 Population replacement.

(b) Our child is inferior to any member of the population that is in group G4, as all members of that group are less infeasible (have a lower unfitness) and are better in objective function terms (have a higher fitness).
(c) Our child is superior with respect to one measure (unfitness or fitness), but inferior with respect to the other, when compared to members of groups G2 or G3.

Now as we have a child we need to add this child to the population. To keep the population size constant (the standard PH assumption) we need to choose a member from the population to kill (ie to replace the chosen member by the child).

We use our four groups (G1–G4) in order to establish in which 'area' of the population we look to find a member to kill. In our population replacement scheme we:

(a) First look in group G1. As stated above, any population member in this group is worse than the child on both measures (fitness and unfitness). Hence we should plainly choose a member of the population from this group to kill if we can do so.
(b) It may be, however, that G1 is empty (ie contains no population members). If this is the case we look in G2, members of this group are inferior to the child in terms of unfitness.
(c) If both G1 and G2 are empty we look in G3, members of this group are inferior to the child in terms of fitness.
(d) Finally if G1, G2 and G3 are all empty we look in G4. Members of this group are, as mentioned above, superior to the child on both measures (fitness and unfitness). It might appear strange therefore to choose a member of this group to kill. However, computational experience with this population replacement scheme has been that unless we perturb the population by placing the child in the population the PH fails to make progress.

Once the appropriate group (G1, G2, G3 or G4) has been identified as outlined above we kill a randomly selected member of the group. In the computational results presented below we used a population of size 100 with the initial population being randomly generated.

Computational results

Data sets

In the investigation undertaken by NATS a number of different data sets relating to landings at London Heathrow were considered. In the interests of brevity in this paper, however, we will present results for just a single data set. This data set is based on a 'snapshot', taken at 0815 on Friday 11 September 1998 by the London Air Traffic Control Centre at West Drayton. This snapshot considered all aircraft within 100 nautical miles of Heathrow which were scheduled to land at London Heathrow and contained 20 aircraft. This particular day/time was chosen for a snapshot as it was considered to be a very busy period.

For all aircraft in the snapshot NATS personnel generated earliest and latest landing times based upon aircraft positions and standard speeds. We took as the target landing time for each aircraft the actual landing time as recorded in the Heathrow runway logs. This enables us to compare the performance of the PH against the actual performance achieved on the day in question. With regard to computation time all of the PH results presented here are produced in approximately two seconds on a 333 MHz Pentium PC.

Results

In order to compare the results achieved by our PH with controller sequencing decisions the case where aircraft land in the same order as they did on the day in question, but respecting time windows and with the *minimum* possible separation time, we refer to as the ACTUAL case. By using the minimum possible separation time here we are able to compare the controller sequencing decisions against the PH sequencing decisions with any effects due to aircraft not landing with the minimum possible separation time removed from the comparison.

Figure 2 shows a pictorial representation of the results of our PH for the nonlinear objective function (fitness function) given previously. In that Figure we have along the vertical axis the scheduled landing time as decided by the PH. Aircraft are numbered in that Figure in the order in which they landed in the ACTUAL case. It can be seen that the third aircraft to land is designated by '2-UMED:e'. This means that it was the second aircraft to land in the ACTUAL sequence, is category Upper-Medium and the ':e' means that it is a European flight (ie originated within Europe). The second aircraft to land is designated by '3-HEAVY'. This designation means that it was the third aircraft to land in the ACTUAL sequence, is category HEAVY and the lack of ':e' means that it is not a European flight (ie did not originate within Europe). We would comment here that figures of the type shown, which have a natural interpretation as aircraft descending onto the runway, have been found to be very helpful in communicating our PH results to others within NATS.

Considering Figure 2 it is clear that the PH has arrived at a substantially different landing sequence than the controller-decided ACTUAL sequence (which was first 1-HEAVY, then 2-UMED:e, then 3-HEAVY, then 4-HEAVY:e, etc). It is perhaps worth stressing here that our PH has no knowledge of the controller sequence when arriving at a solution.

In order to get a numeric evaluation of the results of our PH as compared with the ACTUAL case we consider:

(a) timespan—the time to land all the aircraft, $\max[x_i \mid i = 1, \ldots, P]$

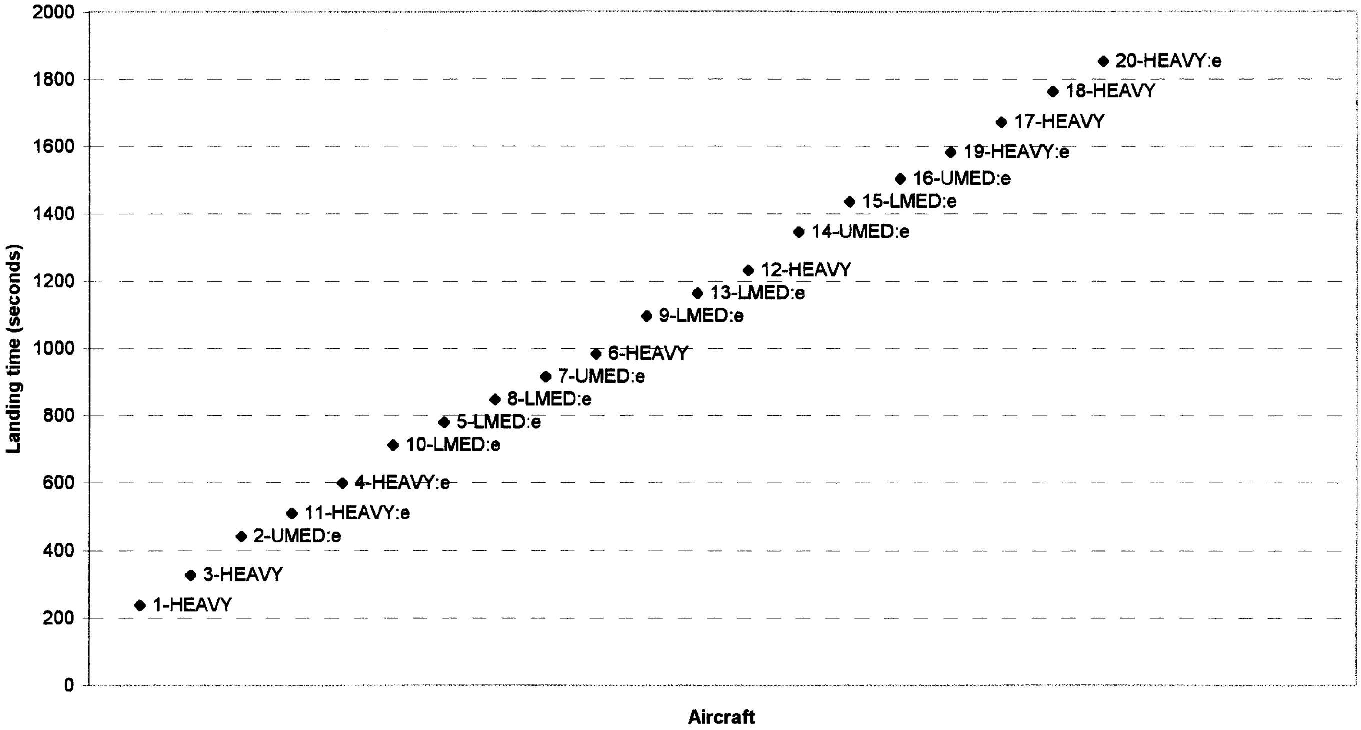

Figure 2 PH landing schedule.

(b) average delay—$\sum_{i=1}^{P} D_i/P$
(c) maximum delay—$\max[D_i \mid i = 1, \ldots, P]$

Timespan was felt to be of importance as it gives some insight into the possibility of gaining extra capacity in the long term. Average delay and maximum delay give insight into the immediate effects of improved sequencing decisions.

Comparing the PH solution shown in Figure 2 with the ACTUAL case timespan decreased from 1894 to 1853 s (ie 41 s were saved), average delay decreased by 40 s, but the maximum delay increased by 187 s.

The significance of these results depends upon your viewpoint. From the viewpoint of an aircraft passenger a reduction in average delay of 40 s is hardly significant/noticeable. However, from the viewpoint of London Heathrow, and of NATS, reducing the timespan by 41 s equates to a percentage time saving of 100(41/1894) = 2.16 %, since in the ACTUAL case the last aircraft landed 1894 s after the start of the time period. Were this to be repeated across time such a saving would have the potential for Heathrow to cope with (approximately) one extra landing per h. This would be a significant improvement.

The results quoted above (reduced timespan and reduced average delay, but increased maximum delay) are indicative of the *tradeoffs* that occur in deciding a landing schedule. Indeed, across the data sets we considered in our investigation, we commonly found that timespan and average delay could be reduced, provided we were prepared to see an increase in the maximum delay.

As an indication of the effectiveness of the PH presented in this paper we have also solved the problem shown in Figure 2, but using the classical improvement heuristic outlined in Algorithm 1. The best solution obtained by that heuristic had a fitness value only 59 % of the final fitness value achieved by the PH. This is a clear indication that the extra complexity, over and above a classical heuristic, involved in our PH is contributing significantly to the final solution obtained.

The reason for presenting the flight origin (European or not) in Figure 2 is that some of the objective functions NATS considered treated European and non-European flights differently. NATS were experimenting to see how the objective function could be adapted to take into account

Algorithm 1 Improvement heuristic

```
(1) Generate three different initial landing sequences for aircraft
    in terms of their earliest, target and latest times.
(2) Improve each of these sequences in turn by:
    (a) for a single aircraft moving it to any position in the
        landing sequence that improves the solution; and
    (b) for pairs of aircraft swapping their positions in the
        landing sequence if that improves the solution; and
    (c) repeating all such moves until no further improvement
        can be achieved.
(3) Return the best solution found after examining and improving
    all three initial sequences.
```

the fact that a ten minute delay (say) to a particular group of flights results in more disruption (missed passenger connections, later flights by the same aircraft/crew being delayed) than the equivalent delay to other flights. The PH solution for the case where European flights are regarded as three times more important than non-European flights is given in Table 1. In that table we show for each aircraft its position in the landing sequence in Figure 2, and its position in the landing sequence when this European weighting is applied. It can be seen that aircraft 3 for example, a non-European flight, is moved significantly later in the landing sequence (from 2nd to 11th). Compared with the solution shown in Figure 2 the timespan is exactly the same: average delay is 7 s less but the maximum delay is increased by 368 s. Solution characteristics for the landing sequences considered here can be seen in Table 2.

Although we have considered numeric time measures (timespan, average delay, maximum delay) above, the PH itself, when attempting to maximise its fitness function, does not directly consider such delays. Instead, when the PH looks for a solution, it *single-mindedly focuses solely on the objective function it is given*. It takes no account of other measures (such as average delay) which we have considered here. In fact, we can take our PH and use it to directly minimise average delay (say). The solution produced by our PH for this case is shown in Table 1. Comparing that PH solution with the ACTUAL case timespan decreased by 98 s, average delay decreased by 66 s, but the maximum delay increased by 164 s. This PH solution equates to a percentage time saving of 100(98/1894) = 5.17%. Were this to be repeated across time such a saving would have the potential for Heathrow to cope with between two and three extra landings per h. This would be a very significant improvement.

The reader may be wondering as to why, if minimising average delay leads to such potential improvement, NATS did not just adopt a simple linear objective such as minimise average delay. The reason lies in a detailed consideration of the schedule produced by such an objective. If one looks beyond simple summary statistics such as average and maximum delay into the *distribution* of delays it may be (as a matter of judgement) that it is felt that too many aircraft are suffering unacceptable delays. For this reason a nonlinear objective, such as the one presented before, which

Table 1 Landing sequences

Aircraft ACTUAL sequence	*Classification*	*Origin*	*PH landing sequence: (as in Figure 2)*	*PH landing sequence: European weighting*	*PH landing sequence: minimise average delay*
1	HEAVY		1	1	1
2	UMED	European	3	2	3
3	HEAVY		2	11	2
4	HEAVY	European	5	3	5
5	LMED	European	7	7	7
6	HEAVY		10	10	10
7	UMED	European	9	9	4
8	LMED	European	8	8	9
9	LMED	European	11	6	8
10	LMED	European	6	5	6
11	HEAVY	European	4	4	11
12	HEAVY		13	14	12
13	LMED	European	12	13	13
14	UMED	European	14	12	15
15	LMED	European	15	15	14
16	UMED	European	16	16	16
17	HEAVY		18	18	17
18	HEAVY		19	20	20
19	HEAVY	European	17	17	18
20	HEAVY	European	20	19	19

Table 2 Solution characteristics

	ACTUAL sequence	*PH landing sequence: (as in Figure 2)*	*PH landing sequence: European weighting*	*PH landing sequence: minimise average delay*
Timespan (s)	1894	1853	1853	1796
Average delay (s)	26	−14	−21	−40
Maximum delay (s)	77	264	632	241
Fitness value	−28 555	329 686	123 520	120 489
European weighting fitness value	−167 587	3 683 454	5 385 048	1 516 337

penalises large delays disproportionately, can produce more acceptable results.

Remarks

As commented previously we have not reported here on all of the objective functions examined/computational experiments done. However, as a result of all the work that we did do we believe that we can make one key remark: *the PH developed is able to quickly (in a matter of seconds) and effectively schedule aircraft landings using a variety of objective functions*. As such it clearly has potential as a technique to improve runway efficiency/utilisation. We would also make a number of additional remarks:

(a) The speed at which decisions as to scheduled landing times can be obtained via the PH (a few seconds) means that it is feasible to implement our PH inside a decision support tool to aid controllers in their task of scheduling aircraft to land at London Heathrow. The PH could be used to produce advisory instructions to enable a controller to achieve a sequence dynamically in real time. Several different solutions with differing characteristics could be generated by the PH (using different objective functions) and presented to the controller. This would permit human judgement to be retained in the process, keeping the controller 'in the loop' and sharing the task between human and computer.

(b) Our PH has, in a matter of seconds, sequenced a set of 20 aircraft that will, in real time, require approximately 30 min to land at Heathrow. Whilst obviously this sequence needs dynamic updating as time passes (aircraft land, new aircraft appear) a controller could well work with this sequence for at least (say) 15 min before needing to dynamically update.

(c) One concern at the start of this work was that the PH would be unable to generate feasible solutions (ie no solutions satisfying time windows and separation would be found). However our computational experience has been that feasible solutions are generated very quickly. We believe that this is due to the fact that the time windows used have sufficient flexibility to ensure feasible solutions are easy to find. In some respects this mirrors the real-life situation: if the time windows were such that feasible landing schedules were hard to find we would regularly have aircraft landing in violation of the separation constraints.

(d) One *key advantage* of adopting an automated approach (such as a PH) to the problem of scheduling aircraft landings may be that *consistency* of performance can be achieved. Simply put, algorithms never get tired and typically produce fairly consistent results. Human controllers, on the other hand, do vary in their performance at a task such as scheduling landings, both over time and amongst themselves.

(e) The PH we have developed can be used as a tool to investigate strategic issues. For example, with regard to landing capacity at London Heathrow through taking example data sets (flight mixes) and seeing how long it takes to land all aircraft. The PH we have developed can also be used to see how changes in the separation criteria would impact upon capacity.

(f) The investigation reported in this paper furthers NATS' research into techniques for improving the utilisation of limited runway capacity. Although we have concentrated here on aircraft arrivals the algorithm developed is applicable to problems involving departures only and to mixed-mode operations.

The PH developed has shown that a decision support tool for scheduling aircraft landings should enable more effective use to be made of the UK's limited runway capacity. NATS is continuing research in this area and additional factors may need to be taken into account in the final design of any decision support tool that is implemented in an operational environment.

We would also comment here that, to the best of our knowledge, the PH presented in this paper places NATS at the forefront of world development in terms of sophisticated algorithms for effective utilisation of runway capacity though improved scheduling.

Conclusions

In this paper we have reported upon an investigation undertaken by NATS in the UK into improving runway utilisation at London Heathrow. The PH developed was described and example results given. We concluded that the PH developed was able to quickly (in a matter of seconds) and effectively schedule aircraft landings using a variety of objective functions.

References

1 Mullins J (1996). Trails of destruction. *New Scientist* **2056**: 28–31.

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

3 Bianco L, Dell'Olmo P and Giordani S (1999). Minimizing total completion time subject to release dates and sequence-dependent processing times. In: Sharaiha YM and Beasley JE (eds). *Advances in Combinatorial Optimization*. *Ann Oper Res* **86**: 393–4l5.

4 Ernst AT, Krishnamoorthy M and Storer RH (1999). Heuristic and exact algorithms for scheduling aircraft landings. *Networks* **34**: 229–241.

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

6 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*. Elsevier Science: Amsterdam, pp 107–150.

7 Holland JH (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificia1 Intelligence*. University of Michigan Press.

8 Bäck T, Fogel DB and Michalewicz Z (eds.) (1997). *Handbook of Evo1utionary Computation*. Oxford University Press: Oxford.

9 Beasley JE (2001). Population heuristics. In: Pardalos PM and Resende MGC (eds). *Handbook of Applied Optimization*. Oxford University Press: Oxford, forthcoming.

10 Mitchell M (1996). *An Introduction to Genetic Algorithms*. MIT Press: Boston, MA.

11 Reeves CR (1993). Genetic algorithms. In: Reeves CR (ed). *Modern Heuristic Techniques for Combinatorial Problems*. Blackwel1 Scientific Publications: Oxford, pp 151–196.

12 Reeves CR (1997). Genetic algorithms for the operations researcher. *INFORMS J Comput* **9**: 231–250.

13 Beasley JE (2000). *A population heuristic for constrained two-dimensional non-guillotine cutting*. Working paper available from the author at The Management School, Imperial College, London SW7 2AZ, UK.

14 Beasley JE and Chu PC (1996). A genetic algorithm for the set covering problem. *Eur J Opl Res* **94**: 392–404.

15 Beasley JE, Meade N and Chang T-J (1999). *Index tracking*. Working paper available from the first author at The Management School, Imperial College, London SW7 2AZ, UK.

16 Chang T-J, Meade N, Beasley JE and Sharaiha YM (2000). Heuristics for cardinality constrained portfolio optimisation. *Comp Oper Res* **27**: 1271–1302.

17 Chu PC and Beasley JE (1997). A genetic algorithm for the generalised assignment problem. *Comp Oper Res* **24**: 17–23.

18 Chu PC and Beasley JE (1998). A genetic algorithm for the multidimensional knapsack problem. *J Heuristics* **4**: 63–86.

19 Chu PC and Beasley JE (1998). Constraint handling in genetic algorithms: the set partitioning problem. *J Heuristics* **4**: 323–357.

Received May 2000;
accepted December 2000 after two revisions