The Sequencing of Aircraft Departures

Problem presented by

John Greenwood

National Air Traffic Services

Problem statement

Aircraft departure sequencing at major airports such a Heathrow is an important example of planning under uncertainty. The departure sequence must satisfy a set of hard constraints, concerned with the sizes and routes of successive departing aircraft, and should also be optimized with respect to a further set of soft constraints. From departure stand to take-off, each aircraft comes under the responsibility of first a Ground Movement Planner, then a Ground Movement Controller and finally a Take-Off Controller. There is most scope for resequencing while aircraft are at the stands, but also most uncertainty in how the decisions taken will affect the final sequence at take-off. In contrast, the Take-Off Controller experiences little uncertainty, but also has limited resequencing options, determined by the geometry of runway holding points. There is currently no complete mathematical formulation. The Study Group was asked to devise models to shed light on the potential gains from resequencing and on the most suitable algorithms to apply at each stage, bearing in mind the differing levels of uncertainty.

Study Group contributors

A. Craig (University of Durham
R. Ketzscher (Cranfield University)
R. A. Leese (Smith Institute)
S. D. Noble (Brunel University)
K. Parrott (University of Greenwich)
J. Preater (Keele University)
R. E. Wilson (University of Bristol)
D. A. Wood (University of Warwick)

Report prepared by

R. A. Leese

1 Overview

(1.1) This report details work carried out at the 40th European Study Group with Industry, held at Keele University between 9th and 12th April 2001. It is concerned with aircraft departure sequencing at Heathrow, with the long-term aim of ascertaining the relative merits of resequencing aircraft at various points between push-back from the stand and take-off. The work was informed by a presentation from John Greenwood of NATS and subsequent discussion on the first day of the Study Group.

(1.2) The central questions are motivated by the following observation. While aircraft are at the stands, there is a great deal of flexibility in their sequencing, but sequencing decisions made at this stage can be 'washed out' by uncertainties in the times taken from stand to runway. In contrast, as aircraft approach take-off, there is only limited scope for resequencing, through the use of branched holding points. The advantage of resequencing at this late stage is that any previous uncertainties have by then been resolved. Some natural questions are therefore as follows. How much flexibility is afforded by branched holding points? What level of variation in taxi times can be restored by good use of the holding points? What is the appropriate level of computational effort to put into sequencing at the stands?

(1.3) The main portion of this report is concerned with making the best use of a branched holding point. A dynamic programming algorithm is constructed to solve this problem, using a cost function incorporating three issues that emerged from discussion with NATS: the impact of wake vortex (WV) classifications and Standard Instrument Departure Sequences (SIDS) on the interval between successive take-offs; a penalty for missing the Calculated Time Of Take-off (CTOT) slots; and an additional penalty for excessive resequencing. A preliminary version of this algorithm was implemented in Visual Basic and is easily capable of handling sequences of 10–15 aircraft, which matches well the typical number of imminent departures that are known with high confidence.

(1.4) Concrete suggestions are also made for how this dynamic programming algorithm can form part of a larger system that would be capable of handling a full day's departures, of which there are between 600 and 700. The proposals are based on the idea of a 'rolling window' of imminent departures, which is maintained in optimal order as new information becomes available. In order to assess appropriate sequencing algorithms for use at the stands, a combination of numerical experimentation and better data for the variation in taxi times would be required.

2 Background

(2.1) The minimum time difference between successive departures is 1 minute, indicating a maximum possible capacity of 60 departures per hour. However, if adjacent aircraft have particular pairs of WV or SIDS classifications then for safety reasons an extra minute must be allowed in the departure sequence. Hence, while 60 departures per hour is an upper bound, there is in practice little prospect of achieving such a level. Heathrow currently achieves approximately 42 departures per hour.

(2.2) Aircraft have four possible WV classifications: heavy, medium, small and light. Medium and small aircraft incur precisely the same pattern of additional gaps in the departure sequence and so may be considered as a single class, called 'medium' in this report. The set of departures for a typical day (4 July 1997) that was made available to the Study Group contained 197 heavy aircraft, 429 medium aircraft and just 11 light aircraft. For clarity of exposition, the light aircraft are not treated explicitly in the remainder of this report. They could be incorporated into the proposed algorithms without much additional difficulty, but one might also take the view that they make up such a small fraction of the total traffic that they could be inserted at the end into an already optimized sequence of heavy and medium departures, without incurring significant additional penalty.

(2.3) There are seven SIDS classifications, identified by abbreviations of the landmarks on the respective departure routes: BPK, CPT, DET, DVR, MID, SAM and WOB. Several pairs have the same pattern of additional gaps in the departure sequence, with the result that there are effectively only four classifications, which will be denoted 1 (comprising BPK and WOB), 2 (comprising CPT and SAM), 3 (comprising DET and DVR) and 4 (comprising MID).

(2.4) The additional gaps that are required as a result of particular pairs of WV or SIDS classifications appearing as neighbours in the departure sequence are summarized by the graph shown in figure 1.

(2.5) The full breakdown of the 626 aircraft (ignoring light aircraft) in the typical day's data that was made available to the Study Group is as follows.

	Heavy	**Medium**
SIDS 1	59	189
SIDS 2	65	62
SIDS 3	51	96
SIDS 4	22	82
Total	**197**	**429**

(2.6) Calculated times of take-off (CTOTs) are allocated by Eurocontrol and correspond to a 15-minute window, running from 5 minutes before the allocated time

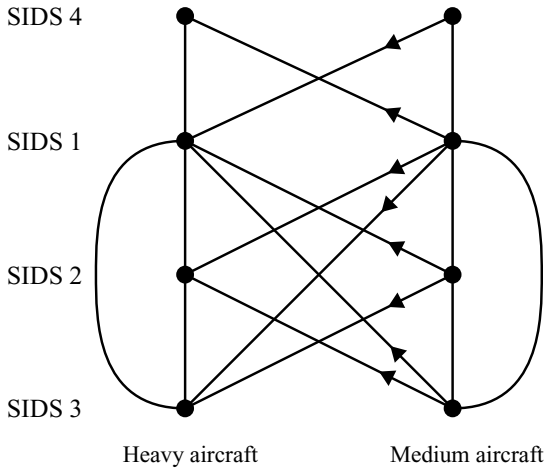

Figure 1: Graphical representation of the additional gaps in the departure sequence resulting from adjacent WV and SIDS classifications. Following an undirected edge in either direction, or a directed edge in the direction of the arrow, gives no additional gap, while any other pair of successive aircraft requires an additional gap of 1 minute.

until 10 minutes afterwards. A significant fraction of aircraft are not allocated a CTOT (or at least if they are then the times are not recorded in the available datasets). The CTOTs are treated as a 'soft' constraint, meaning that there is a penalty attached to missing a slot, but there is no absolute requirement that take-off within a slot must be achieved.

3 Strategy at branched holding points

(3.1) This section considers the problem of feeding a known sequence of aircraft into a branched holding point. We wish to use the limited resequencing mechanism that is possible at the holding point to give a good departure sequence, as measured by an appropriate cost function. Suppose that the original sequence contains n aircraft $a_1, a_2, \ldots, a_n$, with a_1 at the head and a_n at the rear. If after resequencing the result is $a_{\pi_1}, a_{\pi_2}, \ldots, a_{\pi_n}$ then the associated cost is taken to be

$$\phi = \alpha \sum_{i=1}^{n-1} s(a_{\pi_i}, a_{\pi_{i+1}}) + \beta \sum_{i=1}^{n} \max(0, \pi_i - i) + \gamma \sum_{i=1}^{n} g(t_i, \mathrm{CTOT_i}) \,. \qquad (1)$$

Here, $s(a_{\pi_i}, a_{\pi_{i+1}}) \in \{0, 1\}$ is the additional gap, either zero or one minute, that must be inserted in the departure sequence by virtue of having a_{π_i} and $a_{\pi_{i+1}}$ as

adjacent departures; t_i is the departure time of aircraft a_{π_i} taking into account these gaps, $\mathrm{CTOT_i}$ is the corresponding calculated time of take-off and g is a function that describes the penalty for missing a take-off slot. A natural choice is

$$g(t, u) = \max(0, t - u - 10)\,, \tag{2}$$

represented a linearly increasing cost if the end of the CTOT window is passed and zero cost otherwise. It would be straightforward to use other functions, for example ones that also penalized take-offs prior to the start of the CTOT window.

(3.2) The first term in (1) represents the cost in terms of extra delay of imposing the safety constraints related to WV and SIDS classifications. The second term measures the cumulative relegation of aircraft down the sequence and represents the cost of dissatisfaction among the operators of aircraft that are overtaken within the holding point. As already mentioned, the third term represents the cost of missing take-off slots. The three terms are combined in the relative proportions determined by the constants α, β and γ. The precise forms of these three terms were decided following discussions with NATS. In fact the method of dynamic programming gives a great deal of flexibility in choosing the cost function, and it would be straightforward to make modifications, for example to have a more severe (say quadratically increasing) penalty for missing take-off slots.

(3.3) Branched holding points give scope for resequencing through being wide enough to accommodate two planes side-by-side. They can be viewed as a pair of parallel queues, but in the algorithm discussed below it is preferable to think of them as a 'lay-by' in which planes can be temporarily parked while others move past.

(3.4) It is assumed here that the holding point can accommodate a maximum of three aircraft at any one time. In practice, the holding point at Heathrow cannot accommodate three aircraft if they are all heavy, and this restriction could easily be incorporated in our algorithm, although for ease of implementation it is not present in the software demonstrator described below. At any stage, the current state of such a holding point can be described by a sequence of five aircraft $(x_1, x_2, x_3, x_4, x_5)$, interpreted according to the schematic view shown in figure 2.

Aircraft x_1 is the most recent selection for take-off; x_2, x_3 and x_4 are the current occupants of the holding point, with x_2 in the first slot, at the head of the queue, and x_4 at the rear. Finally x_5 is the aircraft that will be the next to arrive at the holding point. We shall often use positive integers to identify the aircraft, so in effect i is shorthand for a_i. Some of the individual elements $(x_1, x_2, x_3, x_4, x_5)$ may be zero, indicating that no aircraft is present in the corresponding location. For example, if there is only one aircraft in

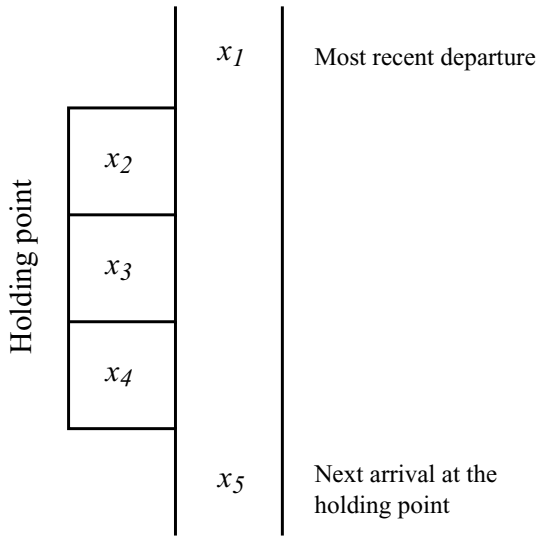

Figure 2: Schematic view of a branched holding point, viewed as a lay-by, with aircraft being temporarily parked on the left hand side. It is assumed that the holding point can accommodate three aircraft, regardless of size.

the holding point then both x_3 and x_4 are zero. The initial state is always $(0, 0, 0, 0, 1)$.

(3.5) A better model would also include the time in the description of each state. With the state description proposed above, the sequence identified by dynamic programming may not be optimal in cases where the optimal sequence involves large accumulated CTOT violations. Including a time in the states would guarantee optimal sequences, but is a refinement that is perhaps not significant in practice.

(3.6) The holding point cannot be used to resequence aircraft arbitrarily. For example, it cannot be used to reverse the order of three aircraft. It is therefore important to know good sequences that *can* be obtained. Dynamic programming provides an attractive method, by effectively constructing the possible chains of states associated with each chain of decisions at the holding point. Each transition from one state to the next corresponds to a single aircraft taking off.

(3.7) From any given state, the set of possible next states depends on the occupancy of the holding point. In general, if the current state has p unoccupied slots in the holding point and if there are q aircraft yet to take off in total, then there are $\min(4, p+2, p+q-2)$ aircraft that are available to be selected as the next one for take-off.

(3.8) The dynamic programming algorithm provides a systematic means of constructing all possible sequences of states and keeping track of the minimum-cost strategy for reaching each of the states. The final states are of the form $(x, 0, 0, 0, 0)$, where x is the final aircraft to depart, and hence the primary interest here is the cheapest way of reaching any state of this form. Including all intermediate states, the total number of states that must be considered for the problem of sequencing n aircraft is $O(n^4)$, increasing to $O(n^5)$ if the state description is extended to include time (see paragraph (3.5) above).

(3.9) It is interesting to compare the number of sequences that may be generated by the holding point to the total number $n!$ of different permutations of n aircraft. Suppose that $S_r(j)$ denotes the number of possible sequences of r remaining aircraft with j of them initially in the holding point. Then we need to find $S_n(0)$. By considering the possible movements in and out of the holding point that are associated with each take-off, we see that

$$\begin{pmatrix} S_r(0) \\ S_r(1) \\ S_r(2) \\ S_r(3) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} S_{r-1}(0) \\ S_{r-1}(1) \\ S_{r-1}(2) \\ S_{r-1}(3) \end{pmatrix},$$

from which it follows that $S_r(0)$ satisfies the recurrence relation

$$S_r(0) = 4S_{r-1}(0) - 3S_{r-2}(0)\,.$$

Observing that $S_1(0) = 1$ and $S_2(0) = 2$, the general solution is

$$S_n(0) = \tfrac{1}{2}(1 + 3^{n-1})\,.$$

The ratio $S_n(0)/n!$ decreases rapidly as n increases. For example, with 10 aircraft the holding point can deliver 9842 different sequences, which is approximately 0.27% of all possibilities. However, although this percentage is small, it is not of itself evidence that the holding point is a poor mechanism for departure sequencing.

(3.10) For the purposes of this report, a preliminary version of the dynamic programming algorithm has been implemented in Visual Basic, with a simple user interface via a spreadsheet in Excel. It has not been optimized for speed, having an execution time of $O(n^7)$, which could be reduced to $O(n^4)$ (or $O(n^5)$ if time is included in the state descriptions) with better searching of the possible intermediate states. If speed were a priority then one would in any case choose a different programming environment. Nevertheless, this software demonstrator can easily sequence 10–15 aircraft in just a few seconds, which corresponds to about 15 minutes of runway operations. Since 15 minutes is a typical taxi time from stand to holding point, this matches well the number of aircraft for which the arrival sequence at the holding point is known with some certainty.

(3.11) The following paragraphs demonstrate the possible sequencing behaviours in an example containing 10 aircraft. They are presented to the holding point in the sequence shown in figure 3, which gives WV and SIDS classifications, in addition to allocated CTOTs. Because the CTOTs are assigned in ascending order to match the initial sequence (and the whole example encompasses little more than a single CTOT slot), the penalties for missing CTOTs, corresponding to the third term in (1), do not come into play. We therefore explicitly set $\gamma = 0$ and concentrate on the interaction between the first two terms in the cost function.

Position	**WV class**	**SIDS class**	**CTOT**
1	Medium	3	0
2	Heavy	3	1
3	Medium	4	2
4	Medium	1	3
5	Medium	4	4
6	Heavy	2	5
7	Heavy	3	6
8	Medium	4	7
9	Heavy	1	8
10	Medium	2	9

Figure 3: The input sequence of 10 aircraft for the example discussed in the main text.

(3.12) Figure 4 shows the output produced by the dynamic programming algorithm for $\alpha = 10$ and three different choices of β, namely $\beta = 1$, $\beta = 3$ and $\beta = 5$. In each case, the departure sequence is in bold type across the top row. The progression of states is given by reading across the columns. The bottom row gives the cumulative cost and so the bottom right hand value in the table is the overall cost of the cheapest strategy found.

(3.13) For $\beta = 5$ the penalty attached to overtaking is sufficiently high to mean that no overtaking takes place and the holding point is not used at all. For $\beta = 3$ the holding point is used just once, to hold back the heavy aircraft 2 until the subsequent run of three medium aircraft has taken off, before it is inserted back into the sequence before aircraft 6, which is another heavy one. It is a very natural move to group heavy aircraft together in this way. For $\beta = 1$, the penalty on overtaking is much less severe and there is extensive use of the holding point. Reading across the progression of states, it is seen that the sequence of events is as follows: aircraft 1 and 2 are placed in the holding point, while 3 and 4 are allowed to depart; then 1 is released and 5 added to the holding point, allowing 6 to depart; then 2 and 5 are released from the

$\alpha = 10$, $\beta = 5$, $\gamma = 0$											
Departure sequence	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Holding point slot 1	0	0	0	0	0	0	0	0	0	0	0
Holding point slot 2	0	0	0	0	0	0	0	0	0	0	0
Holding point slot 3	0	0	0	0	0	0	0	0	0	0	0
Next aircraft on taxiway	1	2	3	4	5	6	7	8	9	10	11
Cumulative cost	0	0	10	20	20	20	30	30	40	40	50

$\alpha = 10$, $\beta = 3$, $\gamma = 0$											
Departure sequence	**0**	**1**	**3**	**4**	**5**	**2**	**6**	**7**	**8**	**9**	**10**
Holding point slot 1	0	0	2	2	2	0	0	0	0	0	0
Holding point slot 2	0	0	0	0	0	0	0	0	0	0	0
Holding point slot 3	0	0	0	0	0	0	0	0	0	0	0
Next aircraft on taxiway	1	2	4	5	6	6	7	8	9	10	11
Cumulative cost	0	0	13	16	19	29	29	29	39	39	49

$\alpha = 10$, $\beta = 1$, $\gamma = 0$											
Departure sequence	**0**	**3**	**4**	**1**	**6**	**2**	**5**	**9**	**7**	**8**	**10**
Holding point slot 1	0	1	1	2	2	5	0	7	8	0	0
Holding point slot 2	0	2	2	0	5	0	0	8	0	0	0
Holding point slot 3	0	0	0	0	0	0	0	0	0	0	0
Next aircraft on taxiway	1	4	5	5	7	7	7	10	10	10	11
Cumulative cost	0	2	4	4	6	6	16	18	18	28	38

Figure 4: Possible departure sequences for the example discussed in the main text, highlighting the effects of choosing different values for β with a fixed value of α.

holding point, at which stage the holding point is empty; it is used just once more, to promote aircraft 9 up the sequence.

(3.14) The three different strategies are summarized in figure 5. The cost of no resequencing is 5α; the cost of just holding back aircraft 2 is $4\alpha + 3\beta$; and the cost of the most elaborate strategy is $3\alpha + 8\beta$. Figure 5 shows how the best strategy is determined by the relative values of α and β. In general, it would be interesting to infer values of α, β and γ by studying the current practice of air traffic controllers.

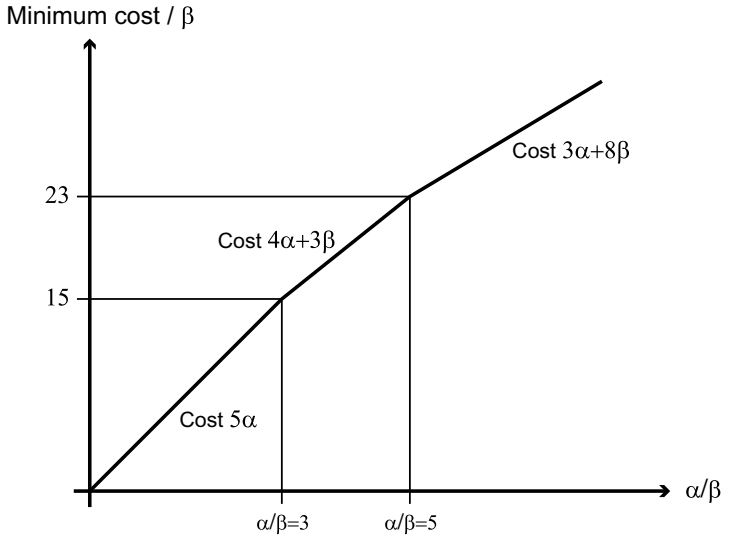

Figure 5: The possible best strategies for the example discussed in the main text, together with their dependence on the relative values of α and β.

4 Resequencing large numbers of aircraft

(4.1) Dynamic programming provides a flexible algorithm for sequencing small numbers of aircraft at the holding point, but there remains the question of handling the few hundred departures that occur in a typical day. It seems that the number of departing aircraft for which a reliable arrival sequence at the holding point is known falls well within the number for which the execution time of the dynamic programming algorithm is a few seconds or less. Hence it is sensible to build the dynamic programming algorithm into a wider scheme.

(4.2) Suppose that the dynamic programming algorithm described above is configured to optimize a block of B aircraft. One possibility for handling more aircraft is simply to process them in disjoint blocks, each containing B aircraft, where B is in the range 10–15 as suggested above. The initial state would not now be $(0, 0, 0, 0, a_1)$, but $(x, 0, 0, 0, a_1)$, where x is the last aircraft to depart in the preceding block. Such a scheme is easy to implement, but its scope for optimizing across the boundary between one block and the next seems rather weak.

(4.3) A better approach is to optimize more frequently, using a 'rolling window'. Each optimization encompasses the aircraft for which information is known with certainty at that time. Following the next departure, as determined by the most recent optimization, the remaining aircraft are reoptimized, incorporating any new ones for which information has become available in the meantime. The

dynamic programming algorithm is certainly fast enough to cope with the extra burden of calculation, and this method is expected to give significantly better performance than a calculation based on disjoint blocks.

(4.4) A potential pitfall of the rolling window implementation is that individual aircraft may be relegated many places down the sequence. In contrast, disjoint blocks will never demote an aircraft further than the end of its own block. There are various possibilities for overcoming such a situation. First, one could interpret the permutation π appearing in (1) as a permutation of all aircraft and not just those in a single run of the algorithm. The second term in the cost function would then tend to protect aircraft against very large relegations. Secondly, there could be imposed an explicit maximum position shift for an individual aircraft. Thirdly, one could base decisions on a larger set of aircraft than those for which certain information is currently known, by adding aircraft that are drawn at random from suitable probability distributions to the end of the input sequence for each dynamic programming run. The idea here is to calculate the expected minimum cost that can be achieved with an uncertain input sequence.

5 Sequencing at the stands

(5.1) There is more flexibility for resequencing at the stands than at the holding point, but one would be making decisions based on uncertain information. The uncertainties arise in the form of variability in the taxi times from pushback to arrival at the holding point, which means that the arrival sequence at the holding point is some perturbation of the departure sequence from the stands. The effects of good decisions implemented at the stands may therefore be 'washed out' by the time aircraft reach the holding point. The data available on the nature of these perturbations is very sketchy, but they do seem to be an important consideration in any overall strategy.

(5.2) There is likely to be more certainty about the projected movements of aircraft departing from stands near the holding points than about those departing from stands further away, owing to the airport geography. Note that aircraft leaving the most distant stands taxi past the nearer stands on their way to the runway. Taking account of the differing uncertainties in taxi times from different stands is a key question in the design of good schedules for this type of problem.

(5.3) In the initial stages of any further investigation, we propose applying simple heuristics for departure sequencing at the stands and coupling them to suitable models for the variability in taxi times. One possible heuristic is to globally sequence aircraft leaving the stands in ascending order of CTOT. A refinement would be to sort first on WV classification within blocks of 10–15 aircraft, and then by CTOT; in this case, it is presumably good to put the heavy aircraft

at the top of all odd-numbered blocks (say) and at the bottom of all even-numbered ones, so that they are clustered more strongly. As a benchmark, any strategy should be compared with a random departure sequence within blocks.

(5.4) To design models for the (random) perturbations that turn departure sequences from the stands into arrival sequences at the holding point, it would be useful to look at the root causes of variability in taxi times. For example, if there are clearly identifiable events that can cause delay, such as waiting for movement at nearby stands, or completing on-board safety checks, then one could have a model that effectively flips a suitably biased coin to decide whether an individual aircraft is delayed, and if so then draws the delay from a suitable probability distribution.

6 Conclusions

(6.1) We have proposed and demonstrated a dynamic programming algorithm for designing strategies at the holding point based on a flexible cost function.

(6.2) We have considered ways in which that algorithm could be extended to cope seamlessly with an arbitrarily long sequence of departures.

(6.3) We have considered possible sequencing strategies at the stands and how they would connect to the holding point strategy.

(6.4) The key questions that remain for further investigation are twofold. First, what quality of sequencing is the holding point capable of delivering? And, in light of the answer to the first question, what level of sophistication is appropriate at the stands? The answers are likely to come from theoretical analysis of possible holding point strategies, coupled with experiments in simulation. Specific suggestions include the following:

- Mimic sophisticated sequencing at the stands by taking good (or even optimal) departure sequences, perturb them in various ways and then attempt to recover them through use of the holding point.
- Experiment with different holding point algorithms on the same input sequence, with particular attention on their extension to very long sequences and on the impact of choosing different values of α, β and γ in the cost function.
- Experiment with different sequencing heuristics at the stands, without any subsequent perturbation and keeping the holding point strategy fixed, to see whether the heuristics implement beneficial moves for which the holding point cannot substitute.