SCHEDULING A HETEROGENEOUS SET OF TRAINS OVER A SINGLE LINE TRACK USING LAGRANGIAN RELAXATION

SCOTT MACKENZIE[1] and GRAHAM MILLS[✉2]

Abstract

There is increasing interest in the cost of railway infrastructure access as a variety of private companies operate trains with different lengths over a common rail network. It is important to have a method for evaluating the cost of adding an additional train to a railway timetable. This is particularly so for single line track with occasional passing loops. The concept of a fixed loop capacity is extended to one that is dependent on the trains. We develop a model for scheduling a heterogeneous set of trains on single line systems with loops. Our method minimizes the total weighted delay. A Lagrangian relaxation technique is used that relaxes the capacity constraints for track segments and super segments. We measure the delay for each train and the total weighted delay for the heterogeneous set of trains. Our model allows us to investigate the robustness of the weighted delay to variation in the departure time of individual trains. The paper demonstrates that a Lagrangian relaxation heuristic provides optimal train schedules for instances of small heterogeneous train sets. The method is used primarily to check the effectiveness of heuristic algorithms commonly used to find schedules for practical problems.

1. Introduction

This work follows on from the 2003 Mathematics in Industry Study Group (MISG) investigation of deadlock avoidance in computer-generated scheduling algorithms for complex rail networks [9]. That investigation showed that a dispatching method using three types of *safe* move avoided deadlock in rail systems where every train has access to every loop. This paper considers a heterogeneous train set of various train lengths

[1]School of Mathematics, University of South Australia, Mawson Lakes Campus, 5095, Australia;
e-mail: scott.mackenzie@unisa.edu.au.
[2]CSIRO Mathematical and Information Sciences, Private Bag 2, Glen Osmond, SA 5064, Australia;
e-mail: graham.mills@bigpond.com.

where some trains cannot use all of the loops and develops a scheduling method that avoids deadlock.

A railway timetable specifies the arrival and departure times of a set of trains at key stations on a network while a train schedule specifies arrival and departure times of an individual train at each station and loop in the network. Scheduling a homogeneous set of trains over a double track network is relatively simply as there is no interference between opposing trains and following trains are not delayed if the inter-departure time is greater than the maximum sectional running time [5]. Scheduling a homogeneous set of trains over a single line network with crossing loops is more difficult as opposing trains can meet only at loops that accommodate each of the trains. Several methods [5, 12] have been developed that schedule a homogeneous set of trains on single line track.

Generally, in practice, both the train set and the network are heterogeneous in that trains vary in speed and length while crossing loops can accommodate some trains but not others. Some methods [3, 6–8] are available for optimal scheduling of trains with different speeds. These methods assume that all trains can use any loop in the network. We are not aware of any work that allows long trains to use only certain selected loops in the system.

Brannlund *et al.* [2] use a Lagrangian relaxation approach to solve a formulation of the railway timetabling problem in which all trains can use all loops. The method avoids deadlock. In the train scheduling problem the idea is to use Lagrangian multipliers to relax the capacity constraints on the track network.

This paper extends the work of Brannlund to a heterogeneous set of trains where some trains cannot use all loops. We generalize the concept of a track segment by dividing the track into constraint zones for each pair of opposing trains. Opposing trains cannot meet within such constraint zones of the network. The problem is then decomposed into a series of shortest path problems for each train with costs at each location determined by Lagrange multipliers. These multipliers are then updated using a sub-gradient type method. The shortest path for each train is a trade-off between the network costs obtained from the Lagrange multipliers and a delay cost incurred by avoiding the network costs.

We measure the delay for each train and the total weighted delay for the heterogeneous set of trains. Our method provides optimal solutions for small problem instances. The heuristic avoids deadlock more effectively than other published heuristics. Our results show that longer trains that cannot use all loops in the system incur significant delays when other long trains are present. The selection of an effective departure time for long trains is critical for reducing the total weighted delay. The Lagrange multipliers associated with track capacity provide a measure for the value of each loop associated with the current schedule.

The remainder of this paper is organized as follows. In Section 2 we review the relevant literature on the scheduling of trains on a single track. In Section 3 we formulate our model by introducing our notation and generalizing the concept of loop capacity to constraint zones. We also describe our objective function and the

constraints for the operation of long trains over a single track. In Section 4 we present our heuristic for solving the problem by relaxing the capacity constrains on track segments and within constraint zones. Our method of updating the multipliers and solving shortest path problems is also described and explained. In Section 5 we discuss our computational experience and the sensitivity of the solution as the departure times of the trains are varied. Finally some concluding remarks and suggestions for extensions of the work are given in Section 6.

2. Literature review

Frank [5] considered homogeneous two-way traffic systems on a railway with single line segments. He allowed either one down train or at most two down trains to wait at a loop while up trains continue on the main line. Frank derived expressions for the traffic capacity and cycle times under the assumption of nonaccumulation of trains at the end nodes (stations). For the case of one train waiting, the capacity of the two-way system is about three times the capacity of a one-way system.

Szpigel [12] developed an integer programming formulation for scheduling trains with given departure times on a single track line that minimizes the weighted delay. He used a branch and bound technique to solve small instances of homogeneous train sets on a small network and he experienced difficulties with problems of reasonable size.

Kraay *et al.* [7] developed a mathematical programming formulation for the pacing of trains over a network with single line segments. The objective was to optimize where and when trains should pass so as to satisfy a balance of time and fuel objectives. Two heuristic solution methods were applied to several illustrative examples and suggestions were made to improve the efficiency of the algorithms.

Mills *et al.* [8] formulated a discrete network flow model by discretizing the departure and arrival time variables for each train at each loop. They use a shortest path algorithm to update the path of each train until a feasible schedule is obtained.

Higgins *et al.* [6] developed a mathematical programming model similar to Kraay's. They used a branch and bound technique, based on the remaining overtake and crossing delay, to solve realistic size instances of the problem. They used the model either for an online scheduling tool or for timetabling and infrastructure planning.

Cai *et al.* [3] extended a greedy heuristic for rapid scheduling of trains on a single track to include most of the practical constraints while retaining the ability to compute a good feasible solution quickly. The most critical constraint in the problem is to avoid any potential collisions among trains. Such a constraint becomes very hard to deal with in a single track railway, where trains are allowed to cross only at some selected passing loops. The algorithm allows on-line use by allowing a train to start from any location and time instant. The algorithm can schedule the physical backup of trains to avoid blockages and ensure a feasible solution. The trains may have different speeds over a segment of track but the capacity of the loops is fixed and independent of the train.

Brannlund *et al.* [2] used a Lagrangian relaxation approach to solve an integer programming formulation of the railway timetabling problem. The track is divided

into blocks that can contain only one train at any time. They discretize time and define a binary variable x_{it}^r as unity if train r occupies block i at time t and zero otherwise. The problem is separated into scheduling single trains by relaxing track capacity constraints and assigning prices (multipliers). The algorithm iterates between finding feasible solutions of the primal and updating the multipliers until the duality gap is reduced within a given tolerance. Their approach is illustrated on a realistic example of a daily schedule for 26 trains over a single line track with 17 stations.

Nilsson [10] considered the management of railway infrastructure access. The use of scarce railway track capacity and the allocation of capacity between competing operators are critical issues for infrastructure management. Nilsson developed an auctioning procedure to assess the operator's willingness to pay for track access. He formulated an optimization problem to allocate capacity using the information from the auction.

Pudney and Wardrop [11] used a problem space search method for generating good train schedules over a single line track. They used an efficient train dispatch heuristic to generate a single train plan. By randomly perturbing the data used to make the dispatch decisions, they generated many train plans. They select the best plan as measured by lateness or cost and discard any plans resulting in deadlock.

Dorfman and Medanic [4] used a discrete event model of train advances along a line to schedule trains on a line or rail network. They claim that their greedy heuristic is computationally efficient. They provide a capacity check algorithm to prevent deadlock of advancing trains assuming a fixed capacity for the loops. Their greedy heuristic does not guarantee optimality but their empirical evidence shows a near optimal performance over several relevant measures.

3. Problem formulation

In this section we define supersegments for each train corresponding to a sequence of segments over which a train has no choice of path. Also we define constraint zones corresponding to a nonsimple sequence of segments that cannot simultaneously be occupied by a pair of opposing trains. This generalizes the concept of segments that can be occupied by only one train at a time to zones that can only be occupied by one of a pair of opposing trains.

The rail network is divided into $|I|$ track segments (blocks) $I = \{1, 2, \ldots, |I|\}$ indexed by i being defined as lengths of track which can only accommodate one train at any moment in time. Consider a set of $|R|$ trains $R = \{1, 2, \ldots, |R|\}$ indexed by r to be scheduled over the railway network. This set is divided into a set of up trains R^u and a set of down trains R^d. We assume each train has a fixed running time over each track segment and d^r is the earliest arrival time of train r at its final destination.

Some trains in the network may not be able to use all track segments in the network. Some loops may be reserved for trains travelling in a particular direction and long trains may not fit onto certain loops. These trains that travel on a restricted network will see supersegments. A supersegment is defined as follows.
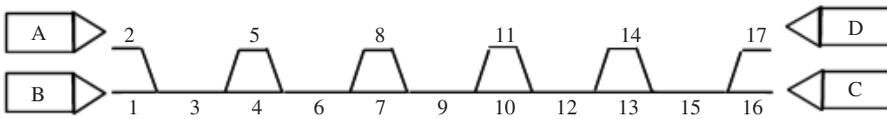
FIGURE 1. Track segments with four trains to be moved across a network consisting of 17 segments.

DEFINITION 1 (Supersegment). For a particular train, if a track segment on its path has only one successor and this track segment is the only predecessor of the successor segment, then the track segment and the successor segment form part of a *supersegment* for this train. A supersegment is the largest contiguous set of such track segments.

To illustrate the above idea consider the example set out in Figure 1. In this example four trains are to be moved across a network consisting of 17 segments. Train A cannot use loops 5 or 14, train B cannot use loops 8 or 11, train C is unable to use loops 5, 11 and 14 and train D cannot use loops 11 or 14. Let S_i^r be a supersegment for train r and the set S^r contain all sets of supersegments that train r sees. The sets of supersegments seen by trains A, B, C and D is given by

$$S^A = \{\{3, 4, 6\}, \{12, 13, 15\}\}, \quad S^B = \{\{6, 7, 9, 10, 12\}\},$$
$$S^C = \{\{3, 4, 6\}, \{9, 10, 12, 13, 15\}\}, \quad S^D = \{\{9, 10, 12, 13, 15\}\},$$

respectively.

We now form a set of constraint zones for each pair of opposing trains. Let Z^{ru} denote the constraint zones for the pair of trains r and u. These constraint zones are the nonsimple sets of elementary segments that are formed by taking the intersections of the supersegments for any particular train r with the supersegments for an opposing train u. The resulting set of constraint zones for train r and opposing train u is given by

$$Z^{ru} = \left\{ i \cap i' \mid i \in S^r, i' \in S^u \right\}.$$

For our example we obtain the constraint zones $Z^{AC} = \{\{3, 4, 6\}, \{12, 13, 15\}\}$ $= Z^{CA}$, $Z^{AD} = \{\{12, 13, 15\}\}$, similarly for B, $Z^{BC} = \{\{9, 10, 12\}\}$, $Z^{BD} = \{\{9, 10, 12\}\}$. The total set of constraint zones seen by train r is given by taking the union of the train r constraint zones with all other opposing train constraint zones giving us $Z^r = \bigcup_u Z^{ru}$. For our example problem we find

$$Z^A = \{\{3, 4, 6\}, \{12, 13, 15\}\}, \quad Z^B = \{\{9, 10, 12\}\},$$
$$Z^C = \{\{3, 4, 6\}, \{9, 10, 12\}, \{12, 13, 15\}\}, \quad Z^D = \{\{9, 10, 12\}, \{12, 13, 15\}\}.$$

Thus, for this small example three distinct constraint zones are formed. We index these constraint zones by j and denote them by Z_j with

$$Z_1 = \{\{3, 4, 6\}\}, \quad Z_2 = \{\{9, 10, 12\}\}, \quad Z_3 = \{\{12, 13, 15\}\}.$$

We now form a set of constraints that keep each pair of trains from occupying a segment at any time. Define the variable x_{it}^r to be one if train r is on track segment i at time t and zero otherwise. Let X^r denote the set of vectors x^r, which represent feasible paths for train r through the network not considering the effects of other trains that may be present. That is, the constraints governing both precedence and traversal times for train r are satisfied by vectors in X^r. The track capacity constraints below require that only one train may be on an individual track segment at any one time,

$$\sum_{r \in R} x_{it}^r \leq 1 \quad \text{for all } i, t.$$

We now form a set of constraints that keep each pair of opposing trains from occupying a constraint zone at any time. Define an indicator function I_j^{ru} as one if trains r and u, travelling in opposite directions, compete for constraint zone Z_j and zero otherwise. This indicator function allows us to set up a constraint that the pair of trains r and u must not occupy this constraint zone at the same time. This constraint zone occupancy constraint on the problem can be stated as

$$\sum_{i \in Z_j} (x_{it}^r + x_{it}^u) \leq 1 \quad \text{for all } t, \quad I_j^{ru} = 1. \tag{3.1}$$

Thus, for trains A and C in the above example on constraint zone Z_1 we require

$$x_{3t}^A + x_{4t}^A + x_{6t}^A + x_{3t}^C + x_{4t}^C + x_{6t}^C \leq 1 \quad \text{for all } t.$$

Similarly for the other pairs of trains in the constraint zones we have the constraints

$$x_{12t}^A + x_{13t}^A + x_{15t}^A + x_{12t}^C + x_{13t}^C + x_{15t}^C \leq 1 \quad \text{for all } t,$$
$$x_{12t}^A + x_{13t}^A + x_{15t}^A + x_{12t}^D + x_{13t}^D + x_{15t}^D \leq 1 \quad \text{for all } t,$$
$$x_{9t}^B + x_{10t}^B + x_{12t}^B + x_{9t}^C + x_{10t}^C + x_{12t}^C \leq 1 \quad \text{for all } t,$$
$$x_{9t}^B + x_{10t}^B + x_{12t}^B + x_{9t}^D + x_{10t}^D + x_{12t}^D \leq 1 \quad \text{for all } t.$$

The objective function to be minimized is the sum of the weighted tardiness of each train, that is,

$$\min J = \sum_{r \in R} \omega^r \max(0, a^r - d^r)$$

where ω^r is a weight representing the priority of train r, a^r is a variable denoting the actual arrival time of train r at the final destination, and d^r is the earliest arrival time at the final destination. In summary, the problem can be stated as

$$\min J = \sum_{r \in R} \omega^r \max(0, a^r - d^r)$$
$$\text{subject to } \sum_{r \in R} x_{it}^r \leq 1 \quad \text{for all } i, t \tag{3.2}$$

(P)

$$\sum_{i \in Z_j} (x_{it}^r + x_{it}^u) \leq 1 \quad \text{for all } j, t, \quad I_j^{ru} = 1 \tag{3.3}$$

$$x^r \in X^r \quad \text{for all } r \in R. \tag{3.4}$$

Constraint (3.4) says that a path for each individual train through the network must be a feasible path for that train not considering its interactions with other trains. That is, track segment traversal order and traversal times across segments are logically feasible. The two constraints (3.2) and (3.3) are both capacity constraints on the system. Constraint (3.2) states that two trains cannot be on the same segment at the same time. Constraint (3.3) is a further capacity constraint that tightens up the solution space. It states that a pair of trains travelling in opposite directions that see a particular constraint zone cannot be on that constraint zone at the same instance of time.

4. Lagrangian relaxation

4.1. Lagrangian decomposition In order to find good solutions to a difficult combinatorial problem it is useful to calculate both upper and lower bounds as close as possible to each other. Lagrangian relaxation [1] is one such technique which is commonly used to find lower bounds. Lagrangian relaxation exploits the fact that many combinatorial problems consist of an easily solvable problem complicated by a series of difficult constraints. By relaxing these difficult constraints and placing them into the objective function we are left with an easier problem to solve and attention can then be paid to finding the appropriate numerical values for the multipliers.

One way of dealing with the pair of difficult capacity constraints on our system is by Lagrangian relaxation. These capacity constraints are relaxed, assigned prices (Lagrange multiplier) and added to the objective function. The relaxed problem is now

$$(P_\lambda) \quad \begin{aligned} \min J &= \sum_{i \in T} (\omega^r \max(0, a^r - d^r)) + \sum_{i,t} \lambda_{it} \left(\sum_{r \in R} x_{it}^r - 1 \right) \\ &+ \sum_{j,t} \sum_{r \in R^u, u \in R^d} I_j^{ru} \mu_{jt}^{ru} \left(\sum_{i \in Z_j} (x_{it}^r + x_{it}^u) - 1 \right) \\ &x^r \in X^r \quad \text{for all } r \in R. \end{aligned} \tag{4.1}$$

This can be interpreted that all trains are charged prices λ_{it} for access to segment i at time t. Likewise the pair of trains r and u are charged μ_{jt}^{ru} for access to constraint zone Z_j at time t. This capacity-relaxed problem now separates into independent subproblems for each train r

$$(P_\lambda^r) \quad \begin{aligned} \min J^r &= \omega^r \max(0, a^r - d^r) + \sum_{it} \lambda_{it} x_{it}^r + \sum_{jt} \sum_{u \in R} I_j^{ru} \mu_{jt}^{ru} \sum_{i \in Z_j} x_{it}^r \\ &x^r \in X^r \quad \text{for all } r \in R. \end{aligned}$$

Let $\hat{x}(\lambda, \mu)$ be the optimal solution to (4.1) and denote its value by

$$\phi(\lambda, \mu) = \sum_{r \in R}(\omega^r \max(0, a^r - d^r)) + \sum_{i,t} \lambda_{it}\left(\sum_{r \in R} \hat{x}_{it}^r - 1\right)$$
$$+ \sum_{j,t} \sum_{r,u} I_j^{ru} \mu_{jt}^{ru}\left(\sum_{i \in Z_j}(\hat{x}_{it}^r + \hat{x}_{it}^u) - 1\right).$$

The last sum is taken over all pairs of trains r and u that occupy supersegment j at time t. This generally infeasible dual problem $\phi(\lambda, \mu)$ for $\lambda, \mu \geq 0$ provides a lower bound on the optimal value of (P). The problem of finding the best possible such bound is the Lagrangian dual problem,

$$(D) \qquad\qquad \begin{array}{c} \max \quad \phi(\lambda, \mu) \\ \text{subject to} \quad \lambda, \mu \geq 0. \end{array}$$

4.2. Subgradient optimization Subgradient optimization [1] is an iterative procedure which attempts to maximize the lower bound from the Lagrangian dual problem by a suitable choice of multipliers.

The basic procedure for this approach is summarized below and is followed by a more detailed discussion of some of the steps.

(1) Set $k = 1$. Initialize all multipliers by setting $\lambda_{it}^1 = 0$, and $\mu_{it}^1 = 0$. Set a limit $k_{\max}$ on the number of iterations.

(2) Solve the relaxed Lagrangian problem (P_λ^k) with the current set of multipliers λ^k and μ^k to obtain a solution $\hat{x}(\lambda^k, \mu^k)$ of value J_{LB}. This J_{LB} is a lower bound on the cost of the original problem.

(3) Find a feasible solution to the original problem (P^k) using the primal heuristic with cost J_{UB}^k. This is an upper bound on the optimal objective value.

(4) Define subgradients G_{it} and H_{jt}^{ru} for the relaxed constraints evaluated at the current dual solution by

$$G_{it} = \sum_{r \in R} x_{it}^r - 1, \qquad H_{jt}^{ru} = \sum_{i \in Z_j}\left(x_{it}^r + x_{it}^u\right) - 1.$$

(5) Define a scalar step size V by

$$V = \frac{\pi(1.05 J_{UB} - J_{LB})}{\|\sum_{it} G_{it}\|^2 + \|\sum_{ru}\sum_{jt} H_{jt}^{ru}\|^2}$$

where J_{UB} is the least upper bound found thus far and π is a user-defined parameter.

(6) Update λ_{it}^k and μ_{jt}^k using

$$\lambda_{it}^{k+1} = \max(0, \lambda_{it}^k + V G_{it}), \qquad (\mu_{jt}^{ru})^{k+1} = \max(0, (\mu_{jt}^{ru})^k + V H_{jt}).$$

Stop if $J_{UB} - J_{LB}$ is small enough or $k = k_{\max}$. Otherwise set $k = k + 1$ and return to step 2.

The updating of the subgradients λ_{it} follows the following rules:

- increase λ_{it} for the next iteration of the dual solution if more than one train occupies segment i at time t;
- decrease λ_{it} if no train is on segment i at time t;
- leave λ_{it} unchanged if exactly one train occupies segment i at time t.

Likewise the updating of μ_{jt} follows similar rules depending on the occupancy of the constraint zone j.

We have modified the multiplier updating to avoid extremely small step sizes from the G_{it} and H^{ru}_{jt} factors in the equation for V. We base our step size on the number of constraint violations. Our dual problem is sparse in that there are very few nonzero variables and most of the components of G_{it} and H^{ru}_{jt} are negative. To increase the step size we have adjusted the subgradients before calculating V using

$$G_{it} = 0, \quad \text{if } \lambda_{it} = 0 \text{ and } G_{it} < 0,$$
$$H^{ru}_{jt} = 0, \quad \text{if } \mu^{ru}_{jt} = 0 \text{ and } H^{ru}_{jt} < 0.$$

The value of π was set to make the initial step size V close to unity. The factor 1.05 multiplying J_{UB} in the equation for V was set so as the dual cost approached the primal cost the step sizes did not become too small [1].

4.3. Minimum cost path problem For each train to be scheduled we must solve a minimum cost path through a network of Lagrange multipliers. In our network there is a node for each segment of the rail network at every time interval. The value of the node is the minimum cost of a train journey from its origin to that node. This network is acyclic and all costs are nonnegative. In such a network finding the shortest path is straightforward. If we evaluate the nodes in time order we know that the predecessors will be labelled with the correct node values. Thus, we know *a priori* the topological ordering of the network.

In this network there are two types of arcs. The first corresponds to a train traversing a segment of track using its known traversal time on that segment to arrive at a successor segment. For this type of arc the cost of traversing a segment i during time periods t_1 to t_2 is given by

$$C^r_i = \sum_{t=t_1}^{t_2} \left(\lambda_{it} + \sum_{j \in Z_j} \sum_u \mu^{ru}_{jt} I^{ru}_j \right)$$

where the indicator function I^{ru}_j is defined by

$$I^{ru}_j = \begin{cases} 1 & \text{if segment } j \in S^r \text{ for train } r, \\ 0 & \text{otherwise.} \end{cases}$$

We say that a train incurs a network cost when it selects this type of arc. The cost of this type of arc results from the dual prices of the segment and constraint zone segment if the segment forms part of the train's constraint zones.

The second type of arc corresponds to a train experiencing a delay of one time unit on a segment to remain on the segment. We say that the train incurs a delay cost when it selects this type of arc. Subsequently the train may be further delayed on the segment or may select a type one arc to a successor segment.

We search through the network of prices to find the lowest cost path for a train from any of its starting segments to any of its destination segments. We stop this searching process once the tardiness cost of a path exceeds the cost of the cheapest path found thus far.

4.4. Primal feasible solutions   By adjusting prices alone we cannot guarantee that a feasible solution will be produced. In order to generate feasible solutions for our train scheduling problem, we use a similar method to solving the relaxed problem. Trains are scheduled one at a time through the network of prices according to a priority list. The first train to be scheduled takes its cheapest path through the network. The nodes on this cheapest path are then removed from the network so that no other train is able to access them. This guarantees that all subsequent train journeys will be feasible with respect to this train. The second train in the list to schedule then searches for its cheapest path through this reduced network. This process is then repeated until all trains have been scheduled. It is a feasible solution since it schedules each train around all previously scheduled trains on the discrete network. Thus, deadlock is avoided.

The priority list is constructed by ordering the trains based on their weights (ω^r) with the higher weight trains being scheduled first. Trains of the same weight are then sorted according to the amount of delay they experienced in the previous dual solution. Trains with less delay being scheduled first.

5. Computational experience

A test case was considered to evaluate the approach. The test case considered involves the scheduling of just four trains, two outbound and two inbound, across the network shown in Figure 1 with 17 segments and 5 crossing loops. Trains A and C in each direction are long trains that do not have access to loop 11 on the network. Trains are released onto this network at times selected so that each train can interact with all other trains. Trains B and D are shorter trains with a higher priority than the long trains by way of their weights of two compared with one for the long trains. The objective function to be minimized is the sum of the weighted tardiness of each train. The due time of each train has been set to be the free running time of train so, in effect, we are minimizing the sum of the weighted delay of each train. Even such a small problem has hundreds of possible feasible solutions.

5.1. Algorithm performance   We monitored the upper bound and the lower bound as the algorithm progressed. Figure 2 shows how the primal and dual costs behave. There is a steady increase in the dual cost while the primal cost oscillates between several solutions. This case gives an optimal solution as the total weighted delay is an integer and the value of the primal and dual solutions differ by less than an integer.
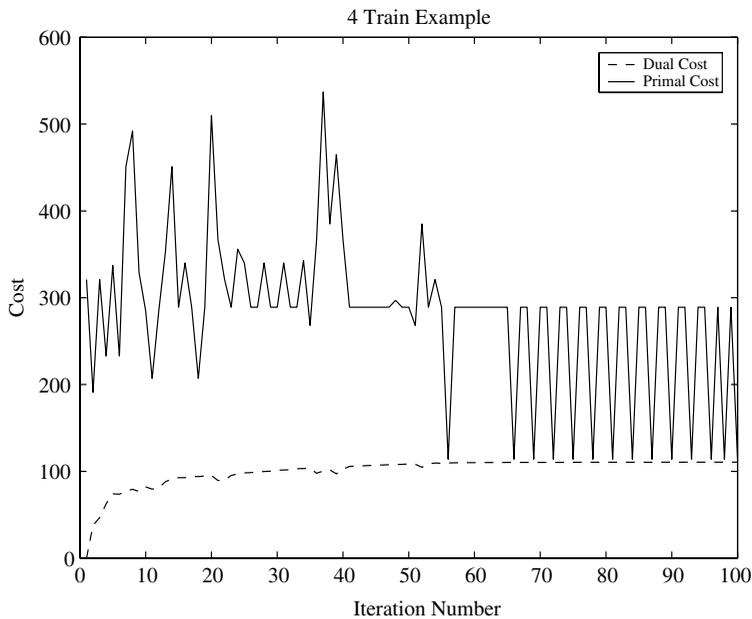
FIGURE 2. Four-train, primal and dual costs versus the iteration number.

Figures 3 and 4 show the best primal solution and the best dual solution. The best primal is feasible. The cost of 99 represents zero delay to train A, 25 units delay to train B with a weight of 2, 41 units delay to train C and 4 units delay to train D with a weight of 2. Figure 3 shows that the long train C waits at loops 13 and 14 for a significant time until the other long train A crosses. The best dual solution gives a similar solution but the crossing of the pair of trains A and D and the pair of trains C and B are not fully resolved at loops 10 and 11.

5.2. Solution robustness   In order to gain some understanding into the robustness of the schedule variations in the release time of each train were made. One train at a time had its release time into the network varied by up to 20 time intervals early and late while all other trains release times remained unchanged. The due time of the perturbed train was also adjusted by the same amount as its release time. The effect of these first-order variations on each trains lateness can be seen below.

We investigated the cost–benefit of a variation if the departure time of individual trains. Figure 5 gives the effect on the total weighted cost from the change in the release time of each individual train. An early release of train A decreases the cost as train C waits less time at loops 13 and 14. Likewise a late release of train A initially increases the cost because train C waits longer at loops 13 and 14. On later release of train A we find that train B departs before train A and the cost is reduced because train B has a higher weight than train A.
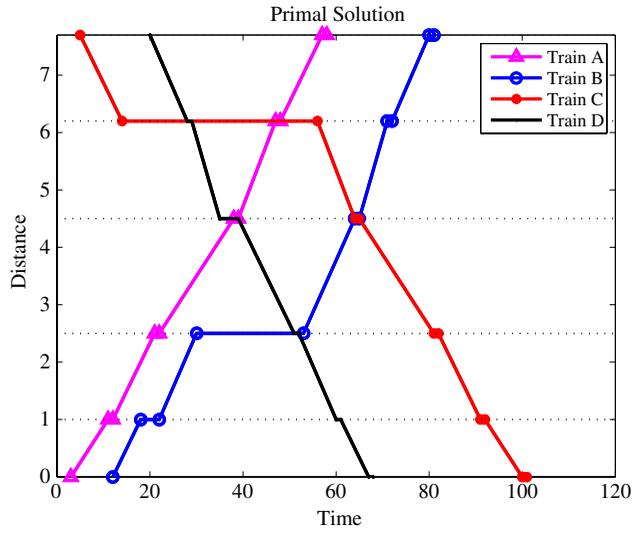
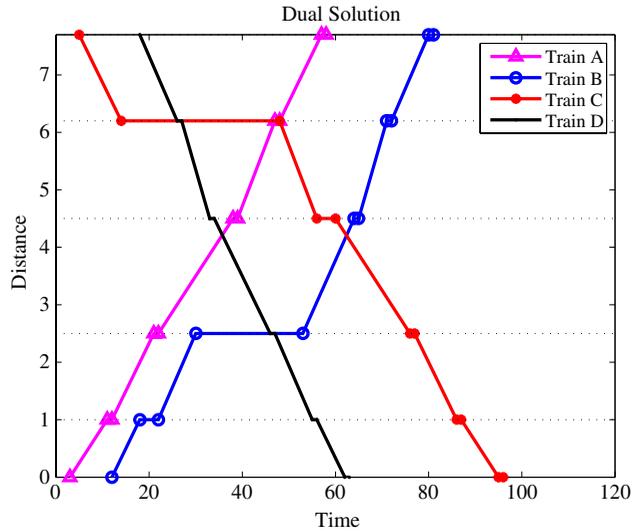FIGURE 3. Four-train best primal solution with cost 99.00.

FIGURE 4. Four-train best dual solution with cost 98.920.

Figure 6 shows how a variation in the release time of one train affects the lateness of the other trains in the network for a re-optimized schedule. We examined the graphs for *small* variation in the departure time of each train.
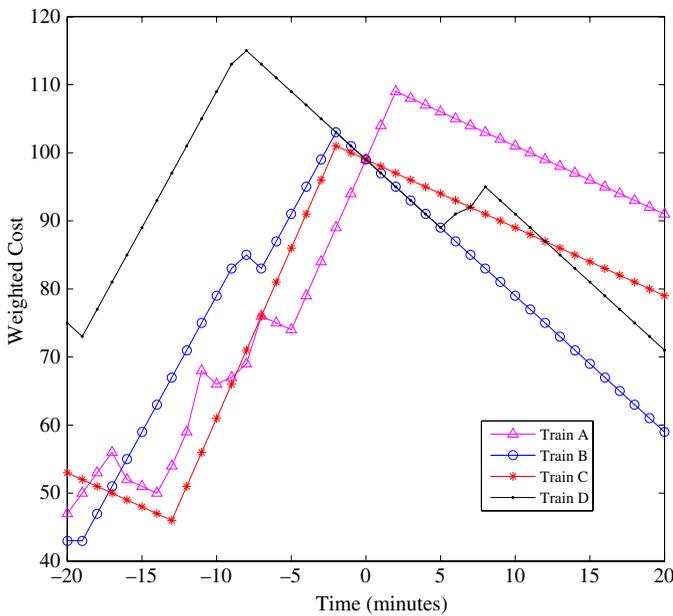
FIGURE 5. Total weighted cost versus the release time perturbation of each train.

- If train A is released earlier, then all other trains arrive earlier because the delay at loops 13 and 14 is reduced. If train A is released later, then train C waits longer at loops 13 and 14 but train B departs ahead of train A and arrives earlier.
- If train B is released earlier, then both trains A and C experience increased delays. However the total cost is lower because trains A and C have low weights of one.
- If train C is released earlier, then it crosses train A at loop 11 rather than loops 13 and 14, so the lateness of train C decreases while that of train A increases. Also trains B and D now cross at loops 10 and 11 so train B arrives earlier. If train C is released later, then it has no effect on the other trains.
- If train D is released earlier, it waits at loops 13 and 14 and affects only itself.

5.3. Infrastructure value   Our method allows an investigation of the value of each loop in the network. Figure 7 shows the sum over the discrete time intervals of the shadow prices (multipliers) at each location on the network as the departure time of train C varies by up to 20 minutes. We constructed the example so that trains conflict on segment 9 and supersegment 18. We expect higher prices on the single track segments 3, 6, 9, 12, 15 and the supersegment 18 rather than on the loops. This is reflected in the high prices associated with these segments.

The prices on segments can be used to infer the benefit of additional infrastructure investment. The higher prices associated with loop 9 and the supersegment 18 indicate there is a high demand for these loops over the range of departure times of train C.
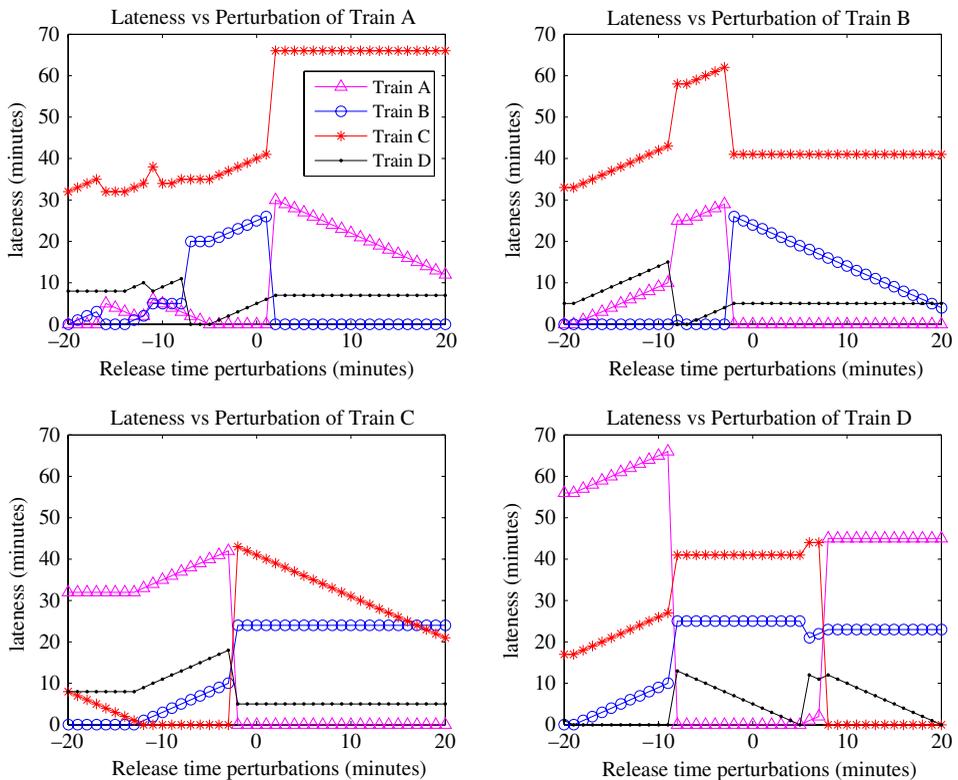
FIGURE 6. Variation in lateness versus release time of each train.

If the departure time of train C is delayed the shadow price for the supersegment 18 is reduced as trains A and C meet outside the supersegment. A consistently high value of a shadow price for a segment over a range of timetables gives an indication that an additional loop within this segment would be an advantage. Similarly a consistently high value of a shadow price for a supersegment indicates that the loops within the supersegment should be modified to allow opposing trains to cross.

6. Conclusions and extensions

We have been able to generalize the capacity at a loop from a fixed value to be train dependent. We introduced the concept of a supersegment for trains that are constrained to follow a series of segments. We defined constraint zones for pairs of opposing trains to prevent double occupancy of supersegments. This formulation allowed our Lagrangian method to relax both the segment capacities and the supersegment capacities.

Our heuristic solution procedure ensured a feasible solution and avoided deadlock. The dual solution gave a lower bound and the duality gap is an indicator of the quality of the solution. Our small test problem gave the optimal solution over the range of

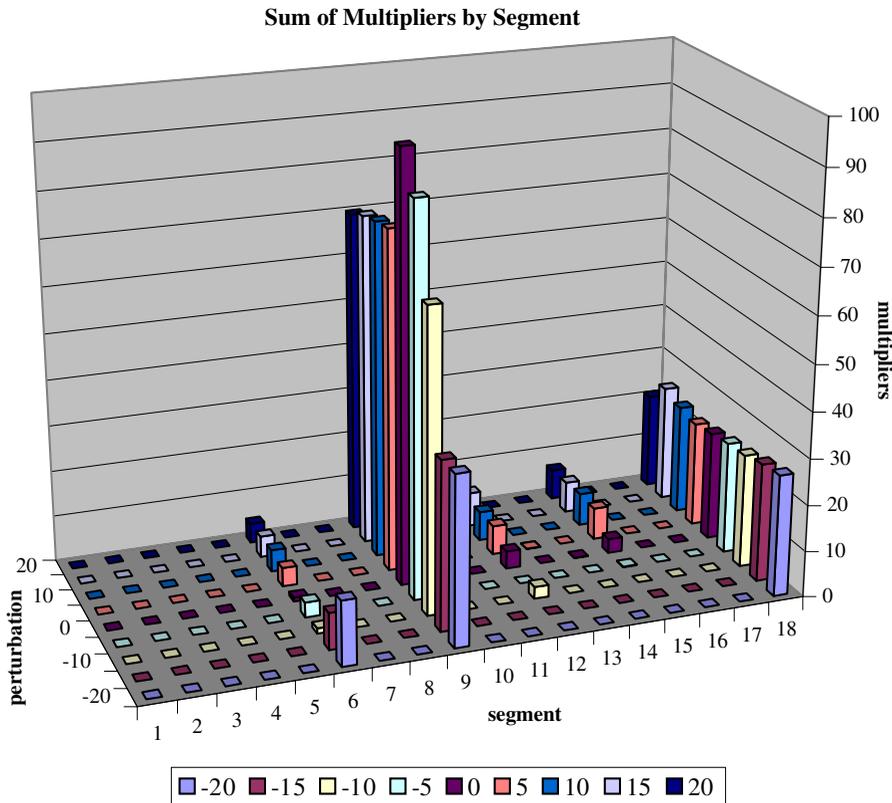**Sum of Multipliers by Segment**

FIGURE 7. Sum of the multipliers over time for each segment as the departure perturbation of Train C varies by up to 20 minutes.

instances examined. We found that the solution was robust to small variations in the departure time of a train in that the system costs changed smoothly.

The multipliers associated with the method provided a network cost for each train. This network cost varied with departure time and provided an estimate of infrastructure access cost associated with different time slots. Also the multipliers provided a value for each segment in the network for the current timetable and suggested infrastructure investment to improve the timetable performance. Our method allows an infrastructure owner to estimate the additional cost of operating a train on a network and to estimate the value of each segment of the infrastructure.

There are several possible extensions of the work. We intend to investigate the supersegment modelling method for larger train sets where pairs of trains oppose each other over part of the network but may follow each other over other parts. Also we intend to examine the trade-off between network cost and delay cost for individual trains as the weighting of the train is varied. It could be worthwhile for a train owner to pay a premium for access resulting in reduced delays.

Acknowledgements

References

[1]   J. E. Beasley, "Lagrangian relaxation", in *Modern heuristic techniques for combinatorial problems* (ed. C. R. Reeves), (McGraw-Hill, New York, 1995) 243–303.

[2]   U. Brannlund, P. O. Lindberg, A. Nou and J.-E. Nilsson, "Railway timetabling using Lagrangian relaxation", *Transp. Sci.* **32** (1998) 358–369.

[3]   X. Cai, C. J. Goh and A. I. Mees, "Greedy heuristics for rapid scheduling of trains on a single track", *IIE Trans.* **30** (1998) 481–493.

[4]   M. J. Dorfman and J. Medanic, "Scheduling trains on a railway network using a discrete event model of railway traffic", *Transp. Res. B* **38** (2004) 81–98.

[5]   O. Frank, "Optimal pacing of trains in freight railroads: Model formulation and solution", *Oper. Res.* **39** (1991) 82–99.

[6]   A. Higgins, E. Kozan and L. Ferreira, "Optimal scheduling of trains on a single line track", *Transp. Res. B* **30** (1996) 147–161.

[7]   D. Kraay, P. T. Harker and B. Chen, "Two-way traffic on a single line of railway", *Oper. Res.* **14** (1991) 801–810.

[8]   R. G. J. Mills, S. E. Perkins and P. J. Pudney, "Dynamic rescheduling of long haul trains for improved timekeeping and energy conservation", *Asia-Pacific J. Oper. Res.* **8** (1991) 146–165.

[9]   R. G. J. Mills and P. J. Pudney, The effects of deadlock avoidance on rail network capacity and performance, *Proc. 2003 Mathematics in Industry Study Group*, University of South Australia, January 2003.

[10] J.-E. Nilsson, "Towards a welfare enhancing process to manage railway infrastructure access", *Transp. Res. A* **36** (2002) 419–436.

[11] P. Pudney and A. Wardop, Generating train plans with problem space search, *Presented at the 9th Int. Conf. on Computer-Aided Scheduling of Public Transport*, San Diego, CA, 9–11 August 2004 [Online] http://fugazi.engr.arizona.edu/caspt/.

[12] B. Szpigel, "Optimal scheduling on a single line railway", in *Operational research '72* (North-Holland, Amsterdam, 1973) 344–352.