
Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

# Railway Timetabling Using Lagrangian Relaxation

U. Brännlund, P. O. Lindberg, A. Nõu, J.-E. Nilsson,

# Railway Timetabling using Lagrangian Relaxation*

U. BRÄNNLUND

*Department of Mathematics, Royal Institute of Technology, Stockholm, Sweden, Email: uffe@math.kth.se*

P. O. LINDBERG

*Department of Mathematics, Linköping University, Linköping, Sweden, Email: polin@mai.liu.se*

A. NÕU

*Department of Mathematics, Royal Institute of Technology, Stockholm, Sweden, Email: andreasn@math.kth.se*

J.-E. NILSSON

*Centre for Research in Transportation and Society, Borlänge, Sweden, Email: jer@cts.du.se*

*We present a novel optimization approach for the timetabling problem of a railway company, i.e., scheduling of a set of trains to obtain a profit maximizing timetable, while not violating track capacity constraints. The scheduling decisions are based on estimates of the value of running different types of service at specified times. We model the problem as a very large integer programming problem. The model is flexible in that it allows for general cost functions. We have used a Lagrangian relaxation solution approach, in which the track capacity constraints are relaxed and assigned prices, so that the problem separates into one dynamic program for each physical train. The number of dual variables is very large. However, it turns out that only a small fraction of these are nonzero, which one may take advantage of in the dual updating schemes. The approach has been tested on a realistic example suggested by the Swedish National Railway Administration. This example contains 18 passenger trains and 8 freight trains to be scheduled during a day on a stretch of single track, consisting of 17 stations. The computation times are rather modest and the obtained timetables are within a few percent of optimality.*

For most railway companies, track capacity is a scarce resource, at least during parts of the day. Several trains of different types compete for the same capacity, e.g., intercity, local, and freight trains. It is a truly delicate task to schedule the trains in such a way, that the needs of different train types are balanced against each other. This difficulty is mainly because the train system is so tightly interknit. Changes in the timetable at one place usually give rise to conflicts at other places.

Today, timetables usually are constructed manually. Typically, the new timetable is constructed by modifying last year's timetable. This implies that the new timetable inherits properties that may be unnecessary and costly. The construction process is usually very time consuming. Thus, for time reasons, there is no possibility to optimize the timetable, and the planner is often satisfied to find a feasible timetable. Hence, we believe there is a need for

* Research supported by the Swedish Transport and Communications Research Board, the Swedish Research Council for the Engineering Sciences, and the Swedish National Railway Administration.



**Transportation Science**
Vol. 32, No. 4, November 1998

0041-1655/98/3204-0358 $01.25

more automated timetabling. Such a tool would imply a faster construction process, giving the planner a possibility to optimize the timetable.

In this paper, we set up an optimization model for the timetabling problem. We suggest an algorithm, and test it on small but realistic examples. The computation times are moderate considering the sizes of the optimization problems. Of course there are many aspects we do not consider, but these hopefully may be added to later, more complex, models. We hope that the suggested procedure will be a first step toward intelligent computer assisted profit/utility maximizing allocation of track capacity in timetabling and other situations. The approaches of SAUDER AND WESTERMAN (1983) and JOVANOVIĆ AND HARKER (1991) to computer-aided train dispatching seem not to be applicable in our setting.

The model and solution technique of Section 1 also have other applications. We list three of them below.

**Allocation of Track Capacity in a Deregulated Train System.** The market for transportation on rail-bound vehicles is increasingly exposed to public and political pressure for deregulation. Previously, it has been common belief that the train market is not well set for free competition. This is due to the huge investments necessary to build railroads, and the rather small marginal costs for running trains. Another important aspect is that scheduling of train traffic is an intricate problem; the strong interaction between different trains on the track makes it very advantageous for the operator to have complete control over the tracks.

Nevertheless, in Sweden, a first step toward deregulation was taken in 1988, when the publicly owned train operator, SJ (Swedish State Railways), was divided into two separate companies, one which operates passenger and freight trains and another, Banverket (Swedish National Railway Administration), which owns and maintains the tracks.

In a deregulated market, the demands for track usage from different train operators interact in a complex way. Thus, a bid from an operator concerning access to the track, cannot just be a desired timetable and an acceptable price. It must concern information of values of timetables close to the nominal one. The problem for the track owner is how to choose between the suggested timetables. Our approach to timetabling can be used in an auction setting in which the operators bid for access to the tracks by specifying their profits as functions of the timetable and where the track owner, using our algorithm, chooses the timetable that maximizes total profits. Auctions for this situation have been studied experimentally by NILSSON (1996), and ISAKSSON AND NILSSON (1996).

**Allocating Trains to Platforms.** Given a timetable, it is a nontrivial task to allocate trains to platforms, at least for large stations. Trains come from given directions at certain times, they have to stop for a given length of time at some platform and they leave in specified directions at prescribed times. For a platform allocation to be feasible, trains must not occupy the same track or switch at the same time and they must adhere to the timetable.

The problem is similar to the timetabling problem and one may use the techniques described below. ZWANEVELD et al. (1996) provide a model formulation of this problem and suggest a branch-and-bound algorithm for its solution.

**Rescheduling of Trains under Disturbances.** Trains may deviate from the published timetable for many reasons: signal errors, accidents, late arrival of locomotives, maintenance work on the track, etc. Delays may often magnify because trains influence each other. Thus, when a delay has occurred, there is a need to reschedule the trains to get back to the original timetable.

Often, the rescheduling has to be performed under time pressure. Therefore, in these situations there is a pronounced need for automatic timetabling tools, which give feasible timetables fast. The methods proposed below can also be used in such situations.

In Section 1, we model the basic optimization problem of how to allocate track capacity to trains in timetabling. Each train is supposed to have a profit function, depending on the time of departure from the terminal minus penalties for waiting along the track. The track is divided into blocks. At each time instance only one train may occupy a block. By lifting these capacity constraints to the objective through Lagrange multipliers, one for each time slot and block, we obtain a relaxed problem. It decomposes into one shortest path problems for each train. The solutions of these are modified to give feasible solutions (timetables).

Sections 2 and 3 treat two central problems of the methods of this paper. The former concerns the problem of solving the relaxed problem, in which capacity constraints are replaced by prices for using the capacity. The latter treats the problem of finding a good feasible timetable, utilizing current track prices and possibly the relaxed solution.

In Section 4, suitable price adjustment schemes are discussed. The number of multipliers is very large in our test problems. However, it turns out that most of them are zero close to the optimum. We

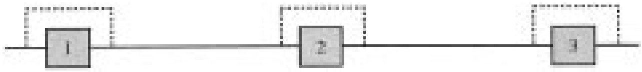



**Fig. 1.** A typical stretch of track. Main track (solid) and side track (dashed).

suggest a scheme in which we do not have to treat all of them explicitly.

Section 5 finally gives our computational results. We have tested our methods on a scheduling problem on a stretch of single track in Sweden. On this stretch, we let trains of different characteristics compete for the capacity. The train types vary from heavy freight trains to lighter passenger trains. The timetables, which are obtained in rather modest computation times, are of good quality (proven to be at most a few percent from optimality).

## 1. THE SCHEDULING PROBLEM AND ITS DUAL

IN THIS SECTION, we give a mathematical model of the timetabling problem for a stretch of single track railway. The restriction to single track is mainly for expository purposes. Double tracks may be handled in a similar manner. Moreover, the problems of scarce track capacity are more pronounced for single track.

The models and methods of this and the following sections may be seen as first attempts to attack profit maximizing train timetabling problems with optimization methods. The single stretch track problem may be refined in different directions, and we believe it will be a central subproblem when constructing timetables for rail networks.

Train tracks are, for reasons of safety and simplicity, divided into blocks, on which there must be at most one train at each given moment. A typical stretch of track is modeled as in Figure 1, where the track between stations is divided into one or several blocks. At each station, there may be a number of side tracks on which a train can stop so that another train may meet or pass. Each such side track is considered as one block.

In our model, time is discretized, typically in minutes. Let $x_{it}^r$ be a binary variable, where $x_{it}^r = 1$ if train $r$ occupies block $i$ at time $t$, and $x_{it}^r = 0$ otherwise. Let $x^r$ and $x$ denote the corresponding vectors. Let $T^r$ denote the set of vectors $x^r$, which result in technically and logically feasible schedules for train $r$, not considering the effect of other trains on the track. The set $T^r$ can include the empty schedule, in which the train does not get scheduled at all and hence $x_{it}^r = 0$ for all $i$ and $t$.

Let $v(x)$ denote the total value of a given timetable, $x$. We model the profit function $v$ as the sum of the profits of the trains in the schedule, $v(x) = \sum_r v^r(x^r)$. We have chosen to model $v^r(x^r)$ as follows. For each train, we let the profit be given by the profit of the departure time minus a per minute cost for unnecessary waiting along the track. Any prolonged travel time in comparison with the fastest feasible schedule is penalized. This applies to speed reductions as well as actual waiting at standstill. One way to model the profit/departure-time tradeoff is illustrated in Figure 2. Note, however, that our solution method allows for any functional form. The per minute waiting cost is set to $v_{\text{dep}}^r / w_{\max}^r$, where $w_{\max}^r$ is the maximum waiting tolerated by train $r$. For the empty schedule ($x^r = 0$) we have chosen $v^r(x^r) = 0$. We could as well, however, have chosen any penalty for not running a train.

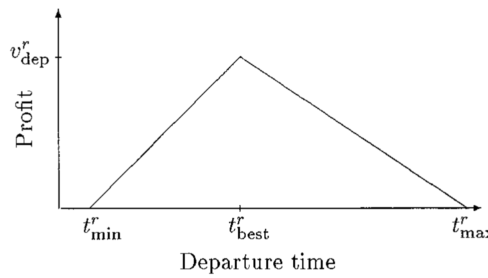



**Fig. 2.** Model of profit for departure time.

The total profit/utility maximizing problem can now be stated as

$$\text{(P)} \qquad \text{maximize} \sum_r v^r(x^r) \tag{1}$$

$$\text{subject to} \quad \sum_r x_{it}^r \leqslant 1 \quad \text{for all } i,\ t, \tag{2}$$

$$x^r \in T^r \quad \text{for all } r. \tag{3}$$

We refer to problem (P) as the *primal* problem. The constraints (2) simply state that, on each block, there can only be one train at each time period, whereas the constraints (3) ensure technical feasibility. We refer to constraints (2) as the *linking constraints*, in the sense that they link the different trains.

In our model, the sets $T^r$ include the empty schedule. If a train renders a utility less than zero it is not scheduled. One may take advantage of this fact when solving for the utility maximizing timetable for a specific train, as is described in Section 2. This option is important in the application of allocating track capacity in a deregulated train system described in the introduction.

One common way of dealing with linking constraints such as constraints (2) of problem (P) is *Lagrangian relaxation*, see, e.g., FISHER (1981) for a lucid exposition. The linking constraints are relaxed and assigned prices $\lambda_{it} \geqslant 0$ and added to the objective function. This may be interpreted that trains are charged $\lambda_{it}$ for access to block $i$ at time $t$. The relaxed problem

$$(\mathrm{P}_\lambda) \qquad \text{maximize } \sum_r v^r(x^r) + \sum_{i,t} \lambda_{it}\left(1 - \sum_r x_{it}^r\right)$$

$$\text{subject to} \quad x^r \in T^r \quad \text{for all } r, \tag{4}$$

separates into independent *subproblems* for each train $r$,

$$(\mathrm{P}_\lambda^r) \qquad \text{maximize } v^r(x^r) - \sum_{i,t} \lambda_{it} x_{it}^r$$

$$\text{subject to} \quad x^r \in T^r. \tag{5}$$

Let $\hat{x}(\lambda)$ be an optimal solution to $(\mathrm{P}_\lambda)$ and denote its value by

$$\phi(\lambda) = \sum_r v^r(\hat{x}^r(\lambda)) + \sum_{i,t} \lambda_{it}\left(1 - \sum_r \hat{x}_{it}^r(\lambda)\right). \tag{6}$$

It is then standard, see Fisher (1981), that $\phi(\lambda)$ for $\lambda \geqslant 0$ provides an upper bound on the optimal value of (P): $\phi(\lambda) \geqslant v(x) = \Sigma_r\, v^r(x^r)$ for all $x$ feasible to (P). The problem of finding the best possible such bound is the so called Lagrangian *dual* problem,

$$(\mathrm{D}) \qquad \text{minimize } \phi(\lambda)$$

$$\text{subject to} \quad \lambda \geqslant 0. \tag{7}$$

The idea is now the following. At iteration $k$, using multipliers, $\lambda^k$, we first solve the relaxed problem $(\mathrm{P}_{\lambda^k})$. Then, we use a primal heuristic to find a feasible solution, $\bar{x}_k$, to (P), for example, by adjusting $\hat{x}(\lambda^k)$. Thereafter, we update the multipliers to get closer to the solution of (D). The price adjustment scheme is easy to understand. If block $i$ at time $t$ is used by more than one train, i.e., $\Sigma_r\, \hat{x}_{it}^r(\lambda^k) > 1$, then the price is raised, $\lambda_{it}^{k+1} > \lambda_{it}^k$, thereby making it less attractive. If, in contrast, $\lambda_{it}^k > 0$ and no train is using block $i$ at time $t$, then the price is lowered, $\lambda_{it}^{k+1} < \lambda_{it}^k$. The best solution so far of (D) with objective value,

$$\phi_{\text{best}}^k = \min_{j \in \{1,\ldots,k\}} \phi(\lambda^j),$$

provides an upper bound on the optimal value of (P), and the hitherto best feasible solution to (P) with objective value

$$v_{\text{best}}^k = \max_{j \in \{1,\ldots,k\}} \sum_r v^r(\bar{x}_j^r)$$

provides a lower bound. Thus, if $\Delta^k = \phi_{\text{best}}^k - v_{\text{best}}^k$ is sufficiently small, we may terminate the iterative process. However, we can not guarantee that $\Delta^k$ will go to zero because either the problem may have a so called *duality gap* or the primal heuristic may not manage to find the optimal solution. If it is necessary to obtain the true optimum, then our approach could be incorporated in a *branch-and-bound* scheme, see e.g., NEMHAUSER AND WOLSEY (1988, p 355ff).

The general idea behind our approach can now be summarized in the form of the following fairly standard algorithm.

**Algorithm 1.** A dual iteration scheme

**Step 0.** Set $k = 1$ and initialize $\lambda^1$, for example by letting $\lambda_{it}^1 = 0$, for all $i$ and $t$. Set a limit, $k_{\max}$, on the number of iterations.

**Step 1.** Given $\lambda^k$, solve the relaxed problem $(\mathrm{P}_{\lambda^k})$, yielding $\phi(\lambda^k)$ and $\hat{x}(\lambda^k)$.

**Step 2.** Find a feasible solution, $\bar{x}_k$, to (P) with value $\Sigma_r\, v^r(\bar{x}_k^r)$.

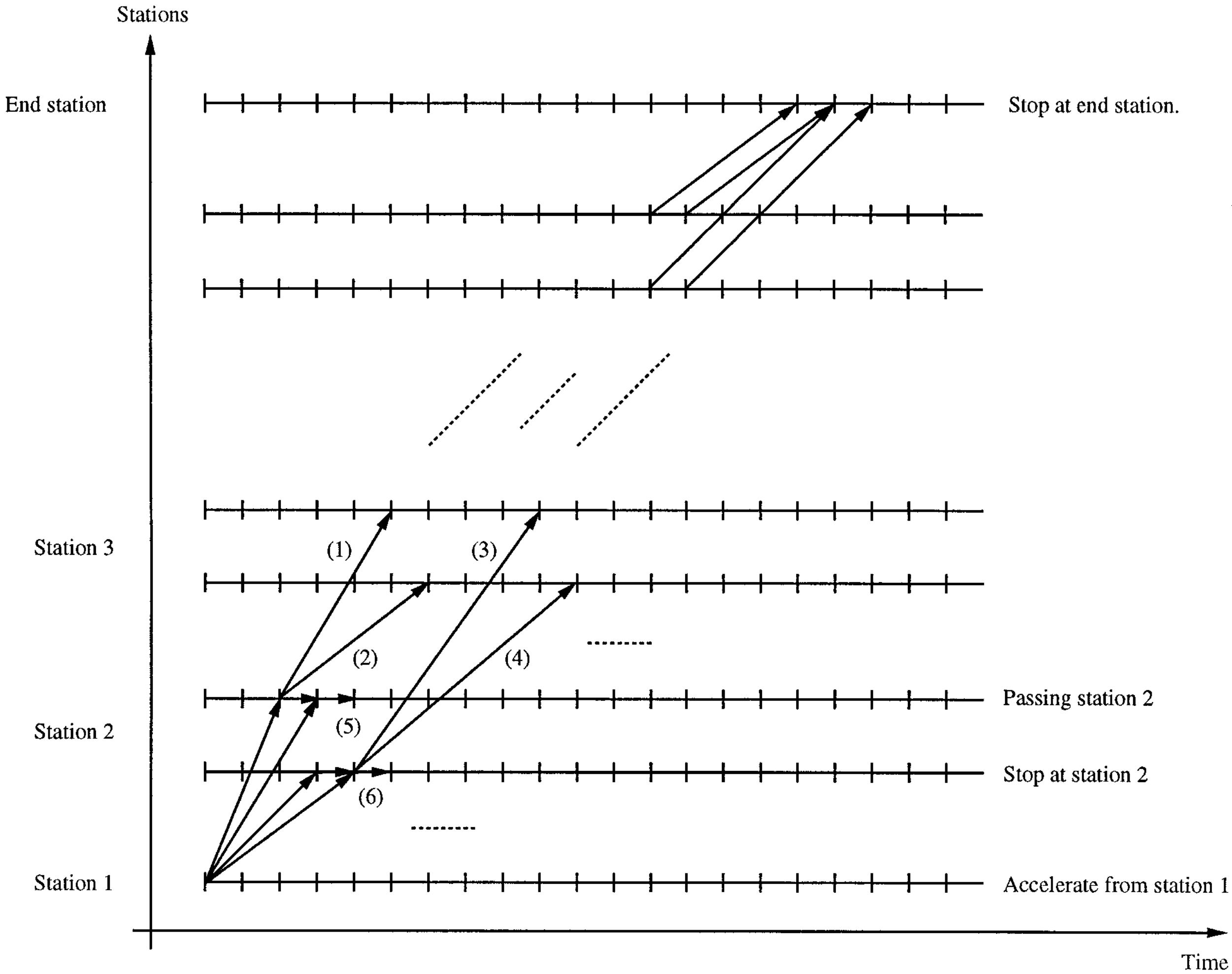
**Step 3.** Stop if $\Delta^k = \phi_{\text{best}}^k - v_{\text{best}}^k$ is small enough, or if $k = k_{\max}$. Otherwise find new prices $\lambda^{k+1}$. Set $k = k + 1$ and return to Step 1.

In the following three sections, we fill out the details of Steps 1 through 3, i.e., we show how to solve $(\mathrm{P}_\lambda)$, how to find good solutions to (P) and how to adjust the multipliers $\lambda$ to solve (D).

## 2. SOLVING THE RELAXED PROBLEM

SUBPROBLEM $(\mathrm{P}_\lambda^r)$, (5), can be viewed as a shortest path problem in a space–time network, specific for train $r$, as depicted in Figure 3. In this network, any path from Station 1 to the end station corresponds to a technically feasible train schedule. In the figure, the vertical axis corresponds to the track, and the horizontal to time, typically discretized in minutes. Each station is modeled as two levels in the space–time network. These levels correspond to the two alternatives that a train has at a station: to stop at the station or to pass it. In the case of a compulsory stop at a certain station, the passing level at that station is omitted.

Most arcs of the network correspond to the train traversing the track between two successive stations. The horizontal component of such an arc is proportional to the train's travel time between the stations (rounded upward), which in turn depends on the train type. In addition to direct travel time, the arcs that correspond to stopping at a station, include additional time due to retardation, whereas the arcs that correspond to starting from a station




Different arc types: (1) Travel time between consecutive stations (T).

(2) (T) plus retardation delay and compulsory station stop.

(3) (T) plus acceleration delay.

(4) (T) plus acceleration and retardation delay and compulsory station stop.

(5) Extra delay on main track.

(6) Extra delay on side track.

**Fig. 3.** The structure of the space–time network.

include acceleration time. Horizontal arcs at the stopping level of a station in the network represent actual waiting on a side track, and those at the passing level represent speed reduction on the incoming track.

The solution of ($P^r_\lambda$), $\hat{x}^r(\lambda)$, corresponds to the profit maximizing path from the starting station to the end station in this space–time network. Nodes representing the origin station result in profit according to the profit function in Figure 2. The cost of an arc is as follows. For an arc, representing traveling on block $i$ during time periods $t_1$ to $t_2$, the cost is $\Sigma_{t=t_1}^{t_2} \lambda_{it}$. Arcs symbolizing unnecessary waiting along the track (i.e., extra delay at a station or speed reduction) get the cost of waiting plus the dual prices of the blocks in question. Arcs corresponding to unnecessary stops also get extra costs proportional to the time of retardation and acceleration.

The network is acyclic and, thus, it is simple to solve for the profit maximizing path by traversing the nodes only once (see e.g., AHUJA, MAGNANTI, and ORLIN (1993, p. 107ff)).

If the value of the profit maximizing path is less than zero for a train, $r$, then the train is not scheduled and, therefore, $\hat{x}^r_{it} = 0$ for all $i$ and $t$. Using this fact, it is possible to calculate upper and lower limits on the scheduling interval for each station, exploiting the earliest and latest possible departure times (see Figure 2), the direct travel time, and the cost for unnecessary waiting. These limits are used to reduce the size of the network.

### 3. FINDING PRIMAL FEASIBLE SOLUTIONS

THE DUAL ITERATION SCHEME, Algorithm 1, calls for a heuristic for finding good feasible solutions. Actually, for dual optimization schemes to be of any value, some way of generating (preferably good) primal feasible solutions is needed, because, normally, one cannot expect to get feasible solutions by just adjusting the prices. In this section, we describe a heuristic approach, which has turned out to give good primal solutions without being too complicated. We do not adjust $\hat{x}(\lambda^k)$ as was suggested in Algorithm 1. Instead, we use the network representation over again and schedule the trains one after the other according to a priority list, taking into account the current prices, $\lambda^k$, for using the tracks.

To begin with, all trains to be scheduled are arranged in a priority list. The intercity trains have priority over passenger trains, which in turn have priority over freight trains. The motivation for this is that extra waiting time is usually not a big issue in the case of freight trains, whereas, in the passenger case, it certainly is. In each class, trains with high values have priority over trains with lower values.

Using the priority list, the trains are scheduled one after the other. This is done with the same technique as described in Section 2, using the current prices for using the tracks. The highest priority train will get its most desired schedule (at the prices $\lambda^k$). Thereafter the space–time network for the next train in the list is modified by removing arcs representing already occupied blocks. The train is then scheduled by finding a profit maximizing path in the reduced network. The corresponding timetable is necessarily feasible with respect to the track capacity constraints through the construction of the reduced network. The process is repeated until all trains have been scheduled. In our setting, we have the option to cancel a train, possibly at the price of a penalty. Hence, the heuristic always generates a feasible solution.

Having restricted ourselves to a priority list heuristic, it is natural to ask what ordering of the trains one should use. There seems to be no simple way to achieve a better ordering than the manually constructed list we have used. However, in a separate paper, NÕU (1997), Nõu applies elaborate local search methods to the ordering problem and is indeed able to find better primal feasible solutions.

### 4. DUAL OPTIMIZATION SCHEMES

THE OBJECTIVE, $\phi$, of the dual problem, (7), is convex, because it is a maximum of affine functions. It is not differentiable at points where $\hat{x}(\lambda)$ is not unique. There exist many methods to minimize such functions. We refer to BRÄNNLUND (1993) for a short overview and to LEMARÉCHAL (1989) for a more extensive one.

It is standard, Fisher (1981), that the vector $g(\lambda) = 1 - \sum_r \hat{x}^r(\lambda)$ is what is called a *subgradient* of $\phi$ at $\lambda$. A common dual optimization technique, which usually works quite well, is the so called *relaxation step technique* or *Polyak II*, in which the multipliers $\lambda$ are iteratively updated according to the formula

$$\lambda^{k+1} = \max(0, \lambda^k - h^k g(\lambda^k)), \tag{8}$$

where $h^k = (\phi(\lambda^k) - \bar{\phi}^k)/\|g(\lambda^k)\|^2$. The *target value* $\bar{\phi}^k$ is some convex combination of the best dual and primal function values found so far. $\|\cdot\|$ denotes the ordinary Euclidean norm.

Our dual problem is a bit different from classical applications in that it has a very large number of variables, of which only a few are nonzero in the optimal solution. In the test cases described in Appendix A, we have on the order of 40,000 dual variables but only a few hundred of them are positive close to the optimum. The subgradients are not sparse in the ordinary sense, because most of the components of $g(\lambda)$ are equal to 1, corresponding to unoccupied blocks. Thus, most components of $-g(\lambda)$ point out of the feasible region. This has the effect that the steps taken by the rule (8) are extremely short.

One way to alleviate this problem is to let $\lambda^{k+1}$ be the solution of the projection problem

$$\begin{aligned} &\text{minimize} \quad \|\lambda - \lambda^k\|^2 \\ &\text{subject to} \quad \bar{\phi}^k \geq \phi(\lambda^k) + g(\lambda^k)^T(\lambda - \lambda^k), \\ &\qquad\qquad\quad \lambda \geq 0. \end{aligned} \tag{9}$$

In Brännlund (1993), it is proven that if $\bar{\phi}^k$ is equal to the optimal dual value, then the $\lambda^{k+1}$ obtained by this modified approach is closer to the optimal set than the $\lambda^{k+1}$ obtained by (8). An efficient algorithm for (9) is also presented there, as well as discussions of other approaches to increase the step-size in the case of constraints on $\lambda$.

Subgradient algorithms like (8) and (9) tend to zig-zag; a step taken in direction $d$ tends to be followed by a step that is close to $-d$. A popular technique to counteract this phenomenon in the unconstrained case is the procedure suggested by CAMERINI, FRATTA, and MAFFIOLI (1975). Their technique can easily be modified to fit with (9). It can then take the following form. With $d^{k-1} = \lambda^k - \lambda^{k-1}$, define

$$\tilde{g}(\lambda^k) = \begin{cases} g(\lambda^k) + \alpha^k d^{k-1} & \text{if} \quad g(\lambda^k)^T d^{k-1} < 0 \\ g(\lambda^k) & \text{otherwise,} \end{cases} \tag{10}$$

where $\alpha^k = -g(\lambda^k)^T d^{k-1} / \|d^{k-1}\|^2$ and use $\tilde{g}(\lambda^k)$ in place of $g(\lambda^k)$ in (9).

*Bundle methods*, Lemaréchal (1989), are methods that accumulate information about the dual function, $\phi$, during the iterations by storing more than one subgradient. A simple bundle method is obtained by letting $\lambda^{k+1}$ be the solution of

$$\begin{aligned} &\text{minimize } \|\lambda - \lambda^k\|^2 \\ &\text{subject to} \quad \bar{\phi}^k \geqslant \phi(\lambda^j) + g(\lambda^j)^T(\lambda - \lambda^j), \quad j \in J \\ &\qquad \lambda \geqslant 0, \end{aligned} \tag{11}$$

where $J$ is an index set corresponding to a subset of the previously generated subgradients, including $g(\lambda^k)$.

In all the methods suggested above, the target value $\bar{\phi}^k$ is updated according to $\bar{\phi}^k = (1 - \gamma^k)\phi^k_{\text{best}} + \gamma^k v^k_{\text{best}}$ where $\gamma^k$ is initially 1 and then halved if $\phi^k_{\text{best}}$ has not improved for the last $K_{\max}$ iterations. This is a heuristic rule, which works well in practice if $K_{\max}$ is chosen appropriately. Updating rules, which guarantee convergence of the dual iterates, can be found in BRÄNNLUND, KIWIEL, and LINDBERG (1995).

## 5. COMPUTATIONAL EXPERIENCE

THIS SECTION IS ORGANIZED as follows. In Section 5.1 we give some general remarks on our implementation, as well as provide some parameters related to Algorithm 1. In Section 5.2 we introduce the test cases we have used. Thereafter, in Section 5.3, we comment on primal and dual results.

### 5.1 Implementational Issues

The model and the algorithms have been programmed in the linear algebra package MATLAB®, MATHWORKS INC, (1996), coupled with subroutines in C and Fortran.

The stopping criterion (cf. Algorithm 1) in these experiments was a relative duality gap, i.e., difference between the best dual and primal objective values divided by the best primal objective value, of $10^{-3}$ (which, however, never became active). The iteration limit, $k_{\max}$, was set to 200. The parameter $K_{\max}$ (cf. Section 4) was, after some initial tests, set equal to 4 for all the dual methods. For the bundle method, the maximum number of subgradients stored, $|J|$, was 50.

### 5.2 Test Cases

To validate our approach computationally, we have obtained one test case, denoted Test Case A, from Banverket. To obtain an even more congested and hence more difficult problem, four additional passenger trains were added to Test Case A, giving Test Case B. Below, we provide some characteristics of the test cases.

Test Case A involves scheduling of 26 trains (18 passenger and 8 freight) on a stretch of single track (physically located in the middle of Sweden) connecting 17 stations. This stretch of tracks connects the cities of Uppsala (U) and Borlänge (BLG). Traveling times for the two different train types were provided by Banverket through the use of a train simulator. They depend on the track topography and are thus direction dependent, as can be seen in Tables I and II. The trains have profit functions as described in Section 1. A detailed specification of the preferred timetables is given in Tables III, IV, and V.

We believe our test cases are large enough to indicate that our approach is viable. We note that they are of comparable size to the scenarios used by HALLOWELL (1993) and Jovanović and Harker (1991). SJ is currently the, by far, largest operator of passenger and freight services in Sweden. The peak load on the Swedish railway network is typically attained weekdays during regular season. The above test cases are, in terms of number of trains, of comparable size to the current peak load on this stretch of track, SJ (1997).

In Figure 4, the preferred timetables for Test Case B are illustrated in the form of a graphical timetable. The total profit of these preferred schedules is 15,200. However, these schedules inflict a total of 235 capacity violations (constituting 52 different conflict situations involving either two or three different trains), and hence not all trains will get their preferred timetables. Note in particular, the conflict between a freight train departing from BLG at 6:20 and a passenger train departing at 6:30 from the same station, and the extreme conflict between two passenger trains departing at the same time, 16:00, from U.

### 5.3 Computational Results

In Figure 5, the best primal solution obtained for Test Case B is illustrated. Here, all the conflicts have been resolved, giving a total profit of 12,346,

TABLE I

*Passenger Train Operation Characteristics*

| From | Travel Times* | | | To/From | Travel Times* | | | To |
|---|---|---|---|---|---|---|---|---|
| U | ×[a] | 1:58 | 0:17 | UNA | 0:29 | 1:38 | ×[b] | U |
| UNA | 0:42 | 5:22 | 0:21 | BNA | 0:40 | 5:20 | 0:22 | UNA |
| BNA | 0:35 | 6:23 | 0:22 | JLA | 0:39 | 6:23 | 0:18 | BNA |
| JLA | 0:40 | 7:20 | 0:20 | MA | 0:37 | 7:21 | 0:21 | JLA |
| MA | 0:37 | 3:17 | 0:21 | HY | 0:37 | 3:16 | 0:20 | MA |
| HY | 0:43 | 3:29 | 0:22 | IST | 0:40 | 3:29 | 0:23 | HY |
| IST | 0:38 | 4:18 | 0:16 | SL | 0:28 | 4:21 | 0:21 | IST |
| SL | 0:31 | 5:12 | 0:21 | BDO | 0:36 | 5:09 | 0:16 | SL |
| BDO | 0:39 | 5:44 | 0:22 | RY | 0:38 | 5:44 | 0:21 | BDO |
| RY | 0:44 | 6:03 | 0:07 | AVKY | 0:10 | 6:26 | 0:23 | RY |
| AVKY | 0:10 | 7:25 | 0:22 | SNB | 0:41 | 7:06 | 0:07 | AVKY |
| SNB | 0:37 | 6:06 | 0:18 | HDM | 0:34 | 6:06 | 0:21 | SNB |
| HDM | 0:31 | 3:41 | 0:21 | VHY | 0:34 | 3:42 | 0:18 | HDM |
| VHY | 0:29 | 5:07 | 0:19 | ST | 0:30 | 5:09 | 0:16 | VHY |
| ST | 0:23 | 5:58 | 0:11 | GTF | 0:38 | 5:54 | 0:13 | ST |
| GTF | 0:37 | 3:24 | 0:16 | SAU | 0:32 | 3:27 | 0:21 | GTF |
| SAU | 0:28 | 4:46 | ×[b] | BLG | ×[a] | 5:07 | 0:17 | SAU |

*Travel times: Acceleration delay/Free running time/Retardation delay, in minutes:seconds.
[a]Included in outbound travel time.
[b]Included in inbound travel time.

which is proven to be within 3.8% of the true optimum (see below). For Test Case A the corresponding best found feasible timetable has a profit of 12,101, which is proven to be within 0.54% of the optimum. In comparison, the total profit of the most preferred (infeasible) timetable is 13,200. We note that the best solution found for Test Case B has only slightly higher profit (12,346 − 12,101 = 245) than the one found for Test Case A, although it potentially could have been better by 1,600, if all four additional trains were to get their preferred timetables. One possible interpretation of this is that the single track is already very congested in Test Case A.

We have implemented four different dual iteration schemes. These are the standard subgradient method (8), the modified subgradient method (9), the modified CFM method (9, 10) and the bundle method (11). However, the bundle method has been implemented a bit differently from that described in Section 4, because solving the quadratic program with the large number of constraints $\lambda \geqslant 0$ by standard software is not straightforward. Instead, we

TABLE II

*Freight Train Operation Characteristics*

| From | Travel Times* | | | To/From | Travel Times* | | | To |
|---|---|---|---|---|---|---|---|---|
| U | ×[a] | 3:04 | 0:30 | UNA | 1:17 | 2:03 | ×[b] | U |
| UNA | 1:59 | 7:00 | 0:50 | BNA | 2:06 | 6:32 | 1:00 | UNA |
| BNA | 1:57 | 7:52 | 0:45 | JLA | 1:35 | 7:48 | 0:52 | BNA |
| JLA | 1:52 | 8:55 | 0:48 | MA | 1:49 | 8:54 | 0:47 | JLA |
| MA | 1:54 | 3:53 | 0:46 | HY | 1:35 | 3:54 | 0:48 | MA |
| HY | 3:27 | 4:02 | 1:04 | IST | 2:17 | 3:55 | 1:12 | HY |
| IST | 1:34 | 5:03 | 0:42 | SL | 1:45 | 5:04 | 0:46 | IST |
| SL | 2:17 | 6:11 | 0:35 | BDO | 1:27 | 6:08 | 0:52 | SL |
| BDO | 2:20 | 7:14 | 0:43 | RY | 1:31 | 7:11 | 0:43 | BDO |
| RY | 2:18 | 7:25 | 0:19 | AVKY | 0:35 | 9:20 | 1:02 | RY |
| AVKY | 0:38 | 9:03 | 0:54 | SNB | 2:12 | 8:09 | 0:19 | AVKY |
| SNB | 1:55 | 6:55 | 0:49 | HDM | 1:50 | 6:53 | 0:53 | SNB |
| HDM | 1:31 | 4:23 | 0:37 | VHY | 1:23 | 4:25 | 0:40 | HDM |
| VHY | 3:18 | 6:07 | 0:49 | ST | 2:01 | 5:47 | 0:54 | VHY |
| ST | 1:13 | 7:10 | 0:40 | GTF | 2:05 | 7:01 | 0:27 | ST |
| GTF | 1:30 | 4:11 | 0:49 | SAU | 1:48 | 4:13 | 0:36 | GTF |
| SAU | 1:28 | 5:31 | ×[b] | BLG | ×[a] | 6:23 | 0:40 | SAU |

*Travel times: Acceleration delay/Free running time/Retardation delay in minutes:seconds.
[a]Included in outbound travel time.
[b]Included in inbound travel time.

TABLE III

*Test Case A: Passenger Trains*

| Train No. | From | To | Compulsory Stop (of 1 minute) at | Departure Time (hrs:min) $t_{best}$ | (min) $t_{min}$ | $t_{max}$ | $v_{dep}$ | (min) $w_{max}$ |
|---|---|---|---|---|---|---|---|---|
| 1 | BLG | U | SL; AVKY | 06:30 | −15 | +15 | 625 | 5 |
| 2 | BLG | U | SL; AVKY | 08:00 | −15 | +15 | 625 | 5 |
| 3 | BLG | U | SL; AVKY | 10:30 | −15 | +15 | 625 | 5 |
| 4 | BLG | U | SL; AVKY | 12:00 | −15 | +15 | 625 | 5 |
| 5 | BLG | U | SL; AVKY | 15:00 | −15 | +15 | 625 | 5 |
| 6 | BLG | U | SL; AVKY | 16:30 | −15 | +15 | 625 | 5 |
| 7 | BLG | U | SL; AVKY | 19:00 | −15 | +15 | 625 | 5 |
| 8 | BLG | U | SL; AVKY | 20:30 | −15 | +15 | 625 | 5 |
| 9 | U | BLG | SL; AVKY | 08:30 | −15 | +15 | 625 | 5 |
| 10 | U | BLG | SL; AVKY | 10:00 | −15 | +15 | 625 | 5 |
| 11 | U | BLG | SL; AVKY | 12:30 | −15 | +15 | 625 | 5 |
| 12 | U | BLG | SL; AVKY | 14:00 | −15 | +15 | 625 | 5 |
| 13 | U | BLG | SL; AVKY | 17:00 | −15 | +15 | 625 | 5 |
| 14 | U | BLG | SL; AVKY | 18:30 | −15 | +15 | 625 | 5 |
| 15 | U | BLG | SL; AVKY | 21:00 | −15 | +15 | 625 | 5 |
| 16 | U | BLG | SL; AVKY | 22:00 | −15 | +15 | 625 | 5 |
| 17 | BLG | U | SL; AVKY; HDM; ST | 07:10 | −30 | +30 | 500 | 5 |
| 18 | U | BLG | SL; AVKY; HDM; ST | 16:00 | −30 | +30 | 500 | 5 |

solve for an approximate solution of (11) in each iteration. This approximation is given by $\lambda^{k+1} = \max\{0, \bar{\lambda}^{k+1}\}$ where $\bar{\lambda}^{k+1}$ solves

$$\text{minimize } \|\lambda - \lambda^k\|^2 \tag{12}$$

$$\text{subject to} \quad \bar{\phi}^k \geqslant \phi(\lambda^j) + \bar{g}(\lambda^j)^T(\lambda - \lambda^j) \qquad j \in J,$$

where $\bar{g}(\lambda^j)$ is a so called *conditional subgradient* at $\lambda^j$, defined by

$$\bar{g}_{it}(\lambda^j) = \begin{cases} 0 & \text{if} \quad \lambda^j_{it} = 0 \quad \text{and} \quad g_{it}(\lambda^j) \geqslant 0, \\ g_{it}(\lambda^j) & \text{otherwise.} \end{cases}$$

In practice, we solve the dual of (12), for which we use the general purpose least-squares solver LSSOL,

TABLE IV

*Test Case A: Freight Trains*

| Train No. | From | To | Departure Time (hrs:min) $t_{best}$ | (min) $t_{min}$ | $t_{max}$ | $v_{dep}$ | (min) $w_{max}$ |
|---|---|---|---|---|---|---|---|
| 1 | AVKY | BLG | 06:00 | 0 | +120 | 300 | 120 |
| 2 | AVKY | BLG | 07:00 | 0 | +120 | 300 | 120 |
| 3 | BLG | AVKY | 17:00 | 0 | +120 | 300 | 120 |
| 4 | BLG | AVKY | 17:30 | 0 | +120 | 300 | 120 |
| 5 | AVKY | BLG | 07:30 | 0 | +30 | 500 | 30 |
| 6 | BLG | AVKY | 18:30 | 0 | +30 | 500 | 30 |
| 7 | BLG | U | 06:20 | −90 | +90 | 200 | 90 |
| 8 | U | BLG | 17:30 | −90 | +90 | 200 | 90 |

TABLE V

*Test Case B: Additional Passenger Trains*

| Train No. | From | To | Compulsory Stop (of 1 minute) at | Departure Time (hrs:min) $t_{best}$ | (min) $t_{min}$ | $t_{max}$ | $v_{dep}$ | (min) $w_{max}$ |
|---|---|---|---|---|---|---|---|---|
| 19 | BLG | U | MA; VHY | 09:00 | −10 | +10 | 400 | 5 |
| 20 | BLG | U | MA; VHY | 16:00 | −10 | +10 | 400 | 5 |
| 21 | U | BLG | MA; VHY | 09:00 | −10 | +10 | 400 | 5 |
| 22 | U | BLG | MA; VHY | 16:00 | −10 | +10 | 400 | 5 |

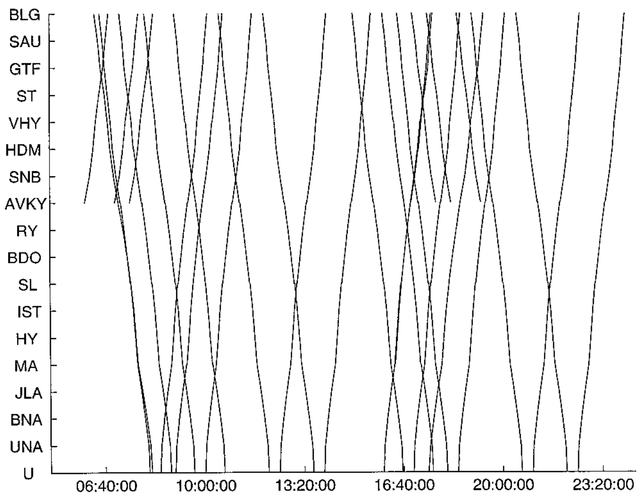



**Fig. 4.** Preferred timetables for Test Case B.

GILL et al. (1986). The number of variables in this dual quadratic program is the number of stored subgradients $|J|$, which typically is far less than the dimension of $\lambda$.

In Figure 6, we show the behavior of our algorithms on Test Case A. The upper curves correspond to the best dual objective values and the lower curves to the objective values of the best feasible solutions. There are a couple of observations to be made from this figure. First, we notice the slow convergence of the standard subgradient technique (8). This is due to the small steps caused by the nonnegativity constraints on the multipliers, as explained in Section 4. Therefore, this method is not considered any further. Second, the other three methods converge quickly, especially with respect to the large number of dual variables (which is over 40,000), as well as considering the number of nonzero dual variables at the optimum, which is 270. Third, there is not a big difference in the performance of the three best methods. The bundle method does manage to obtain a slightly better dual value (12,166) than the other two methods (12,182 for modified subgradient and 12,176 for modified CFM). However, considering the duality gap, $(12{,}166 - 12{,}101)/12{,}101 = 0.54\%$, it is doubtful whether it is worthwhile to use such an elaborate scheme.

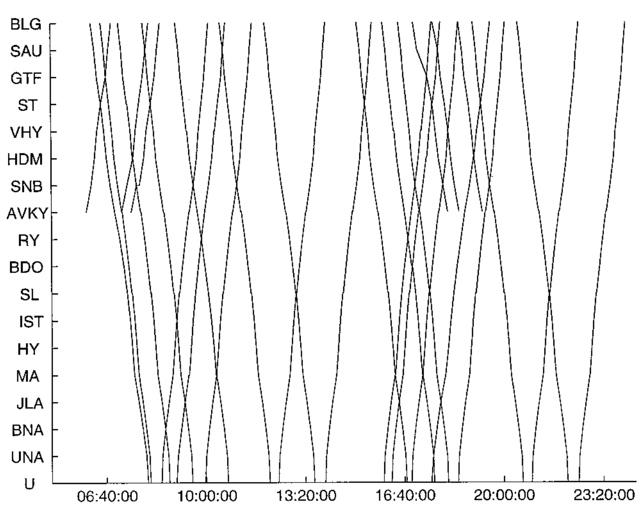



**Fig. 5.** Best obtained feasible timetable for Test Case B.

In Figure 7, the results for Test Case B are demonstrated. Here, the number of nonzero multipliers at the last iteration of the bundle method is 420, to be compared to 270 for Test Case A. The best dual function value found by the bundle method is 12,818 and the best primal solution has a value of 12,346. Hence, we see that the relative duality gap has increased to $(12{,}818 - 12{,}346)/12{,}346 = 3.8\%$. We do not know if this is because our primal heuristic is too simple, or if there is a significant true duality gap in this test case.

The implementation has not been optimized with respect to speed, because our main objective has been to see if our approach is computationally viable. However, we realize that CPU-timing is important for the approach to be accepted in practice. Therefore, to get a feeling for the computational burden, please consider the following. The computational tests have been performed on a SUN Sparcstation 20 (SPECint95 = 3.11, SPECfp95 = 3.10). One hundred iterations of Algorithm 1 on Test Case A using any of the dual updating schemes take roughly one minute. The solution of the QP-problems, needed for the bundle update, use less than 10% of the total computation time.

## 6. EXTENSIONS AND CONCLUSIONS

THE CONTRIBUTION of this paper is three-fold. First, a novel approach to train scheduling has been proposed. This is all the more important, considering that the typical way to handle these problems does not include any formal optimization techniques, but relies on paper-and-pencil techniques based on rules-of-thumb. Second, it has been shown how to heuristically solve the proposed optimization formulation in rather short computation times by means of Lagrangian relaxation. Moreover, Lagrangian relaxation seems to give feasible solutions within a few percent of optimality, at least in our test cases. This implies that it could be feasible to apply branch and bound to compute the true model optimum. On the other hand, one may very well stay satisfied with a solution within a few percent of the bounds, considering uncertainty in the data. The third more tech-

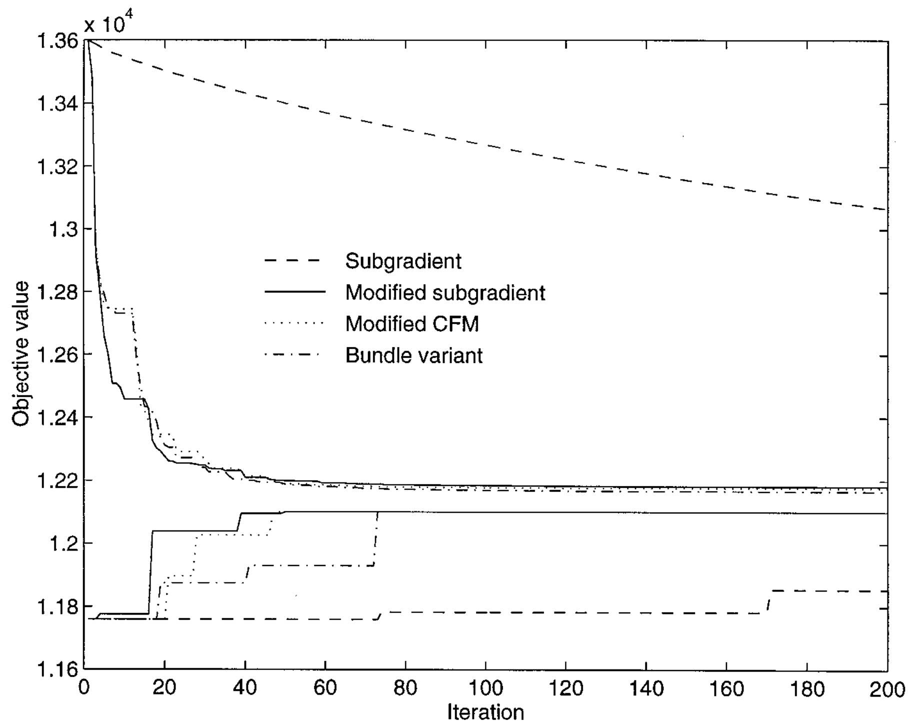



**Fig. 6.** Best primal and dual values for all four dual schemes (Test Case A).

nical contribution is that we have shown experimentally that the unusually large numbers of dual variables encountered in this problem need not be disastrous, if nonnegativity constraints are treated properly.

The presented model can be extended in several

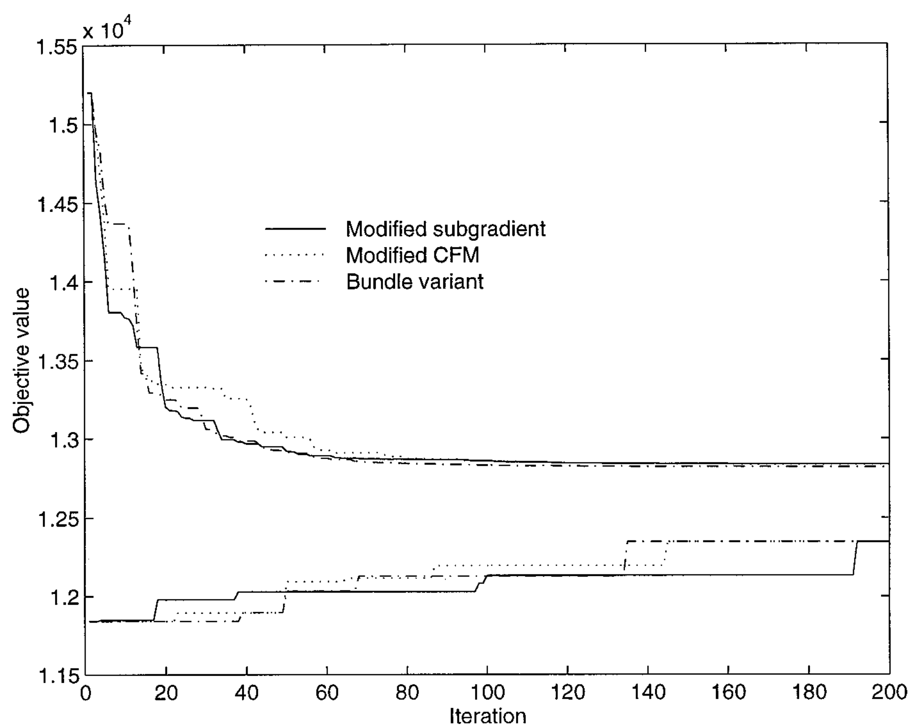



**Fig. 7.** Best primal and dual values for the three best schemes (Test Case B).

directions, and would indeed need to be, to become of practical use. Each such extension, though, would necessitate further research. We hope to be able to come back to these questions later.

The most important extension is maybe to extend the methods to a network of single track lines. This can perhaps be achieved by decomposing the network into single track lines by some form of Lagrangian decomposition and then using the technique of the present paper. Further, the profit functions of the trains probably need to be modeled more accurately. For instance, consecutive trains interact in that they compete for passengers. Here, however, we have to strike a balance between accurate modeling and computational tractability.

A train's schedule usually includes more than one trip between end stations during a day. Then, restrictions on minimum turn-around time at the terminals must be observed. This situation can be incorporated into our model in the following manner. For each physical train, one constructs only one space–time network. Hence, all the trips during a scheduling interval are handled in the same network. At end stations, a turning level is introduced. Arcs are added from the stopping level at the end station to the turning level, and their horizontal extension corresponds to the minimum turn-around time.

Finally, when constructing timetables, one usually tries to make them robust, i.e., such that delays do not propagate. This is usually achieved by introducing slack into the timetable. We have not studied these questions. To settle such questions, one would need to simulate traffic with disturbances, as is done by Jovanović and Harker (1991). Hopefully, the need for slack will diminish, if there are fast methods to reschedule the trains in case of disturbances.

## REFERENCES


AHUJA, R. K., T. L. MAGNANTI, AND J. B. ORLIN, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs (1993).

BRÄNNLUND, U., On Relaxation Methods for Nonsmooth Convex Optimization, Ph.D. dissertation, Department of Mathematics, Royal Institute of Technology, Stockholm, 1993.

BRÄNNLUND, U., K. C. KIWIEL, AND P. O. LINDBERG, "A Descent Proximal Level Bundle Method for Convex Nondifferentiable Optimization," *Operations Research Letters* **17,** 121–126 (1995).

CAMERINI, P. M., L. FRATTA, AND F. MAFFIOLI, "On Improving Relaxation Methods by Modified Gradient Techniques," *Mathematical Programming Study* **3,** 79–97 (1975).

FISHER, M. L., "The Lagrangian Relaxation Method for Solving Integer Programming Problems," *Management Science* **27,** 1–18 (1981).

GILL, P. E., S. J. HAMMARLING, W. MURRAY, M. A. SAUNDERS, AND M. H. WRIGHT, "User's Guide for LSSOL (Version 1.0): a Fortran Package for Constrained Linear Least-Squares and Convex Quadratic Programming," Report SOL 86-1, Department of Operations Research, Stanford University, 1986.

HALLOWELL, S., Optimal Dispatching under Uncertainty: With Application to Railroad Scheduling, Ph.D. dissertation, Systems Engineering Department, University of Pennsylvania, 1993.

ISAKSSON, G., AND J.-E. NILSSON, Experimental Comparison of the First-Price and Second-Price Auctions—The Case of Track Capacity Allocation, Working paper, Centre for Research in Transportation and Society, Dalarna University College, Borlänge, Sweden, 1996.

JOVANOVIĆ, D. AND P. T. HARKER, "Tactical Scheduling of Rail Operations: The SCAN I System," *Transportation Science* **25,** 46–64 (1991).

LEMARÉCHAL, C., "Nondifferentiable Optimization," in *Handbooks in Operations Research and Management Science, Vol. 1, Optimization*, Nemhauser, G. L., A. H. G. Rinnooy Kan, and M. J. Todd (eds), Elsevier Science, Amsterdam, 529–572, 1989.

MATHWORKS INC., *Using MATLAB*, The MathWorks Inc., Natick, MA., USA (1996).

NEMHAUSER, G. L., AND L. A. WOLSEY, *Integer and Combinatorial Optimization*, John Wiley & Sons, Inc., New York, 1988.

NILSSON, J.-E., Allocation of Track Capacity. Experimental Evidence on the Use of Priority Auctioning in the Railway Industry, Working paper, Centre For Research in Transportation and Society, Dalarna University College, Borlänge, Sweden (1996).

NÕU, A., Railway Timetabling—Lagrangian Heuristics, Technical report, TRITA/MAT-97-OS12, Department of Mathematics, Royal Institute of Technology, 1997.

SAUDER, R. L., AND W. M. WESTERMAN, "Computer Aided Train Dispatching: Decision Support through Optimization," *Interfaces* **13,** 24–37 (1983).

SJ, Swedish State Railways, private communication, 1997.

ZWANEVELD, P. J., L. G. KROON, H. E. ROMEIJN, M. SALOMON, S. DAUZÈRE-PÉRÈS, S. P. M. VAN HOESEL, AND H. W. AMBERGEN, "Routing Trains through Railway Stations: Model Formulation and Algorithms," *Transportation Science* **30,** 181–194 (1996).


(Received: November 1996; revisions received: December 1997; accepted: December 1997).