An Efficient MIP Model for Locomotive Scheduling with Time Windows

Martin Aronsson[1], Per Kreuger[1] and Jonatan Gjerdrum[2]

[1] Swedish Institute of Computer Science (SICS)
Box 1263, SE-164 29 Kista, Sweden
{martin.aronsson,per.kreuger}@sics.se
[2] Green Cargo AB
Box 39, SE-171 11 Solna, Sweden
Jonatan.Gjerdrum@greencargo.com

Abstract This paper presents an IP model for a vehicle routing and scheduling problem from the domain of freight railways. The problem is non-capacitated but allows non-binary integer flows of vehicles between transports with departure times variable within fixed intervals. The model has been developed with and has found practical use at Green Cargo, the largest freight rail operator in Sweden.

Keywords Vehicle routing and scheduling, rail traffic resource management, resource levelling

1 Introduction

The increasing competition within the railway transportation sector requires long-term sustainable and effective resource utilisation methods for companies such as Green Cargo, the largest rail freight operator in Sweden.

In many countries in Europe, railroads have traditionally been state-owned organisations with diverse interests in e.g. passenger traffic, freight traffic, infrastructure and real estate investments. The Swedish state railway was properly deregulated in all these aras around the millennium, creating separate companies with dedicated resources. Before the deregulation, locomotives were used for passenger traffic in the day-time and freight traffic at night. Today, vehicles are dedicated to one type of traffic, which has brought about utilisation patterns such as in figure 1. In 2005, Green Cargo is facing heavy reinvestments in its locomotive fleet.

If Green Cargo would be able to even out (*level*) the production resource requirements at peak times less investments would be needed. Green Cargo thus aims at a levelled week-day production.

ATMOS 2006
6th Workshop on Algorithmic Methods and Models for Optimization of Railways
http://drops.dagstuhl.de/opus/volltexte/2006/683

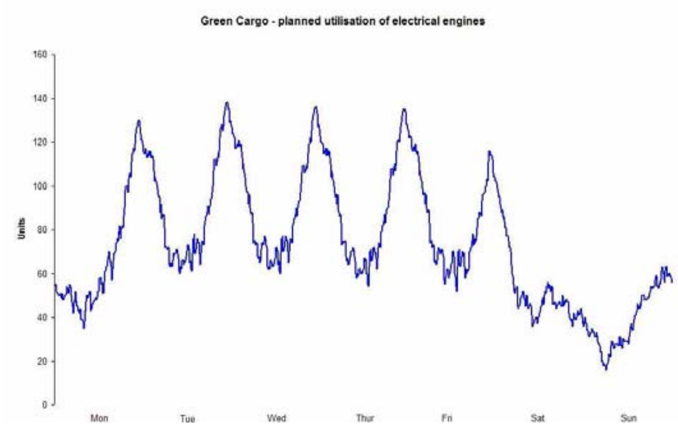


Fig. 1. Vehicle utilization pattern

1.1 Timetable

Swedish railway companies have to adhere to timetables partly designed by the government authorities. Green Cargo as well as other operators bids for allocation slots in the track network based on information about traffic patterns and customer requirements. If no slot conflicts arise, the operators receive their bids. However, the service potential is in some sense beyond the control of the individual operators.

1.2 Locomotive optimisation

The locomotive optimisation process determines the turn-round plan for all locomotives. In the fleet assignment process, a sequence of timetable slots is assigned to each locomotive by optimising planning tools based on network models. Furthermore, multiple locomotives and deadhead passive transports usage is determined, and options for maintenance is provided.

There are a number of rules or preconditions that a solution to this problem has to comply with: only specific points for switching locomotives exist, specific locomotives switching times are required, pulling power at specific lines are prespecified as is the type of locomotives that can be utilised.

Normally, the *timetable slots* are considered as *given* in this process. However, the flexibility of scheduling rail freight timetables is greater than that of passenger railways as long as customer requirements are catered for. If the slots were allowed to be shifted in time, this would enable several locomotive turns to take place which would otherwise be considered infeasible. I.e. it is not enough to only generally reduce the peaks of the traffic. To reduce the number of locomotives in the turn-round plans, the turns have to be made at the right moments as well.

Locomotives thus need to be at the right location at the right time considering that customer, resource sharing (e.g. crew and waggons) and turn-round requirements as well as input from operators sharing the track resources also have to be taken into account when determining the slots. Both duration (travelling time) and end points of the slot are determined by these factors.

There have been no commercially available tools to help the planners bridge the gap between timetable slot planning and locomotive turn-round planning. The idea presented here introduce the opportunity to modify the slot positioning in the early phase of the timetabling process so as to enable the locomotive optimisation process to achieve better results while retaining quality in terms of customer service level performance. A further optimised locomotive turn-round plan will also result in less reinvestment requirements making it an important incentive for conducting this research.

2 Problem description

Minimum cost network flow models have been extensively used (see e.g. [1]) to compute an optimal allocation of a set of scheduled transports to vehicles. The transports are normally represented as nodes in a network and the fact that a vehicle, used by one transport, can also be used by an other one, as a (directed) arc between their corresponding nodes. Classical network flow models of this kind usually have set partitioning structure and binary flow variables so that each transport is allocated to a unique vehicle. They also usually include one or more depot-nodes and are acyclic either in the plane or on a cylinder.

A straightforward generalisation of this type of flow model for cyclic schedules without depots, allows (small) integer values for the flows and have been used for engine routing in rail transportation (see e.g. [2]). In such models, additional integer variables are associated with each node to encode how many vehicles travel with each transport. Flow is conserved on each node without any depots giving cyclic schedules for each vehicle. Lower and upper bounds on the node variables variables capture the minimum and maximum number of vehicles required and usable by each transport. In addition, multiple commodities can be used to encode heterogeneous vehicle fleets.

Lower bounds on the node variables vary in known cases from 0 on (potential) passive transports (service trains) to 2 for heavy freight transports which require at least two engines. Upper bounds larger than the corresponding lower ones encode the possibility to relocate additional *accompanying* vehicles with a planned transport that is already served with the required number of vehicles. With a cost function penalising the total number of vehicles needed and any nonproductive relocation of the vehicles we get a straight forward and practical model which has seen several years of practical use in e.g. the Swedish rail industry.

Normally, the network is statically generated using temporal non-overlap and distance conditions on the transports. It would, however, be of great practical value if this kind of model could be generalised to allow for rescheduling of the transports in cases where this would significantly reduce the cost of the vehicle usage. Using time windows for the departure times of the transports and an initial network with connections between any two transports which arrive and depart from the same location, breaks the locality (and hence, the network structure) of the model since a transfer of a vehicle (turn) from one transport to another may pose requirements on the placement of the transports within its time window which can be incompatible with requirements posed by another transfer.

Problems of this general type are variants of the "multiple Travelling Salesman Problem" (m-TSP). The case with time windows is normally referred to as a "multiple Travelling Salesman Problem with Time Windows" m-TSPTW. See e.g. [3,1,4,5,6,7]. This problems is normally (e.g. [8]) considered as a special case of the extensively studied class "Vehicle Routing Problems" (VRPs) [9,10]. One could also argue that the m-TSPTW is an uncapacitated variant of the "Vehicle Routing Problems with Time Windows" (VRPTW) which is also well studied, albeit using mainly other methods than the one proposed here. See e.g. [11,12,13,14,15,16,17,18]. They are also part of the larger problem of assigning engines to pre-scheduled transports based on more general transport and vehicle properties referred to as "Locomotive Scheduling Problems" (see e.g. [5]).

The problem under study here differs from most of those studied in this literature by allowing non binary (but small integer) flows between nodes in the network. This corresponds to multiple (required and/or optional) temporally overlapping visits in the m-TSPTW. The problem does not include any finite capacities on vehicles as in VRPTW but neither does it have the simple set partitioning structure of the m-TSPTW type of problem. The model presented here uses a single commodity but should be straight forward to generalise for heterogeneous vehicle fleets. However, the practicality of such a generalisation has not been investigated.

The paper presents an IP-model for this problem which can be used to efficiently and exactly solve practical problems up to the size of those occurring in real life transportation planning for moderate sizes (< 3 hours) of departure time windows using a state-of-the-art commercial solver.

The model and its implementation for the solution of a fully operational large scale practical case is presented. The transports in this case are train transports with a fixed schedule whose departure times are relaxed from ± 15 up to ± 90 minutes and the vehicles considered are the engines used to pull the trains. Performance results for solving several version of the practical problem using CPLEX 9 [19] on a PC-type workstation are also reported.

3 Model parameters

The model is parametrised by a number of constants and variables with associated bounds which will be summarised here. The constraints and objective function will be presented in section 5 below. Note that we have chosen to present the variable bounds, which are, of course, also parameters to the model, in connection with the respective variables below. Note also that the problem is periodic, i.e. that the transport schedule is repeated after a fixed period CT. The individual vehicle schedules may, on the other hand, take several such periods before they are repeated.

3.1 Constants

n	The number of transports in the problem.
CT	Cycle Time (period after which the transport schedule is repeated).
t_i	Travel time for transport i. We require each t_i to be positive and strictly smaller than CT.
p_i	Penalty per vehicle *accompanying* a transport above that of its vehicle requirement.
lo_i, ld_i	Origin and destination locations for transport i.
r_{ij}	Setup time (turn-time) for the exchange of one or more vehicles between transport i and j. We require each r_{ij} to be positive and fulfil the inequality $t_i + r_{ij} < CT$. This requirement is quite natural in most cyclic transportation problems where travel times and setup times are typically small in comparison with the cycle period.
M	Any "sufficiently large" constant ("big M") used to linearise the model.

3.2 Decision variables (discrete)

X_{ij}	Integer variable, determining how many vehicles are *turned* (transferred) from transport i to transport j. In the train case considered here, the lower bound, $\underline{X_{i,j}}$ is normally 0 and the upper bound $\overline{X_{i,j}}$ either 1 or 2.
C_{ij}, C'_{ij}	Boolean variables, used to determine if a *turn* (if any) from transport i to transport j crosses the period limit CT one or more times.
Y_{ij}, Y'_{ij}	Integer variables, which for any optimal solution will have the values $Y_{ij} = C_{ij}X_{ij}$ and $Y'_{ij} = C'_{ij}X_{ij}$ respectively.
S_i	Integer variable used to encode the number of vehicles allocated to transport i. A lower bound $\underline{S_i}$ on this variable encodes the vehicle requirement of the transport while an upper bound $\overline{S_i}$ limits the number of vehicles usable/transportable by it. For the train case where only engine vehicles are considered, these limits are normally $\underline{S_i} = 1$ and $\overline{S_i} = 2$ unless the transport is a scheduled potential vehicle relocation transport, in which case the the lower bound may be $\underline{S_i} = 0$, indicating that the transport need not be performed unless needed to balance the vehicle flow of the model.
E_i	Integer variable used to encode the number of vehicles *accompanying* a transport in addition to the number required $\underline{S_i}$ by the transport itself.

3.3 Time point variables (continuous)

d_i Continuous variable denoting the departure time of train i. The departure time window of the transport i is encoded as the bounds $\underline{d_i}$ and $\overline{d_i}$ of d_i. For any i, we require that $0 \leq \underline{d_i} \leq \overline{d_i} < CT$.

This formulation does not guarantee that the arrival times $d_i + t_i$ will always be smaller than CT which influence the formulation of the constraints relating the arrival and departure events of the transports. The following section gives a case analysis of the situations that can occur and motivates the constraint formulation given in the section following it.

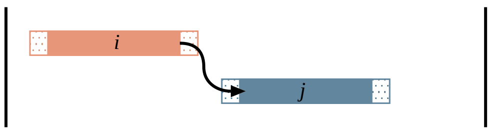
4 Turning over the cycle time border

The cases are illustrated by figures where a coloured vertical bar represents the transports. The length of the coloured bar is the travel time of the transport (the interval between scheduled departure time and arrival time). The surrounding transparent bar illustrates the departure time window of the transport so that the coloured bar may be placed anywhere within the transparent one.

There are four main cases for a turn to transport i to transport j to consider, each one described below.

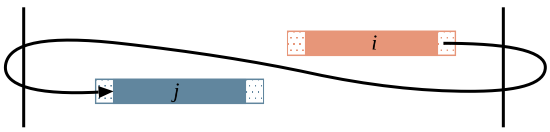
A_o The turn, if chosen, will never cross the cycle time boarder i.e.

$$\overline{d_i} + t_i + r_{ij} \leq \underline{d_j}$$



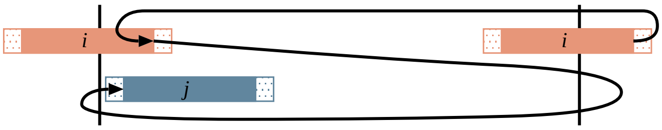
A_1 The turn, if chosen, is certain to cross the cycle time boarder exactly once. i.e:

$$\left(\underline{d_i} + t_i + r_{ij} > \overline{d_j}\right) \wedge \left(\overline{d_i} + t_i + r_{ij} - CT \leq \underline{d_j}\right)$$



A_2 A more rare case which has nevertheless to be taken into account is when the turn, if chosen, is certain to cover two periods. Note that in this case (as well as sometimes, in A_1), two instances of the transport that crosses the boarder have to be considered, one leaving the period and one entering the period, i.e:

$$\left(\underline{d_i} + t_i + r_{ij} - CT > \overline{d_j}\right)$$



This is hardly ever desirable, at least not if the period time is long in comparison with the longest travel time. In the Swedish rail freight problem the period time is a week and the longest transport travel time normally less than 24 hours.
In the model below we will penalise this case twice as hard as A_1 which in practise means that turns of this type are almost never found in an optimal solution. An alternative model could with some loss of generality instead explicitly forbid this type of turn by introducing constraints forcing $X_{ij} = 0$ whenever $\left(\underline{d_i} + t_i + r_{ij} - CT > \overline{d_j}\right)$ which would in the general case reduce the number of booleans in the model to half. In practise, however, the gain is marginal since we introduce the booleans only where they are needed and this case is as already mentioned rare.
The case where $d_i + t_i + r_{ij} > 2CT$ can be safely ignored since we require the constants to fulfil $t_i + r_{ij} \leq CT$ and $t_i < CT$.

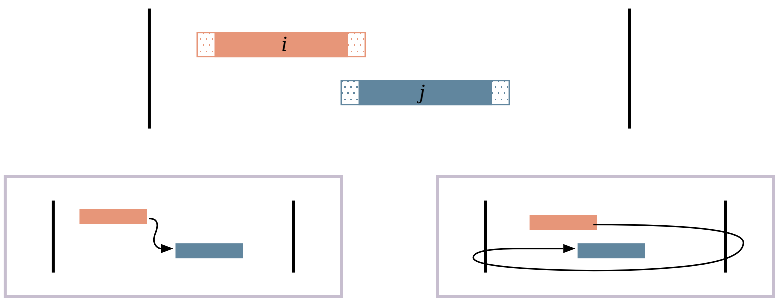
A more complex case occur when the time windows overlap so that the turn may or may not cross the cycle time boarder one or more times but the exact number depends on the assignment of the departure time variables.

For typical distributions of time windows in the rail freight case, we see mainly cases where the time window limits will place us in situations where we cannot determine if we are in case A_0 or A_1, only rarely whether we are in case A_1 or A_2 but almost never in ones where we cannot exclude at least one of the three. This fact can be used to reduce the number of boolean variables needed in the model significantly.

In the general case it is possible to distinguish the following sub-cases:

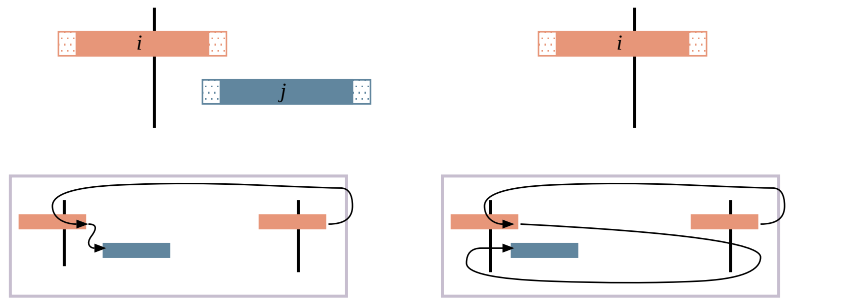
B_1 The turn may cross the cycle time limit once or not at all, i.e:

$$\left(\underline{d_i} + t_i + r_{ij} \leq \overline{d_j}\right) \wedge \left(\overline{d_i} + t_i + r_{ij} > \underline{d_j}\right) \wedge \left(\overline{d_i} + t_i + r_{ij} - CT \leq \underline{d_j}\right)$$



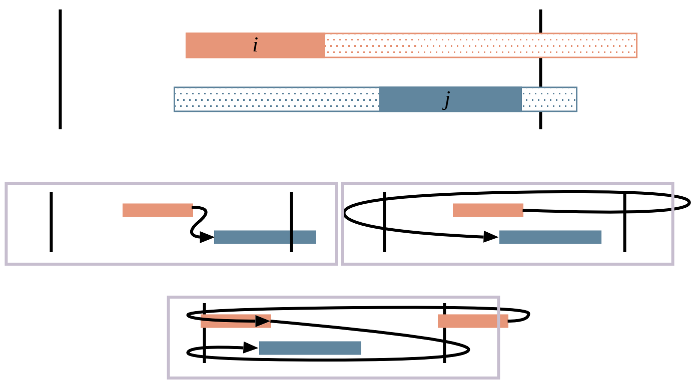
B_2 The turn may cross the cycle time limit twice but maybe only once. I.e:

$$\left(\underline{d_i} + t_i + r_{ij} > \overline{d_j}\right) \wedge \left(\underline{d_i} + t_i + r_{ij} - CT \leq \overline{d_j}\right) \wedge \left(\overline{d_i} + t_i + r_{ij} - CT > \underline{d_j}\right)$$



B_3 The turn may cross the cycle time border twice, once or not at all. I.e:

$$\left(\underline{d_i} + t_i + r_{ij} \leq \overline{d_j}\right) \wedge \left(\overline{d_i} + t_i + r_{ij} - CT > \underline{d_j}\right)$$



In the constraints given below we will *not* distinguish between these three sub-cases but treat them collectively as a single case $\mathcal{B}$ which will simplify the presentation of the model. In a practical implementation it *does* make sense to distinguish between them since we need to introduce two booleans per possible turn only in the B_3 case which is very rare.

5 Model constraints and objective

The cases labelled A_0 through A_2 above are all, if used as turns in a solution, determined to cross the cycle time limit either once, twice or not at all. The cases labelled B_i on the other hand are indeterminate and will be collectively encoded using the two boolean decision variables C_{ij} and C'_{ij}. To be able to treat the

A and B cases separately we will define four mutually exclusive subsets of the possible turns.

Let

$$\mathcal{A}_0 = \{\langle i,j\rangle \mid (0 < i,j \leq n) \wedge \overline{d_i} + t_i + r_{ij} \leq \underline{d_j}\}$$
$$\mathcal{A}_1 = \{\langle i,j\rangle \mid (0 < i,j \leq n) \wedge \left(\underline{d_i} + t_i + r_{ij} > \overline{d_j}\right) \wedge \left(\overline{d_i} + t_i + r_{ij} - CT \leq \underline{d_j}\right)\}$$
$$\mathcal{A}_2 = \{\langle i,j\rangle \mid (0 < i,j \leq n) \wedge \left(\underline{d_i} + t_i + r_{ij} - CT > \overline{d_j}\right)\}$$

and

$$\mathcal{B} = \{\langle i,j\rangle \mid 0 < i,j \leq n\} \setminus (\mathcal{A}_0 \cup \mathcal{A}_1 \cup \mathcal{A}_2)$$

We will need to explicitly represent the decision variables only for the case $\mathcal{B}$. Observe also that $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{B}$ have no elements in common.

Since the main objective of the model is to minimise the number of vehicles used by a solution and this corresponds exactly to the number of vehicles turned over the cycle time limit, the objective function will treat each of these cases (except $\mathcal{A}_0$ which can never contribute to the cost) separately. We also introduce a term in the cost function which penalises the use of additional vehicles for transports that do not need them. These are in most cases necessary to balance the flow of the model but should be avoided if possible. The penalty is weighted by the (temporal) length t_i of the transport and a factor p_i specific to each transport.

This factor should in most cases be smaller than 1 to give the number of vehicles the appropriate influence on the total cost. Potential *passive transports* (with vehicle demand $\underline{S_i} = 0$) will generally have a larger factor p_i than those in which a vehicle relocation is accompanying an existing transport.

Minimise

$$\Sigma_{\langle i,j\rangle \in \mathcal{A}_1} X_{ij} + \Sigma_{\langle i,j\rangle \in \mathcal{A}_2} 2X_{ij} + \Sigma_{\langle i,j\rangle \in \mathcal{B}} \left(Y_{ij} + Y'_{ij}\right) + \Sigma_{0<i\leq n} E_i p_i t_i$$

subject to

1. The number of vehicles turned *from* transport i is equal to the number used by it

$$\forall i((\Sigma_{j\in\{j|ld_i=lo_j\}} X_{ij}) - S_i = 0)$$

and the number of vehicles turned *to* transport j is equal to the number used by it

$$\forall j((\Sigma_{i\in\{ki|ld_i=lo_j\}} X_{ij}) - S_j = 0)$$

The candidate turns are chosen such that the destination ld_i of the source transport i and the origin lo_j of the sink j is identical. One way to relax this condition somewhat is described in section 6 below.

2. Turn time constraints.

$$\begin{cases} d_j - d_i + CT\ C_{ij} + CT\ C'_{ij} > t_i + r_{ij} \\ X_{ij} - Y_{ij} + M\ C_{ij} \leq M \\ X_{ij} - Y'_{ij} + M\ C'_{ij} \leq M \end{cases} \qquad \forall i, j\ (\langle i, j \rangle \in \mathcal{B})$$

and

$$S_i - E_i = \underline{S_i} \qquad \forall i$$

3. C_{ij}, C'_{ij} *boolean*, $C'_{ij} \leq C_{ij}$
4. S_i, X_{ij} (implicitly) *integer*
5. Variable bounds $\underline{d_i} \leq d_i \leq \overline{d_i}$, $\underline{S_i} \leq S_i \leq \overline{S_i}$, $\underline{X_{ij}} \leq X_{ij} \leq \overline{X_{ij}}$ for $\forall ij$

5.1 Constraint notes

The flow (conservation) constraints (1) ensure that each transport is supplied with as many vehicles as it needs and that the flow is balanced. To ensure that this is always possible we need to introduce a “sufficiently large” set of “potential” passive transports into the problem.

How this is done in general is not further discussed in this paper. However a straight forward heuristic to introduce additional such transports of a fixed maximum duration is outlined in section 6 below.

The turn time constraints (2) and their use of the boolean variables (3) are the core of the model. Note that $C_{ij} = C'_{ij} = 0$ if and only if $d_i + t_i + r_{ij} \leq d_j$, that $C_{ij} = 1 > C'_{ij}$ if and only if $d_i + t_i + r_{ij} - CT \leq d_j$ and finally that $C_{ij} = C'_{ij} = 1$ if and only if $d_i + t_i + r_{ij} - CT > d_j$ corresponding exactly to the three A-cases above. Note also that unnecessarily assigning 1 to C_{ij} while $X_{ij} > 0$ will be penalised by forcing Y_{ij} to become equal to X_{ij} and similarly for C'_{ij} and Y'_{ij}.

E_i is defined by the equation $S_i - E_i = \underline{S_i}$ to be the *excess* number of vehicles travelling with transport i. The requirement that $C'_{ij} \leq C_{ij}$ removes an obvious symmetry from the model. In practise the effect of this constraint is minor since the case where both booleans are needed, almost never occurs.

A key feature of the model and the main reason that it scales relatively well in practise is that the integrality constraints on S_i, and X_{ij} (4) need not be enforced by the solver. In each leaf in the search tree branching on the boolean variables C_{ij} and C'_{ij} the part of the coefficient matrix involving these variables will be a pure minimal cost flow). The same is obviously not the case for the part of the coefficient matrix involving the departure time variables d_i but since these variables are related to the decision variables S_i and X_{ij} only through the booleans (C_{ij}, C'_{ij}), each assignment of the d_i that is consistent with a complete (integral) assignment of the booleans will also be consistent with the optimal assignment to the decision variables S_i and X_{ij}.

This means that the optimal solution to the problem obtained by relaxing the integrality constraints on S_i and X_{ij} (but not on C_{ij} and C'_{ij}) will also be an optimal solution to the original problem.

5.2 Minimising deviation from a given timetable

In a setting where we start with a given time table and relax the departure and arrival times to allow a reduction of the total vehicle requirement of the schedule we may also want to minimise the deviation from the original time table starting from one in which the vehicle cost has been minimised. The following supplementary optimisation step can then be used for a given solution in which the Y_{ij}, Y'_{ij}, C_{ij} and C'_{ij} variables have have been determined. Let d_i^{orig} denote the original (unrelaxed) departure times and let the corresponding d_i variables have the same bounds as in section 5. Then introduce the following additional variables:

w_i	the amount of time that train i is changed with respect to the original time table.
w_i^-	the amount of time that the departure of train i is moved earlier
w_i^+	the amount of time that the departure of train i is moved later

Minimise

$$\sum_{0<i\leq N} w_i$$

subject to

1. $\forall i(d_i + w_i^- \geq d_i^{orig})$ determining the amount of time that transport i is early
2. $\forall i(d_i - w_i^+ \leq d_i^{orig})$ determining the amount of time that transport i is delayed
3. $w_i^+ + w_i^- - w_i = 0$ relating positive, negative and absolute movement of departure of train i
4. $\forall i, j(d_i - d_j \leq 2M - M * Y_{ij} - M * Y'_{ij} + CT * C_{ij} + CT * C'_{ij} - t_i - r_{ij})$ enforcing the turns of a previously determined solution.

In principle it should be possible to combine these two models into a third one weighing departure time deviation against vehicle cost but doing so in the straightforward way breaks the clean separation of the discrete decision variables X_{ij} and S_i on the one hand and the continuous departure time variables d_i on the other. In practise such a model does not scale at all well. In addition it would probably be very difficult in practise to determine suitable weights for the deviation variables w_i in the combined cost function. In the empirical results reported below the result of minimising the deviation for each given solution to the main problem is given in the deviation column.

6 System generated service trains

By changing the flow equations (1) in the above model, we can capture the introduction of (additional) passive transports (service trains), to reduce the overall need of vehicles somewhat.

Let $\text{dist}(l_1, l_2)$ be a function defining geographical distance between locations l_1 and l_2 and m a limit on the distance traversed by an (additional) passive transport. Replace the flow equations in the model above with:

$$\forall i((\Sigma_{j\in\{j|\text{dist}(ld_i,lo_j)\leq m\}}X_{ij}) - S_i = 0)$$

$$\forall j((\Sigma_{i\in\{i|\text{dist}(ld_i,lo_j)\leq m\}}X_{ij}) - S_j = 0)$$

The turn time r_{ij} for such turns i, j must be adapted to reflect the additional time taken to relocate the vehicle and the cost function extended with a term that reflects the cost for driving a service train:

$$\Sigma_{\langle i,j\rangle\in C}d_{ij}X_{ij}$$

where $C = \{\langle i, j\rangle\,|\text{dist}(lo_i, ld_j) > 0\}$ and d_{ij} is a cost factor that may or may not reflect the actual distance between ld_i and lo_j. In the experiments below m was set to one hour and each d_{ij} to a factor corresponding to $p_i t_i$ for a preplanned passive transport of duration t_i of one hour which in turn was twice of that of a vehicle accompanying an active transport of that duration.

The rest of this extended model is identical to that of section 5.

7 Empirical results

The following performance results have all been produced using data extracted from production data of the largest Swedish rail freight company Green Cargo. The case contains almost all transports handled by the most common type of vehicle in use, the electrical RC locomotive, and covers a full week. The problems solved below were generated by introducing a fixed amount of slack for each departure time in the production plan.

The number of transports in each of the generated problems is 1304. This includes a small number of statically generated potential passive transports of vehicles used to balance the flow in the network. In general relocation of vehicles are performed either as *accompanying transports* travelling with "real" transports or as *passive transports* (deadheads) where the vehicle travels by itself through the network. In both these cases the number of vehicles travelling with a scheduled transport exceeds the minimum required to perform the transport.

In the solutions reported below, *accompanying transports* has been freely introduced (though penalised) and moved around between transports that allow them. Passive transports on the other hand are eliminated wherever that leads to an improved objective. Dynamically generated new passive transports (as section 6) are introduced only in a separate set of problems and even then they are limited to a maximum traversal time of 60 minutes (while the average transport time is about four hours). Allowing additional passive transports in this way typically reduces the number of vehicles somewhat but introduce the need to schedule additional tasks on the infrastructure resources.

Note that introducing slack uniformly is not completely realistic. In reality customer requirements or limits on infrastructure capacity may not allow free rescheduling of the transports within their time windows. To some extent this can be improved by introducing individual slacks for each transport and weighted binary relations between arrival and departure events that encode e.g. transfers of cars and cargo. In the performance results reported here, no such additional constraints were used. Nevertheless a production version the software used to generate these problems is currently in use at Green Cargo in their planning of locomotives.

The tables below reports for each slack size (in minutes) the number of booleans needed to encode the turn time constraints which should give a rough indication of the MIP size. Furthermore the tables reports properties of the optimal solution found in terms of the number of vehicles, the total amount of "accompanying" and (additional) "passive" time in minutes. Performance results include the number of nodes, iterations and runtime in seconds as reported by CPLEX 9 for each slack size. The runtime reported is those reported by CPLEX on an 2.4 GHz Pentium 4 processor using about 2 GB of main memory. For the larger cases caching the node tree to disk was used whenever it became larger than the main memory. The strategy used was the default branch and bound (cut) heuristic of CPLEX 9 [19].

Once the optimal solution for the locomotive turnrounds has been found a new timetable is generated minimising the sum of deviations from the original timetable. This problem is linear and no performance results of these runs are given. The resulting deviation (in minutes) is given in each table to give an indication of how much the original time table had to be changed to achieve the corresponding improvement of the main objective.

The result reported in table 1 is without additional passive transports as outlined in section 5 while those in table 2 were obtained using the method outlined in section 6 to generate additional passive transports where this improved the objective.

Table 1. Without additional passive transports

Slack minutes	Booleans	Vehicles	Accompanying minutes	Deviation minutes	Nodes	Iterations	Runtime h:mm:ss
± 0	-	117	50835	-	-	6803 (0)	0:00:05
± 15	1027	116	50206	5107	0	7208	0:00:07
± 30	1995	112	51107	13763	50 (1)	9056	0:00:54
± 45	2836	105	51177	20841	40	10158	0:01:19
± 60	3913	99	49402	35651	714 (251)	28565	0:08:22
± 75	4930	97	49411	48486	21101 (3547)	630926	1:53:40
± 90	5876	90	50385	69067	156441 (74849)	9245253	23:22:43

In both cases the ± 0 case is for network solving, not MIP. The result reported for the ± 90 cases was obtained with a integrality gap of 0.05% which guaranties a

Table 2. Allowing short additional passive transports

Slack minutes	Booleans	Vehicles	Passive minutes	Accompanying minutes	Deviation minutes	Nodes	Iterations	Runtime h:mm:ss
± 0	-	116	276	49006	-	-	7105 (0)	0:00:07
± 15	1127	113	247	50459	5274	0	7182	0:00:08
± 30	2178	110	247	50429	15067	50	9323	0:01:06
± 45	3132	104	116	51354	23456	50 (40)	10624	0:01:36
± 60	4346	98	218	49264	40258	647 (435)	31029	0:08:53
± 75	5456	95	203	50185	5541	14506 (1818)	403299	1:15:32
± 90	6538	88	612	52293	64147	182364 (60220)	9870645	26:04:30

solution with an optimal number of vehicles and a solution within approximately 3 days of passive transport time from the optimum. Using the 0.01% default gap of CPLEX 9 gave a runtime about four times as long and only a marginal improvement of the objective.

8 Conclusions

The model presented generalises the m-TSPTW problem to multiple required and upper bounded optional visits to each location. It is shown how it can be applied to an important practical problem in rail transportation and that cases of realistic size can be solved using a standard (though state-of-the-art) commercial solver.

Innovative features of the model include the use of boolean variables to separate the integer and continuous parts of the problem and maintaining the flow character of the integer part of the problem for each complete assignment of the booleans.

Application of the model produces a modified timetable which accommodates the requirements for an efficient locomotive turn-round plan. The practical usefulness of the model and its scalability is demonstrated on a set of problems derived from a real case in the Swedish rail freight industry.

Significant savings can be realised for a uniform fleet of locomotives, in terms of locomotives planned, by utilising the presented method.

Future work would include identifying methods to compute better lower bounds and possibly to investigate if decomposition methods traditionally used for VRPTW type problems can be used to further improve the performance of solvers using the model. Multicommodity variants of the model may also be of interest.

References

1. Desrosiers, J., Dumas, Y., Solomon, M., Soumis, F.: Time Constrained Routing and Scheduling. In: Network Routing. Volume 8 of Handbooks in Operations Research and Management Science. North-Holland (1995) 35–139

2. Drott, J., Hasselberg, E., Kohl, N., Kremer, M.: A planning system for locomotive scheduling. Technical report, Swedish State Railways, Stab Tågplanering, Stockholm, Sweden, and Carmen Systems AB (1997)
3. Solomon, M., Desrosiers, J.: Time window constrained routing and scheduling problems. Transportation Science **22** (1988) 1–13
4. Zwaneveld, P., Kroon, L., Romeijn, H., Salomon, M., Dauzère-Pérès, S., van Hoesel, S., Ambergen, H.: Routing trains through railway stations: Model formulation and algorithms. Transportation Science **30** (1996) 181–194
5. Ahuja, R.K., Liu, J.N., Orlin, James B.and Sharma, D., Shughart, L.A.: Solving real-life locomotive scheduling problems. Working Paper 4389-02, MIT Sloan (2002)
6. Mitrovic-Minic, S., Krishnamurti, R.: The multiple traveling salesman problem with time windows: Bounds for the minimum number of vehicles. Technical report, School of Computing Science, Simon Fraser University (2002)
7. Bektas, T.: The multiple traveling salesman problem: an overview of formulations and solution procedures. Omega (2006) 209–219
8. Cordeau, J.F., Desaulniers, G., Desrosiers, J., Solomon, M.M., Soumis, F.: Chapter 7: The VRP with Time Windows. In: The Vehicle Routing Problem [10]. SIAM Monographs on Discrete Mathematics and Applications,SIAM, Philadelpia, Pa. (2002)
9. Ball, M., Magnanti, T., Monma, C., Nemhauser, G., eds.: Network Routing. In Ball, M., Magnanti, T., Monma, C., Nemhauser, G., eds.: Handbooks in Operations Research and Management Science. Volume 8., North-Holland (1995)
10. Toth, P., Vigo, D., eds.: The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematis and Applications. SIAM (2002)
11. Savelsbergh, M.: Local search in routing problems with time windows. Annals of Operations Research **4** (1985) 285–305
12. Solomon, M.: On the worst-case performance of some heuristics for the vehicle routing and scheduling problem with time window constraints. Networks **16** (1986) 161–174
13. Solomon, M.: Algorithms for the vehicle routing and scheduling problem with time window constraints. Operations Research **35** (1987) 254–265
14. Desrochers, M., Lenstra, J., Savelsbergh, M., Soumis, F.: Vehicle routing with time windows: optimization and approximation. In Golden, B., Assad, A., eds.: Vehicle Routing: Methods and Studies, Amsterdam, North Holland (1988) 65–84
15. Desrochers, M., Desrosiers, J., Solomon, M.: Using column generation to solve the vehicle routing problem with time windows. Operations Research **18** (1990) 411–419
16. Desrochers, M., Desrosiers, J., Solomon, M.: A new optimization algorithm for the vehicle routing problem with time windows. Operations Research **40** (1992) 342–354
17. Kohl, N., Madsen, O.B.G.: An optimization algorithm for the vehicle routing problem with time windows based on lagrangian relaxation. Operations Research **45** (1997) 395–406
18. Potvin, J.Y., Rousseau, J.M.: An exchange heuristic for routing problems with time windows. Journal of the Operational Research Society **46** (1995) 1433–1446
19. ILOG: ILOG CPLEX Callable Library 9.0 Reference Manual. ILOG (2003)