

Improving Timetable Quality in Scheduled Transit Networks

Valérie Guihaire^{1,2} and Jin-Kao Hao²

¹ HEURISIS, 8 rue Le Notre, 49100 Angers, France
vguihaire@perinfo.eu

² LERIA, Université d'Angers, 2 boulevard Lavoisier, 49045 Angers, France
hao@info.univ-angers.fr

Abstract. This work deals with an original problem with regard to the traditionally sequential planning process in public transit networks. This problem aims at modifying the network's timetables without rendering the vehicle and driver schedules obsolete. The objective is to improve the quality of service for passengers through number and quality of transfers. This approach goes in the opposite direction compared to the usual approach which schedules resources once timetables are set. We propose a model and a solution method based on tabu search and a neighborhood specifically developed. Experiments are led on five instances related to a real transit network. Important gains are obtained on the considered case study, allowing for better mobility of users inside the network and on the intermodal level.

Keywords: Transit network, synchronization, timetabling.

1 Introduction

We address the problem of modifying the timetables of a running (timetabled and scheduled) transit network to improve its quality of service through transfers while maintaining the vehicle and driver trip assignments. This problem is met by transit planners in practice but is counter-intuitive regarding the treatment of transit planning in the literature, according to which lines timetabling is operated first and resource scheduling is only operated next [1]. We call this problem Schedules-based re-Timetabling (SbrT) in the rest of this paper.

In a running transit network, i.e. in which all the planning steps have been completed, the context and environment continue to evolve. For instance, changes in the demand or in the offer of coordinated modes can occur, leading to a situation in which changes in the timetable could benefit to the quality of service. The traditional process would be to define and apply changes to the timetables and then restart the resource planning process. However, planning vehicle and driver schedules is an extremely complex task that needs to take into account a great number of constraints, objectives and parameters. This can be too demanding a task to even consider applying timetable changes in the first place. Therefore, to deal with this problem, we propose a model and a method to boost

the service quality of transit networks while leaving unchanged the sequences of trips assigned to vehicles and drivers.

With regard to our problem here, one study is noteworthy, although the SbrT is not its specific purpose. In [4], Jansen et al. propose to improve the quality and number of transfers in a transit network through shifts in the line starting times. This implies that all the runs of a given line are shifted by the same value and headways (frequencies) are fixed. It takes no account of resource usage, implying that this method is rather directed toward urban transit networks, and assumes that interlining is not practiced, so that vehicle allocation is not impacted by the changes. In extra-urban networks, the risk is to obtain a timetable that is not compatible with the current vehicle assignments or worse, with the available number of resources. Another approach to reorganize the timetables without requiring additional vehicles can be to include minimizing the fleet size to the objectives [6]. However, the risk is high that the resource schedules will need a total reconstruction, thus not answering our problem here.

2 Problem Description

The problem consists in slightly modifying the timetables so as to improve both quantity and quality of the transfer opportunities, while respecting the trip sequences assigned to vehicles as well as to drivers. Results for this problem are new timetables for the lines.

In order to better understand the scheduling part of the planning process, let us first present the constitution of a resource trip assignment. Such an assignment (see Figure 1) is a sequence of trips, deadheads (to and from the depot as well as between trips), turnaround time and stopping or pausing time according to the type of resource (vehicle or driver).

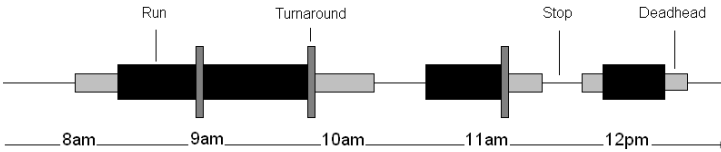


Fig. 1. Vehicle trip assignment

2.1 Objective: Quality and Quantity of Transfer Opportunities

The objective consists in creating numerous and high-quality transfer opportunities for users. A non-captive users policy is applied, meaning a transfer opportunity happens only when the transfer waiting time is inside a predefined interval. A minimal, ideal and maximal waiting time are all specified for each kind of transfer, as well as a level of importance. The quality of the transfer opportunity is proportional to the closeness with the ideal waiting time. Transfers among lines of the network as well as with external modes are considered here.

2.2 Constraints

A set of constraints applies to the timetable:

Maximal Shift. The current timetable has very likely been designed to comply with the other main criterion for quality of service: headways. Headway determination permits to provide a regular service for one, and second, to adapt to demand peaks and prevent overcrowding. In order to preserve this adaptation to the demand fluctuations, we only allow a shifting value comprised inside a definite interval.

Driver Trip Assignment. The existing driver assignment must be compatible with the new timetable. Concretely, each driver must be able to serve the same sequence of line runs, turnarounds, deadheads and pausing times, without any of these elements overlapping. In terms of duration, only pausing times may vary. To respect social issues linked to pause duration while introducing some flexibility in the model, we allow each waiting time to vary inside a pre-defined interval. The human planner thus provides a list of intervals (in minutes) as data for the problem. For instance, if the list is: $[0-15];[16-30];[31-45];[46-N]$, then the duration of an initial 28-minute long pause may vary between 16 and 30 minutes in the final solution.

Vehicle Trip Assignment. The existing vehicle assignment must be compatible with the new timetable. Concretely, each vehicle must be able to serve the same sequence of line runs, turnarounds, deadheads and stopping times, without any of these elements overlapping. In terms of duration, only stopping times may vary.

3 Problem Formulation

We define a model allowing for a high level of flexibility, in which the set of decision variables matches the starting time of all the runs in the network. Additionally, a set of state variables is kept up to date to facilitate the computation of costs linked to transfers. This latter set includes the stopping times of the runs at intermediate stops along the runs. To compute the value of these variables, we consider fixed running times by runs along the search process. This is acceptable due to the short shifting span allowed for each starting time.

3.1 Notations

Let \mathcal{L} be the set of lines, \mathcal{R} the set of runs, \mathcal{S} the set of stops, \mathcal{V} the set of vehicles, \mathcal{D} the set of drivers, \mathcal{T} the set of transfers, and \mathcal{H} the planning horizon.

- For each line $l \in \mathcal{L}$:
 - $\mathcal{R}_l \subset \mathcal{R}$: set of runs served.
- For each run $r \in \mathcal{R}$:
 - $\mathcal{S}_r \subset \mathcal{S}$: set of served stops.
 - s_r^α, s_r^ω : first and last stop.

- $h_{r,s}^{\leftarrow}, h_{r,s}^{\rightarrow}$: arriving and leaving time for $s \in \mathcal{S}_r$ in the initial timetable.
 - $h(r)$: initial starting time for run r ($h(r) = h_{r,s_r^\alpha}^{\rightarrow}$).
 - v_r : vehicle assigned to r .
 - d_r : driver assigned to r .
 - r_v^{\rightarrow} : run following r in the schedule of vehicle $v = v_r$.
 - r_d^{\rightarrow} : run following r in the schedule of driver $d = d_r$.
- For each transfer $t = (l_1, s_1, l_2, s_2) \in \mathcal{T}$ taking place at stops $s_1, s_2 \in \mathcal{S}$ on lines $l_1, l_2 \in \mathcal{L}$:
- il_t : level of importance
 - $[wt_t], [wt_t], [wt_t]$: minimal, ideal and maximal transfer waiting time.
- $tth_{r,r'}$ and $ttd_{r,r'}$: turnaround time between two consecutive runs r and r' inside a vehicle and a driver trip assignment respectively.
- $dhv_{r,r'}$ and $dhd_{r,r'}$: deadhead time between two consecutive runs r and r' inside a vehicle and a driver trip assignment respectively. This deadhead can be direct between two runs, or divided in two parts when a stop at the depot is involved.
- $[dw_{r,r'}]$ and $[dw_{r,r'}]$: minimal and maximal duration for the pausing time between two consecutive line runs inside a driver trip assignment. It is deduced from the initial pausing time ($h(r') - h_{r,s_r^\omega}^{\leftarrow} - ttd_{r,r'} - dhd_{r,r'}$), and from the intervals for pause durations (Section 2.2).
- $[S]$: maximal time shift allowed for each run starting time.

Decision Variables and Values. The decision variables are the starting times $\pi(r) \in \mathcal{H}$ of each run $r \in \mathcal{R}$.

The set of state variables include the arriving and departing times of runs at stops along the routes. Let $\pi_{r,s}^{\leftarrow} \in \mathcal{H}$ and $\pi_{r,s}^{\rightarrow} \in \mathcal{H}, \forall r \in \mathcal{R}, \forall s \in \mathcal{S}_r$ represent these state variables. The decision variable $\pi(r)$ corresponds to state variable $\pi_{r,s_r^\alpha}^{\rightarrow}$.

A *configuration* σ is a complete assignment of values in \mathcal{H} to the set of decision variables.

3.2 Objective: Quality and Quantity of Transfer Opportunities

The cost function relative to transfers is a nonlinear function of the waiting time and favors the most heavily close-to-ideal waiting times. The cost incurred to the configuration is also weighed by the relative level of importance of the transfer. We compute the transfer cost for each couple of run for the arriving line and run for the departing line. Each gap belonging to the allowed interval means a transfer opportunity and generates an addition to the cost of the configuration. This is in the context of a maximization problem.

$$f(\sigma) = \sum_{t=(l_1, s_1, l_2, s_2)}^{t \in \mathcal{T}} \left[il_t * \sum_{\substack{r_1 \in \mathcal{R}_{l_1} \\ r_2 \in \mathcal{R}_{l_2}}} fTr(t, \pi_{r_1, s_1}^{\leftarrow}, \pi_{r_2, s_2}^{\rightarrow}) \right]$$

where

$$fTr(t, h_1 \in \mathcal{H}, h_2 \in \mathcal{H}) = \begin{cases} \text{increasing from 0.5 to 1 if} & \lfloor wt_t \rfloor < h_2 - h_1 < \lceil wt_t \rceil \\ \text{decreasing from 1 to 0 if} & \lceil wt_t \rceil < h_2 - h_1 < \lfloor wt_t \rfloor \\ 0 & \text{o.w.} \end{cases}$$

Only the costs related to transfers among lines of the network are presented here, but those related to intermodal transfers are computed in the same way and included in the objective function.

3.3 Constraints

Complete Assignment

$$\forall r \in \mathcal{R}, \pi(r) \in \mathcal{H} \quad (1)$$

Maximal Shift

$$\forall r \in \mathcal{R}, |\pi(r) - h(r)| \leq \lceil S \rceil \quad (2)$$

Driver Trip Assignment. Let $att(r, r') = \pi(r') - \pi_{r, s_r}^{\leftarrow} - dh_{d, r, r'} - tt_{d, r, r'}$ be the waiting time for the driver between consecutive runs r and r' in his service.

$$\forall d \in \mathcal{D}, \forall (r, r') \in \mathcal{R}^2, (r' = r_d^+) \Rightarrow (\lfloor dw_{r, r'} \rfloor \leq att(r, r') \leq \lceil dw_{r, r'} \rceil) \quad (3)$$

Vehicle Trip Assignment. Let $sta(r, r') = \pi(r') - \pi_{r, s_r}^{\leftarrow} - dh_{v, r, r'} - tt_{v, r, r'}$ be the stopping time of the vehicle between consecutive runs r and r' in its service.

$$\forall v \in \mathcal{V}, \forall (r, r') \in \mathcal{R}^2, (r' = r_v^+) \Rightarrow (sta(r, r') \geq 0) \quad (4)$$

4 Solution Approach

A Tabu Search method [3] is used. The initial solution is based on the timetables and resource (vehicle and driver) schedules from the running transit network. The search starts from this feasible solution and remains at all time inside the feasibility domain. A preprocessing phase is applied to reduce the domain of the variables. A neighborhood mechanism is then used to modify the departure times of the trips while preventing overlaps inside the resources schedules. Parameters for our Tabu Search implementation are exposed in the following sections. The output provided by the method consists in new line timetables for the transit network, with better quality with respect to the objective function, while the driver and vehicle trip sequences remain feasible.

4.1 Search Space and Preprocessing

The goal of the preprocessing phase is to reduce the domains of the variables by constraints propagation, and to compute what can be computed once and for all at the beginning of the process, so as to fasten the search.

Domain reduction - node consistency. Given the variables' initial domain \mathcal{H} and the maximal shift constraint, the size of the domain of each variable can be reduced to twice the size of the maximal shift.

$$\forall r \in \mathcal{R}, \pi(r) \in [h(r) - \lceil S \rceil; h(r) + \lceil S \rceil]$$

Preprocessing - arc consistency. In the model, constraints on the respect of the initial assignments are crucial. It is imperative to insure that none of the elements in these assignments overlaps its neighbors during the search. In order to prevent such a situation, we must consider the duration of each element of the assignments. Each of them has a fixed duration, except for stopping and pausing times. It is thus possible to compute the minimal time interval separating the starting time of a run from the starting time of the runs before and after it inside the vehicle and driver assignments. This interval constrains the domains of the variables along the search. It includes the duration of the run, the turnaround time, the deadheading time, and in the case of the driver schedule, the minimal pausing time. Additionally, with respect to the driver schedule, a maximal time interval is defined that takes into account the maximal pausing time allowed with respect to its initial value and the user-parameterized list of intervals.

4.2 Neighborhood - TripShift

We define a neighborhood mechanism, *TripShift*, to generate simple and fast moves while respecting the constraints. A move from *TripShift* modifies the value of a single variable (i.e. a trip). A neighbor solution is obtained by shifting its value by n minutes. n is chosen inside \mathbb{Z}^* and to respect the constraints, so that the search remains inside the feasibility domain.

Let $mv_{TShift} = (r, n)$ be a move from this neighborhood.

Let $\sigma = (t_1, t_2, \dots, t_{|\mathcal{R}|})$ be the current solution and $\sigma' = \sigma \circ mv_{TShift}$ a neighbor solution. Then, $\sigma' = (t'_1, t'_2, \dots, t'_{|\mathcal{R}|})$ obeys to the following description:

$$\forall r^* \in \mathcal{R}, \begin{cases} t'_{r^*} = t_{r^*} + n & \text{if } r^* = r \\ t'_{r^*} = t_{r^*} & \text{o.w.} \end{cases}$$

4.3 Parameters of Tabu Search for the SbrT

- The *initial solution* consists in the current line timetables, vehicle schedules and driver schedules.
- The *neighborhood* used is *TripShift*. It is fully explored at each iteration.

- For the sake of *diversification* and to prevent the search from stagnating, the selection of moves to apply is random among those of maximal cost.
- Each move remains inside the tabu list for a random number of iterations. This number, which determines the *tabu tenure* at each iteration, varies between two pre-defined bounds (Section 5.3).
- The *stop criterion* used is the computational time.

5 Experimentations and Numerical Results

5.1 Data Instances

To define data instances, we used the structure and timetables from a real extra-urban transit network, located in a French area called Loiret. The network serves three average-size cities and many small villages. It counts 50 oriented lines, 673 stops, 30 activities (trains, schools) in connection with the network, and 282 types of possible transfers. A type of transfer is a pair of lines crossing at a stop and among which a transfer possibility may thus be created. The timetable for a typical day has 318 runs. Three transport operators serve the network (on 8, 16 and 26 lines each). Concerning the transfer objective, 180 possibilities occur in the current network, for an objective function value of 947.462. Since resource usage data is confidential, we do not have access to the constitution of the vehicle and driver trip assignments defined by the transit operators. We used a classical approach to determine five sets of resource schedules, whose characteristics are presented in Table 1:

Table 1. Number of resources involved in each instance

	Instance1	Instance2	Instance3	Instance4	Instance5
Number of vehicles	91	91	91	91	91
Number of drivers	170	171	163	171	159

- the vehicle trip assignments are generated using an exact auction algorithm [2] depending on a weighing strategy; five sets of assignments are generated, with at least 30% difference among each pair of them (with respect to consecutive trips in the assignments).
- the driver trip assignments are generated based on the vehicle trip assignments using a classical bi-level driver scheduling algorithm [5].

5.2 Numerical Results

Our algorithm was coded in C++, compiled with VC++ 9.0, on a laptop equipped with Windows Vista, a 2.10 Ghz Intel(R) Core(TM)2 Duo CPU processor and 4Go RAM. For these tests, we used a computational time of 1 minute as unique stop criterion for our algorithm. A series of 30 tests was launched on each instance, given the non deterministic behavior of the tabu algorithm.

The tabu tenure is dynamic. The number of iterations during which each move remains in the list varies between the arbitrary values of 7 and 20 (see Section 5.3). The maximal allowed shift by run is 10 minutes (in each direction), a value which represents a compromise between size of the search space (which needs to be large enough to provide opportunities for improvement) and respect of the initial resources organisation (especially in terms of headway repartition).

Table 2 presents the number of transfer possibilities by data instance, in the initial timetable (row 2) and in the solution obtained by the algorithm (row 3, mean and standard deviation). Row 4 shows the percentage of improvement between the two solutions. Table 3 presents the results of the same series of tests, in terms of value of the objective function.

Table 2. Mean number (standard deviation) of transfer opportunities, and improvement compared to the initial solution

	Initial Solution	Tabu Search with TripShift	
		mean (s-d)	%
Instance 1	180	245.07 (0.25)	36.15
Instance 2		247 (0)	37.22
Instance 3		245.2 (0.41)	36.22
Instance 4		243 (0)	35.00
Instance 5		245.1 (0.31)	36.17

Table 3. Mean value (standard deviation) of the objective function, and improvement compared to the initial solution

	Initial Solution	Tabu Search with TripShift	
		mean (s-d)	%
Instance 1	947.462	1057.37 (0.12)	11.60
Instance 2		1061.63 (0)	12.05
Instance 3		1072.58 (0.08)	13.21
Instance 4		1047.14 (0)	10.52
Instance 5		1072.35 (0.06)	13.18

Those tables show that the algorithm brings a sizeable improvement to the initial solution, in terms of both criteria used. On average, it brings a 36.15% increase in the number of transfer possibilities, and a 12.11% improvement in terms of value of the objective function. This difference is explained by the additional consideration in the objective function of the quality of the new and existing transfer possibilities.

These test results show that the considered problem, going backwards in the traditional transit planning process, owns the potential for improvement in the level of quality of service. Furthermore, the proposed model offers a good flexibility for re-timetabling the network without condemning the resources schedules.

5.3 Discussion

Computational time. The computational time needed to attain the final solution is very brief. The algorithm reaches its best value after 19.3 seconds on average in the performed series of tests. This computational time is acceptable for the transit operators, considering that SbrT is a punctual operation that does not require real-time results.

Tabu Tenure. We chose to use a variable number of iterations of presence in the tabu list for each generated move. This number is randomly chosen inside a pre-defined interval. A set of different intervals was tested, and

the impact of this parameter revealed to be very weak. Table 4 shows the results obtained with the extreme values tested for the interval, the upper bound of the largest interval matching approximately half the average size of the neighborhood. The method slightly benefitted from the increase in the number of iterations of presence. Complementarily, Table 5 shows that when moves remain in the tabu list between 500 and 1000 iterations, the complete neighborhood exploration (with associated evaluations) is on average, 2.65 times longer than when moves remain in the list for 7 to 20 iterations.

Table 4. Objective function value by instance and tabu list size

	TripShift	
	[7-20]	[500-1000]
Instance 1	1057.37	1057.52
Instance 2	1061.63	1062.62
Instance 3	1072.58	1072.65
Instance 4	1047.14	1047.25
Instance 5	1072.35	1072.37

Table 5. Average time (in seconds) needed for full exploration of the neighborhood by instance and tabu list size

	TripShift	
	[7-20]	[500-1000]
Instance 1	0.012	0.031
Instance 2	0.010	0.028
Instance 3	0.013	0.033
Instance 4	0.011	0.030
Instance 5	0.012	0.032

We observe that the results are of better quality using the interval [500-1000] compared with interval [7-20] in all five cases, however the observed variation is small. At the same time, larger values for the number of iterations of presence slows the search down, since a great number of verifications needs to be performed. In view of the combination of these two effects, we chose to keep the [7-20] interval in our method, to prevent negative effects in larger networks.

Upper Bound Computation. We tried to compute an upper bound for the problem, by taking advantage of the maximal shift constraint, which restricts the domain of each variable. We determined the set of all the transfer opportunities that could be generated independently (ie when the domains of both variables - runs implied in the transfer - encompassed values that fit into the allowed waiting time interval for the transfer). For each one, we computed the best cost that could be achieved. The upper bound obtained consists in 399 transfer opportunities and a 1332.11 value for the objective function. The drawback of this method is that it does not take into account the fact that a run can be involved in many transfer opportunities and assigns a potentially different value to the run for each transfer opportunity, providing unrealistic goals for the algorithm. This explains the large difference between the bound and the results actually achieved.

Neighborhood. We developed and tested a second neighborhood, which consists in shifting at once all the elements of a driver trip assignment by the same value, while preserving the feasibility of the vehicle trip assignments. This neighborhood proved to define a somewhat too small search space, preserving most of the initial schedules to the expense of fewer improvement

possibilities. *TripShift* defines in comparison an interesting compromise between the improvement of the quality of service and the preservation of the existing resource usage.

6 Conclusion

An original problem was considered here, that goes backwards in the traditional transit planning process: the Scheduled-based re-Timetabling problem. We define a model based on the assignment of new starting times to all the runs in the timetables. Major constraints impose that the existing vehicle and driver trip assignments are preserved, and that their elements do not overlap despite the shifts in their starting times. The advantages of this model are its flexibility and its high level of preservation of the initial schedules (both in terms of number of resources needed and social acceptability).

We developed an adapted neighborhood mechanism that we integrated inside a method relying on tabu search. Tests were carried out on five generated instances based on a real network, and allowed a sensible improvement in terms of transfer opportunities on the case study. This combination of model and method shows the potential of the SbrT on a running schedule in terms of quality of service for transit users.

Acknowledgments. We wish to thank the reviewers for their helpful comments.

References

1. Ceder, A., Wilson, N.H.M.: Bus Network Design. *Transportation Research Part B* 20(4), 331–344 (1986)
2. Freling, R., Wagelmans, A., Paixao, J.M.P.: Models and algorithms for single-depot vehicle scheduling. *Transportation Science* 35(2), 165–180 (2001)
3. Glover, F.: Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research* 13(5), 533–549 (1986)
4. Jansen, L.N., Pedersen, M.B., Nielsen, O.A.: Minimizing passenger transfer times in public transport timetables. In: 7th Conference of the Hong Kong Society for Transportation Studies: Transportation in the Information Age, pp. 229–239 (2002)
5. Ceder, A.: Crew scheduling. In: *Public Transit Planning and Operation: Theory, Modeling and Practice*, pp. 279–318. Butterworth-Heinemann (2007)
6. Guihaire, V., Hao, J.K.: Transit Network Re-timetabling and Vehicle Scheduling. In: *MCO*, vol. 14, pp. 135–144. Springer, Heidelberg (2008)