# Locating optimal timetables and vehicle schedules in a transit line

**Juan A. Mesa · Francisco A. Ortega · Miguel A. Pozo**

**Abstract** This paper deals with the Transit Network Timetabling and Scheduling Problem (TNTSP) in a public transit line. The TNTSP aims at determining optimal timetables for each line in a transit network by establishing departure and arrival times of each vehicle at each station. We assume that customers know departure times of line runs offered by the system. However, each user, traveling after or before than their desired travel time, will give rise to an inconvenience cost, or a penalty cost if that user cannot be served according to the scheduled timetable. The provided formulation allocates each user to the best possible timetable considering capacity constraints. The problem is formulated using a p-median based approach and solved using a clustering technique. Computational results that show useful applications of this methodology are also included.

**Keywords** Timetabling · Vehicle scheduling · Schedule delay · Location-allocation

## 1 Introduction

The Transit Network Timetabling and Scheduling Problem (TNTSP) aims at determining optimal timetables for each line in a transit network by establishing departure and arrival times of each vehicle at each station. The TNTSP is based on the following general input: An infrastructure of a transport system described by a node set (network stations) and an edge set (tracks between adjacent stations), a trip demand matrix between pairs of nodes of

J.A. Mesa
Department of Applied Mathematics II, University of Seville, Seville, Spain
e-mail: jmesa@us.es

F.A. Ortega
Department of Applied Mathematics I, University of Seville, Seville, Spain
e-mail: riejos@us.es

M.A. Pozo (✉)
Department of Statistics and Operational Research, University of Seville, Seville, Spain
e-mail: miguelpozo@us.es

the infrastructure, a set of transit lines with associated frequencies which have already been determined in order to satisfy such trip demand and, finally, a vehicle fleet with specific characteristics. The objective of the TNTSP consists of finding arrival and departure times of each vehicle at each station such that the demand satisfaction, required fleet size and vehicle capacities can be optimized/bounded.

The TNTSP integrates two stages of the global transit planning process, that is usually divided in a sequence of five steps (Ceder and Wilson 1986): line planning, frequencies setting, timetabling, vehicle scheduling and crew rostering. Even separately, solving any of these problems implies a challenge in terms of computational complexity (Magnanti and Wong 1984; Quak 2003); however, a considerable amount of work points out the integration of several of these planning stages at the same time (Guihaire and Hao 2008), in order to achieve interaction and feedback in the process as well as better quality results.

The TNTSP might be solved under different time contexts: for the strategic context (e.g. promoting the extension of the network infrastructure by means of including new line segments between stations), in the tactical planning (where regular vehicle scheduling for a given frequency—daily or weekly—is determined) and in real-time scenarios (where dispatchers must manage the traffic by making optimal decision about which vehicles to stop and where, according to data about vehicle positions that are continuously subject to an updating process).

Timetabling is the process of implementing the service frequency on each fixed route, providing arrival/departure times at each station. Timetables can be periodic (e.g. Liebchen et al. 2010) if they are repeated in time intervals. Although periodicity makes timetables easy-to-remember, non-periodic timetables (e.g. de Palma and Lindsey 2001) can be implemented to more adequately fit within the current time-dependent demand pattern. Two different types of infrastructures can be analyzed to implement timetables: a single corridor (see e.g. Brannlund et al. 1998; Caprara et al. 2002; Zhou and Zhong 2007) or an entire network including transfers (see e.g. Caprara et al. 2006). In both scenarios, the main objectives that are usually taken into consideration are addressed at maximizing the transfer synchronization, in order to minimize waiting times at transfers (e.g. Guo and Wilson 2011) and minimizing the *schedule delay* (Small 1982). The concept of schedule delay arises with the fact that arriving early is likely to involve some wasted time while for most users, arriving late has more severe repercussions. In this way, timetabling can be seen as a p-median problem (Hakimi 1964) where the objective is to minimize the time/distance between passenger desired departure times and actual ones. Narrow headways lead users to arrive randomly to stations giving rise to a waiting cost for the user. On the other hand, when headways are wider, customers tend to strictly follow timetables, arriving only few minutes before the departure time. This last situation does not provoke a waiting cost for the user but an inconvenience cost to fit desired travel time to actual timetables (Grosfeld-Nir and Bookbinder 1995; Fosgerau 2009).

The vehicle scheduling problem consists of allocating a set of vehicles to a set of timetables, taking into consideration some practical requirements like depot/s location/s, vehicle features (speed, size, maintenance costs, fuel costs) and other extensions. An optimal schedule minimizes the fleet size as well as operational costs. Vehicle scheduling problems have received considerable attention in the literature considering a single corridor (see e.g. Higgins et al. 1996; Oliveira and Smith 2000; Zhou and Zhong 2005) and an entire network (Cai and Goh 1994; Chew et al. 2001). A recent overview in vehicle scheduling problem has been provided by Bunte and Kliewer (2009).

Dealing with the scheduling problem separately from the timetabling problem implies that fleet size requirements cannot be bounded, only minimized. Moreover, if timetables are too plentiful for a small fleet size, a feasible vehicle schedule could not even be guaranteed. In this sense, little work has been developed searching for the integration of timetabling and vehicle scheduling (Guihaire and Hao 2010). One of the first approaches that integrates both timetabling and vehicle scheduling problem is the one provided by Chakroborty et al. (2001), where periodic timetables are determined in order to minimize waiting times at transfer stations as well as the required fleet size. Castelli et al. (2004) deal with non periodic timetables assuming that routes, means of transport and quality of services are fixed in advance. The operator's main objective is to minimize their costs, while serving, at the same time, as many customers as possible. The solution procedure schedules a single line at a time, possibly re-optimizing or correcting the previous decisions at each step. Chang and Chung (2005) consider a single, one-way track train timetabling problem for a rapid transit system. Liu and Shen (2007) integrate timetabling and vehicle scheduling by using a bilevel formulation, where the upper level is referred to the operators' objectives while the lower level reflects the user interests. Guihaire and Hao (2010) develop an integrated approach without considering periodic timetables but they include evenness of the line headways as one of the optimization criterium. Finally, Cadarso and Marín (2012) integrate railway timetabling and scheduling updating the frequencies known from the railway line planning problem. Frequencies for some arcs are maintained in a determined frequency window as well as headways that are maintained for every train line in the network.

This paper deals with a customer-oriented timetabling-scheduling model applied to the setting of a public transit corridor with two lines (one per each direction). A pre-set number of line runs (vehicle expeditions along a line) will be located in each line along the time horizon, under constraints determined by the fleet size and the maximum number of allowed line runs. Additionally, we will assume non-periodic timetables that will be known by users in advance. Therefore, timetables will be determined in order to optimally serve the existing demand by considering that those individuals, traveling later than or ahead of their desired travel time, will suffer a user's inconvenience cost. Customers are supposed to suffer an inconvenience cost, if their preferred pickup/delivery times vary from the actual ones, and a penalty cost, if the requests are not served in a time window. In this sense, the user behavior, explained by means of inconvenience costs, is motivated from the contexts of the vehicle routing problem with time windows (Cordeau et al. 2007b) and transportation on demand (Cordeau et al. 2007a), where disaggregated demand and the human factor acquire great significance. To the best of our knowledge, inconvenience costs have not been considered in the TNTSP literature. Additionally, since we are dealing with a public transportation where users choose freely in which vehicle they want to board, the situation where users are freely allocated to vehicles, ignoring capacity levels, will be studied.

The remainder of this article is structured as follows. Section 2 is devoted to describing the context where the model is formulated. In Sect. 3 a variation of the p-median problem is proposed to optimally locate a number of line runs ensuring vehicle schedules. In order to decrease the size of the problem, a clustering algorithm is described in Sect. 4. Computational experiments are provided in Sect. 5 in order to show the usefulness and applicability of this methodology. The extension of this methodology to transportation networks composed of several transit lines is described in Sect. 6. Finally, conclusions and future lines of research will be described in Sect. 7.

## 2 Problem description

### 2.1 Infrastructure

Let $L$ be a transit corridor consisting of a node set $S$ (stations) and an edge set $E$ (tracks). Let $l \in \mathcal{L}$ be the set of feasible lines in $L$. For the setting considered in this paper, $\mathcal{L}$ only will contain two lines; that is, $\mathcal{L} = \{1, 2\}$, where $l = 1$ is a directed transit line running along $L$ and $l = 2$ is also running along $L$ but in the opposite direction. We denote by $\langle s, l\rangle$ the station in position $s \in \{1, \dots, |S_l|\}$ belonging to the set of stations $S_l \subseteq S$ of a given line $l \in \mathcal{L}$. Additionally, let $E_l$ be the subset of $E$ that contains all edges used by line $l$. Each identical vehicle will operate along $L$ during a time horizon that will be discretized into a set of time slots $t \in T = \{1, \dots, |T|\}$. Each vehicle performs a number of line runs or expeditions along a line. Line runs have to be located in time for each line. The total number of line runs to locate in line $l$ ($\rho_l$), the vehicle capacity ($Q$) and the fleet size $\kappa$ will be assumed to be input data of the problem.

### 2.2 Demand

Let $I$ be the set of transportation requests formulated by customers of transit corridor $L$. Each request $i \in I$ involves the following information:

1. A pair of origin and destination stations, denoted by $\langle s_i, l_i\rangle$ and $\langle s_i', l_i\rangle$, respectively. Such stations must be associated to a line $l_i$ with edges $e \in E_l$ that will be used as a path to satisfy request $i$. With this, it can be defined a parameter $m_{ie}$ equal to one if edge $e \in E_{l_i}$ is used when request $i \in I$ is served or equal to zero otherwise.
2. A preferred departure time $t_i$ to locate a line run in station $\langle 1, l_i\rangle$. Furthermore, $t_i \in [t_i^-, t_i^+]$, which denotes the earliest and the latest times that are admissible for serving request $i$.
3. A penalization cost $c_i$ that will be paid if the corresponding request is not served.
4. Trips along the transit line are direct and travel times are known in advance, so the users' inconvenience with respect to the arrival time can be obviated. Denoting by $T_i = \{t_i^-, t_i^- + 1, \dots, t_i^+\}$ the set of feasible time slots where a line run can be located in order to serve request $i$, parameter $\varphi_{it}$ will compute the relative cost of allocating request $i$ to the line run which departs from station $\langle 1, l_i\rangle$ at time $t \in T_i$. The total inconvenience cost of a transportation request is defined as $c_i\varphi_{it}$ if $i$ is allocated to a line run in $t$, or $c_i$ if $i$ is not allocated to any line run.
5. Following previous definitions, $I_{lt}$ denotes the subset of requests that can be served locating a line run for line $l$ in the time slot $t$, that is $I_{lt} = \{i \in I : t \in T_i \wedge l_i = l\}$.

### 2.3 Timetables and vehicle schedules

The concept of timetable must be formalized as follows. Given the set of line runs $r \in R_l$ defined in line $l$, with $|R_l| = \rho_l$, a timetable $\Theta$ along partition $T$ is defined as the set of arrival/departure times at each station for each line run: $\Theta = \{(\theta^+_{\langle s,l\rangle r}, \theta^-_{\langle s,l\rangle r}), l \in \mathcal{L}, \langle s, l\rangle \in S_l, r \in R_l\}$.

Potentially, all timetables can be generated over the three defined sets. However, the number of feasible timetables can be highly reduced by means of the following result:

**Property 1** Assuming that:

1. stopping time at each station is known/prefixed:

$$\theta^-_{\langle s,l\rangle r} - \theta^+_{\langle s,l\rangle r} = \lambda_{\langle s,l\rangle}, \quad l \in \mathcal{L}, \langle s,l\rangle \in S_l, r \in R_l$$

2. travel times between consecutive stations is known/prefixed:

$$\theta^+_{\langle s+1,l\rangle r} - \theta^-_{\langle s,l\rangle r} = \mu_{\langle s,l\rangle}, \quad l \in \mathcal{L}, \langle s,l\rangle \in S_l : s < |S_l|, r \in R_l$$

the following properties can be stated:

- a fixed travel time $\tau_l$ can be assumed in order to complete a line run:

$$\theta^+_{\langle |S_l|,l\rangle r} - \theta^-_{\langle 1,l\rangle r} = \tau_l, \quad l \in \mathcal{L}, r \in R_l$$

- timetables can be redefined as follows:

$$\Theta \equiv x = \{x_{lt}, l \in \mathcal{L}, t \in T\}$$

  where $x_{lt} \in \{0, 1\}$ is equal to 1 if and only if a line run is allocated to line $l$ at time slot $t$.

Note that the number of vehicles required to perform the timetable cannot be greater than the fleet size, even when a vehicle can perform several line runs in the time period under consideration. For this reason, we must characterize when a timetable can be performed by the given fleet size.

**Definition 1** A timetable $x$ is a feasible $\kappa$-vehicle schedule if the number of vehicles required to perform $x$ is less or equal than the fleet size $\kappa$, thus

$$0 \le \sum_{t'=1}^{t} x_{1t'} - \sum_{t'=1}^{t-\tau_2} x_{2t'} \le \kappa \quad t \in T \tag{1}$$

$$0 \le \sum_{t'=1}^{t} x_{2t'} - \sum_{t'=1}^{t-\tau_1} x_{1t'} \le \kappa \quad t \in T. \tag{2}$$

This definition ensures that between time slots $t = 1$ and $t = \tau_l$ no more than $\kappa$ vehicles leave the depot of any line. After $t = \tau_l$, vehicles that arrive to the end of one line can be used by the other, but at any moment the difference between those vehicles which have left a depot and those which have arrived to such a depot cannot be negative nor greater than $\kappa$. This will lead us to a characterization between an optimal timetable and an optimal vehicle schedule.

**Property 2** Assuming that before starting a line run, vehicles are always empty, they can remain stopped as long as necessary without any additional cost. Therefore, an optimal timetable $x$ satisfying (1) and (2) is also an optimal vehicle schedule.

# 3 Formulation

## 3.1 The $\rho$-median problem

Two sets of binary variables are considered in the formulation:

$x_{lt} \in \{0, 1\}$ equal to 1 when a vehicle starts a line run in line $l$ at time $t$
$y_{it} \in \{0, 1\}$ equal to 1 when request $i$ is allocated to a vehicle which starts a line run at time $t$.

The mathematical model that describes our problem is defined as follows:

$$z = \min \sum_{i \in I} \left[ \sum_{t \in T_i} c_i \varphi_{it} y_{it} + \sum_{t \in T_i} c_i (1 - y_{it}) \right] \tag{3a}$$

$$\text{s.t.:} \quad \sum_{t \in T} x_{lt} \leq \rho_l \qquad l \in \mathcal{L} \tag{3b}$$

$$\sum_{t \in T_i} y_{it} \leq 1 \qquad i \in I \tag{3c}$$

$$y_{it} - x_{l_i t} \leq 0 \qquad i \in I, \ t \in T_i \tag{3d}$$

$$0 \leq \sum_{t'=1}^{t} x_{1t'} - \sum_{t'=1}^{t-\tau_2} x_{2t'} \leq \kappa \qquad t \in T \tag{3e}$$

$$0 \leq \sum_{t'=1}^{t} x_{2t'} - \sum_{t'=1}^{t-\tau_1} x_{1t'} \leq \kappa \qquad t \in T \tag{3f}$$

$$\sum_{i \in I_{lt}} y_{it} m_{ie} \leq Q x_{lt} \qquad l \in \mathcal{L}, \ e \in E_l, \ t \in T \tag{3g}$$

$$x_{lt} \in \{0, 1\} \qquad l \in \mathcal{L}, \ t \in T \tag{3h}$$

$$y_{it} \in \{0, 1\} \qquad i \in I, \ t \in T_i. \tag{3i}$$

Objective function (3a) minimizes the total users' inconvenience. For the sake of understandability, we present (3a) as defined in Sect. 2.2. Constraint (3b) ensures that no more than $\rho_l$ line runs are located in line $l$. Constraint (3c) guarantees that request $i$ is not allocated to more than one line run, avoiding negative terms in the second part of the objective function. Constraint (3d) ensures that requests can only be allocated to time slots where a line run has been located. Constraints (3e) and (3f) forces that no more than $\kappa$ vehicles are used in each line. Constraint (3g) guarantees that no more than $Q$ requests use edge $e$ of line $l$ in a line run located at time $t$.

The provided formulation exhibits significant similarities with the classic p-median problem (Hakimi 1964, 1965), where the total sum of weighted distances between a given set of customers, and a set of locations for potential facilities is minimized. In our formulation, facilities are line-runs, customers are travelers and distances are measured by scheduled delays (we recall Sect. 1; Small 1982). In our model, a request $i$ can be refused in exchange for the penalty cost $c_i$, and deviations from desired travel times are penalized by using the discrete parameter $\varphi_{it}$. Additionally, constraints for controlling schedules and capacities have been stated.

### 3.2 Public context for the $\rho$-median problem

Since the access to vehicles is performed inside stations, all the information regarding timetables is available for passengers so they are free to choose between those line runs that better fit into their own interests. This idea contrasts with other transport systems where the operator is the unique decision maker who can choose which request should be served, when and using which vehicle. In order to emphasize the importance of the passenger point-of-view in the modelization, additional constraints can be added. First, we impose that a passenger must be allocated to a line run if there is at least one line run located in $T_i$:

$$\sum_{t\in T_i} x_{l_i t} \le \rho_{l_i} \sum_{t\in T_i} y_{it} \quad i \in I. \tag{3j}$$

Additionally, constraint (3k) ensures that each request is allocated to a line run if and only if such line run is the one with lowest inconvenience cost in $T_i$:

$$\sum_{\substack{t'\in T_i \\ \varphi_{it'} > \varphi_{it}}} y_{it'} + x_{l_i t} \le 1 \quad i \in I, t \in T_i. \tag{3k}$$

Constraint (3k) is adapted from Wagner and Falkson (1975) and belongs to the so called closest assignment constraints. According to Espejo et al. (2012), this kind of constraints can be modelled in many different ways, giving rise to better/worse linear programming relaxations. As shown in that paper, the constraint provided by Wagner and Falkson (1975) can be strengthened by using the fixed numbers of facilities (line runs) to locate. Even with such improvements, we must note that the inclusion of this constraint together with (3j) considerably increases the computational complexity of the proposed model. In order to provide good solutions within a reasonable time of computation, a request clustering algorithm is introduced in the next section in order to reduce the number of variables and constraints of the model.

## 4 Clustering algorithm

The number of requests $|I|$ can be highly reduced considering subsets of requests $\bar{I}_{lt} \subseteq I$ with a common preferred departure time for a line run, thus $\bar{I}_{lt} = \{i \in I : l_i = l \wedge t_i = t\}$. We define a new set of clustered requests $j \in J$ in such a way that each subset $\bar{I}_{lt}$ is identified with a transportation request $j$ weighted by a factor $q_j$. In this way, we can solve the problem by using the set $j \in J$ instead of $i \in I$. Note that clustered requests can be built if $|I| > |T|$ or if $\exists l \in \mathcal{L}, t \in T$ such that $|\bar{I}_{lt}| > 1$. In order to define the clustering algorithm, we require the following parameters:

- $\bar{t}_j$: preferred departure time for clustered request $j$ in order to locate a line run in station $\langle 1, l_j \rangle$
- $\bar{c}_j$: penalization cost that will be paid if the corresponding clustered request is not served
- $\bar{m}_{je}$: number of requests in $I$ that are grouped in $j$ and use edge $e$ $(\bar{m}_j = [\bar{m}_{j1}, \bar{m}_{j2}, \ldots, \bar{m}_{j|E_{l_j}|}])$
- $q^{max}$: maximum weight for a clustered request, that is $q_j \le \min\{q^{max}, Q\}$.

Algorithm 1 describes how these parameters are constructed.

**Algorithm 1**: Clustering requests

```
input  : Requests $i \in I$, clusters $J := \{\}$, parameters $c_i, m_{ie}$ and maximum weight
         $q^{max}$ for a clustered request
output: Clustered requests $j \in J$, and parameters $\bar{t}_j, \bar{c}_j, \bar{m}_j$
for each request $i \in I$ do
    if $\nexists$ $j \in J$ clustered request such that $\bar{t}_j = t_i$ and $q_j < q^{max}$ then
        Create the clustered request $j = |J| + 1$: $[J, q_j, \bar{c}_j, \bar{m}_j] := [J \cup \{j\}, 1, c_i, m_i]$;
        else
            Update $[q_j, \bar{c}_j, \bar{m}_j] := [q_j + 1, \bar{c}_j + c_i, \bar{m}_j + m_i]$
```

For each request, we have to check if there exists a clustered request such that $\bar{t}_j = t_i$ and $q_j \leq q^{max}$. Since we can obviate searching along requests $j \in J$ such that $q_j = q^{max}$, in the worst case we check as many requests as time slots in $T$. Thus, the complexity of Algorithm 1 is $O(|I||T|)$.

Next, in order to solve the $\rho$-median problem with clustered requests (c-$\rho$-median problem), we apply the following procedure:

**Procedure 1** (c-$\rho$-median problem)

1. Solve (3a)–(3k) considering the set of clustered requests $J$ instead of $I$. Let $x'$ be the timetable solution returned.
2. Solve (3a)–(3k) considering the set of requests $I$ and fixing $x$ variables as $x := x'$ (thus, only variables $y$ and $z$ have to be found). Let $z^{UB}$ be the best upper bound returned and let $y'$ be the request assignment. The solution given by the c-$\rho$-median problem is $\{z = z^{UB}, x = x', y = y'\}$.
3. Let $z^{LB}$ be the best lower bound returned by the $\rho$-median problem. The gap induced by the c-$\rho$-median problem can be obtained by means of

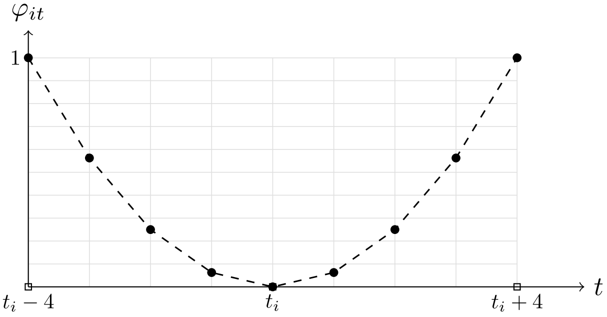
$$c\text{-}gap = \frac{100(z^{UB} - z^{LB})}{z^{LB}}.$$

Note that without considering capacity constraints, optimal solutions for the c-$\rho$-median and $\rho$-median problems are equivalent. On the other hand, under capacity constraints clusters cannot be split and only complete clusters can be allocated/rejected. Consequently, requests within a cluster cannot be allocated to different line-runs with equal inconvenience costs. Therefore, the solution provided by the c-$\rho$-median problem is an upper bound of the $\rho$-median problem.

## 5 Computational experience

In order to show the applicability of the previous model and algorithm, a scenario composed of one transit corridor with two lines (each one running on a different direction) along eight stations has been considered. A random instance of $|I| = 1000$ requests has been generated in the time interval with desired arrival times following a uniform probability distribution. All requests have been assumed to have equal penalization costs $c_i$ and inconvenience costs ($\varphi_{it}$) defined in $T_i = \{\max\{0, t_i - 4\}, \ldots, \max\{0, t_i - 1\}, t_i, \min\{|T|, t_i +$

**Fig. 1** Discrete inconvenience costs under consideration



1}, ..., min{$|T|, t_i + 4$}} that is, $|T_i| \simeq 9$ options for each $i \in I$. The discretized inconvenience function (see Fig. 1) has been taken as follows:

$$\varphi_{it} = \min\left\{1, \left(\frac{\max\{0, t_i - t\}}{\max\{1, t_i - t_i^-\}}\right)^2 + \left(\frac{\max\{0, t - t_i\}}{\max\{1, t_i^+ - t_i\}}\right)^2\right\}.$$

The time horizon has been split into 60 time slots ($|T| = 60$) of standardized length $\|t\|$. Distances (in time) between stations are considered equal to $4.5\|t\|$ and a time for boarding and alighting equal to $0.5\|t\|$ will be required. Fleet sizes ($\kappa$) will be assumed equal to 2, 3 and 4. Line runs will vary in $\rho$=1, 2, 3, 4, 5, 6, 8, 10, 12, that would provide in a time interval of $60\|t\|$ an approximate frequency of 60, 30, 20, 15, 12, 10, 7.5 and 5 time (measured in time units of length $\|t\|$).

The computational experience also includes different scenarios depending on the capacity requirements established. Scenario $S1$ assumes the version of the problem without capacity constraints. In scenarios $S2_{(Q)}$, $Q \in \{40, 45\}$ constraint (3g) is activated for establishing a version of selective capacity in the problem. In scenarios $S3_{(Q)}$, $Q \in \{40, 45\}$ constraints (3j) and (3k) are activated for establishing the optimal timetable choices for single requests. Finally, scenarios $S3_{(Q)}^{(q^{max})}$, $Q \in \{40, 45\}, q^{max} \in \{3, 5\}$ show that constraints (3j) and (3k) are activated and requests have been clustered considering a maximum cluster size equal to $q^{max}$.

All instances have been solved using ILOG CPLEX 12.2 on a personal computer with an Intel(R) Core(TM)i7 CPU 3.4 GHz processor and 16 GB RAM. Default solver values were used for all parameters.

Table 1 shows the comparative objective values obtained for different scenarios, varying the fleet size, line runs and vehicle capacities. For the sake of favoring the comparison of results arising from heterogeneous contexts, the initial objective value $z$ has been normalized and complemented by means of the function $\theta_1 \equiv 100(1 - z/|I|)$. Moreover, the percentage of covered requests with any level of satisfaction has been also included by means of the function $\theta_2 \equiv 100(1 - (\sum_i \sum_{|T_i|}) y_{it}/|I|)$. Logically, the inconvenience perception determines the level of satisfaction attained, holding in all scenarios $\theta_2 \geq \theta_1$. Additionally, the coverage level increases as long as $Q$, $\rho$ or $\kappa$ increase. However, $\theta_1$ and $\theta_2$ cannot improve at a certain level without including more vehicles. Irregularities in objective values for scenarios $S3_{(45)}$ and $S3_{(40)}$ at $\kappa = 4$, $\rho = 6, 8, 10, 12$ are due to the fact that optimality is not reached before the time limit (3600 seconds) under consideration.

Tables 2 and 3 show running times in seconds (*sec*) and gaps (*gap*, *c-gap*) obtained for the different scenarios. We denote as *gap*, the relative gap computed with the best lower

**Table 1** Demand coverage for the different scenarios

| $\kappa$ | $\rho$ | $S1$ | | $S2_{(45)}$ | | $S3_{(45)}$ | | $S2_{(40)}$ | | $S3_{(40)}$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\theta_1$ | $\theta_2$ | $\theta_1$ | $\theta_2$ | $\theta_1$ | $\theta_2$ | $\theta_1$ | $\theta_2$ | $\theta_1$ | $\theta_2$ |
| 2 | 1 | 18.27 | 29.2 | 16.26 | 22.4 | 9.97 | 15.4 | 15.7 | 21.8 | 6.6 | 13.1 |
| 2 | 2 | 35.97 | 55.5 | 31.72 | 42.6 | 25.87 | 35.3 | 30.39 | 40.8 | 23.26 | 31.1 |
| 2 | 3 | 49.13 | 73.2 | 44.71 | 58.2 | 34.35 | 49.1 | 42.81 | 55.4 | 28.03 | 40.9 |
| 2 | 4 | 49.13 | 73.2 | 46.01 | 59.2 | 36.55 | 51 | 45.02 | 56.4 | 28.03 | 40.9 |
| 2 | 5 | 49.13 | 73.2 | 46.01 | 59.2 | 36.55 | 51 | 45.02 | 56.4 | 28.03 | 40.9 |
| 2 | 6 | 49.13 | 73.2 | 46.01 | 59.2 | 36.55 | 51 | 45.02 | 56.4 | 28.03 | 40.9 |
| 2 | 8 | 49.13 | 73.2 | 46.01 | 59.2 | 36.55 | 51 | 45.02 | 56.4 | 28.03 | 40.9 |
| 2 | 10 | 49.13 | 73.2 | 46.01 | 59.2 | 36.55 | 51 | 45.02 | 56.4 | 28.03 | 40.9 |
| 2 | 12 | 49.13 | 73.2 | 46.01 | 59.2 | 36.55 | 51 | 45.02 | 57.1 | 28.03 | 40.9 |
| 3 | 1 | 18.27 | 29.2 | 16.26 | 22.4 | 9.97 | 15.4 | 15.7 | 21.8 | 6.6 | 13.1 |
| 3 | 2 | 35.97 | 55.5 | 31.72 | 42.6 | 25.87 | 35.3 | 30.39 | 40.8 | 23.26 | 31.1 |
| 3 | 3 | 52.4 | 81 | 46.89 | 61.4 | 37.84 | 50.2 | 44.84 | 58 | 35.19 | 44.3 |
| 3 | 4 | 66.38 | 91.3 | 59.73 | 76.4 | 45.57 | 61 | 57.11 | 72.7 | 39.92 | 54.2 |
| 3 | 5 | 66.38 | 91.3 | 61.29 | 78.9 | 48.83 | 62.7 | 59.55 | 74.9 | 42.57 | 52.8 |
| 3 | 6 | 66.38 | 91.3 | 61.29 | 78.9 | 48.34 | 59.5 | 59.55 | 74.9 | 43.72 | 52.5 |
| 3 | 8 | 66.38 | 91.3 | 61.29 | 78.9 | 48.76 | 65.6 | 59.55 | 74.9 | 42.91 | 52.1 |
| 3 | 10 | 66.38 | 91.3 | 61.29 | 78.9 | 48.6 | 63.6 | 59.55 | 74.9 | 43.72 | 52.5 |
| 3 | 12 | 66.38 | 91.3 | 61.29 | 78.9 | 48.6 | 63.6 | 59.55 | 74.9 | 43.33 | 52.1 |
| 4 | 1 | 18.27 | 29.2 | 16.26 | 22.4 | 9.97 | 15.4 | 15.7 | 21.8 | 6.6 | 13.1 |
| 4 | 2 | 35.97 | 55.5 | 31.72 | 42.6 | 25.87 | 35.3 | 30.39 | 40.8 | 23.26 | 31.1 |
| 4 | 3 | 52.4 | 81 | 46.89 | 61.4 | 37.84 | 50.2 | 44.84 | 58 | 35.19 | 44.3 |
| 4 | 4 | 68.09 | 97.5 | 61.08 | 77.2 | 47.02 | 59.1 | 58.37 | 73 | 41.11 | 54 |
| 4 | 5 | 79.59 | 100 | 73.7 | 89.7 | 56.08 | 70.8 | 70.77 | 85.2 | 48.61 | 61.5 |
| 4 | 6 | 79.94 | 100 | 75.33 | 92.1 | 63.15 | 81.7 | 73.16 | 88.8 | 52.23 | 60.1 |
| 4 | 8 | 79.94 | 100 | 75.33 | 92.1 | 60.33 | 75.5 | 73.16 | 88.8 | 53.77 | 69.8 |
| 4 | 10 | 79.94 | 100 | 75.33 | 92.1 | 60.34 | 74.8 | 73.16 | 88.8 | 53.04 | 60.9 |
| 4 | 12 | 79.94 | 100 | 75.33 | 92.1 | 61.48 | 80.5 | 73.16 | 88.8 | 53.88 | 63.7 |

bound obtained by Cplex within a time limit of 3600 seconds and the best upper bound obtained by Cplex within a time limit of 3600 seconds or 1200 seconds (indicated in the table). Additionally, *c-gap* denotes the gap induced by the clustering algorithm as described in Sect. 4 taking as reference the best upper bound provided by Cplex for $S3_{(Q)}$ within a time limit of 3600 s.

Note first that we do not report running times or gaps for $S1$ since all instances were solved to optimality in less than 0.2 seconds. Even $S2_{(Q)}$ can be solved for each instance in a few seconds. We remark that theoretical scenarios $S1$ and $S2$ are interesting for establishing a comparative analysis on values reached in the objective function. On the other hand, scenario $S3$ is closer to the real operability although obtaining optimal solutions for that context requires a considerable computational effort. In order to assess the time consumed within the computation of optimal solutions for scenario $S3$, different scenarios for maximum running times of 3600 seconds and 1200 seconds have been analyzed. As mentioned in the introduction (third paragraph), the TNTSP might be called

**Table 2** Running times and gaps obtained for the different scenarios and $Q = 45$

| $\kappa$ | $\rho$ | time limit = 3600 s | | | | time limit = 1200 s | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $S2_{(45)}$ | | $S3_{(45)}$ | | $S3_{(45)}$ | | $S3^{(3)}_{(45)}$ | | $S3^{(5)}_{(45)}$ | |
| | | *sec* | *gap* | *sec* | *gap* | *sec* | *gap* | *sec* | *c-gap* | *sec* | *c-gap* |
| 2 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 2 | 2.3 | 0 | 60.6 | 0 | 59.8 | 0 | 9.1 | 0 | 18.4 | 0 |
| 2 | 3 | 2 | 0 | 301.1 | 0 | 300.9 | 0 | 36.9 | 0 | 31.3 | 0 |
| 2 | 4 | 2.5 | 0 | 1946.6 | 0 | 1200 | 0 | 28 | 0 | 14.8 | 0 |
| 2 | 5 | 2.5 | 0 | 308.6 | 0 | 308 | 0 | 79.2 | 0 | 46 | 0 |
| 2 | 6 | 2.5 | 0 | 1329.3 | 0 | 1200 | 0 | 36.9 | 0 | 44.8 | 0 |
| 2 | 8 | 2.5 | 0 | 1088.9 | 0 | 1090 | 0 | 75.5 | 0 | 61.6 | 0 |
| 2 | 10 | 2.6 | 0 | 2656.8 | 0 | 1200 | 0.01 | 30.6 | 0 | 29.4 | 0 |
| 2 | 12 | 2.5 | 0 | 1307.2 | 0 | 1200 | 0 | 30.7 | 0 | 29.2 | 0 |
| 3 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 2 | 2 | 0 | 68.6 | 0 | 68.7 | 0 | 10.7 | 0 | 24.5 | 0 |
| 3 | 3 | 2.4 | 0 | 2141.1 | 0 | 1200 | 0 | 141.1 | 0 | 23.6 | 0 |
| 3 | 4 | 2.8 | 0 | 3600 | 1.73 | 1200 | 1.99 | 412.3 | 1.99 | 223.8 | 1.99 |
| 3 | 5 | 2.9 | 0 | 3600 | 1.52 | 1200 | 1.88 | 328.7 | 1.77 | 109.5 | 1.77 |
| 3 | 6 | 2.7 | 0 | 3600 | 1.92 | 1200 | 2.92 | 166.3 | 2.23 | 118.6 | 2.23 |
| 3 | 8 | 3.1 | 0 | 3600 | 1.42 | 1200 | 1.7 | 369.6 | 1.62 | 244 | 1.62 |
| 3 | 10 | 3 | 0 | 3600 | 1.05 | 1200 | 1.2 | 277.6 | 1.09 | 120.1 | 1.09 |
| 3 | 12 | 2.8 | 0 | 3600 | 1.88 | 1200 | 2.31 | 327.2 | 2.22 | 295.9 | 2.22 |
| 4 | 1 | 3.8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 2 | 2 | 0 | 380.7 | 0 | 381 | 0 | 11.5 | 0 | 5 | 0 |
| 4 | 3 | 2.4 | 0 | 2527.5 | 0 | 1200 | 0.06 | 184.4 | 0 | 60.7 | 0 |
| 4 | 4 | 3.9 | 0 | 3600 | 0.63 | 1200 | 0.87 | 441.8 | 0.65 | 464.8 | 0.65 |
| 4 | 5 | 3.3 | 0 | 3600 | 2.72 | 1200 | 4.34 | 1200 | 4.08 | 1200 | 3.81 |
| 4 | 6 | 3.7 | 0 | 3600 | 2.61 | 1200 | 5.14 | 1200 | 3.9 | 1200 | 4.24 |
| 4 | 8 | 3.6 | 0 | 3600 | 3.15 | 1200 | 5.98 | 1200 | 3.89 | 1200 | 4.3 |
| 4 | 10 | 3.9 | 0 | 3600 | 3.2 | 1200 | 5.13 | 1200 | 3.9 | 427 | 3.98 |
| 4 | 12 | 3.8 | 0 | 3600 | 2.84 | 1200 | 5.56 | 1200 | 3.99 | 242.8 | 3.68 |

to solve decision problems in contexts characterized by strict limitations of time, going from those where speed might not be important to those where a fast solution might be required.

Under a time limit of 3600 seconds Cplex is able to solve to optimality all instances for $\kappa = 2$ leaving small gaps for $\kappa = \{3, 4\}$. Note that these gaps are due to the effect of capacities and closest assignment. Decreasing the time limit to 1200 seconds the gaps increase around a 2 %. Under a time limit of 1200 seconds, the resolution of $S3^{(3)}_{(Q)}$ reaches optimality in all instances for $\kappa = 2$ increasing slightly the gaps of $S3_{(Q)}$ after 3600 seconds for $\kappa = \{3, 4\}$. However, $S3^{(3)}_{(Q)}$ outperforms $S3_{(Q)}$ under a time limit of 1200 seconds in terms of gap and running time. The same observation can be made for $S3^{(5)}_{(Q)}$ with lower running times.

**Table 3** Running times and gaps obtained for the different scenarios and $Q = 40$

| $\kappa$ | $\rho$ | Time limit = 3600 s | | | | Time limit = 1200 s | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $S2_{(40)}$ | | $S3_{(40)}$ | | $S3_{(40)}$ | | $S3^{(3)}_{(40)}$ | | $S3^{(5)}_{(40)}$ | |
| | | *sec* | *gap* | *sec* | *gap* | *sec* | *gap* | *sec* | *c-gap* | *sec* | *c-gap* |
| 2 | 1 | 3.3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 2 | 2.3 | 0 | 62 | 0 | 61.9 | 0 | 14.7 | 0 | 5.6 | 0 |
| 2 | 3 | 2.5 | 0 | 206.4 | 0 | 206.4 | 0 | 34.8 | 0 | 20 | 0 |
| 2 | 4 | 3 | 0 | 1186.4 | 0 | 1190.6 | 0 | 85.5 | 0 | 41.3 | 0 |
| 2 | 5 | 2.9 | 0 | 561.1 | 0 | 560.5 | 0 | 119.5 | 0 | 31.8 | 0 |
| 2 | 6 | 2.7 | 0 | 190.6 | 0 | 190.6 | 0 | 71.8 | 0 | 37.9 | 0 |
| 2 | 8 | 2.9 | 0 | 846.3 | 0 | 846.8 | 0 | 102.4 | 0 | 16.7 | 0 |
| 2 | 10 | 3.7 | 0 | 248.4 | 0 | 247.9 | 0 | 91.8 | 0 | 39.3 | 0 |
| 2 | 12 | 3.6 | 0 | 2017.3 | 0 | 1200 | 0 | 92.2 | 0 | 39.5 | 0 |
| 3 | 1 | 3.1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 2 | 2.3 | 0 | 61.3 | 0 | 61.7 | 0 | 15 | 0 | 5.9 | 0 |
| 3 | 3 | 2.7 | 0 | 967.8 | 0 | 967.9 | 0 | 72 | 0 | 30.8 | 0 |
| 3 | 4 | 3.6 | 0 | 3600 | 1.42 | 1200 | 1.72 | 443.9 | 1.57 | 91.2 | 1.57 |
| 3 | 5 | 5.4 | 0 | 3600 | 2.06 | 1200 | 3.3 | 316.5 | 2.33 | 118.9 | 2.33 |
| 3 | 6 | 4.5 | 0 | 3600 | 1.52 | 1200 | 2.1 | 210.8 | 1.8 | 213.5 | 1.8 |
| 3 | 8 | 4.9 | 0 | 3600 | 1.35 | 1200 | 1.86 | 1051.7 | 1.49 | 132.4 | 1.49 |
| 3 | 10 | 5.1 | 0 | 3600 | 1.56 | 1200 | 2.73 | 537.9 | 1.92 | 171.4 | 1.92 |
| 3 | 12 | 5 | 0 | 3600 | 1.43 | 1200 | 1.75 | 176.6 | 1.62 | 153.4 | 1.62 |
| 4 | 1 | 3.1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 2 | 2.3 | 0 | 67.1 | 0 | 67 | 0 | 11 | 0 | 6 | 0 |
| 4 | 3 | 2.5 | 0 | 1239.4 | 0 | 1200 | 0 | 237.8 | 0 | 35.8 | 0 |
| 4 | 4 | 3 | 0 | 3581.9 | 0 | 1200 | 0.14 | 365.3 | 0 | 173.9 | 0 |
| 4 | 5 | 4.2 | 0 | 3600 | 3.1 | 1200 | 4.87 | 1200 | 4.55 | 198 | 4.55 |
| 4 | 6 | 5.7 | 0 | 3600 | 3.72 | 1200 | 5.8 | 1200 | 5.22 | 271.9 | 5.22 |
| 4 | 8 | 5.9 | 0 | 3600 | 3.5 | 1200 | 5.48 | 1200 | 5.11 | 659.1 | 5.4 |
| 4 | 10 | 6.5 | 0 | 3600 | 3.54 | 1200 | 5.53 | 1200 | 4.72 | 542.4 | 5.23 |
| 4 | 12 | 6.7 | 0 | 3600 | 3.42 | 1200 | 5.53 | 1200 | 4.72 | 1050.3 | 5.37 |

In conclusion, Tables 2 and 3 prove the usefulness of introducing the clustering algorithm in scenario $S3$, because running times become considerably reduced in exchange for a small increase of the gap and, sometimes, even improves the actual ones. This methodology is of special interest for solving bigger size instances not considered in this paper. In addition cluster sizes allow an adaptation of the procedure in accordance with the instance size.

Summarizing, Table 4 provides the results obtained for Scenario 3. In this table, we include averages and maximum values of computational times ($\overline{sec}$, $sec^*$), gaps ($\overline{gap}$, $gap^*$) and clustering gaps ($\overline{c\text{-}gap}$, $c\text{-}gap^*$) computed for different values of $\kappa$ and all values of $\rho$.

**Table 4** Summary of the results obtained for Scenario 3

| $\kappa$ | | Time limit = 3600 s | | Time limit = 1200 s | | | Time limit = 1200 s | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | $S3_{(45)}$ | $S3_{(40)}$ | $S3_{(45)}$ | $S3_{(40)}$ | | $S3^{(3)}_{(45)}$ | $S3^{(3)}_{(40)}$ | $S3^{(5)}_{(45)}$ | $S3^{(5)}_{(40)}$ |
| 2 | $\overline{sec}$ | 999.9 | 590.94 | 728.74 | 500.52 | $\overline{sec}$ | 36.32 | 68.08 | 30.61 | 25.79 |
| | $sec^*$ | 2656.8 | 2017.3 | 1200 | 1200 | $sec^*$ | 79.2 | 119.5 | 61.6 | 41.3 |
| | $\overline{gap}$ | 0 | 0 | 0 | 0 | $c\text{-}gap$ | 0 | 0 | 0 | 0 |
| | $gap^*$ | 0 | 0 | 0.01 | 0 | $c\text{-}gap^*$ | 0 | 0 | 0 | 0 |
| 3 | $\overline{sec}$ | 2645.52 | 2514.34 | 940.97 | 914.4 | $\overline{sec}$ | 225.94 | 313.82 | 128.89 | 101.94 |
| | $sec^*$ | 3600 | 3600 | 1200 | 1200 | $sec^*$ | 412.3 | 1051.7 | 295.9 | 213.5 |
| | $\overline{gap}$ | 1.06 | 1.04 | 1.33 | 1.5 | $\overline{c\text{-}gap}$ | 1.21 | 1.19 | 1.21 | 1.19 |
| | $gap^*$ | 1.92 | 2.06 | 2.92 | 3.3 | $c\text{-}gap^*$ | 2.23 | 2.33 | 2.23 | 2.33 |
| 4 | $\overline{sec}$ | 2723.13 | 2543.16 | 975.67 | 940.78 | $\overline{sec}$ | 737.52 | 734.9 | 533.37 | 326.38 |
| | $sec^*$ | 3600 | 3600 | 1200 | 1200 | $sec^*$ | 1200 | 1200 | 1200 | 1050.3 |
| | $\overline{gap}$ | 1.68 | 1.92 | 3.01 | 3.04 | $\overline{c\text{-}gap}$ | 2.27 | 2.7 | 2.3 | 2.86 |
| | $gap^*$ | 3.2 | 3.72 | 5.98 | 5.8 | $c\text{-}gap^*$ | 4.08 | 5.22 | 4.3 | 5.4 |

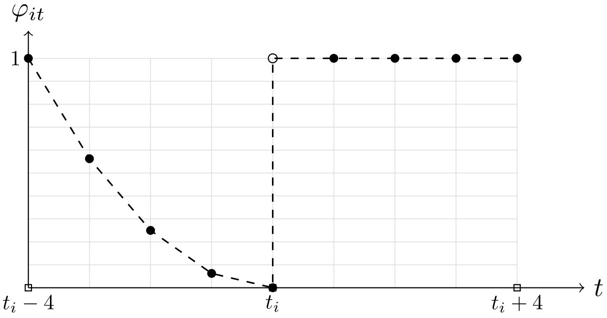


**Fig. 2** Inconvenience costs for a transportation request $i$, going from station $\langle s_i, l_i \rangle$ towards $\langle s_i', l_i \rangle$, that requires a transfer at $\langle s_i', l_i \rangle$

## 6 Extension to transportation networks composed of several transit lines

Previous sections describe how to jointly plan timetables and vehicle schedules along a single transit line for potential customers traveling between origin and destinations of such a line. The extension of the scenario of a single transit line to the more general case where multiple lines are considered in the network can be performed including two new kinds of transportation requests coming from previous lines or going towards other lines.

- **Case 1:** Let $i$ be a transportation request going from $\langle s_i, l_i \rangle$ towards $\langle s_i', l_i \rangle$. If we assume that $i$ requires to transfer to a second line at $\langle s_i', l_i \rangle$, then $t_i$ is the time slot that minimizes the waiting time at $\langle s_i', l_i \rangle$. However, $i$ cannot arrive to $\langle s_i', l_i \rangle$ later than the departure time of the second vehicle. Thus, the inconvenience costs of this kind of request is maximum after $t_i$ as depicted in Fig. 2.
- **Case 2:** Let $i$ be a transportation request going from station $\langle s_i, l_i \rangle$ towards $\langle s_i', l_i \rangle$. If we assume that $i$ comes from a previous line and has transferred at $\langle s_i, l_i \rangle$, then $t_i$ is the time slot that minimizes the waiting time at $\langle s_i, l_i \rangle$. However, a line run located earlier than $t_i$ cannot serve request $i$. Thus, the inconvenience costs of this kind of request is maximum before $t_i$ as depicted in Fig. 3.

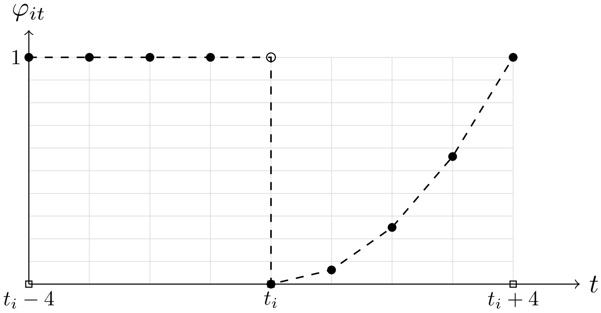


**Fig. 3** Inconvenience costs for a transportation request $i$, going from station $\langle s_i, l_i \rangle$ towards $\langle s'_i, l_i \rangle$, requiring a transfer at $\langle s_i, l_i \rangle$

According to the survey conducted by Stern (1996) on various transit agencies in the US, it is infrequent that passengers use more than a single transfer during their origin-destination trips on the regional transit network. In these circumstances, it is reasonable to classify requests into two subsets: requests that use a single line to reach their destinations and those that require to perform a single transfer in order to achieve the final destination. For the first type, the symmetrical function shown in Fig. 1 can be applied to assess the user's inconvenience and, for the second type of request, the previous asymmetrical functions can provide an extension to the scenario where the presence of transfers can be treated with the methodology developed throughout the paper. Moreover, the penalization costs associated to the non-served requests can be weighted according to the case where the affected user belongs.

## 7 Conclusions

A new approach for jointly planning timetables and vehicle schedules along a single transit line has been developed by emphasizing the point of view of potential customers. The setting analyzed in this paper assumes a model of fully disaggregated demand for a scenario that includes capacity constraints and demand behavior according to different criteria. A p-median based formulation has been proposed including specific constraints for the scheduling problem for a given fleet size of vehicles. In addition, demand behavior is associated with the inclusion of closest assignment type constraints.

A clustering algorithm has been developed in order to provide an alternative methodology for solving instances of the problem when computational time must be limited. The performed computational experience shows the difficulty of including closest assignment constraints in a transportation problem and the advantages of deriving a clustering algorithm that allows an appropriate preprocessing of the information.

The infrastructure analyzed in the paper consists of a single corridor. However, we have developed some hints on how to extend our methodology to more general transportation networks where multiple transit lines operate. In those contexts, transfers between lines required by the passengers can modify the cost associated to the user's inconvenience. The optimization model is shown as a consistent approach, since its applicability remains despite of the change of the network infrastructure.

**Acknowledgements** This work was partially supported by *Ministerio de Ciencia e Innovación* under grant MTM2010-19576-C02-01, MTM2009-14243 and MTM2012-37048, by the *Junta de Andalucía* under grant P09-TEP-5022 and FQM-5849 and by the FEDER funds of the European Union. We would like to thank Gilbert Laporte for his helpful comments and guidelines.

## Appendix: Notation

*Data*

| | |
|---|---|
| $L$ | Transit corridor |
| $S$ | Node set (stations) |
| $e \in E$ | Edge set (tracks) |
| $l \in \mathcal{L}$ | Set of feasible lines in $L$ ($\mathcal{L} = \{1, 2\}$) |
| $l = 1$ | Directed transit line running along $L$ |
| $l = 2$ | Directed transit line running along $L$ in the opposite direction of $l = 1$ |
| $S_l \subseteq S$ | Set of stations for a given line $l \in \mathcal{L}$ |
| $\langle s, l \rangle \in S_l$ | Station in position $s \in \{1, \dots, \|S_l\|\}$ |
| $E_l \subset E$ | Subset that contains all edges used by line $l$ |
| $t \in T$ | Set of time slots ($T = \{1, \dots, \|T\|\}$) |
| $\rho_l$ | Number of line runs to locate |
| $Q$ | Vehicle capacity |
| $\kappa$ | Fleet size |
| $i \in I$ | Set of transportation requests |
| $l_i$ | Line used by request $i$ |
| $\langle s_i, l_i \rangle$, $\langle s'_i, l_i \rangle$ | Origin and destination stations for request $i$ |
| $m_{ie}$ | Parameter equal to one if edge $e \in E_{l_i}$ is used when request $i \in I$ is served or equal to zero otherwise |
| $t_i$ | Preferred departure time for request $i$ to locate a line run in station $\langle 1, l_i \rangle$ |
| $t_i^-, t_i^+$ | Earliest and latest times that are admissible for serving request $i$ |
| $c_i$ | Penalization cost that will be paid if request $i$ is not served |
| $T_i \subseteq T$ | Set of feasible time slots where a line run can be located in order to serve request $i$ ($T_i = \{t_i^-, t_i^- + 1, \dots, t_i^+\}$) |
| $\varphi_{it}$ | Relative cost of allocating request $i$ to the line run which departs from station $\langle 1, l_i \rangle$ at time $t \in T_i$ |
| $I_{lt} \subset I$ | Subset of requests that can be served locating a line run for line $l$ in time slot $t$ |
| $r \in R_l$ | Set of line runs defined in line $l$ ($\|R_l\| = \rho_l$) |
| $\Theta$ | Timetable along partition $T$ |
| $\theta^+_{\langle s,l \rangle r}, \theta^-_{\langle s,l \rangle r}$ | Arrival/departure times at station $\langle s, l \rangle$ for line run $r \in R_l$ in line $l \in \mathcal{L}$ |
| $\lambda_{\langle s,l \rangle}$ | Stopping time required at station $\langle s, l \rangle$ |
| $\mu_{\langle s,l \rangle}$ | Travel time between stations $\langle s, l \rangle$ and $\langle s + 1, l \rangle$ |
| $\tau_l$ | Fixed travel time required to complete a line run in line $l$ |

*Decision variables*

| | |
|---|---|
| $x_{lt} \in \{0, 1\}$ | Binary variable equal to 1 if and only if a line run is allocated to line $l$ at time slot $t$ |
| $y_{it} \in \{0, 1\}$ | Binary variable equal to 1 if and only if request $i$ is allocated to a vehicle which starts a line run at time $t$ |

*Clustering algorithm data*

| | |
|---|---|
| $\bar{I}_{lt} \subset I$ | Subset of requests with a common preferred departure time for locating a line run in line $l$ at the time slot $t$ |
| $j \in J$ | Set of clustered requests |
| $q_j$ | Weight of a clustered request |
| $q^{max}$ | Maximum weight for a clustered request |

| | |
|---|---|
| $\bar{t}_j$ | Preferred departure time for request $i$ to locate a line run in station $\langle 1, l_i \rangle$ |
| $\bar{c}_j$ | Penalization cost that will be paid if the clustered request $j$ is not served |
| $\bar{m}_{je}$ | Number of requests in $I$ that are grouped in $j$ and use edge $e$ ($\bar{m}_j = [\bar{m}_{j1}, \bar{m}_{j2}, \ldots, \bar{m}_{j\|E_{l_j}\|}]$) |

## References

Brannlund, U., Lindberg, P. O., Nou, A., & Nilsson, J.-E. (1998). Railway timetabling using Lagrangian relaxation. *Transportation Science*, *32*(4), 358–369.

Bunte, S., & Kliewer, N. (2009). An overview on vehicle scheduling models. *Public Transport*, *1*, 299–317.

Cadarso, L., & Marín, A. (2012). Integration of timetable planning and rolling stock in rapid transit networks. *Annals of Operations Research*, *199*(1), 113–135.

Cai, X., & Goh, C. J. (1994). A fast heuristic for the train scheduling problem. *Computers & Operations Research*, *21*(5), 499–510.

Caprara, A., Fischetti, M., & Toth, P. (2002). Modeling and solving the train timetabling problem. *Operations Research*, *50*(5), 851–861.

Caprara, A., Monaci, M., Toth, P., & Guida, P. L. (2006). A Lagrangian heuristic algorithm for a real-world train timetabling problem. *Discrete Applied Mathematics*, *154*(5), 738–753.

Castelli, L., Pesenti, R., & Ukovich, W. (2004). Scheduling multimodal transportation systems. *European Journal of Operational Research*, *155*(3), 603–615.

Ceder, A., & Wilson, N. H. M. (1986). Bus network design. *Transportation Research. Part B: Methodological*, *20*(4), 331–344.

Chakroborty, P., Deb, K., & Sharma, R. K. (2001). Optimal fleet size distribution and scheduling of transit systems using genetic algorithms. *Transportation Planning and Technology*, *24*(3), 209–226.

Chang, S.-C., & Chung, Y.-C. (2005). From timetabling to train regulation–a new train operation model. *Information and Software Technology*, *47*(9), 575–585.

Chew, K., Pang, J., Liu, Q., Ou, J., & Teo, C. (2001). An optimization based approach to the train operator scheduling problem at Singapore MRT. *Annals of Operations Research*, *108*(1), 111–118.

Cordeau, J.-F., Laporte, G., Potvin, J.-Y., & Savelsbergh, M. W. (2007a). Transportation on demand. In C. Barnhart & G. Laporte (Eds.), *Handbooks in operations research and management science: Vol. 14. Transportation* (pp. 429–466). Amsterdam: Elsevier. Chap. 7.

Cordeau, J.-F., Laporte, G., Savelsbergh, M. W., & Vigo, D. (2007b). Vehicle routing. In C. Barnhart & G. Laporte (Eds.), *Handbooks in operations research and management science: Vol. 14. Transportation* (pp. 367–428). Amsterdam: Elsevier. Chap. 6.

de Palma, A., & Lindsey, R. (2001). Optimal timetables for public transportation. *Transportation Research. Part B: Methodological*, *35*(8), 789–813.

Espejo, I., Marín, A., & Rodríguez-Chía, A. M. (2012). Closest assignment constraints in discrete location problems. *European Journal of Operational Research*, *219*(1), 49–58.

Fosgerau, M. (2009). The marginal social cost of headway for a scheduled service. *Transportation Research. Part B: Methodological*, *43*(8–9), 813–820.

Grosfeld-Nir, A., & Bookbinder, J. (1995). The planning of headways in urban public transit. *Annals of Operations Research*, *60*, 145–160.

Guihaire, V., & Hao, J.-K. (2008). Transit network design and scheduling: a global review. *Transportation Research. Part A, Policy and Practice*, *42*(10), 1251–1273.

Guihaire, V., & Hao, J.-K. (2010). Transit network timetabling and vehicle assignment for regulating authorities. *Computers & Industrial Engineering*, *59*(1), 16–23.

Guo, Z., & Wilson, N. H. (2011). Assessing the cost of transfer inconvenience in public transport systems: a case study of the London underground. *Transportation Research. Part A, Policy and Practice*, *45*(2), 91–104.

Hakimi, S. (1964). Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, *12*(3), 450–459.

Hakimi, S. (1965). Optimum distribution of switching centers in a communications network and some related graph theoretic problems. *Operations Research*, *13*(3), 462–475.

Higgins, A., Kozan, E., & Ferreira, L. (1996). Optimal scheduling of trains on a single line track. *Transportation Research. Part B: Methodological*, *30*(2), 147–161.

Liebchen, C., Schachtebeck, M., Schöbel, A., Stiller, S., & Prigge, A. (2010). Computing delay resistant railway timetables. *Computers & Operations Research*, *37*(5), 857–868.

Liu, Z.-G., & Shen, J.-S. (2007). Regional bus operation bi-level programming model integrating timetabling and vehicle scheduling. *Systems Engineering - Theory & Practice*, *27*(11), 135–141.

Magnanti, T., & Wong, R. (1984). Network design and transportation planning: models and algorithms. *Transportation Science*, *18*(1), 1–55.

Oliveira, E., & Smith, B. M. (2000). *A job-shop scheduling model for the single-track railway scheduling problem* (Tech. rep. 2000.21). University of Leeds, Leeds, UK.

Quak, C. (2003). *Bus line planning*. Ph.D. thesis, Delft University of Technology, The Netherlands.

Small, K. A. (1982). The scheduling of consumer activities: work trips. *The American Economic Review*, *72*(3), 467–479.

Stern, R. (1996). *Passenger transfer system review. Synthesis of transit practice*. *Transportation research board*, vol. 19.

Wagner, J. L., & Falkson, L. M. (1975). The optimal nodal location of public facilities with price-sensitive demand. *Geographical Analysis*, *7*(1), 69–83.

Zhou, X., & Zhong, M. (2005). Bicriteria train scheduling for high-speed passenger railroad planning applications. *European Journal of Operational Research*, *167*(3), 752–771.

Zhou, X., & Zhong, M. (2007). Single-track train timetabling with guaranteed optimality: branch-and-bound algorithms with enhanced lower bounds. *Transportation Research. Part B: Methodological*, *41*(3), 320–341.