Pesquisa Operacional (2019) 39(1): 85-108
© 2019 Brazilian Operations Research Society
Printed version ISSN 0101-7438 / Online version ISSN 1678-5142
www.scielo.br/pope
doi: 10.1590/0101-7438.2019.039.01.0085

EXPLOITING THE TIMETABLING FLEXIBILITY IN THE CONTEXT OF THE VEHICLE SCHEDULING PROBLEM WITH HETEROGENEOUS FLEET

Monize Sâmara Visentini[1*], Olinto César Bassi de Araújo[2],
Denis Borenstein[3] and Alberto Francisco Kummer Neto[4]

Received March 26, 2018 / Accepted December 8, 2018

ABSTRACT. In the context of a heterogeneous fleet, the single-depot Vehicle Scheduling Problem is known as the Vehicle Type Scheduling Problem (VTSP). In this paper, we extend a VTSP model proposed in the literature to enable timetabling flexibility. Two strategies were developed to promote flexibility: aggregated trips and time windows. The former allows to aggregate two or more trips, occurring close in time, into a single one through the use of larger vehicle types, while the latter allows shifting the departure time of scheduled trips backward in few minutes. Both approaches were tested on generated instances that simulate the traffic behavior of a Brazilian city, where clock-face departure pattern is not required and demand peak times occur. We verified significant savings in the daily operation of the public transportation service with acceptable impact on passengers' experience.

Keywords: Heterogeneous Fleet, Scheduling, Demand.

1 INTRODUCTION

The Vehicle Scheduling Problem (VSP) consists of minimizing the assignment costs of vehicles to a given set of timetabled trips while satisfying two primary constraints: (i) each trip is assigned precisely to a single vehicle; and (ii) each vehicle performs a feasible sequence of trips [1, 10]. Each vehicle starts and ends its operations in the same depot with travel times and stations fixed and previously defined. Different approaches to model the VSP have been developed, as well as solution methods and extensions to enhance reality representation, e.g., the inclusion of multiple-depots and heterogeneous fleet [2]. The literature on vehicle scheduling predominantly covers a single type of vehicle; however, in practice, more than one type is used [4]. In the context of a heterogeneous fleet, the VSP is known as the Vehicle-Type Scheduling Problem (VTSP). The

*Corresponding author – https://orcid.org/0000-0001-6233-6106

[1] Universidade Federal da Fronteira Sul – Cerro Largo, RS, Brazil. E-mail: monize.visentini@uffs.edu.br

[2] Universidade Federal de Santa Maria – Santa Maria, RS, Brazil. E-mail: olinto@ctism.ufsm.br

[3] Universidade Federal do Rio Grande do Sul – Porto Alegre, RS, Brazil. E-mail: denisb@ea.ufrgs.br

[4] Universidade Federal do Rio Grande do Sul – Porto Alegre, RS, Brazil. E-mail: alberto.neto@inf.ufrgs.br

latter, when compared with the VSP, increases the degrees of freedom for planning decisions and therefore, the problem complexity.

The management of heterogeneous vehicle fleet is a common practice in public transportation and is present in most cities around the world. As an example, [28] showed that transit agencies from a hundred cities in the US, Canada, and Puerto Rico have a heterogeneous fleet and, at least, half have articulated buses. By that time, New York City was shown to possess the largest heterogeneous fleet with 4,344 buses in total, 3,704 them with more than 35 feet of length and 640 are articulated. A different study [18] claimed that in Austria 52.50% of the buses are articulated, while in Belgium this figure is 12.10%. In São Paulo, Brazil, the fleet is composed of seven types, characterized by different vehicles capacities as stated in [8].

The heterogeneous fleet is even more striking in urban public transportation systems such as those in Brazil, where bus timetables are not clock-face and there are demand peak times during a day. In this configuration, it is often necessary to schedule several vehicles with few minutes of difference, and it is not uncommon passengers go to the bus stop in order to pick up any bus within a time window, not a specific one. This scenario is the main motivation of this work because if we consider heterogeneous fleet, it is possible to aggregate two or more trips close in time through the use of a larger vehicle and, considering irregular headways, it is also possible to shift the departure time of trips in few minutes forward or backward. Both approaches have little or no impact on the perception of the quality of the service provided from the user's perspective and reduce operating costs, as well as congestion and pollution.

Important to note that scheduling of heterogeneous fleet is not only closely linked to the economic interest of management companies, but it also impacts on passengers' experience and traffic conditions of urban regions as a whole. For example, the employment of one articulated bus could meet the demand of two consecutive trips that are close in time, each one made by smaller buses, keeping service level to passengers untouched. Therefore, a timetable can be changed through the allocation of different bus types, observing frequency and vehicles' capacity. The optimal solution of the corresponding VTSP tends to maximize the system's gains, resulting in a reduction of the total number of scheduled vehicles.

Although there are models in the literature that addresses the cost minimization of vehicle scheduling by modifying the timetable [13, 17, 19], the research is scant on approaches that focus on both heterogeneous fleet and demand of service trips. Furthermore, the majority of the developed approaches have difficulties to solve very large instances, or its implementation is beyond the technical capabilities of private transport operators. Both factors restrict their application to real-world public transportations system.

[16] point out that the level of service (captured by the timetabling) and the operating costs (regarding vehicle usage) are relevant in the management of the transport system because a high level of service guarantees a social benefit and saving on costs is necessary to have a profitable system. From this perspective this paper aims to exploit timetabling flexibility in the VTSP, evaluating its benefits towards minimizing the total costs involved. To promote timetabling flexi-

bility two strategies were considered: aggregated trips and time windows (TW). The first strategy enables the aggregation of two or more trips closes in time, allowing redistribution of their demands to reduce the total number of scheduled vehicles through the use of larger vehicle types. The second strategy adds time windows, which allows shifting the departure time of scheduled trips backward in a few minutes. To this end, we extend a VTSP formulation and include these strategies as constraints, developing an integer linear programming formulations for the VTSP, based on [31]. The input for the developed models is a timetable, the estimation of passenger peak demand of each timetabled trip, and a time window parameter. We tested our approaches on generated instances that simulate the traffic behavior of a Brazilian city. In these instances irregular headways and demand peak-time are considered.

Our approach was designed to be used by a service provider to strategic planning, e.g., the definition of fleet size considering several peak demand scenarios. It can also be used for tactical planning, e.g., to define an off-line vehicle scheduling and a timetabling for daily or weekly frequencies. Further, the results obtained in both planning can be employed by the service provider towards the negotiation of lower cost timetabling with the regulating authorities. Since the level of service and the operating costs are naturally in conflict [16], our model takes into account the trade-off between these criteria by optimizing the timetabling and vehicle scheduling. Thus, development of this study makes relevant the extent to which public transport not only can improve the quality of life of urban dwellers, providing adequate mobility, but also contributes to the solution of urban problems such as noise, congestion, lack of public spaces, pollution, etc [25].

The significant contributions of this paper are the following: (i) VTSP modeling with timetabling flexibility, using the demand as a parameter, an attribute largely overlooked in vehicle scheduling literature; and (ii) a simple and efficient TW implementation which demands small modifications of the underlying network structure.

The paper is organized as follows. A brief review of the literature is presented in Section 2. In Section 3, we introduce our modeling approaches. Following, we describe the configuration of our experiments and computational results for the generated instances in Sections 4 and 5, respectively. Finally, we present the conclusion of the paper with final remarks and suggestions for further researches in Section 6.

2 LITERATURE REVIEW

Although much attention has been paid to vehicle scheduling and timetabling of the bus-based public transportation literature, few papers addressed the problem of minimizing the scheduled vehicle costs by modifying the timetable [12, 9, 16, 15]. Approaches that simultaneously handle these two critical phases of the transit system planning can lead to efficient solutions, mainly in the presence of heterogeneous fleet [31].

Especially relevant to our work is the paper of [20], in which the authors presented an ILP model applying TW to the multiple-depot VSP (MDVSP) to reduce the total amount of scheduled vehicles. The shift of some timetabled trips few minutes backward or forwards introduces flexibility in the beginning or end of a trip. The authors implemented "shadow arcs" of the original service arcs, which leads to a substantial increase in the network size. In a later publication, [19] expanded their previous work integrating TWs to vehicle and crew scheduling problems. Due to network size and problem complexity, two heuristics were developed to determine a subset of critical trips and eliminate uninteresting shadow arcs, reducing the number of variables of model drastically. A combination of Lagrangian relaxation and column generation was employed to solve the integrated problem. Considering instances available in the literature, the authors reported savings in the number of vehicles and crew comparing with previously reported results [14, 30].

[31] employed a time-space network (TSN) to integrate the timetabling and the VTSP, solving the integrated problem by sequentially applying mathematical programming and metaheuristics. A solution of the VTSP model is provided as input to a simulated annealing algorithm, which promotes small changes in the timetable, shifting the departure time of trips in few minutes forward or backward for some randomly selected bus line. Their approach does not take into consideration the service level offered to passengers.

[4] integrated heterogeneous fleet and timetable using the Deficit-Function Theory (DFT). The problem was formulated as a cost-flow network problem, aiming to reduce the number of scheduled vehicles to a particular timetable. A heuristic incorporating DFT components was developed and tested with two small examples. Recently, [5] developed a multi-objective methodology to create bus schedules using vehicles of several sizes, with the following simultaneous objectives: minimize both the waiting time of passengers on stations and the difference between the number of expected and observed passengers in each timetabled trip. However, their methodology has the drawback of being applied to only individual bus lines since they did not address interline vehicle scheduling.

[12] also considered the integrated problem of timetabling and vehicle scheduling towards minimizing passengers' total waiting time via transfer synchronization. They proposed an iterated local search heuristic to solve such a problem. Experiments were conducted in urban traffic of a region of France, composed of some cities and several villages. The study has been expanded in [13] introducing the additional objective of minimizing time spent on empty trips. Their proposed heuristics were also applied to change the timetable according to a vehicle schedule solution. Such an integrated approach resulted in a reduction of 17.15% of vehicles used when compared to the sequential approach. The authors solved the integrated problem in the context of the unconstrained capacity homogeneous fleet.

[27] introduced the simultaneous vehicle scheduling and passenger service problem that seeks to optimize, simultaneously, the passenger's service level, and operating costs. The problem was modeled as an ILP model and solved using large neighborhood search. Although this study presents a very interesting formulation, the required data for the transfer stations are so intensive that it is difficult to apply for transit systems predominantly based on buses, as the case of Brazil. [16] developed a bi-objective formulation to integrate timetabling and vehicle scheduling

problems towards analyzing the trade-off between operating costs and level of service of transit systems. Solutions were provided for instances inspired in the transit network of Monterrey, Mexico, with up 50 lines and five synchronization nodes. Despite the solution method captures the multi-criteria aspects and heterogeneity of the urban transit networks planning, the capacity of the vehicles are not considered in the presented formulations.

[29] presented a vehicle routing with time windows based formulation for the integrated timetabling and VSP with balanced departure times, considering homogeneous fleet. Due to the complexity of the formulation, the authors developed a hybrid metaheuristic framework, decomposing the problem into scheduling and a balancing component. The authors applied their approach in two lines with higher trip frequency and three lines with lower trip frequency in a rural public transport network in Germany. Given the high CPU times spent to solve these small problems, it is unclear whether if the developed method can be directly applied to more generic real-world instances, which can involve thousands of trips and hundreds of vehicles. [23] and [24] integrated passengers into the problem of integrating VSP and timetabling. However, to cope with the problem complexity introduced extreme assumptions, such as the nonexistence of deadheads, restricting their application to handle real-world problems.

More recently, [7] solved the MDVSP with trip shifting, in which the overall quality of the timetabling is controlled using the following criteria: the number of shifted trips, the headway between the consecutive trips of a line, and the quality of some passenger connections. To solve the problem, the authors developed a two-phase heuristic. The first phase finds suitable vehicle schedules (using a column generation heuristic), while the second phase optimizes the schedules of the first phase to generate a timetabling. The authors considered a homogeneous fleet and did not consider the peak demand for each trip.

We can conclude that works on conceding flexibility to the VSP with multiple vehicle types are somewhat limited and that [31] presented a problem that is most similar to our study. However, our approach allows to exogenously control the level of service to passengers by limiting the maximum interval in which timetabled trips can be shifted. Further, our approach can be used more strategically to fleet redimensioning, in situations of demand volatility, as is the current case in Brazil [6], by considering the peak demand for each trip.

3 PROBLEM STATEMENT AND MODELING

The VTSP can be defined as follows. For a given set of trips from a predefined timetable, estimated peak passenger demand for each trip, travel times between all pairs of stations, and a set of vehicles of different types, find a feasible minimum-cost bus schedule, in which (i) each trip is assigned to a single vehicle; (ii) the capacity of each scheduled vehicle must not be exceeded the peak passenger demand of the assigned trip; and (iii) each scheduled vehicle performs a sequence of compatible trips [31]. Trips i and j are a compatible pair of trips if the same vehicle can reach the starting point of trip j after it finishes trip i.

In this section, we introduce a variant of the VTSP known as VTSP with Sequential Change of Timetabling, or simply VTSP-SCT, in which it is possible to introduce changes in a predefined timetabling. This variant enables two or more trips to be combined into a single trip, as long as the resulting peak demand be compatible with the capacity of some vehicle of the fleet.

Following [14], we assume that there is available information about possible scenarios to cope with the assumption that demand is known beforehand. For example, a holiday scenario will likely have different necessities than a regular working day; Thus the referring demands will also differ.

Regarding the VTSP-SCT, our objective is to minimize fixed and operating costs over an underlying feasible network, exploiting several scheduling possibilities for different vehicle types. An efficient network representation is needed to reach such an objective. Before describing the mathematical formulation for the problem, we first introduce the applied network model.

3.1 Underlying network

The TSN was firstly described for the vehicle scheduling problem in [21] and subsequently applied in [20], [22], [19], and [30]. Previously, the VSP was formulated using a connection network [3]. The TSN can be defined as a cyclic directed graph $G = (N,A)$ where N is the set of nodes and A is the set of arcs. Each node $n \in N$ represents a specific station s or the depot d at a particular time t. Let S be the set of stations and T the planning horizon, set N can be defined as $N = \{(s,t) : s \in S \cup \{d\},\ t \in T\}$. Set A is composed of six subsets, $A = A^{se} \cup A^{wait} \cup A^{dh} \cup A^{pout} \cup A^{pin} \cup A^{c}$, defined as follows:

- A^{se} is the set of service arcs that connect the corresponding departure and arrival nodes at the start and end locations of a timetabled trip. There is an exact correspondence for every timetabled trip to one service arc. Thus, each arc $(i,j) \in A^{se}$ has a demand of P_{ij} passengers.
- A^{wait} is the set of waiting arcs in which a vehicle is waiting at a station/depot before starting a new trip.
- A^{dh} is the set of deadheading arcs which represents trips with no passengers between a compatible pair of service trips, specifically from the end node of the first arc to the start node of the second one.
- A^{pout} is the set of pull-out arcs, expressing arcs departing from depot d to some station $s \in S$.
- A^{pin} is the set of pull-in arcs, expressing arcs departing from some station $s \in S$ to depot d.
- A^{c} is the set of circulation arcs, such that each arc connects the last node in the depot timeline to the first node in this timeline. As the TSN was designed to represent one working day, the amount of unit flows of circulation arc represents the total of scheduled vehicles.

Figure 1 depicts a TSN to one depot (*Depot*) and three stations (A, B and C). For a heterogeneous fleet, the network has a multi-layer structure, with one layer for each vehicle type. The service, deadhead, pull-in/out and circulation arcs denote the bus in movement while waiting arcs represent the bus stopped at a station or in the depot [30].

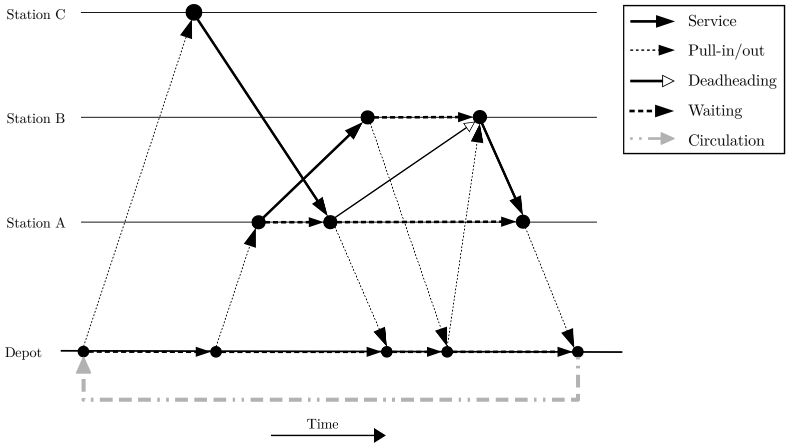


Figure 1 – Example of a time-space network to one depot and three stations

3.2 Mathematical Formulation for the VTSP

The VTSP can be modeled as follows based on [31]. Let P_{ij} be the demand for a service arc $(i,j) \in A^{se}$ and let F be the set of vehicle types. Let p_f be the capacity of each vehicle type $f \in F$ and c_{ijf} be the cost of vehicle type $f \in F$ traveling arc $(i,j) \in A$. Introducing integer decision variable x_{ijf}, representing the flow of vehicle type $f \in F$ in arc $(i,j) \in A$, the model is formalized as follows:

Model VTSP:

$$\min \sum_{(i,j)\in A} \sum_{f\in F} c_{ijf} x_{ijf} \tag{1}$$

s.t.

$$\sum_{(i,j)\in A} x_{ijf} - \sum_{(j,l)\in A} x_{jlf} = 0 \qquad \forall j \in N, \forall f \in F \tag{2}$$

$$\sum_{f\in F:\, p_f \geq P_{ij}} x_{ijf} = 1 \qquad \forall (i,j) \in A^{se} \tag{3}$$

$$x_{ijf} \in \mathbb{N} \qquad \forall (i,j) \in A, \forall f \in F \tag{4}$$

The objective function (1) minimizes the total vehicle costs. Constraints (2) ensure the flow properties of the network. Constraints (3) assure that all service trips are operated precisely once for a single vehicle type with enough capacity to attend the trip demand. Constraints (4) define the domain of the decision variables.

The multilayer network used with the VTSP to consider different vehicle types cannot be decomposed to solve each layer independently. The VTSP can be described as a multi-commodity flow problem, analogously to the MDVSP, a well-known NP-hard problem [1]. As a result, the VTSP is also an NP-hard problem [31].

As the VTSP is challenging to solve, requiring a large number of decision variables to be modeled, [20] and [31] proposed preprocessing routines that conveniently decreases the number of deadheading arcs and depot nodes in the TSN, allowing to solve the problem using a standard MILP solver (see Section 5). For a detailed description of the TSN reduction procedures, please refer to the above-aforementioned papers.

3.3 Preprocessing and Mathematical Formulation for the VTSP-SCT

As introduced in Section 3, the VTSP-SCT enables timetabling flexibility by allowing the combination of two or more service trips with departure times within a predetermined interval. Algorithm 1 is used to detect possible trips to be grouped, composing time intervals $k \in K$. Each time interval $k \in K$ has an associated set of arcs A_k^{se} which have the same departure and arrival stations. Furthermore, the most substantial difference between the start time of any pair of arcs in A_k^{se} is limited by a time parameter Δt. Sets A_k^{se}, $k \in K$, are disjoint.

Algorithm 1 Generation of time intervals used in VTSP-SCT.

```
procedure GenerateSCT($A^{se}$, $\Delta t$)
    $K \leftarrow \emptyset$
    $k \leftarrow 0$
    $A^{se} \leftarrow$ sort trips of $A^{se}$ by their start times
    if $\Delta t > 0$ then
        for all $s \in S$ do
            for all $w \in S$ do
                for $t \in T$ do
                    $A_k^{se} \leftarrow$ all arcs from $A^{se}$ departing from $s$ to $w$ with start time in $[t,\ t+\Delta t]$
                    if $|A_k^{se}| > 1$ then
                        $P_k \leftarrow \sum_{(i,j)\in A_k^{se}} P_{ij}$
                        $K \leftarrow K \cup \{k\}$
                        $A^{se} \leftarrow A^{se} \setminus A_k^{se}$
                        $k \leftarrow k+1$
                    end if
                end for
            end for
        end for
    end if
    return $(K, A_k^{se})$
end procedure
```

The Algorithm 1 takes the set A^{se} and time parameter Δt as input. For each pair of departure and arrival stations $s \in S$, $w \in S$ and a time $t \in T$, the routine selects all arcs from $A^{se} \setminus \bigcup_{k' \in K} A^{se}_{k'}$ which depart from s to w and have start time in $[t,\, t+\Delta t]$. This selection is further used to create set A^{se}_k of combinable arcs of time interval k subjected to demand parameter P_k. By its design, the routine outputs $K = \emptyset$ to $\Delta t = 0$.

Introducing parameter P_k as the total demand of each time interval $k \in K$, the VTSP-SCT can be formulated as follows:

VTSP-SCT Model:

$$\min \sum_{(i,j)\in A} \sum_{f \in F} c_{ijf} x_{ijf} \tag{1}$$

s.t.

$$\sum_{(i,j)\in A} x_{ijf} - \sum_{(j,l)\in A} x_{jlf} = 0 \qquad \forall j \in N, \forall f \in F \tag{2}$$

$$\sum_{(i,j)\in A^{se}_k} \sum_{f \in F} p_f x_{ijf} \geq P_k \qquad \forall k \in K \tag{5}$$

$$\sum_{f \in F} x_{ijf} \leq 1 \qquad \forall k \in K, \forall (i,j) \in A^{se}_k \tag{6}$$

$$\sum_{f \in F : p_f \geq P_{ij}} x_{ijf} = 1 \qquad \forall (i,j) \in A^{se} \setminus \bigcup_{k \in K} A^{se}_k \tag{7}$$

$$x_{ijf} \in \mathbb{N} \qquad \forall (i,j) \in A, \forall f \in F \tag{4}$$

This model has the same objective function (1) of the VTSP, and the same constraints (2) and (4) related to the conservation of flow and decision variables domain, respectively. Constraints (5) ensure that the combined capacity of vehicles that perform the service trips of time interval k met at least the total demand P_k. Constraints (6) guarantee that each service arc belonging to any interval $k \in K$ is operated at most once for a single vehicle type. These two last sets of constraints allow changes in the timetable. Constraints (7) assure that service trips that are not in any time interval $k \in K$ are operated precisely once by a single vehicle type with enough capacity to attend the trip demand.

Since the number of vehicles which met the total demand P_k can be smaller than the number of service trips in the interval k, some unperformed arcs of A^{se}_k might be removed from the timetabling in the optimal solution, having their demand redistributed between the remaining arcs of the interval.

3.4 Including TW constraints to the VTSP-SCT

We propose a new TW implementation in which TW arcs are generated similarly to waiting arcs, which ease its integration to the VTSP and the VTSP-SCT. In our approach, we extend the set of arcs of the underlying network [21] with the set of TW arcs, $A = A \cup \{A_{tw}\}$. With a simple implementation, a TW arc is used to link two service arcs that have at least one station in common and have a difference between departure and arrival time less than T_{tw} minutes. TW can be expressed as *reversed waiting arcs*, i.e., waiting arcs implemented in reverse “time” direction. This approach increases the number of arcs, but it keeps the number of nodes unchanged.

In Figure 2, we illustrate the insertion of a TW arc (color blue) in the network of Figure 1. This arc enables the compatibility between two service trips and allows the saving of one vehicle from the VTSP optimal solution.

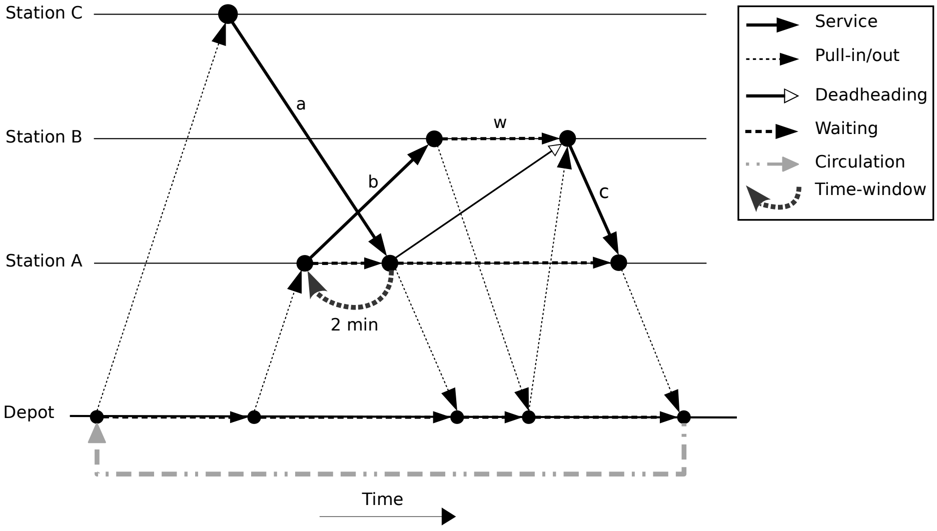


Figure 2 – Network on Figure 1 with time window arc.

For a maximum T_{tw}=2 minutes, Figure 2 (highlighted in blue) enables the synchronization of two trips[1], *a* and *b*, subtly modifying the timetable for the itinerary served by these trips. Such synchronization can minimize the number of scheduled vehicles and reduce the number of express trips, resulting in lower operating costs. For a better understanding of the time window arcs application, Table 1 simulates a possible timetable applied to the example of Figure 2.

As shown in Table 1, a solution without (w/o) time windows need two vehicles to run the trips *a* and *c*, while the solution with (w/) time windows (T_{tw}=2 minutes) needs just one. Note that for trip *b*, the solution with time windows entails in an adjustment at timetabling, modifying the departure time to 08:20, at station A, and arrival time to 08:46, at station B. It should be noted that to reduce the impact of the small delays caused by the use of time window arcs (at trip *b*, for example), we impose the use of subsequent waiting arcs (arcs *w*) following the delayed service trips (arcs *b*). The duration time of these subsequent waiting arcs must be, at least, equal to the duration time of used time windows arc (T_{tw}). Thus, the delay of the arrival time of the

[1] For ease of reading, in this case, trips *a*, *b*, and *c* will be used as synonyms of trips of arcs *a*, *b*, and *c*, respectively.

Table 1 – Two solutions for the timetable for network of Figure 2.

Trip ID	Start time	Finish time	Origin	Dest.	Solution w/o time windows	Solution w/ time windows
a	08:00	08:20	C	A	v_1	v_1
b	08:18	08:44	A	B	v_2	v_1
c	08:55	09:26	B	A	v_1	v_1
Amount of vehicles used					2	1

trip *b* is deducted from the subsequent waiting time, and the operating times of the trip *c* remain unchanged.

Constraints (8) and (9) are included in VTSP-SCT-TW mathematical model to ensure this condition.

Small T_{tw} values are used to reduce changes in the timetabling, avoiding possible disturbances to passengers. Thus, delays may remain unnoticed by passengers due to changes in speed, traffic jams and the time spent at the bus stop for embarking and disembarking passengers. Finally, another strategy applied to minimize the disturbance of timetabling is to penalize TW arcs with a high cost to discourage the use of this type of arc.

The proposed TW implementation can be easily integrated with VTSP and VTSP-SCT models. The two new models can be defined as follows:

VTSP-TW Model:

$$\min \sum_{(i,j)\in A} \sum_{f\in F} c_{ijf} x_{ijf} \tag{1}$$

s.t.

$$\sum_{(i,j)\in A} x_{ijf} - \sum_{(j,l)\in A} x_{jlf} = 0 \qquad \forall j \in N, \forall f \in F \tag{2}$$

$$\sum_{f\in F : p_f \geq P_{ij}} x_{ijf} = 1 \qquad \forall (i,j) \in A^{se} \tag{3}$$

$$x_{hif} \leq x_{ijf} \qquad \forall (h,i) \in A^{tw}, \forall (i,j) \in A^{se}, \forall f \in F \tag{8}$$

$$x_{hif} \leq x_{jlf} \qquad \forall (h,i) \in A^{tw}, \forall (j,l) \in A^{wait}, \forall (i,j) \in A^{se}, \forall f \in F \tag{9}$$

$$x_{ijf} \in \mathbb{N} \qquad \forall (i,j) \in A, \forall f \in F \tag{4}$$

VTSP-SCT-TW Model:

$$\min \sum_{(i,j)\in A} \sum_{f\in F} c_{ijf} x_{ijf} \tag{1}$$

s.t.

$$\sum_{(i,j)\in A} x_{ijf} - \sum_{(j,l)\in A} x_{jlf} = 0 \qquad \forall j \in N, \forall f \in F \tag{2}$$

$$\sum_{(i,j)\in A_k^{se}} \sum_{f\in F} p_f x_{ijf} \geq P_k \qquad \forall k \in K \tag{5}$$

$$\sum_{f\in F} x_{ijf} \leq 1 \qquad \forall k \in K, \forall (i,j) \in A_k^{se} \tag{6}$$

$$\sum_{f\in F: p_f \geq P_{ij}} x_{ijf} = 1 \qquad \forall (i,j) \in A^{se} \setminus \bigcup_{k\in K} A_k^{se} \tag{7}$$

$$x_{hif} \leq x_{ijf} \qquad \forall (h,i) \in A^{tw}, \forall (i,j) \in A^{se}, \forall f \in F \tag{8}$$

$$x_{hif} \leq x_{jlf} \qquad \forall (h,i) \in A^{tw}, \forall (j,l) \in A^{wait}, \forall (i,j) \in A^{se}, \forall f \in F \tag{9}$$

$$x_{ijf} \in \mathbb{N} \qquad \forall (i,j) \in A, \forall f \in F \tag{4}$$

Objective function (1) and constraints (2–4) and (5–7) are the same of the previous models. The novelty is in the inclusion of constraints (8) and (9). They ensure that when a TW arc is used, compatible service and its subsequence waiting arcs must also be used.

4 EXPERIMENTS CONFIGURATION

The models presented in the previous section were implemented in C++ programming language and solved with IBM ILOG CPLEX Optimization Studio 12.7.1. The experiments were made in PC on a 2.70 GHz Intel Xeon E2697 CPU and 32 GB RAM.

As the use of artificial instances is a well-consolidated approach to conduct experiments and evaluate solution methods for the VSPs [3, 26], we generated our benchmarking instances based on real data from the transportation system of a Brazilian city. Unfortunately, there is no available method to generate instances considering demands for trips, and therefore suitable for the VTSP and its variants. We had to develop our method in which an instance is generated through three steps.

We adapted [3]'s method to generate a set of bus lines. Each line has a starting and ending relief points, i and j, respectively, generated by a uniform integer distribution. A relief point represents a point (x,y) on the Cartesian plane with coordinates $1 \leq x \leq 60$ and $1 \leq y \leq 60$. A bus line must belong to one of the following classes: *short* or *long itineraries*. A *short itinerary* had their trip length t_{len} given by a uniform distribution $U(\tau_{ij}+5,\ \tau_{ij}+40)$, where τ_{ij} is the Euclidean distance between starting and ending relief points of the trip. The class of *long itineraries*, as the name suggests, have a larger trip length t_{len} given by uniform integer distribution $U(180, 300)$.

Following [3], we imposed that $i = j$ for long itineraries, and used a proportion of 60% of long itinerary bus lines and 40% of short ones.

In the second step of the generation procedure, we start the construction of the timetable. Each trip i performs the itinerary of some bus line, which is used to compute the start (st_i) and finish (et_i) times of the trip. The starting time of trip i for a short itinerary is generated with the following probabilities: 15% of probability of $st_i \in U(60, 360)$; 70% of probability of $st_i \in U(360, 1080)$; and 15% of probability of $st_i \in U(1080, 1380)$. The procedure is more straightforward for trips that perform long itineraries. The starting time of trip i follows an uniform integer distribution $U(300, 1200)$. Finally, the ending time et_i of trip i is computed as $st_i + t_{len}$.

The final step of the instance generation covers the demand simulation. We used the approach developed by [11] to generate the demand of each trip of the timetabling based on the trip starting time. The trip demand is computed by a weighted sum of three Gaussian distributions, with averages of 7, 12, and 18, standard deviations of 2, 1.5, and 3, and weights of 4.5, 1, and 5, respectively. Thus, the demand of each trip is computed based on the probability value and the capacity of the largest vehicle of the fleet. The resulting curve is shown in Figure 3, where the averages represent peak demand hours.

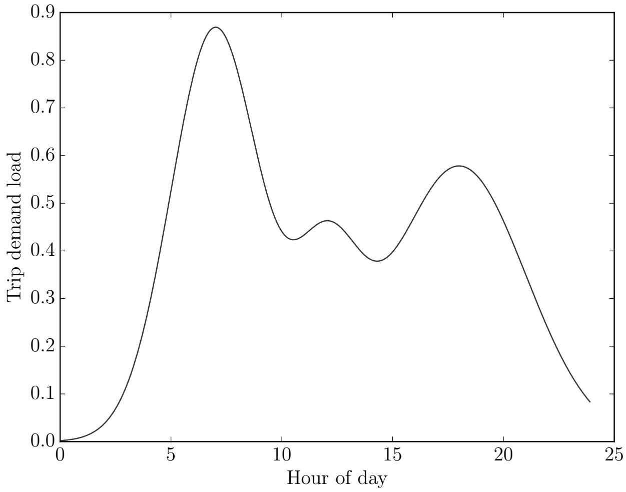


Figure 3 – Probability density function used in demand simulation.

In the computational experiments, we used a heterogeneous fleet composed of three different vehicle types: type A, corresponding to an articulated bus, with a capacity to 141 passengers; type B with a capacity to 100 passengers; and type C, able to carry up to 83 passengers. From data provided by Brazilian transportation companies, vehicle costs were defined according to their capacities. Vehicle types A, B, and C have a fixed cost factor (c_f) of 1.7, 1.2 and 1, respectively.

The costs of service, waiting and deadheading arcs were calculated based on their respective length. Circulation and TW arcs were penalized with a high cost value to guarantee the objectives of minimum cost flow problem, following [26], and to reduce the introduction of delays in the timetable. Arc costs were computed as follows:

$$c_{ijf} = c_f * \begin{cases} 250 & \forall(i,j) \in A^{pin} \cup A^{pout} \\ 10^6 & \forall(i,j) \in A^{c} \\ 2000 & \forall(i,j) \in A^{tw} \\ 10*(t_j - t_i) & \forall(i,j) \in A^{se} \\ 8*(t_j - t_i) & \forall(i,j) \in A^{dh} \\ 1*(t_j - t_i) & \forall(i,j) \in A^{wait}, s_i = s_j \neq d \\ 0*(t_j - t_i) & \forall(i,j) \in A^{wait}, s_i = s_j = d \end{cases} \tag{10}$$

Two different experiments were carried out on benchmark instances to evaluate the developed models as follows: (i) to compare the results of the proposed VTSP-TW and the approach described in [20, 19]; and (ii) to compare the results of VTSP-SCT, VTSP-TW, and VTSP-SCT-TW models.

For the first computational experiment, we implemented in C++ the approach described in [20, 19]. We refer to this implementation as KBS. Following [20, 19], we also applied a higher fixed cost to TW arcs to ensure that only arcs that imply in vehicles savings will appear in the optimal solution.

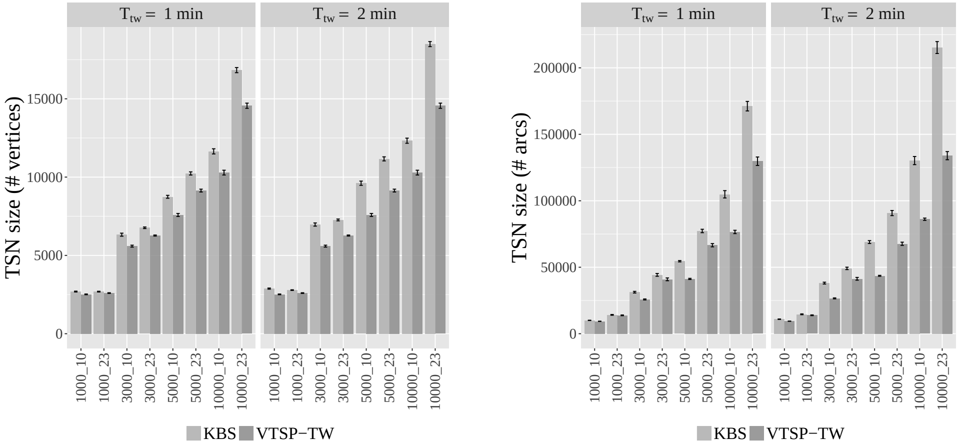
Table 2 shows the characteristics of the generated benchmark instances, which totalize 40 instances grouped in eight configurations, named according to the number of trips and number of stations.

Table 2 – Characteristics of benchmark instances.

Configuration	Trips	Stations
1000_10	1000	10
1000_23	1000	23
3000_10	3000	10
3000_23	3000	23
5000_10	5000	10
5000_23	5000	23
10000_10	10,000	10
10000_23	10,000	23

5 COMPUTATIONAL EXPERIMENTS

Our first analysis refers to the comparison of the size of the network structure for KBS and VTSP-TW models. Figure 4 shows a comparison of TSN network size to benchmark instances, regarding vertices and arcs, for $T_{tw} = \{1,2\}$ minutes.



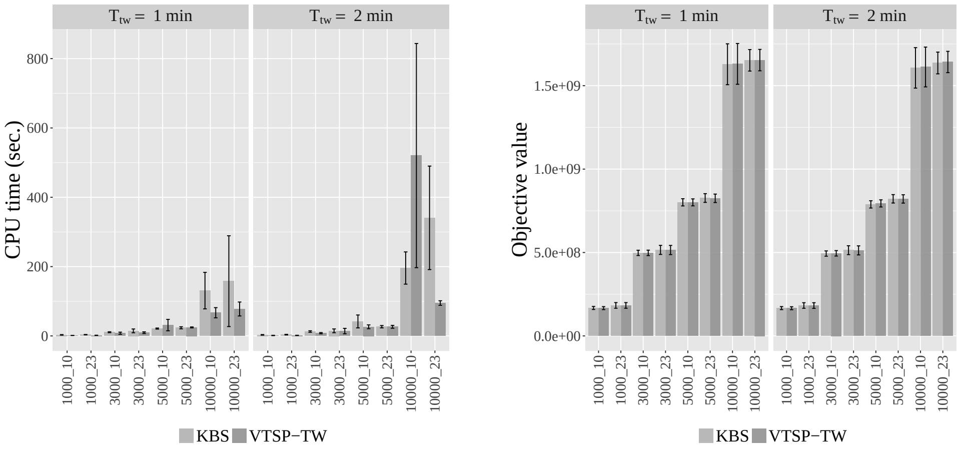
(a) (b)

Figure 4 – Comparison of network size for KBS and VTSP-TW models.

As claimed on Section 2, "shadow" arcs, implemented in KBS model as time window arcs, lead to a substantial increase of the network size than the proposed implementation of time windows at VTSP-TW model. The difference of network sizes for both KBS and VTSP-TW is negligible to instances with 1000 trips. On instances with 3000 or more trips, KBS model creates a greater network size, due to the application of "shadow" arcs. On average, KBS model has at most 36.2% more arcs and 17.4% more vertices than VTSP-TW model. It is also important to note that for the VTSP-TW model there is a smaller growth rate in the network size when increasing both the size of the instances and the time window intervals.

In Table 3 we compares the results obtained by VTSP-TW and KBS models with $T_{tw} \in \{1,2\}$ for five instances of each configuration. Column *O.F.* refers to the average optimal solution value found by the MILP solver, column *CPU* (*sec.*) corresponds to the average CPU time (in seconds) required to solve the problem to optimality, column *# Vehicles* indicates the average number of scheduled vehicles of each vehicle type, and column *Active TW* corresponds to the average number of active TW arcs in the optimal solution.

Regarding the computational time of the tested models, usually, VTSP-TW and KBS tend to consume more CPU time to solve configurations with 10 than 23 stations, mainly due to the increasing number of trip compatibilities. Figure 5 shown a comparative for both models considering the CPU times and the objective function value of instances detailed in Table 2.



(a)

(b)

Figure 5 – Comparison of cost and solution time for KBS and VTSP-TW models.

As an example, instances with 10.000 trips and 10 stations took at least 5 times more CPU usage than instances with 10.000 trips and 23 stations with VTSP-TW and $T_{tw} = 2$ min. VTSP-TW increased the CPU usage 3.7 times, on average, in comparison with the KBS model. In general, instances up to 5000 trips were solved to optimality within 30 seconds, where the configuration 5000_10 /T_{tw} = 2 took up to 41.82 seconds to be solved by KBS model. An independent-samples t-test was conducted to compare the CPU times required to solve instances with 10 and 23 stations for both models. There was no significant difference in the scores for 10, arithmetic mean ($\bar{x}$)=67.69, standard deviation (s)=148.82, and 23 stations ($\bar{x}$=52.25, s=96.70), for t(160)=0.7776, and p = 0.2191.

As shown in Figure 5b, this result probably occurs due to the time spent by VTSP-TW model to solve some of the 10000_10 instances (T_{tw} = 2 minutes), which significantly increased the average CPU time of such configuration.

Excepting configuration 10000_10 / T_{tw} = 2 minutes, VTSP-TW model was, on average, about 1.5 times faster than KBS model. This result is confirmed by an independent-sample t-test comparing CPU time required to solve both VTSP-TW and KBS models for all instances (excepting instance 10000_10, T_{tw} = 2 minutes). There was a significant difference of scores between VTSP-TW ($\bar{x}$=21.34, s=24.61) and KBS models ($\bar{x}$=33.46, s=58.28), for statistical values of t(140)=1.6017, and p = 0.05631.

Regarding the total number of scheduled vehicles, results were on average the same for both models, as follows: VTSP-TW ($\bar{x}$=711.12, s=498.35); KBS model ($\bar{x}$=719.36, s=497.44). For some instances, the number of scheduled vehicles by VTSP-TW is slightly different from the obtained by KBS model, but considering the average of all instances, the difference value is smaller than 1%. This result can be clearly verified when comparing the means and standard deviations from Figure 5a for both models.

Table 3 – Computational results for different TW approaches.

Model	Instance	T_{tw}	O.F.	CPU (sec.)	# Vehicles				# Active TW
					A	B	C	Total	
KBS	1000_10	1	1.67699×10^8	3.09	30.40	111.40	1.40	143.20	11.00
	1000_10	2	1.66921×10^8	3.11	30.20	111.20	2.80	144.20	10.80
	1000_23	1	1.82981×10^8	3.72	34.60	115.80	0.60	151.00	14.40
	1000_23	2	1.82581×10^8	3.75	34.60	115.40	1.40	151.40	14.40
	3000_10	1	4.97884×10^8	10.75	85.00	336.40	8.80	430.20	33.00
	3000_10	2	4.93905×10^8	12.73	83.40	335.00	31.40	449.80	32.60
	3000_23	1	5.15722×10^8	14.61	97.60	334.60	3.00	435.20	35.60
	3000_23	2	5.13968×10^8	14.71	97.80	332.60	8.60	439.00	35.60
	5000_10	1	8.00922×10^8	21.28	137.60	536.40	19.00	693.00	55.20
	5000_10	2	7.88420×10^8	41.82	134.80	528.20	99.80	762.80	54.60
	5000_23	1	8.26756×10^8	23.52	135.60	565.60	10.40	711.60	54.60
	5000_23	2	8.21933×10^8	26.66	134.40	562.20	30.40	727.00	54.60
	10000_10	1	1.62851×10^9	130.73	107.00	277.40	1102.20	1486.60	87.20
	10000_10	2	1.60690×10^9	195.92	106.00	274.40	1085.40	1465.80	318.40
	10000_23	1	1.65201×10^9	157.97	116.20	283.60	1102.80	1502.60	39.00
	10000_23	2	1.63620×10^9	340.73	115.60	281.60	1090.20	1487.40	132.20
VTSP-TW	1000_10	1	1.66903×10^8	1.41	11.00	30.40	110.60	152.00	2.20
	1000_10	2	1.66206×10^8	1.40	10.80	30.60	110.00	151.40	3.20
	1000_23	1	1.82982×10^8	1.76	14.40	34.60	115.80	164.80	0.60
	1000_23	2	1.82184×10^8	1.60	14.40	34.60	115.00	164.00	1.60
	3000_10	1	4.97973×10^8	7.85	33.00	85.40	336.00	454.40	8.00
	3000_10	2	4.95296×10^8	8.00	32.80	84.40	334.60	451.80	19.00
	3000_23	1	5.15088×10^8	9.41	35.60	96.40	335.40	467.40	4.20
	3000_23	2	5.12743×10^8	14.07	35.60	96.60	332.80	465.00	10.20
	5000_10	1	8.00679×10^8	31.31	55.40	139.60	533.40	728.40	16.40
	5000_10	2	7.94416×10^8	26.37	55.40	137.00	530.20	722.60	42.20
	5000_23	1	8.25036×10^8	24.59	54.80	137.20	561.60	753.60	13.20
	5000_23	2	8.21231×10^8	26.38	54.80	134.00	561.60	750.40	30.00
	10000_10	1	1.63073×10^9	66.99	108.00	280.20	1099.40	1487.60	71.00
	10000_10	2	1.61211×10^9	520.29	107.20	279.60	1082.40	1469.20	279.00
	10000_23	1	1.65342×10^9	77.74	116.40	288.00	1098.60	1503.00	32.40
	10000_23	2	1.64206×10^9	94.93	116.00	286.00	1090.20	1492.20	87.00

Analyzing the number of used TW arcs, VTSP-TW used about 35% less TW arcs than KBS model, but this difference was not statistically significant in a t-test (t(158)=-9.928, p = 0.355). Although VTSP-TW model is less efficient than KBS, the former introduced fewer disruptions to the original timetabling than the latter. Therefore, VTSP-TW is a competitive option to KBS, offering good compromise solutions concerning CPU usage, total fixed and variable costs, and flexible schedules with low impact on passengers.

Another analysis is made in Table 4 compares the performance of models VTSP, VTSP-SCT, VTSP-TW, and VTSP-SCT-TW for 5 instances of each configuration. In this table, we included column *Grouped trips*, which refers to the average number of trips grouped from the original timetable, and replaced column *Solution* by column *SSV* that refers to the average Scaled Solution Value obtained by the models for each problem size, so that the objective function obtained by model VTSP is equal to 1. Computational time for all experiments was limited to 7200 seconds.

As expected, model VTSP is the most efficient to be solved to optimality, since it has fewer constraints. Among the models incorporating flexibility, model VTSP-SCT-TW, in general, was the most efficient, with exception of three test cases of some instances of 10000 trips (signaled with a '*' in Table 4). VTSP-SCT model led to higher cost savings than VTSP model, on average, 7.2%, 10.4%, and 13.4% for $\Delta t = 1$, $\Delta t = 2$ and $\Delta t = 3$, respectively. For instance, with 10000 trips, the savings on the average number of vehicles exceeded 23%, considering $\Delta t = 3$. As the number of trips increased, greater was the savings of vehicles obtained by VTSP-SCT. This result was expected, because in the smallest instances trips tend to be separated by larger time intervals, decreasing the number of trips that can be grouped.

The 10000_10 instances required, on average, more time to be processed. In addition to larger problem sizes, the required CPU times also increased with Δt, due to the increase in the number of possibilities for grouping trips. Regarding the strategic and tactical applications, a large CPU time has a less critical role when facing the potential savings introduced of the developed models. Moreover, because many managers solve the VSP manually, which is a very time-consuming task, a period of one to two hours to obtain an optimal computational solution can be considered reasonable.

Comparing VTSP and VTSP-TW, $T_{tw} = 1$ enabled 0.6% less scheduled vehicles, on average, while $T_{tw} = 2$ reduces the scheduled vehicles by 1.3%. The number of vehicles saved with VTSP-TW was lower than with VTSP-SCT. Comparing the number of #Grouped Trips and #Active TW, it was clear that VTSP-TW model performed fewer changes in the timetabling than VTSP-SCT model, explaining the proportion of saved vehicles. In contrast, the CPU time required to solve VTSP-TW was potentially lower than to solve VTSP-SCT, especially for instances with more trips.

Overall, VTSP-SCT-TW model led to the most substantial vehicle savings. For all tested instances, this model showed the largest savings concerning vehicle numbers, without introducing excessive modifications to the timetabling. The results in Table 4 indicated at least 1% more savings, on average, in the number of vehicles by applying VTSP-SCT-TW in comparison with VTSP-SCT model. Instances with 10000 trips obtained more than 24% savings compared to VTSP when $\Delta t = 3$ and TW arcs were allowed. Interesting to note that VTSP-SCT-TW grouped, on average, 0.2% fewer trips than VTSP-SCT, considering the proportional Δt interval, and activated 17.6% more TW arcs than VTSP-TW. Based on these results, it is possible to affirm that VTSP-SCT-TW model introduces minimal changes to the timetabling and offered promising savings in the number of scheduled vehicles when compared to the remaining models.

Table 4 – Computational results for generated instances.

Instance	Model	Δt	T_{tw}	SSV	CPU	# Vehicles				# Grouped	# Active
					(sec.)	A	B	C	Total	trips	TW
1000_10	VTSP	0	0	1.000	1.65	11.17	30.17	112.00	153.33	0.00	0.00
	VTSP-SCT	1	0	0.981	1.65	12.20	31.00	106.20	149.40	0.00	16.40
		2	0	0.973	2.11	12.80	29.60	105.40	147.80	0.00	28.00
		3	0	0.964	2.14	14.20	29.00	102.20	145.40	0.00	38.20
	VTSP-SCT-TW	1	1	0.973	1.51	12.20	30.80	105.00	148.00	2.20	16.20
		2	1	0.965	2.03	12.80	29.40	104.40	146.60	2.00	28.20
		3	1	0.956	2.06	14.20	28.80	101.20	144.20	2.00	38.00
		1	2	0.969	1.55	12.00	30.80	104.60	147.40	3.60	16.60
		2	2	0.961	1.80	12.60	29.40	104.00	146.00	3.40	28.20
		3	2	0.952	1.97	14.00	28.80	100.80	143.60	3.80	38.40
1000_23	VTSP	0	0	1.000	2.07	14.40	35.00	115.80	165.20	0.00	0.00
	VTSP-SCT	1	0	0.987	2.40	15.40	34.40	112.40	162.20	0.00	18.80
		2	0	0.982	2.52	16.40	33.20	111.20	160.80	0.00	26.60
		3	0	0.975	2.93	16.60	33.80	109.00	159.40	0.00	39.00
	VTSP-SCT-TW	1	1	0.984	2.45	15.40	34.00	112.40	161.80	0.60	18.80
		2	1	0.979	2.47	16.40	32.60	111.40	160.40	0.80	27.40
		3	1	0.971	2.67	16.40	33.80	108.60	158.80	1.00	39.80
		1	2	0.980	2.65	15.40	33.80	111.80	161.00	1.80	19.00
		2	2	0.974	2.50	16.40	32.60	110.60	159.60	1.80	27.40
		3	2	0.968	2.69	16.40	33.60	108.20	158.20	1.80	39.40
3000_10	VTSP	0	0	1.000	6.21	33.20	86.20	337.60	457.00	0.00	0.00
	VTSP-SCT	1	0	0.966	20.27	41.60	84.00	309.00	434.60	0.00	146.40
		2	0	0.946	23.25	46.00	81.40	294.80	422.20	0.00	240.00
		3	0	0.930	33.41	49.00	81.60	281.40	412.00	0.00	321.40
	VTSP-SCT-TW	1	1	0.960	22.93	41.20	83.40	307.40	432.00	10.20	147.40
		2	1	0.941	41.01	45.60	80.40	294.20	420.20	9.80	238.00
		3	1	0.924	37.79	47.60	80.20	282.60	410.40	10.40	317.00
		1	2	0.954	41.77	41.20	83.40	304.60	429.20	23.40	147.60
		2	2	0.936	44.50	46.00	81.40	289.80	417.20	22.80	241.40
		3	2	0.920	40.58	47.80	81.00	279.00	407.80	22.20	320.00
3000_23	VTSP	0	0	1.000	7.29	35.60	98.00	335.20	468.80	0.00	0.00
	VTSP-SCT	1	0	0.959	12.34	43.40	90.80	309.60	443.80	0.00	153.40
		2	0	0.942	15.52	47.80	90.20	294.00	432.00	0.00	233.60
		3	0	0.919	23.35	51.60	87.60	278.60	417.80	0.00	311.00
	VTSP-SCT-TW	1	1	0.956	14.50	43.60	89.80	308.60	442.00	4.60	153.00
		2	1	0.938	27.14	48.00	90.20	291.80	430.00	4.80	234.00
		3	1	0.915	65.78	52.40	87.80	275.00	415.20	6.40	313.80
		1	2	0.952	23.43	43.80	89.60	306.40	439.80	12.60	151.60
		2	2	0.934	26.94	48.20	90.00	289.60	427.80	12.20	233.20
		3	2	0.911	39.35	52.40	87.80	273.00	413.20	15.80	314.60
5000_10	VTSP	0	0	1.000	15.35	55.60	139.40	540.00	735.00	0.00	0.00
	VTSP-SCT	1	0	0.944	42.26	71.60	138.40	468.80	678.80	0.00	399.20
		2	0	0.914	41.43	78.00	139.40	433.00	650.40	0.00	619.20
		3	0	0.886	76.11	89.40	138.20	392.80	620.40	0.00	819.20
	VTSP-SCT-TW	1	1	0.936	31.96	72.20	136.80	463.40	672.40	26.80	398.00
		2	1	0.907	49.66	77.80	138.80	428.40	645.00	26.60	619.60
		3	1	0.879	75.35	88.40	138.20	388.40	615.00	24.80	820.20
		1	2	0.928	63.25	70.80	136.60	459.80	667.20	68.20	396.00
		2	2	0.900	164.98	76.60	137.60	426.20	640.40	59.80	615.80
		3	2	0.872	97.38	87.80	137.80	384.60	610.20	56.80	817.80

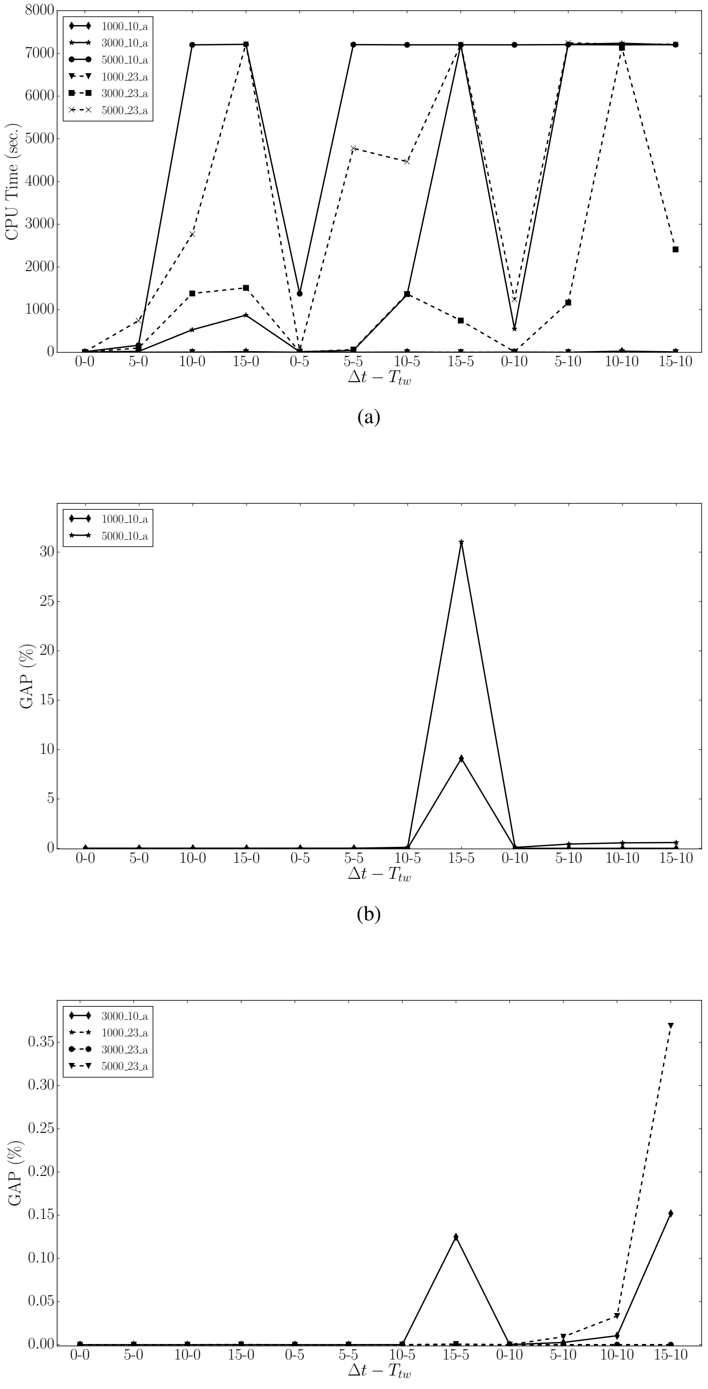
(continued on next page)

Table 4 – (cont.) Computational results for generated instances.

Instance	Model	Δt	T_{tw}	SSV	CPU (sec.)	# Vehicles				# Grouped trips	# Active TW
						A	B	C	Total		
5000_23	VTSP	0	0	1.000	12.43	54.60	136.60	568.20	759.40	0.00	0.00
	VTSP-SCT	1	0	0.945	36.17	76.40	132.60	490.80	699.80	0.00	403.60
		2	0	0.919	108.91	81.80	132.20	460.40	674.40	0.00	623.20
		3	0	0.893	75.99	90.00	130.20	427.20	647.40	0.00	811.20
	VTSP-SCT-TW	1	1	0.939	30.70	74.60	132.20	489.00	695.80	17.60	404.60
		2	1	0.913	90.25	81.20	131.00	457.60	669.80	18.60	620.00
		3	1	0.887	96.92	90.00	129.80	422.80	642.60	16.20	812.40
		1	2	0.934	38.39	75.60	131.00	484.80	691.40	38.40	403.40
		2	2	0.908	74.39	81.80	130.00	454.20	666.00	37.00	616.40
		3	2	0.882	127.88	89.80	131.40	417.00	638.20	40.00	814.00
10000_10	VTSP	0	0	1.000	83.95	108.20	284.00	1108.60	1500.80	0.00	0.00
	VTSP-SCT	1	0	0.904	332.70	178.60	264.60	855.60	1298.80	0.00	1476.00
		2	0	0.902	462.96	202.60	271.60	733.80	1208.00	0.00	2176.80
		3	0	0.860	1494.43	217.40	279.40	648.40	1145.20	0.00	2735.20
	VTSP-SCT-TW	1	1	0.859	472.74	177.20	262.80	847.80	1287.80	88.60	1470.40
		2	1	0.828	729.13	200.00	268.40	729.80	1198.20	89.00	2169.80
		3	1	0.825	3628.73	215.40	275.60	644.80	1135.80	81.60	2731.00
		1	2	0.897*	4662.24	176.00	260.60	835.40	1272.00	314.60	1467.20
		2	2	0.896*	5397.34	197.00	269.00	719.20	1185.20	301.40	2167.20
		3	2	0.852	5214.92	397.00	262.60	539.20	1198.80	215.60	2740.20
10000_23	VTSP	0	0	1.000	107.22	116.40	289.00	1107.00	1512.40	0.00	0.00
	VTSP-SCT	1	0	0.854	361.75	179.60	264.80	866.20	1310.60	0.00	1448.60
		2	0	0.822	1606.85	203.20	268.00	751.80	1223.00	0.00	2152.60
		3	0	0.820	2883.99	219.60	275.20	659.60	1154.40	0.00	2712.00
	VTSP-SCT-TW	1	1	0.887	411.41	178.60	265.00	858.80	1302.40	52.00	1437.40
		2	1	0.890	726.23	204.20	263.80	746.00	1214.00	53.80	2146.20
		3	1	0.844	1955.73	218.80	272.20	656.20	1147.20	44.60	2703.00
		1	2	0.847	934.32	178.80	259.80	853.40	1292.00	133.00	1440.20
		2	2	0.938	1841.96	204.00	263.60	736.00	1203.60	130.40	2142.80
		3	2	0.815*	4843.70	213.60	274.80	652.60	1141.00	115.00	2705.60

We also tested benchmark instances up to 5000 trips applying VTSP-SCT-TW model to larger values of $\Delta t \in \{5,10,15\}$ and $T_{tw} \in \{5,10\}$, towards evaluating the robustness of a commercial MILP to solve more difficult configurations. Figure 6 presents the results for instances with 1000, 3000 and 5000 trips regarding CPLEX gaps and required CPU time.

As expected, VTSP-SCT-TW model became more difficult to be solved with larger Δt and T_{tw} parameter values. The higher of both possibilities of aggregated trips and insertion of TW arcs significantly increased the number of constraints in the model, directly affecting the CPU usage. Instances with 10 stations remained challenging to be solved, reflecting higher CPU times (Figure 6a) and higher gap values (Figures 6b and 6c), since less station concentrates more service trips. Considering that the developed flexibility approaches focuses mainly on strategic planning, the implemented model can be solved for a quite broad range of configuration by a robust commercial MILP solver. Further, it should be noted that high values of Δt and T_{tw} might become unacceptable to regulating authorities due to their undesired impact on passengers' everyday life.



(a)

(b)

(c)

Figure 6 – Impact of increasing Δt and $\mathrm{T_{tw}}$ in (6a) CPU time and (6b, 6c) *GAP* values.

6 CONCLUSION

We extended the ILP model of [31] to solve a variant of the single-depot VSP, in which heterogeneous fleet and timetabling flexibility are simultaneously considered. To effectively enable timetabling flexibility, we proposed two different approaches: the inclusion of TW arcs in the network, and the grouping of trips, considering the possibility of a higher capacity vehicle to perform two or more closed starting time trips in the original scheduling.

The developed approach for the insertion of TW arcs creates small changes in the network, offering the following advantages in comparison with previous implementations described in the literature [20, 19]: (i) preprocessing to remove uninteresting TW arcs is not needed, and (ii) VTSP-TW model leads to fewer disruptions to the original timetabling than the KBS model. Based on the results, VTSP-TW model emerges as a suitable alternative approach to solve large instances of this problem, with very similar CPU times and optimal solution values.

The other developed approach considers the possibility of grouping multiple trips into a single one, making use of a larger vehicle type. Since the timetable presents irregular headways, in which the departure times of trips do not have a fixed interval, grouping trips are possible in short time intervals. This approach, based on computational experiments, resulted in small changes to the timetable and no significant effect on passengers' experience.

The results obtained in the computational experiments demonstrated that our approach has potential to significantly reduce the total costs related to the operation of public transit systems, by decreasing the number of scheduled vehicles, while satisfying the demand. For a problem including 10000 trips, the processing time can take up to two hours, which is quite acceptable, considering the effort necessary to solve the problem manually.

This first approach to the VTSP problem, considering timetable integration, is a case study applied to the Brazilian reality. However, the main idea of the problem tends to be applied to different timetabling and vehicle scheduling problems. Then, we intend to extend our future research efforts in two directions as follows: (i) to implement a heuristic approach towards increasing the solution efficiency of the models, allowing to increase the number of scenarios to be evaluated; and (ii) to contemplate multiple depots in the modeling approach.

ACKNOWLEDGEMENTS

This research has been supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Brazil, by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Brazil, and by Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS). The last author would also thanks to CAPES for his PhD scholarship.

REFERENCES

[1] BERTOSSI AA, CARRARESI P & GALLO G. 1987. On some matching problems arising in vehicle scheduling models. *Networks*, **17**(3): 271–281.

[2] BUNTE S & KLIEWER N. 2009. An overview on vehicle scheduling models. *Journal of Public Transport*, **1**(4): 299–317.

[3] CARPANETO G, DELL'AMICO M, FISCHETTI M & TOTH P. 1989. A branch and bound algorithm for the multiple depot vehicle scheduling problem. *Networks*, **19**(5): 531–548.

[4] CEDER A. 2011. Public-transport vehicle scheduling with multi vehicle type. *Transportation Research Part C: Emerging Technologies*, **19**(3): 485–497.

[5] CEDER A, HASSOLD S & DANO B. 2013. Approaching even-load and even-headway transit timetables using different bus sizes. *Public Transport*, **5**(3): 193–217.

[6] CNT. 2017, Confederação Nacional do Transporte. Ônibus urbanos enfrentam endividamento e queda de demanda. `http://www.cnt.org.br/imprensa/noticia/onibus-urbanos-endividamento-queda-demand`, acesso em: 26 jun. 2018.

[7] DESFONTAINES L & DESAULNIERS G. 2018. Multiple depot vehicle scheduling with controlled trip shifting. *Transportation Research Part B: Methodological*, **113**: 34 – 53, `https://www.sciencedirect.com/science/article/pii/S0191261517310652`.

[8] DT. 2017, Diário do Transporte. Frota de ônibus de São Paulo: em 4,5 anos, número de articulados sobe mais de 80% na cidade. `https://diariodotransporte.com.br/2017/07/28/frota-de-onibus-de-sao-paulo-articulados/`, acesso em: 26 jun. 2018.

[9] FARAHANI RZ, MIANDOABCHI E, SZETO W & RASHIDI H. 2013. A review of urban transportation network design problems. *European Journal of Operational Research*, **229**(2): 281–302.

[10] FRELING R, WAGELMANS APM & PAIXÃO JMP. 2001. Models and algorithms for single-depot vehicle scheduling. *Transportation Science*, **35**(2): 165–180.

[11] GUEDES PC & BORENSTEIN D. 2015. Column generation based heuristic framework for the multiple-depot vehicle type scheduling problem. *Computers & Industrial Engineering*, **90**: 361–370.

[12] GUIHAIRE V & HAO JK. 2008. Transit network re-timetabling and vehicle scheduling. In: *Proceedings of the 2nd International Conference on Modeling, Computation and Optimization in Information Systems and Management Science*, Springer, Heidelberg, pp. 135–144.

[13] GUIHAIRE V & HAO JK. 2010. Transit network timetabling and vehicle assignment for regulating authorities. *Computers & Industrial Engineering*, **59**(1): 16–23.

[14] HUISMAN D, FRELING R & WAGELMANS APM. 2005. Multiple-depot integrated vehicle and crew scheduling. *Transportation Science*, **39**(4): 491–502.

[15] IBARRA-ROJAS O, DELGADO F, GIESEN R & MUOZ J. 2015. Planning, operation, and control of bus transport systems: A literature review. *Transportation Research Part B: Methodological*, **77**: 38 – 75, `http://www.sciencedirect.com/science/article/pii/S0191261515000454`.

[16] IBARRA-ROJAS O, GIESEN R & RIO-SOLIS Y. 2014. An integrated approach for timetabling and vehicle scheduling problems to analyze the trade-off between level of service and operating costs of transit networks. *Transportation Research Part B*, **70**: 35–46.

[17] IBARRA-ROJAS OJ & RIOS-SOLIS YA. 2012. Synchronization of bus timetabling. *Transportation Research Part B: Methodological*, **46**(5): 599–614.

[18] INTERNATIONAL ASSOCIATION OF PUBLIC TRANSPORT. 2010. Data & Statistics. European Union (EU): Latest Data on Urban Bus Fleets. Tech. Rep. 1, New Delhi, India, `www.globalmasstransit.net/templates/print\_preview.html`.

[19] KLIEWER N, AMBERG B & AMBERG B. 2012. Multiple depot vehicle and crew scheduling with time windows for scheduled trips. *Public Transport*, **3**(3): 213–244.

[20] KLIEWER N, BUNTE S & SUHL L. 2006. Time windows for scheduled trips in multiple depot vehicle scheduling. In: *Proceedings of the 11th EURO Working Group on Transportation*, Technical University of Bari, Bari, Italy, pp. 340–346.

[21] KLIEWER N, MELLOULI T & SUHL L. 2002. A new-solution model for multi-depot multi-vehicle-type vehicle scheduling in (sub)urban public transport. In: *13th Mini-EURO Conference and the 9th meeting of the EURO working group on transportation*, Technical University of Bari, Bari, Italy, pp. 604–609.

[22] KLIEWER N, MELLOULI T & SUHL L. 2006. A time space network based exact optimization model for multi-depot bus scheduling. *European Journal of Operational Research*, **175**: 1616–1627.

[23] LAPORTE G, ORTEGA FA, POZO MA & PUERTO J. 2017. Multi-objective integration of timetables, vehicle schedules and user routings in a transit network. *Transportation Research Part B: Methodological*, **98**: 94–112.

[24] LIU T & CEDER AA. 2017. Integrated Public Transport Timetable Synchronization and Vehicle Scheduling with Demand Assignment: A Bi-objective Bi-level Model Using Deficit Function Approach. *Transportation Research Procedia*, **23**: 341 – 361.

[25] MARTNEZ F, BALDOQUN MG & MAUTTONE A. 2017. And solution method to a simultaneous route design and frequency setting problem for a bus rapid transit system in Colombia. *Pesquisa Operacional*, **37**: 403 – 434.

[26] PEPIN AS, DESAULNIERS G, HERTZ A & HUISMAN D. 2009. Comparison of heuristic approaches for the multiple depot vehicle scheduling problem. *Journal of Scheduling*, **12**(1): 17–30.

[27] PETERSEN H, LARSEN A, MADSEN O, PETERSEN B & ROPKE S. 2013. The simultaneous vehicle scheduling and passenger service problem. *Transportation Science*, **47**(4): 603–616.

[28] ROMAN A. 2012. Top 100 Bus Fleets. *Metro Magazine*, (40): 25–30.

[29] SCHMID V & EHMKE JF. 2015. Integrated timetabling and vehicle scheduling with balanced departure times. *OR Spectrum*, **37**(4): 903–928.

[30] STEINZEN I, GINTNER V, SUHL L & KLIEWER N. 2010. A time-space network approach for the integrated vehicle and crew scheduling problem with multiple depots. *Transportation Science*, **44**(3): 367–382.

[31] VAN DEN HEUVEL APR, VAN DEN AKKER JM & VAN KOOTEN NIEKERK ME. 2008. Integrating timetabling and vehicle scheduling in public bus transportation. Tech. Rep. February, Department of Information and Computing Sciences, Utrecht University, The Netherlands.