# On the impact of spatio-temporal granularity of traffic conditions on the quality of pickup and delivery optimal tours

Omar Rifki, Nicolas Chiabaut, Christine Solnon

# On the impact of spatio-temporal granularity of traffic conditions on the quality of pickup and delivery optimal tours

Omar Rifki[1,3], Nicolas Chiabaut[1], Christine Solnon[2]

**Acknowledgements.** This work is supported by the LABEX IMU (ANR-10-LABX-0088) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR). We would like to thank anonymous referees for helpful comments and suggestions.

[☆]Corresponding author. Complete address : ENTPE, laboratoire LICIT, 3 Rue Maurice Audin, 69120 Vaulx-en-Velin, France. Email : omar.rifki@gmail.com

[1]Univ. Gustave Eiffel / Univ. de Lyon, ENTPE / IFSTTAR, LICIT, UMR _T 9401, F-69518, Lyon, France.

[2]Université de Lyon, INSA-Lyon, Inria, CITI, CNRS, LIRIS, F-69621, Villeurbanne, France.

[3]Université de Lyon, cole Urbaine de Lyon, France.

**Abstract**

Optimizing the duration of delivery tours is a crucial issue in urban logistics. In most cases, travel times between locations are considered as constant for the whole optimization horizon. Making these travel times time-dependent is particularly relevant in real urban traffic environments as traffic conditions and thus travel speeds vary according to the time of the day. In this paper, we review the literature on time-dependent routing problems, with a specific focus on benchmarks and performance criteria used to experimentally evaluate the interest of exploiting time-dependent data, showing the lack of studies on the impact of spatio-temporal features of the benchmark on solutions. Hence, we introduce a new benchmark produced from a realistic traffic flow micro-simulation of Lyon city, allowing us to consider different levels of spatial granularity (*i.e.*, number of sensors used to measure traffic conditions) and temporal granularity (*i.e.*, frequency of measures). Finally, we experimentally evaluate the impact of the spatio-temporal granularity on the quality of solutions for different classical problems, including the traveling salesman problem, the pickup and delivery problem, and the dial-a-ride problem.

**Keywords.** Urban freight, Spatio-temporal granularity, Traveling salesman problem, Pickup and delivery problem, Dial-a-ride problem, Time-dependent cost functions

## 1. Introduction

Mobility of goods is vital to urban life [1] but is one of the leading causes of congestion in cities and does, in return, suffer from this issue which has enormous implications [2]. Optimizing freight deliveries is thus a key to rendering mobility systems efficient and make our cities prosper and livable again. This issue is the main objective of the general pickup and delivery problem (GPDP), the goal of which is to minimize the travel time for visiting a given set of locations to pick up and/or deliver goods. Different constraints may be added on the visiting

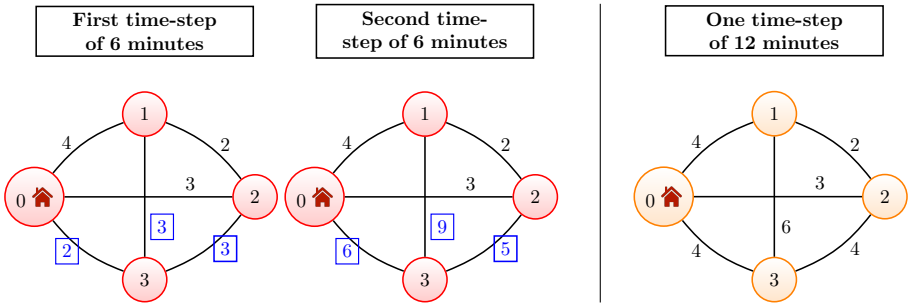


Figure 1: Illustration of the impact of time-step granularity. Left: With two time-steps of 6 minutes, such that costs of $(0, 3)$, $(1, 3)$, and $(3, 2)$ increase between the first and the second time-step due to congestion, the best tour is $T_1 = \langle 0, 3, 1, 2, 0 \rangle$, with a duration of 10 minutes. Right: With one time-step of 12 minutes (such that costs of edges are averaged over the two 6-minute time-steps), the best tour is $T_2 = \langle 0, 1, 2, 3, 0 \rangle$ with a duration of 14 minutes.

order, the loading capacity of the vehicles, the number of available vehicles, or the time windows to respect when visiting locations.

Classically, travel times between locations to visit are assumed to be constant. This is not realistic because traffic conditions are not constant throughout the day, especially in a urban context. As a consequence, quickest paths (*i.e.*, successions of road links), and travel times between locations may change along the day. To fill this lack of realism in the classical GPDP, cost functions that define travel times must be adapted to traffic dynamics and become *time-dependent*. Optimizing freight tours with a realistic traffic dynamic description is thus related to solving the time-dependent GPDP (TD-GPDP), which takes into account variations of travel times during the day [3].

A crucial question is: what is the effect of integrating time-dependency in the quality of the optimal tours, and does this effect depend on the granularity of the function that defines travel times? Fig. 1 shows through a simple toy example how a sole change of the temporal granularity unveils different tours with different durations. This issue may be amplified for complex road networks of thousands of links and for whole day time horizons.

A spatial dimension should also be studied. Indeed, data used to estimate travel times are often coming from sensors, and the number and location of these sensors may have an impact on the quality of travel time estimations which in

turn may have an impact on the quality of optimal tours.

To study the impact of the spatio-temporal granularity of data on tour quality, we introduce a new benchmark that has been built by using a micro-simulation software of the Lyon conurbation as a proxy of the reality. We consider different levels of spatial granularity, by varying the number and position of sensors, and different levels of temporal granularity, by varying the frequency of the measures. This benchmark is used to experimentally evaluate the impact of the spatio-temporal granularity of data on the quality of tours.

The rest of the paper is organized into five sections. In Section 2, we review the literature on TD-GPDPs, with a focus on benchmarks and performance criteria used in experimental evaluations. We show the interest of introducing our new benchmark and motivate our experimental study. In Section 3, we formally define the TD-GPDP, and show how this general problem may be instantiated into well-known specific problems. In Section 4, we describe the dynamic programming approach used to solve TD-GPDPs in our experimental study. In Section 5, we introduce a new benchmark for TD-GPDPs that has been built by using new techniques and simulation models originating from traffic flow theory. In Section 6, we experimentally evaluate the interest of exploiting time-dependent cost functions, and we evaluate the impact of the spatio-temporal granularity of data on the quality of solutions for different TD-GPDP variants.

## 2. Literature Review

A review of time-dependent routing problems is provided in [3]. This review mainly focuses on travel time and speed models (deterministic and stochastic ones), on time-dependent point-to-point route planning that aims at finding the quickest path from an origin to a destination point, on time-dependent multi-point routing problems and on existing approaches for solving these problems up to 2015. However, performance criteria and benchmarks used to evaluate these approaches are not described in this review.

| | Problem | | Constraints | | Solving | Data | | | |
|---|---|---|---|---|---|---|---|---|---|
| Ref (Year) | TSP | VRP | TW | Q | approach | Artificial | #pts | #steps | size |
| [4] (1992) | ✓ | ✓ | ✓ | ✓ | ILP | ✓ | 25 | 3 | - |
| [5] (1996) | ✓ | | | | DP | ✓ | 55 | 3 | - |
| [6] (2002) | ✓ | | | | LS | ✓ | 127 | 2 | - |
| [7] (2003) | | ✓ | ✓ | | LS | ✓ | 100 | 3 | - |
| [8] (2004) | | ✓ | ✓ | | CH | | 786 | 50 | 20 |
| [9] (2005) | | ✓ | ✓ | ✓ | GA | ✓ | 100 | 30 | 60 |
| [10] (2005) | ✓ | | | | CH | ✓ | 127 | 2 | 180 |
| [11] (2006) | | ✓ | ✓ | ✓ | LS | | 19 | 15 | 15 |
| [12] (2008) | | ✓ | | ✓ | LS | ✓ | 80 | 144 | 10 |
| [13] (2008) | | ✓ | ✓ | ✓ | ACO | | 30 | 10 | 60 |
| [14] (2012) | ✓ | ✓ | | | CH | | 40 | 672 | 60 |
| [15] (2012) | ✓ | | | | ACO | ✓ | 318 | 140 | 5 |
| [16] (2012) | | ✓ | ✓ | ✓ | CH | ✓ | 200 | 5 | - |
| [17] (2014) | ✓ | | | | ILP | ✓ | 40 | 3 | - |
| [18] (2015) | ✓ | | | | CP | | 30 | 65 | 6 |
| [19] (2017) | ✓ | | ✓ | | ILP | ✓ | 40 | 3 | - |
| [20] (2018) | ✓ | | ✓ | | ILP | ✓ | 40 | 73 | - |
| [21] (2019) | ✓ | | ✓ | | ILP | ✓ | 40 | 73 | - |
| [22] (2020) | | ✓ | ✓ | ✓ | LS | ✓ | 151 | 5 | 180 |
| Current paper | ✓ | ✓ | ✓ | ✓ | DP | | 61 | 120 | 6 |

Table 1: Literature review. For each reference, we specify the kind of problem and constraints considered (TW = Time Windows; Q = Capacity constraints), the solving approach (DP = Dynamic Programming; CP = Constraint Programming; ILP = Integer Linear Programming; GA = Genetic Algorithm; LS = Local Search; ACO = Ant Colony Optimization; CH = Constructive Heuristic), and the kind of data used in the experiments (#pts = maximum number of visit points; #steps = maximum number of time-steps per day (resp. per week) for all references but [14] (resp. for [14]); size = size of the corresponding time-step in minutes). The size for [8] is based on a specific aggregation algorithm and we estimate it to be roughly around 20 minutes. The account of the current paper is specified in the last line. We solve a particular variant of the VRP corresponding to the TD-TW-mDARP, as described in Section 3.2.

In this section, we review 19 papers of the literature on time-dependent multi-point routing problems, listed in Table 1. These papers consider the Travelling Salesman Problem (TSP) where a single vehicle must visit a given set of points, or Vehicle Routing Problems (VRPs) which extend the TSP by considering a fleet of vehicles instead of a single vehicle. In some cases, problems have additional constraints: time window constraints, which ensure that points are visited within given time intervals, or capacity constraints, which ensure that the load of a vehicle never exceeds its capacity. In [22], precedence constraints are added to model a pickup and delivery problem.

In Section 2.1, we briefly describe the approaches used to solve these problems. In Sections 2.2 and 2.3, we review the performance criteria and benchmarks used to evaluate these approaches. This review motivates us to introduce a new benchmark, and these motivations are presented in Section 2.4.

### *2.1. Solving approaches*

Time-dependent TSPs and VRPs are $\mathcal{NP}$-hard problems as they are generalizations of the TSP, which is $\mathcal{NP}$-hard [23].

In some papers, (meta-)heuristic approaches are considered: simulated annealing in [6], ant colony optimization in [13, 15], tabu search in [7, 11, 12], genetic algorithms in [9], adaptive large neighborhood search [22] and construction based heuristics in [8, 10, 14, 16]. These meta-heuristic approaches have polynomial-time complexities but, as a counterpart, there is no guarantee on the computed solution's quality.

Other papers consider exact approaches which compute optimal solutions (and prove optimality) but have exponential-time complexities. An approach based on Constraint Programming (CP) has been proposed in [18], and an approach based on Dynamic Programming (DP) has been proposed in [5]. All other exact approaches are based on Integer Linear Programming (ILP). A first ILP formulation is due to Malandraki and Daskin [4]. Later on, several approaches have been devised, for example, [17, 19, 21]. Recently, Vu *et al.* [20] have proposed an approach based on time-expanded networks which outper-

forms all other ILP approaches for the time-dependent TSP with time windows.

### *2.2. Performance criteria*

Experiments reported in [4, 5, 6, 10, 12, 15, 16, 17, 19, 20, 21, 22] mainly aim at evaluating the efficiency of a new solving approach. In this case, the main performance criterion is the search effort (which is usually measured by means of CPU time) and solutions computed with time-dependent data are usually not compared with solutions computed with constant data.

Time-dependent TSPs and VRPs are much more challenging problems than their constant counterparts. Hence, it is worth studying the interest of solving these problems. In other words, if solutions computed with time-dependent data are not better than solutions computed with constant data, then it is not interesting to design algorithms for solving time-dependent problems.

A few papers have evaluated the interest of exploiting time-dependent data by comparing solutions computed with time-dependent and constant data. In [7, 9, 11, 14, 22], travel times are simply compared. These studies show that travel times are underestimated when using constant data (by around 7% in [11], for a range of 16-20% in [14], up to 50% in [9], and up to 78% in [7]).

However, this simple performance criterion does not tell us if the solution computed with constant data is different from the solution computed with time-dependent data: it may be possible that in both solutions the points are visited in the same order, and that travel time differences only come from the fact that travel times are computed with different cost functions. For example, in Fig. 1, the travel time of tour $T_1$ increases from 10 to 15 when evaluating it with the right side cost function instead of the left side one, whereas the travel time of tour $T_2$ increases from 14 to 17 when evaluating it with the left side cost function instead of the right side one.

Hence, another performance criterion is considered in [8, 13, 18]: to compare a constant and a time-dependent solution, travel times of both solutions are evaluated with the time-dependent cost function (which is closer to real traffic conditions). In this case, if the two solutions visit points in the same order,

they have identical travel times. In [8] (resp. [13]), it is shown that constant solutions are 10% (resp. 7%) longer than time-dependent ones when evaluating all solutions with time-dependent cost functions. In [18], it is shown that, for 40% of the instances, constant and time-dependent solutions are identical and, for 10% of the instances, constant solutions are more than 5% longer than time-dependent solutions.

In some papers, performance is also measured by means of time window feasibility. For example, in [11], it is shown that some time windows are missed when optimizing with constant data, and in [22], it is shown that tours optimized with constant data violate 3% of the time windows.

### *2.3. Benchmarks*

As shown in column *Artificial* of Table 1, many benchmarks are based on artificial data, which have been randomly generated. These benchmarks do not reflect the reality of urban traffic and cannot be used to evaluate the interest of exploiting time-dependent data. For example, the benchmark introduced in [17] (which is widely used to evaluate algorithm run times) has been randomly generated and it only considers three time-steps: the first and third time-steps correspond to morning and evening rush hours, respectively, and the second one corresponds to the middle of the day when the traffic demand is lower.

A realistic benchmark is described in [8], based on stationary control centers that collect speed information from a number of specially equipped vehicles in the Berlin metropolitan area. The collected data is then aggregated and transmitted to all vehicles as a driving recommendation. This data, however, is quite old (from 1988 to 1996), and the used transmission techniques are currently out of date. In [11] and [14], *Floating Car Data (FCD)* are used to generate benchmarks targeting a large regional area of the northwest of England for [11] and the area of Stuttgart in [14]. In [13], an automated traffic control system is used to collect traffic information on the Padua road network. In [8] (resp. [11], [13], and [14]), the size of the finest time-step is fixed to 20 (resp. 15, 60 and 60) minutes, *i.e.*, travel times are updated every 20 (resp. 15, 60, and 60)

minutes. These sizes are too large to fully account for the dynamic of urban traffic.

Another realistic benchmark is introduced in [18]. This benchmark has been generated by using real traffic data coming from the city of Lyon, and the size of the time-steps is 6 minutes. However, this benchmark has two main weaknesses. First, many road sections are not equipped with sensors and, for these sections, the speed is interpolated from the closest sensors, which may be an unreliable estimation. Second, time-dependent cost functions between visit points are obtained by computing time-dependent shortest paths, and this is done by considering that the duration of a path is equal to the sum of the travel times of its road sections. This underestimates real travel times as the time spent in vertices (corresponding to intersection delays) is not considered: we know from experience that the time to cross intersections or to turn left, for example, is an essential factor in the increase of travel times during rush hours.

### *2.4. Motivations for introducing a new benchmark*

In this paper, we introduce a new benchmark based on data coming from a realistic microscopic traffic flow simulation. Simulation is a convenient proxy of real-world traffic conditions which presents at least two main advantages. First, travel times are consistent with the traffic flow theory, which is barely the case in existing benchmarks. Second, simulation gives access to the finest level of details. Compared to experimental measurements, this approach provides full coverage of the physical phenomena. In particular, since sensors are virtualized, we can control the number of sensors and their positions (spatial dimension) as well as the frequency of the measures (temporal dimension). By varying the granularity of spatial and temporal dimensions, we evaluate the impact that traffic infrastructure has on the quality of the computed solutions. To the best of our knowledge, no other approach has discussed this dimension.

Our new benchmark also includes precedence constraints between visit points, capacity constraints on vehicle loads, and time window constraints on the time a point can be visited. These constraints allow us to evaluate the interest of

exploiting time-dependent data on six problems described in the next section.

## 3. The Time-Dependent General Pickup and Delivery Problem

In this section, we first introduce a mathematical model of the TD-GPDP, and then we define its six variants considered in our experimental study.

### *3.1. Mathematical model of the TD-GPDP*

We use lowercase letters to denote input values, calligraphic letters to denote sets, and uppercase letters to denote decision variables.

*Input data.* $\mathcal{P}$ denotes the set of all points to visit. Each visit point $i \in \mathcal{P}$ has a service time $s_i \in \mathbb{N}$, an earliest visit time $e_i \in \mathbb{N}$ and a latest visit time $l_i \in \mathbb{N}$.

$\mathcal{R}$ denotes the set of pickup and delivery requests. Each request $r \in \mathcal{R}$ has a weight $w_r \in \mathbb{N}$, a pickup point $p_r \in \mathcal{P}$ and a delivery point $d_r \in \mathcal{P}$. Pickup and delivery points are all different points.

$\mathcal{V}$ denotes the set of vehicles. Each vehicle $v \in \mathcal{V}$ has a capacity $q_v \in \mathbb{N}$. The route of a vehicle $v \in \mathcal{V}$ starts from a depot denoted $v_{start}$, and ends at a depot denoted $v_{end}$. The set of all start depots is $\mathcal{D}_{start} = \{v_{start} : v \in \mathcal{V}\}$, and the set of all end depots is $\mathcal{D}_{end} = \{v_{end} : v \in \mathcal{V}\}$. We associate different start and end points to each vehicle to simplify the model and make it more general. However, these different points may correspond to the same geographical location (which is the case in our benchmark).

Finally, $\mathcal{T}$ denotes the set of all possible times, and $t_0 \in \mathcal{T}$ is the starting time of all vehicles. For each couple of points $i, j \in \mathcal{P} \cup \mathcal{D}_{start} \cup \mathcal{D}_{end}$ and each time $t \in \mathcal{T}$, $c_{ij}^t$ denotes the duration to travel from $i$ to $j$ when leaving $i$ at time $t$ (we describe how travel durations are computed in Section 5).

*Decision variables.* For each point $i \in \mathcal{P} \cup \mathcal{D}_{start} \cup \mathcal{D}_{end}$, we define the following decision variables:

- $V_i$ represents the vehicle which visits $i$ ($V_i \in \mathcal{V}$);

- $Q_i$ represents the load of the vehicle when arriving on $i$ ($Q_i \in \mathbb{N}$);
- $A_i$ represents the arrival time on $i$ ($A_i \in \mathcal{T}$);
- $D_i$ represents the departure time from $i$ ($D_i \in \mathcal{T}$);
- $S_i$ represents the successor of $i$, *i.e.*, the point which is visited just after $i$ by the vehicle which has visited $i$ ($S_i \in \mathcal{P} \cup \mathcal{D}_{end}$).

*Objective function.* In our study, the goal is to minimise the total travel time, *i.e.*, the sum for every point $i \in \mathcal{P} \cup \mathcal{D}_{start}$ of the difference between the arrival time on the successor of $i$ and the departure time from $i$:

$$\min \sum_{i \in \mathcal{P} \cup \mathcal{D}_{start}} A_{S_i} - D_i \tag{1}$$

Other objective functions could be considered such as, for example, the actual number of used vehicles (where a vehicle $v$ is not used if it travels directly from its start depot to its end depot, *i.e.*, $S_{v_{start}} = v_{end}$). However, these other objective functions are less relevant for evaluating the impact of the definition of time-dependent functions $c_{ij}^t$ on the total travel time.

*Constraints.* In Fig. 2, we list all constraints of the TD-GPDP. Constraints (2)-(3) ensure that every vehicle $v$ starts from $v_{start}$ at time $t_0$ and ends on $v_{end}$. Constraint (4) ensures that the successor of a point $i$ is visited by the same vehicle as $i$. Constraint (5) ensures that two different points cannot have the same successor. It ensures that every point is visited exactly once because every point $i \in \mathcal{P} \cup \mathcal{D}_{start}$ has exactly one successor $S_i \in \mathcal{P} \cup \mathcal{D}_{end}$. Constraint (6) ensures that arrival and departure times allow serving $i$. Constraint (7) ensures that there is enough time to travel from a point $i$ to its successor $S_i$. Constraint (8) ensures that the arrival time on $i$ satisfies its time window. Constraints (9)-(10) ensure that a request is delivered after its pickup and by the same vehicle. Finally, Constraints (11)-(13) define vehicle loads and Constraint (14) ensures that these loads never exceed vehicle capacities.

$$V_{v_{start}} = V_{v_{end}} = v \qquad \forall v \in \mathcal{V} \tag{2}$$
$$D_{v_{start}} = t_0 \qquad \forall v \in \mathcal{V} \tag{3}$$
$$V_i = V_{S_i} \qquad \forall i \in \mathcal{P} \cup \mathcal{D}_{start} \tag{4}$$
$$S_i \neq S_j \qquad \forall i, j \in \mathcal{P} \cup \mathcal{D}_{start}, i \neq j \tag{5}$$
$$D_i \geq A_i + s_i \qquad \forall i \in \mathcal{P} \tag{6}$$
$$A_{S_i} \geq D_i + c_{iS_i}^{D_i} \qquad \forall i \in \mathcal{P} \cup \mathcal{D}_{start} \tag{7}$$
$$e_i \leq A_i \leq l_i \qquad \forall i \in \mathcal{P} \tag{8}$$
$$D_{p_r} \leq A_{d_r} \qquad \forall r \in \mathcal{R} \tag{9}$$
$$V_{p_r} = V_{d_r} \qquad \forall r \in \mathcal{R} \tag{10}$$
$$Q_{v_{start}} = 0 \qquad \forall v \in \mathcal{V} \tag{11}$$
$$Q_{S_{p_r}} = Q_{p_r} + w_r \qquad \forall r \in \mathcal{R} \tag{12}$$
$$Q_{S_{d_r}} = Q_{d_r} - w_r \qquad \forall r \in \mathcal{R} \tag{13}$$
$$Q_i \leq q_{V_i} \qquad \forall i \in \mathcal{P} \tag{14}$$

Figure 2: Constraints of the TD-GPDP.

### *3.2. Variants of the TD-GPDP*

Based on this general model, we consider different variants of the TD-GPDP. Let us first define five single-vehicle problems, where the set $\mathcal{V}$ contains only one vehicle (which is assigned to every decision variable $V_i$):

**Problem** $P_1$ is the Time-Dependent Travelling Salesman Problem (TD-TSP), and it is obtained from the TD-GPDP by ignoring Constraints (8)-(14);

**Problem** $P_2$ is a basic single-vehicle Time-Dependent Pickup-and-Delivery Problem (TD-PDP) without capacity nor time window constraints, and it is obtained from $P_1$ by adding Constraints (9)-(10);

**Problem** $P_3$ is a basic single-vehicle Time-Dependent Dial-A-Ride Problem (TD-DARP), and it is obtained from $P_2$ by adding Constraints (11)-(14);

**Problem** $P_4$ is a TD-PDP with time windows (denoted TD-TW-PDP), and it is obtained from $P_2$ by adding Constraint (8);

**Problem** $P_5$ is a TD-DARP with time windows (denoted TD-TW-DARP), and it is obtained from $P_3$ by adding Constraint (8).

Each of these single-vehicle problems may be generalized by considering that $\mathcal{V}$ contains more than one vehicle. In our study, we mainly focus on single vehicle problems to evaluate whether we can compute a better tour with time-dependent cost functions. We evaluate the impact of considering several vehicles on the following problem:

**Problem** $P_6$ is a generalization of $P_5$, denoted TD-TW-mDARP, where $\mathcal{M}$ contains more than one vehicle, *i.e.*, $\#\mathcal{M} > 1$.

The constraints of the six variants of the TD-GPDP are summarized in Table 2.

| | Precedence (9)-(10) | Capacity (11)-(14) | Time window (8) | Number of vehicles in $\mathcal{M}$ |
|---|---|---|---|---|
| $P_1$ (TD-TSP) | - | - | - | 1 |
| $P_2$ (TD-PDP) | Yes | - | - | 1 |
| $P_3$ (TD-DARP) | Yes | Yes | - | 1 |
| $P_4$ (TD-TW-PDP) | Yes | - | Yes | 1 |
| $P_5$ (TD-TW-DARP) | Yes | Yes | Yes | 1 |
| $P_6$ (TD-TW-mDARP) | Yes | Yes | Yes | $m > 1$ |

Table 2: Constraints of the 6 variants of the TD-GPDP.

Time-dependent cost functions are a generalization of constant cost functions (such that $c^t_{i,j}$ returns the same value for every possible starting time $t$). Hence, for every time-dependent problem $P_1$ to $P_6$, we can obtain the corresponding classical problem by using a constant cost function instead of a time-dependent one in the objective function (1) and in Constraint (7). We refer to these classical problems as constant problems (by opposition to time-dependent problems).

## 4. Solving approach

Our goal is not to introduce a new approach for solving TD-GPDPs, but to evaluate whether we can obtain better tours when solving TD-GPDPs instead of their constant counterparts. To this aim, we need to solve TD-GPDPs with an exact approach able to find optimal solutions. State-of-the-art ILP and CP approaches described in Section 2.1 do not scale well when considering instances

without time windows. For example, the ILP approach of [20][4] cannot solve our instances of $P_1$ (TD-TSP without time windows) within a reasonable amount of time even for small instances with 20 points to visit. Similarly, the CP approach of [18] does not scale well when there are no time window constraints.

Hence, we consider the dynamic programming approach introduced in [5] to solve the TD-TSP. In this section, we first recall the basic idea of this approach, and then show how to extend it to solve Problems $P_2$ to $P_6$.

### *4.1. Dynamic Programming Approach for solving the TD-TSP*

As pointed out in [5], the dynamic programming approach proposed by Held and Karp in [24] to solve the TSP may be extended to the TD-TSP in a straightforward way. More precisely, let $v_{start}$ and $v_{end}$ be the start and end depot of the single vehicle. For each point $i$ and each subset of points $\mathcal{S}$, let $p(i, \mathcal{S})$ denote the earliest arrival time of a path that starts from $v_{start}$ at time $t_0$, visits each point of $\mathcal{S}$ exactly once, and finishes on point $i$. The Bellman equations that recursively define $p(i, \mathcal{S})$ are:

$$\begin{cases} \text{if } \mathcal{S} = \emptyset,\ p(i, \mathcal{S}) = c^{t_0}_{v_{start}\, i} \\ \text{otherwise, } p(i, \mathcal{S}) = \min_{j \in \mathcal{S}} p(j, \mathcal{S} \setminus \{j\}) + s_j + c^{p(j, \mathcal{S}\setminus\{j\}) + s_j}_{ji} \end{cases}$$

The earliest time for arriving on $v_{end}$ when leaving $v_{start}$ at time $t_0$ is given by $p(v_{end}, \mathcal{P})$. We have to subtract $t_0$ and remove all service times from this earliest arrival time to obtain the total travel time, *i.e.*, the optimal value of the objective function defined by (1) is equal to $p(v_{end}, \mathcal{P}) - t_0 - \sum_{i \in \mathcal{P}} s_i$.

The algorithm is derived from these recursive equations in a straightforward way. The time complexity of this algorithm (similarly as the initial algorithm of Held and Karp) is $\mathcal{O}(n^2 \cdot 2^n)$, and its space complexity is $\mathcal{O}(n \cdot 2^n)$, where $n = \#\mathcal{P}$ is the number of points to visit. This algorithm is able to solve instances of the TD-TSP with $n = 10$ (resp. 20 and 30) points in less than 0.01 (resp. 1 and 1000 seconds) when there are no additional constraints[5]. This algorithm

[4] We thank Vu *et al.* for sharing their source code.

[5] All experiments of the paper have been performed on an `Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz` processor with 32 GB RAM memory.

outperforms state-of-the-art ILP and CP approaches for the TD-TSP as these approaches are not able to solve instances with $n = 30$ within a reasonable amount of time when there is no additional constraint.

### *4.2. Extension of the Dynamic Programming Approach to the TD-GPDP*

In [25], Desrosiers *et al.* show that dynamic programming can be easily extended to handle different kinds of constraints, and they introduce a dynamic programming approach for solving a single-vehicle dial-a-ride problem with time windows. Their approach can be extended to our time-dependent problems in a straightforward way. The basic idea is to assign $\infty$ to every state $p(i, \mathcal{S})$ that violates some constraints.

- For problems $P_2$ to $P_6$, a state $p(i, \mathcal{S})$ violates precedence constraints (9)-(10) if there exists a delivery point $d_r \in \mathcal{S} \cup \{i\}$ such that the corresponding pickup point $p_r$ does not belong to $\mathcal{S}$, *i.e.*,

$$\text{if } \exists r \in \mathcal{R}, d_r \in \mathcal{S} \cup \{i\} \wedge p_r \notin \mathcal{S}, \text{ then } p(i, \mathcal{S}) = \infty$$

- For problems $P_3$, $P_5$, and $P_6$, a state $p(i, \mathcal{S})$ violates capacity constraints (11)-(14), if the weight of all ongoing requests exceeds the capacity, *i.e.*,

$$\text{if } \sum_{r \in \mathcal{R}, p_r \in \{i\} \cup \mathcal{S}, d_r \notin \{i\} \cup \mathcal{S}} w_r > q_v, \text{ then } p(i, \mathcal{S}) = \infty.$$

- For problems $P_4$ to $P_6$, a state $p(i, \mathcal{S})$ violates the time window constraint (8), if there is a point in $\mathcal{S}$ that cannot be visited without violating the time window of $i$, *i.e.*,

$$\text{if } \exists j \in \mathcal{S}, e_j + s_j + \min_{t \in \mathcal{T}} c_{ji}^t > l_i, \text{ then } p(i, \mathcal{S}) = \infty$$

  or if it is not possible to arrive on $i$ before the end of its time window, *i.e.*,

$$\text{if } \min_{j \in \mathcal{S}} p(j, \mathcal{S} \setminus \{j\}) + s_j + c_{ji}^{p(j, \mathcal{S} \setminus \{j\}) + s_j} > l_i, \text{ then } p(i, \mathcal{S}) = \infty.$$

  Furthermore, we must ensure that $p(i, \mathcal{S})$ is greater than or equal to the beginning of the time window of $i$, *i.e.*,

$$p(i, \mathcal{S}) = \max\{e_i, \min_{j \in \mathcal{S}} p(j, \mathcal{S} \setminus \{j\}) + s_j + c_{ji}^{p(j, \mathcal{S} \setminus \{j\}) + s_j}\}.$$

When adding constraints, the number of states to compute is drastically reduced because we no longer have to compute states associated with subsets of $\mathcal{S}$ whenever $p(i, \mathcal{S})$ is assigned to $\infty$. Hence, we can solve instances with more vertices. For example, we can solve instances with $n = 40$ (resp. 50 and 60) in less than 0.014 (resp. 0.15 and 2) seconds for Problem $P_4$ when the $n$ points are partitioned in three subsets of similar sizes, and two points $i$ and $j$ in the same subset have the same time window (*i.e.*, $e_i = e_j < l_i = l_j$) whereas two points $i$ and $j$ in two different subsets have non overlapping time windows (*i.e.*, $e_i < l_i < e_j < l_j$ or $e_j < l_j < e_i < l_i$).

## 5. Description of the benchmark

One objective of this study is to provide a benchmark to the research community with the following goals in mind: (i) accounting for different spatial granularities, from perfect knowledge to a realistic coverage, to understand the impact of the number of sensors and their positions on the quality of tours; (ii) accounting for different temporal granularities to evaluate the interest of exploiting time-dependent data when optimizing tours; and (iii) considering different kinds of constraints to evaluate their impact on the results.

### *5.1. Estimation of time-dependent travel times of road links*

To obtain full access to ground data, we use a dynamic microscopic simulator of traffic flows, called SYMUVIA [26], on a sub-part of the Lyon transportation network (see Fig. 3). This software can simulate the whole complexity of the urban traffic flow by taking into account different classes of vehicles, individual driving behaviors, lane-changing phenomenons, intersections, etc. It is built on a car-following law based on Newell's model [27] and its numerous extensions [28, 29]. If the simulation is only a proxy of the real world, it provides access to the finest level of details and every possible measurement of traffic dynamics may be emulated: individual travel times, link speeds, loop detector data, etc.

The traffic demand for a whole day (24 hours) has been estimated based on real traffic conditions (measured by actual sensors displayed in Fig. 3) and on

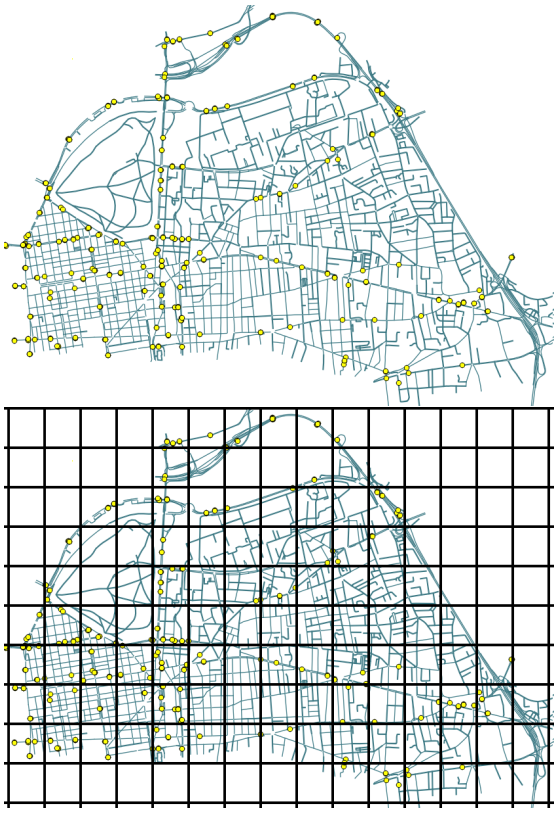

Figure 3: Top: Lyon road network considered in the study with the actual positions of sensors (in yellow). Bottom: 15 x 10 grid subdivision considered in the parametric dataset.

the estimation of the origin-destination matrix of the transportation demand of the area. This latter was constructed by capturing the actual movements of the population of the city through the 2016 population census results of France [30] and the various mobility surveys conducted by local authorities. Therefore, this demand is representative of a realistic classical weekday in Lyon city.

We use the following consistent spatio-temporal mean formulation to calculate the travel time $f(k,t)$ for every road link $k$ of the network and every time-step starting at time $t$ and ending at time $t + \Delta t$:

$$f(k,t) = \frac{\lambda_k}{V_k(t)} \qquad \text{and} \qquad V_k(t) = \frac{Q_k(t)}{K_k(t)},$$

where $\lambda_k$ is the length of link $k$, $Q_k$ is the spatio-temporal mean of the flow in

link $k$ at time $t$ and $K_k$ is the spatio-temporal mean of the density in link $k$ at $t$, calculated according to definitions given by [31]:

$$Q_k(t) = \frac{\sum_r \delta_r}{\lambda_k \Delta t}, \qquad K_k(t) = \frac{\sum_r \tau_r}{\lambda_k \Delta t},$$

where $\delta_r$ and $\tau_r$ are respectively the distance traveled and the time spent by vehicle $r$ within the link $k$ from $t$ to $t + \Delta t$ (these values are easily obtained from our micro-simulation). Note that those definitions are entirely consistent with the dynamic of traffic flow because they weight accurately the different traffic conditions that can be observed within a road link [32], on the contrary of classical loop detector data.

### *5.2. Spatio-temporal features of the benchmark*

When a carrier plans a delivery tour, it does not know the exact traffic conditions that will be observed while implementing the tour. Thus, it must rely on a predictive model to estimate these traffic conditions given data coming from sensors. The design of urban predictive models is an active research topic that is not discussed here (see [33], for instance, for a comparison of predictive models using data of the sensors visualized in Fig. 3). In this study, we assume a perfect predictive model for each traffic flow sensor and consider two datasets:

- A realistic dataset, denoted $D_{Lyon}$, where sensors are positioned exactly in the same places as in the current Lyon network [34] (cf. Fig. 3). The coverage is quite low, with 7.35% equipped links. In this case, travel times of links not covered by sensors are estimated using interpolation, taking the value of the closest sensor with respect to the Euclidean distance.

- A parametric dataset, denoted $D_\sigma$ with $\sigma \in \{10, 20, \ldots, 100\}$, where sensors are evenly distributed in the network with respect to a spatial granularity level which is controlled by $\sigma$. More precisely, we fit our transportation network in a regular grid of 15 x 10 cells, where each cell is a 396m x 425m area, as shown in Fig. 3. In each cell, we randomly select $\sigma$% of the road links to be equipped with a sensor, according to a uniform distribution. Travel times of the remaining $(100 - \sigma)$% links of the

cells are obtained by interpolation within the same cell. This grid subdivision allows us to consider increasing levels of spatial granularity in a structured manner. Note that $D_{100}$ represents the dataset with complete spatial information.

For each dataset $D_\sigma$ with $\sigma \in \{Lyon, 10, 20, \ldots, 100\}$, we have generated five cost functions denoted $D_\sigma Sl$, where $l \in \{6, 12, 24, 60, 720\}$ is the length of the time-step (in minutes). More precisely, the time horizon $\mathcal{T}$ starts at 7:00 and ends at 19:00 and it is divided in $\frac{\mathcal{T}}{l}$ consecutive time-steps such that the duration of each time-step is equal to $l$. Hence, $\mathcal{T}$ is divided in 120 (resp. 60, 30, 12, and 1) time-steps when $l = 6$ (resp. 12, 24, 60, and 720). For each road link $k$ and each time step $s \in [0, \frac{\mathcal{T}}{l}[$, $D_\sigma Sl(k, s)$ is equal to the average travel time of $k$ during the time interval that starts at 7:00+$s \times l$ and ends at 7:00+$(s + 1) \times l$.

### *5.3. Computation of point-to-point travel times*

To solve TD-GPDPs, we have to compute the cost function $c_{ij}^t$ which returns the travel time from point $i$ to point $j$ when leaving $i$ at time $t$, for each couple of points $(i, j) \in \mathcal{P} \cup \mathcal{D}_{start} \times \mathcal{P} \cup \mathcal{D}_{end}$. In our benchmark, $c_{ij}^t$ is modeled as a step-wise function: for each time $t$ within a same time-step, $c_{ij}^t$ is constant. Other models could be used [35]. However, this model is well suited for traffic data since it fits the usual scheme of travel time estimation.

To compute $c_{ij}^t$, we have to compute the duration of the quickest path from $i$ to $j$, and this must be done for each dataset $D_\sigma Sl$ and each time-step $s \in [0, \frac{\mathcal{T}}{l}[$. A valuable property of $D_\sigma Sl$ to efficiently compute quickest paths is the *no-passing* or *First-In-First-Out (FIFO)* condition: $D_\sigma Sl$ satisfies the FIFO condition if, for each road link $k$ and each couple of starting times $t_1, t_2$ such that $t_1 < t_2$, we have $t_1 + D_\sigma Sl(k, t_1) < t_2 + D_\sigma Sl(k, t_2)$. In other words, it is not possible to arrive sooner when leaving later. If $D_\sigma Sl$ satisfies the FIFO condition, then shortest paths can be efficiently computed by adapting Dijkstra algorithm, otherwise, the problem becomes $\mathcal{NP}$-hard [36]. If $D_\sigma Sl$ does not satisfy the FIFO condition, then we use the algorithms of [7, 18] to transform $D_\sigma Sl$ into a FIFO cost function.

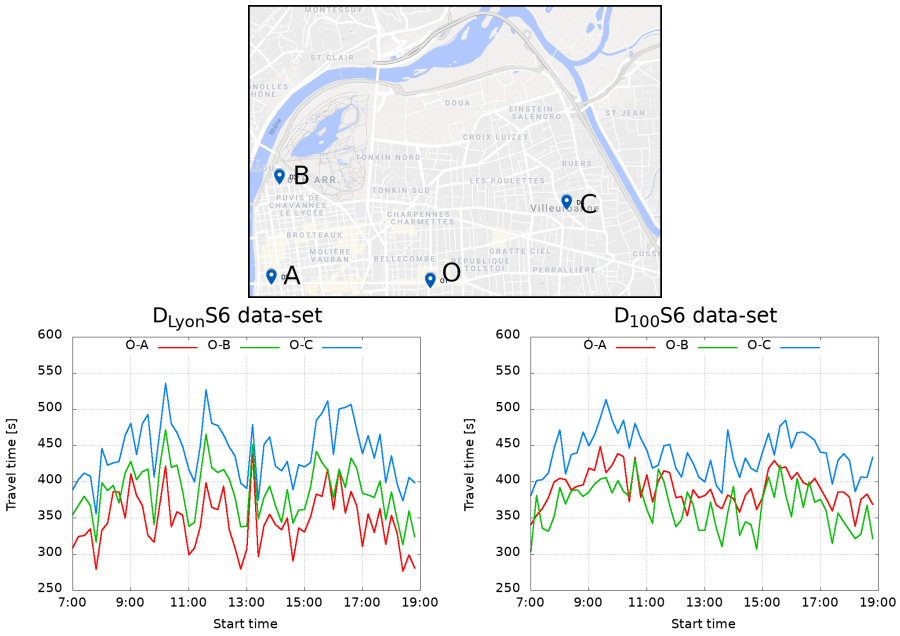


Figure 4: Example of time-dependent cost functions for three origin-destination addresses O-A, O-B, and O-C positioned in the top map, and computed with $D_{Lyon}S6$ (left) and $D_{100}S6$ (right) datasets.

Fig. 4 shows an example of three time-dependent cost functions when considering one origin $O$ and three destinations $A$, $B$, and $C$, for the complete ($\sigma = 100$) and the realistic ($\sigma = Lyon$) spatial coverages when $l = 6$ minutes.

### *5.4. Description of the benchmark instances*[6]

We denote $n$ the number of points to visit (excluding depot points), *i.e.*, $n = \#\mathcal{P}$. For each value of $n \in \{10, 20, 30, 40, 50, 60\}$, we have generated 30 instances by randomly selecting a set $\mathcal{P}$ of $n$ urban addresses from the city of Lyon. For all instances, we consider the same address for all start and end depots ($v_{start}$ and $v_{end}$): this address is located in the city center, in front of the Lyon-Part-Dieu train station. A fixed stop duration $s_i = 3$ minutes is associated with each address $i \in \mathcal{P}$.

[6] The code and data used to generate our instances are available at: http://perso.citilab.fr/csolnon/TDGPDP.html

Hence, our benchmark is composed of $6 \times 30 = 180$ instances of Problem $P_1$ (TD-TSP instances without additional constraints) and, for each of these instances, we have $5 \times 11 = 55$ time-dependent cost functions such that each cost function has been obtained with a different time-step length $l \in \{6, 12, 24, 60, 720\}$ and a different spatial coverage $\sigma \in \{Lyon, 10, 20, \ldots, 100\}$.

From these instances of Problem $P_1$, we derive instances of Problems $P_2$ to $P_6$ by adding constraints. The number of requests is set to $n/2$ and, for each request $r \in [1, n/2]$, we set the weight to $w_r = 1$. Every pickup point $p_r$ and delivery point $d_r$ (with $r \in [1, n/2]$) is a different point of $\mathcal{P}$.

For Problems $P_3$, $P_5$, and $P_6$, we consider different values for the capacity of a vehicle $v$, *i.e.*, $q_v \in \{2, 4, 6, 8\}$.

Finally, for Problems $P_4$ to $P_6$ we add time window constraints. We control the tightness of these constraints with two parameters denoted $nTW$ and $xTW$: $nTW$ controls the number of different time windows and it ranges from 2 to 6 in our experiments, and $xTW$ controls the average travel time allowed to travel from one point to its successor point and it ranges from 100 to 200 seconds in our experiments. More precisely, for each request $r \in [1, n/2]$, $p_r$ and $d_r$ have the same earliest and latest visit times which are defined as follows:

$$e_{p_r} = e_{d_r} = t_0 + w \times (r \ \% \ nTW) \text{ and } l_{p_r} = l_{d_r} = e_{p_r} + w$$

where % is the remainder of the euclidean division, $w = (xTW + s) \times n/nTW$ is the width of the time window and $s$ is the fixed stop duration (set to 3 minutes in all instances). In other words, there are $nTW$ consecutive time windows of $w$ seconds, and there are roughly $n/nTW$ points to visit within each time window.

## 6. Experimental analysis

In this section, we evaluate the interest of optimizing tours with time-dependent costs on our benchmark. More specifically, we address the following questions from an empirical perspective:

**Q1:** Can we find better tours when using time-dependent cost functions instead of constant ones?

**Q2:** What is the impact of the spatio-temporal granularity $(\sigma, l)$ of cost functions on the travel time of optimal tours?

**Q3:** Does this impact change when adding precedence or capacity constraints?

**Q4:** What is the impact of $(\sigma, l)$ on the satisfaction of time window constraints?

**Q5:** Does this impact change when increasing the number of vehicles?

We introduce performance measures used to answer these questions in Section 6.1. Questions are addressed in Sections 6.2 to 6.6.

### *6.1. Performance measures*

*Realistic travel time of a tour.* The travel time of a tour depends on the time-dependent cost function used to compute it, and we want to compare tours computed with different cost functions. Ideally, the travel time of a tour should be evaluated with respect to real traffic conditions: the best tour is the one with the smallest travel time when performing it in real conditions. Hence, travel times are computed with respect to the best approximation of real traffic conditions, *i.e.*, $D_{100}S6$ which has a full spatial cover and the smallest time-steps. Given a tour $T$, its "realistic" travel time computed with the time-dependent cost function $D_{100}S6$ is denoted $rtt(T)$.

*Measure used to answer Q1.* To evaluate the interest of exploiting time-dependent cost functions, we compare the optimal tour computed with a constant cost function (when $l = 720$) with optimal tours computed with time-dependent cost functions (when $6 \leq l \leq 60$). More precisely, we compute the gap in percentage between a constant tour $T_{const}$ and a time-dependent tour $T_{td}$ as follows:

$$gap(T_{const}, T_{td}) = \frac{rtt(T_{const}) - rtt(T_{td})}{rtt(T_{td})} \times 100 \tag{15}$$

Positive (resp. negative) gap values correspond to cases where time-dependent tours are faster (resp. longer) than constant tours when realizing them in realistic traffic conditions.

*Measure used to answer Q2, Q3, and Q5.* Our benchmark is composed of 55 cost functions $D_\sigma Sl$ where $\sigma$ defines the spatial granularity and $l$ the temporal granularity. We denote $T^{D_\sigma Sl}$ the optimal tour computed with $D_\sigma Sl$. To evaluate the impact of $\sigma$ and $l$ on travel times, we measure the quality of $T^{D_\sigma Sl}$, denoted $Q(T^{D_\sigma Sl})$, by means of its gap in percentage to $T^{D_{100}S6}$, *i.e.*,

$$Q(T^{D_\sigma Sl}) = \frac{rtt(T^{D_\sigma Sl}) - rtt(T^{D_{100}S6})}{rtt(T^{D_{100}S6})} \times 100 \tag{16}$$

The smaller the quality gap $Q(T^{D_\sigma Sl})$, the faster the tour when realizing it in realistic traffic conditions, *i.e.*, the better the tour.

*Measure used to answer Q4 and Q5.* Precedence and capacity constraints are not impacted by travel time cost functions: if a tour satisfies these constraints, then it still satisfies them when computing arrival times with any cost function $D_\sigma Sl$. This is no longer the case for time window constraints as the satisfaction of these constraints depends on arrival times, and arrival times depend on the cost function. We say that an instance is $(\sigma, l)$-*feasible* if there exists at least one tour which satisfies all time windows when evaluating travel times with $D_\sigma Sl$. However, when an instance is $(\sigma, l)$-feasible, it may be possible that its optimal tour $T^{D_\sigma Sl}$ no longer satisfies all time windows when computing travel times with $D_{100}S6$. In this case, we say that $T^{D_\sigma Sl}$ is *(100,6)-inconsistent*. Finally, given an instance $i$, we define the predicate *isFeasible*$(i, D_\sigma Sl)$:

$$\textit{isFeasible}(i, D_\sigma Sl) \Leftrightarrow i \text{ is } (\sigma, l)\textit{-feasible} \text{ and } T^{D_\sigma Sl} \text{ is } (100, 6)\textit{-consistent} \tag{17}$$

and we evaluate the feasibility of a set $\mathcal{I}$ of benchmark instances for a cost function $D_\sigma Sl$ with a measure denoted $F^{\%}(D_\sigma Sl)$ such that:

$$F^{\%}(D_\sigma Sl) = \frac{\#\{i \in \mathcal{I} : \textit{isFeasible}(i, D_\sigma Sl)\}}{\#\mathcal{I}} \times 100 \tag{18}$$

In other words, $F^{\%}(D_\sigma Sl)$ is the percentage of instances which are $(\sigma, l)$-feasible and for which the optimal tour $T^{D_\sigma Sl}$ is *(100,6)-consistent*.

*Illustration on an example.* Consider the two cost functions defined in Fig. 1, and suppose that costs on the left (resp. right) side of the figure correspond to

$D_{100}S6$ (resp. $D_{100}S12$). In this case, optimal tours are $T^{D_{100}S6} = \langle 0, 3, 1, 2, 0\rangle$ and $T^{D_{100}S12} = \langle 0, 1, 2, 3, 0\rangle$. Let us assume that the best tour with constant data is $T^{D_{100}S720} = \langle 0, 2, 3, 1, 0\rangle$. When computing travel times with $D_{100}S6$, we have $rtt(T^{D_{100}S6}) = 10$, $rtt(T^{D_{100}S12}) = 17$, and $rtt(T^{D_{100}S720}) = 19$. To answer Q1, we compute:

$$gap(T^{D_{100}S720}, T^{D_{100}S6}) = 90\% \text{ and } gap(T^{D_{100}S720}, T^{D_{100}S12}) = 12\%$$

In other words, the tour computed with constant costs is 90% (resp. 12%) longer than the one computed with time-dependent costs with 6 (resp. 12) minute time-steps when travel times are computed with the finer data.

To answer Q2 or Q3 when $\sigma = 100$ and $l = 12$, we compute $Q(T^{D_{100}S12}) = 70\%$. In other words, $T^{D_{100}S12}$ is 70% longer than $T^{D_{100}S6}$ when all travel times are computed with the finer data.

To answer Q4 or Q5, we evaluate *isFeasible*$(i, D_\sigma Sl)$ where $i$ is the instance of Fig. 1. If the time window of point 3 is $[2, 3]$ and the time window of all other points is $[0, 20]$, then *isFeasible*$(i, D_{100}S6)$ is true ($\langle 0, 3, 1, 2, 0\rangle$ satisfies all time windows) whereas *isFeasible*$(i, D_{100}S12)$ is false ($i$ is not $(100, 12)$-feasible as it is not possible to travel from 0 to 3 in less than 4 minutes). If the time window of point 3 is $[2, 10]$ and the time window of all other points is $[0, 20]$, then $i$ is $(100, 12)$-feasible but *isFeasible*$(i, D_{100}S12)$ is false ($T^{D_{100}S12} = \langle 0, 1, 2, 3, 0\rangle$ is $(100, 6)$-inconsistent as the vehicle arrives at time 11 on point 3 when computing arrival times with $D_{100}S6$).

### *6.2. Q1: Can we find better tours when using time-dependent cost functions instead of constant ones?*

To answer this question, we only consider problem $P_1$ that does not have any additional constraint, in order not to bias the study with side effects due to constraints. The impact of adding constraints is studied in the next sections.

In Fig. 5, we consider cost functions with complete spatial information (when $\sigma = 100$), and we display the gap (as defined in Eq. 15) between tours optimized with constant cost functions ($T^{D_{100}S720}$) and tours optimized with time-dependent cost functions ($T^{D_{100}Sl}$ with $l \in \{6, 12, 24, 60\}$).

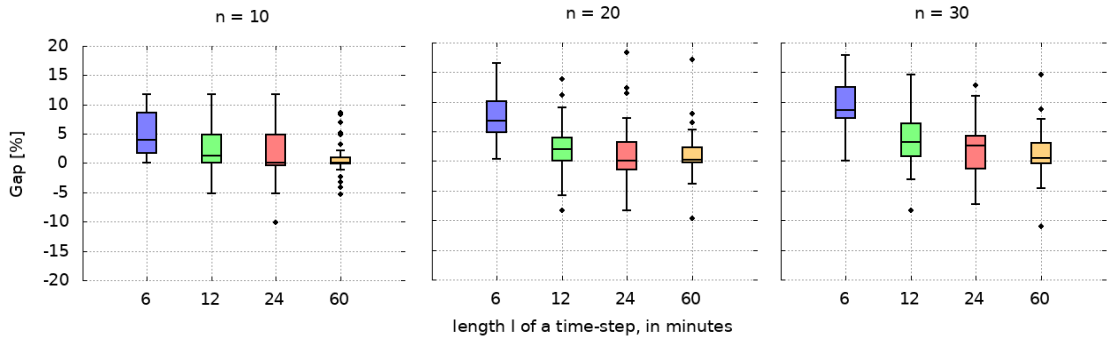


Figure 5: Distribution of $gap(T^{D_{100}S720}, T^{D_{100}Sl})$ with $l \in \{6, 12, 24, 60\}$ for Problem $P_1$.

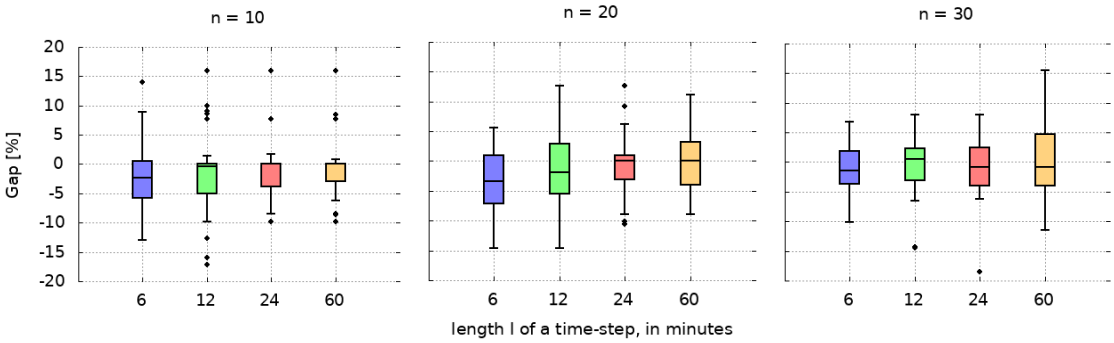


Figure 6: Distribution of $gap(T^{D_{Lyon}S720}, T^{D_{Lyon}Sl})$ with $l \in \{6, 12, 24, 60\}$ for Problem $P_1$.

When $l = 6$, the gap is always positive, and it increases when increasing the number $n$ of points to visit. The median gap is equal to 4% (resp. 6.8 and 8.6%) when $n = 10$ (resp. 20 and 30). The largest gap is equal to 18% when $n = 30$, meaning that there is an instance for which the tour optimized with constant costs is 18% longer than the tour optimized with $D_{100}S6$.

However, gaps decrease when $l$ increases. When $n = 30$, the median gap is decreased from 8.6 to 3.2, 2.6, and 0.5% when $l$ is increased from 6 to 12, 24, and 60 minutes, respectively. If the median gap is always positive, minimum gaps become negative when $l \geq 12$. For example, when $l = 60$ and $n = 30$, the smallest gap is $-11\%$, meaning that there is an instance for which the tour optimized with $D_{100}S720$ is 11% faster than the tour optimized with $D_{100}S60$.

In Fig. 6, we consider cost functions with realistic spatial information, *i.e.*,

$D_{Lyon}Sl$. In this case, median gaps are close to zero. For example, when $n = 30$, the median gap is equal to -1.5% (resp. 0.5, -0.9, and -0.8%) when $l = 6$ (resp. 12, 24, and 60). The minimum gap is always lower than zero, and for some instances, the loss can be greater than 15%.

By studying the position of the sensors considered in $D_{Lyon}Sl$ (corresponding to the sensors deployed on the Lyon road network), we find that these sensors are often placed on the congested axes of the network, which leads to an overestimation of travel times to cross road links not equipped with sensors (since these travel times are estimated by interpolation with respect to the links equipped with sensors, generally more congested). Fig. 4 confirms this observation: travel times of O-B are over-estimated with $D_{Lyon}S6$ and they are greater than those of O-A with $D_{Lyon}S6$ whereas they are smaller with $D_{100}S6$.

As a conclusion, the answer to Q1 depends on $\sigma$.

- When $\sigma = 100$ (*i.e.*, every road link has a sensor), the answer is: yes, it is worth exploiting time-dependent cost functions, and the smaller the time-step $l$, the better the tour.

- When $\sigma = Lyon$ (*i.e.*, sensors are located like in the actual Lyon road network, and cost functions of links that are not equipped with sensors are generated using interpolation), the answer is: no, it is not interesting to optimize tours with time-dependent data as tours are not really better.

### *6.3. Q2: What is the impact of $(\sigma, l)$ on the travel time of optimal tours?*

As the answer to Q1 depends on whether $\sigma = 100$ or $\sigma = Lyon$, we now investigate other spatial distributions by varying $\sigma$ from 10 to 100: In this case, $\sigma$% of the road links are equipped with sensors, with a balanced yet random distribution of these sensors.

The upper row of Fig. 7 displays the quality gap $Q(T^{D_\sigma Sl})$, as defined by Eq. 16, for the TD-TSP. The increase of the gap when increasing the time-step size $l$ for the spatial cover $\sigma = 100$ is expected: As we have seen in Fig. 5, larger time-steps $l$ produce worse tours and lead to larger quality gaps. This

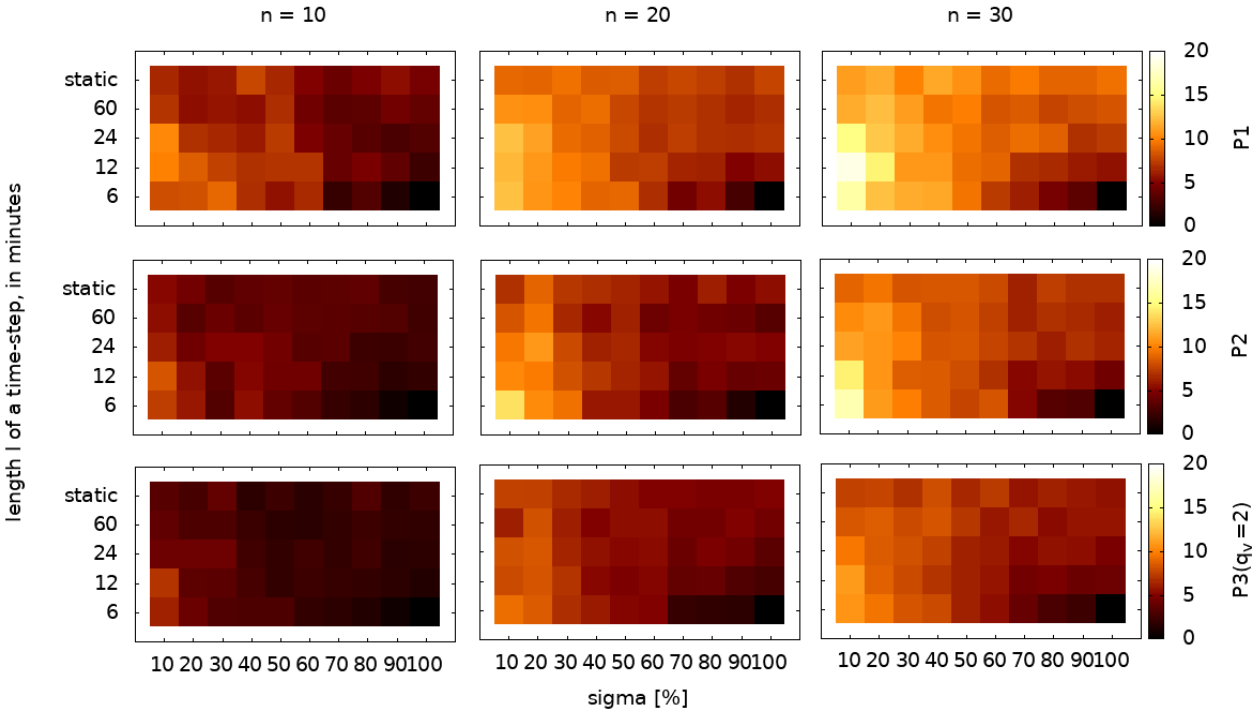


Figure 7: Impact of $(\sigma, l)$ on the quality gap $Q$, for $P_1$ (upper row), $P_2$ (middle row), and $P_3$ (lower row) problems, and for $n = 10$ (left column), $n = 20$ (middle column) and $n = 30$ (right column). For each of these 9 cases, we display a rectangle composed of $5 \times 10$ cells such that the color of each cell $(\sigma, l)$ (with $\sigma \in \{10, 20, \ldots, 100\}$ and $l \in \{6, 12, 24, 60, 720\}$) gives the value of $Q(T^{D_\sigma Sl})$ (on average for the 30 instances).

pattern holds in general for dense spatial covers, when $\sigma \geq 60$. For example, when $\sigma = 80\%$ and $n = 30$, the gap $Q(T^{D_\sigma Sl})$ is equal to 4.6% (resp. 6.6, 8.8, 7.7 and 8.9%) for $l = 6$ (resp. 12, 24, 60 and 720). However, this is no longer true for sparse spatial covers, when $\sigma \leq 50$, and a rather inverted order is observed. When $\sigma = 10\%$, for instance, the gap is equal to 16.5% (resp. 18.9, 15.2, 11.7 and 11.3%) for $l = 6$ (resp. 12, 24, 60 and 720). Hence, when $\sigma \leq 50$, optimizing with small time-steps is not interesting, and better tours are computed with constant cost functions (when $l = 720$).

Actually, when $l = 6$, the gap increases when $\sigma$ decreases: tours computed with $\sigma = 100$ are much better than those computed with $\sigma = 10$. This phenomenon is still observed with $l \in \{12, 24\}$, though it is less obvious. However,

when using the constant cost function ($l = 720$), the quality of tours no longer depends on the spatial coverage $\sigma$ and gaps of tours computed with $\sigma = 100$ are not very different from gaps of tours computed with $\sigma = 10$.

As a conclusion, the answer to Q2 is: when $l = 6$, the spatial coverage has a strong influence and the larger $\sigma$ the better the tours, but when increasing $l$ the influence of the spatial coverage decreases and with constant cost functions (when $l = 720$), the quality of tours does not really depend on the spatial coverage.

### *6.4. Q3: Does the impact of $(\sigma, l)$ change when adding precedence or capacity constraints?*

To answer this question, we first compare results for Problems $P_1$, $P_2$ and $P_3$ in Fig. 7 as these problems are gradual in terms of constraints: $P_2$ adds precedence constraints to $P_1$, and $P_3$ adds capacity constraints to $P_2$. We observe very similar behaviors for the three problems. However, when adding constraints, the quality gap decreases: it is higher for $P_1$ than for $P_2$, and higher for $P_2$ than for $P_3$, in almost all cases when fixing the spatio-temporal granularity $(\sigma, l)$. In other words, the interest of exploiting time-dependent cost functions decreases when adding constraints. This is explained by the fact that adding constraints drastically reduces the number of feasible tours. Indeed, if there exists only one feasible tour, then the same tour is computed whatever the cost function is.

Fig. 8 additionally confirms this fact: it shows us that when we increase the vehicle capacity for Problem $P_3$ from $q_v = 2$ to $q_v = 8$ (thus increasing the number of feasible tours), the quality gap tends to increase.

As a conclusion, the answer to Q3 is: when adding precedence or capacity constraints, the interest of exploiting time-dependent cost functions decreases.

### *6.5. Q4: What is the impact of $(\sigma, l)$ on the satisfaction of time windows?*

Fig. 9 shows the evolution of $F^{\%}$ (as defined in Eq. 18) with respect to the spatio-temporal granularity $(\sigma, l)$ for problem $P_4$ with $n = 40$ points to visit when varying $nTW$ and $xTW$. As expected, we observe that $F^{\%}$ decreases when decreasing $xTW$, as this decreases the width of time windows: When $xTW =$

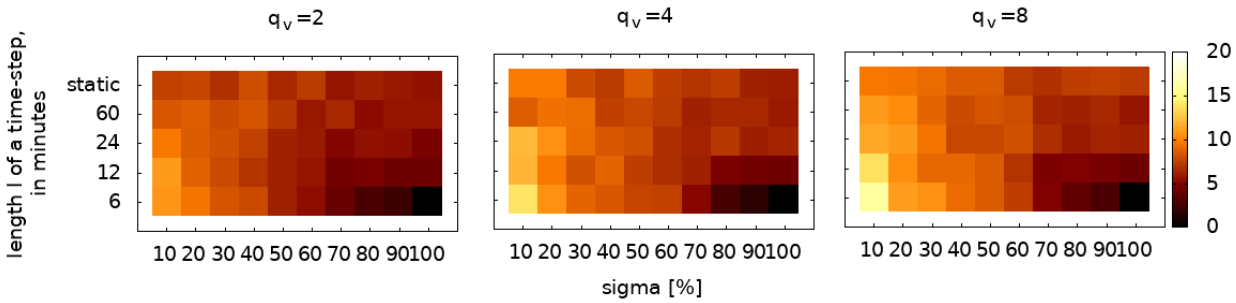


Figure 8: Impact of $(\sigma, l)$ on the quality gap $Q$ for Problem $P_3$ when $n = 30$ and the capacity $q_v$ ranges from 2 to 8. The color scale of cells is kept the same as in Fig. 7.

100 (resp. 200), $F^{\%}$ is very often equal to 0% (resp. 100%), for all spatio-temporal granularities. We also observe that $F^{\%}$ increases when decreasing the number of time windows $nTW$. When decreasing $nTW$, the number of points that must be visited within a same time window is increased so that the average time allowed for each travel within a same time window stays equal to $xTW$. However, when the number of points that must be visited within a same time window increases, the number of possible permutations for visiting these points increases exponentially and, therefore, the probability that there exists one of these permutations which fits in the time window increases.

Now, let us focus on the central case where $nTW = 4$ and $xTW = 150$. In this case, the spatio-temporal granularity has a strong influence on feasibility. With a perfect information (when $\sigma = 100$ and $l = 6$), $F^{\%} = 100\%$, *i.e.*, every instance is $(100, 6)$-feasible, and when the time step size increases, $F^{\%}$ decreases: $F^{\%}$ is equal to 100% (resp. 57, 20, 7, and 0%) when $l = 6$ (resp. 12, 24, 60, and 720). For the spatial dimension, different behaviors are observed depending on whether $l = 6$ or $l \geq 12$. When $l = 6$, $F^{\%}$ tends to decrease when decreasing $\sigma$: for example, when $\sigma = 100\%$ (resp. 50 and 10%), $F^{\%} = 100\%$ (resp. 67 and 50%). However, when $l \geq 12$, $F^{\%}$ tends to increase when decreasing $\sigma$: for example, when $l = 24$ and $\sigma = 100\%$ (resp. 50 and 10), $F^{\%} = 20\%$ (resp. 33 and 60%).

In Fig. 10, we fix $nTW$ to 4 and $xTW$ to 150, and we display the evolution

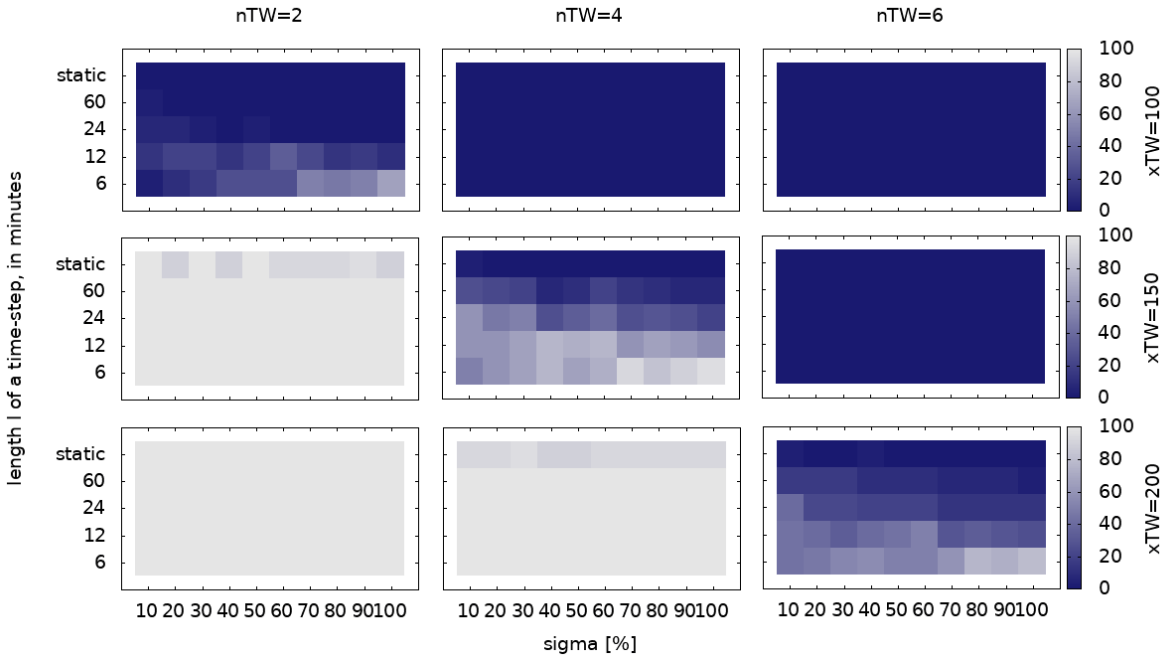


Figure 9: Impact of $(\sigma, l)$ on $F^{\%}$ for problem $P_4$ when $n = 40$. The time window tightness $xTW$ ranges from 100 (upper row) to 200 (lower row), and the number of time windows $nTW$ ranges from 2 (left column) to 6 (right column). For each of these 9 cases, the color of each cell $(\sigma, l)$ (with $\sigma \in \{10, 20, \ldots, 100\}$ and $l \in \{6, 12, 24, 60, 720\}$) gives the value of $F^{\%}$.

of $F^{\%}$ with respect to the spatio-temporal granularity $(\sigma, l)$ for Problem $P_5$, when varying the capacity $q_v$ of the vehicle from 2 to 6 and the number $n$ of points to visit from 40 to 60. As expected, increasing the capacity $q_v$ increases $F^{\%}$ as this increases the number of valid tours. Also, as observed in Fig. 9 for $P_4$, $F^{\%}$ decreases when increasing $l$ and, for the spatial dimension $\sigma$, two different behaviors are observed depending on $l$: when $l = 6$, $F^{\%}$ decreases when decreasing $\sigma$ whereas when $l \geq 12$, $F^{\%}$ increases when decreasing $\sigma$. This behaviour is observed for the three values of $n$ (from 40 to 60).

To provide explanations of this phenomenon, we display average shortest path travel times for each time-dependent cost function $D_{\sigma}Sl$ in Fig. 11. Green (resp. red) cells correspond to cases where average travel times of $D_{\sigma}Sl$ are larger (resp. smaller) than those of $D_{100}S6$. We observe that travel times

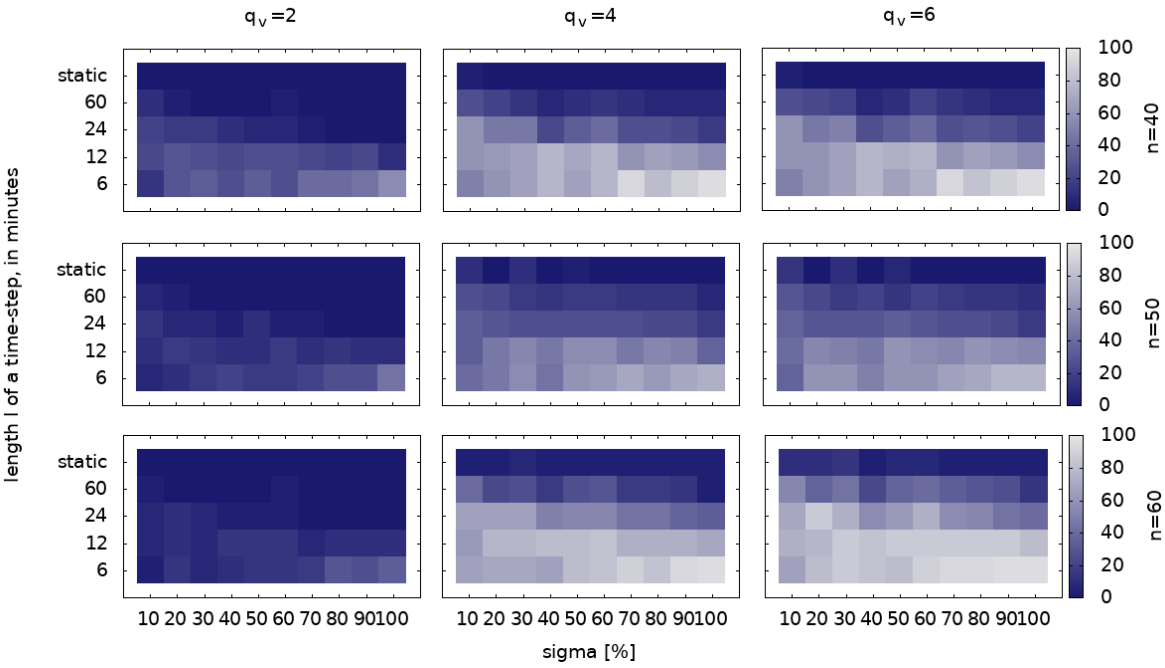


Figure 10: Impact of $(\sigma, l)$ on $F^{\%}$ for problem $P_5$ when $nTW = 4$ and $xTW = 150$. The number of points $n$ ranges from 40 (upper row) to 60 (lower row), and the capacity $q_v$ ranges from 2 (left column) to 6 (right column). The color of each cell $(\sigma, l)$ gives the value of $F^{\%}$.

decrease when $\sigma$ decreases whereas they increase when $l$ increases[7].

When $l = 6$ and $\sigma = 100$, $F^{\%} = 100\%$, meaning that all instances are $(100, 6)$-feasible. However, time windows are rather tight. Hence, when $(\sigma, l)$ is such that travel times increase (green cells), many instances become $(\sigma, l)$-infeasible. For example, when $l = 720$, $F^{\%}$ is close to 0% for all values of $\sigma$ because shortest path travel times increase of 30%, on average, making it impossible to find a tour which satisfies all time windows when travel times are computed with $D_\sigma Sl$.

On the contrary, when $(\sigma, l)$ is such that travel times decrease (red cells),

[7]Note that this phenomenon is not observed at the level of road links, *i.e.*, the average travel time of a road link during the whole time horizon is roughly the same for all values of $l$ and $\sigma$. The fact that average travel times of shortest paths change when changing $l$ or $\sigma$ comes from the fact that road links have different travel time variations, even though they have the same average value when considering the whole time horizon.

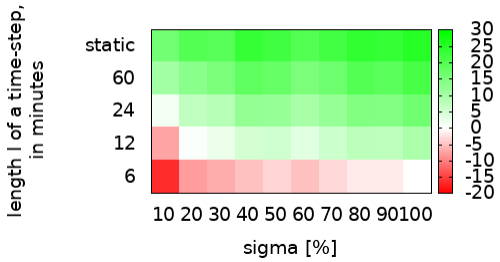


Figure 11: Shortest path travel times. The color of each cell $(\sigma, l)$ corresponds to $(sp(D_\sigma Sl) - sp(D_{100}S6)) \times 100/sp(D_{100}S6)$ where $sp(D)$ is the travel time of a shortest path computed with the cost function $D$, on average for the whole time horizon and for every couple of points of the 30 instances with $n = 40$ points.

all instances are $(\sigma, l)$-feasible. However, it may happen that the optimal tour $T^{D_\sigma Sl}$ is $(100, 6)$-inconsistent as travel times of $D_{100}S6$ are greater than those of $D_\sigma Sl$. For example, travel times of $D_{10}S6$ are 20% faster than travel times of $D_{100}S6$, on average. Therefore when evaluating $T^{D_{10}S6}$ with $D_{100}S6$, some time windows that were satisfied with $D_{10}S6$ are no longer satisfied with $D_{100}S6$.

As a conclusion, the answer to question Q4 is: the impact of the spatio-temporal granularity on feasibility depends on the tightness of time windows.

- When time windows are very tight, most instances are infeasible for all values of $\sigma$ and $l$ including $\sigma = 100$ and $l = 6$.
- When time windows are very large, most instances are feasible and optimal tours are still consistent when evaluating travel times with $D_{100}S6$, for all values of $\sigma$ and $l$ including $\sigma = 10$ and $l = 720$.
- Between these two extreme cases, the temporal granularity $l$ has a strong and consistent impact on feasibility: $F^{\%}$ decreases when increasing $l$. The spatial granularity $\sigma$ also has a strong impact on feasibility, but this impact depends on $l$: when $l = 6$ (resp. $l \geq 12$), $F^{\%}$ decreases (resp. increases) when decreasing $\sigma$. This impact is explained by the fact that shortest path travel times decrease when $\sigma$ decreases whereas they increase when $l$ increases.

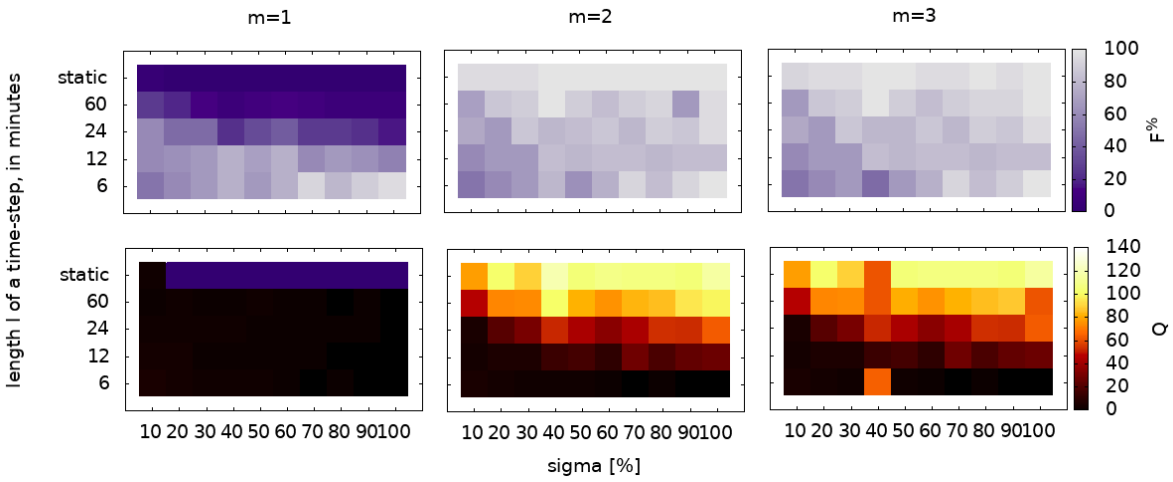


Figure 12: Impact of $(\sigma, l)$ for $P_6$ when $n = 40$, $nTW = 4$, $xTW = 150$, $q_v = 4$, and $m$ ranges from 1 to 3. Top row: The color of each cell $(\sigma, l)$ gives the value of $F^{\%}(D_\sigma Sl)$. Bottom row: The color of each cell $(\sigma, l)$ gives the value of $Q(T^{D_\sigma Sl})$, on average for all $(\sigma, l)$-feasible instances for which $T^{D_\sigma Sl}$ is $(100, 6)$-consistent (cells are colored in blue when $F^{\%} = 0\%$).

### *6.6. Q5: Does the impact of $(\sigma, l)$ change when increasing the number of vehicles?*

In the top row of Fig. 12, we display the evolution of $F^{\%}$ for problem $P_6$ (with $n = 40$) when increasing the number $m$ of vehicles from 1 to 3. In most cases, $F^{\%}$ increases when increasing $m$. This increase is very strong for constant cost functions (when $l = 720$): $F^{\%}$ is always equal to 0% when $m = 1$ whereas it is always close to 100% when $m \geq 2$, for all values of $\sigma$. Surprisingly, when $m \geq 2$, $F^{\%}$ is nearly always larger for $l = 720$ than for $l = 6$.

Again, this rather counter-intuitive phenomenon may be explained by looking at shortest path travel times displayed in Fig. 11. Indeed, when increasing the number $m$ of vehicles from 1 to 2 or 3, time windows become easier to satisfy. Therefore, most instances are $(\sigma, 720)$-feasible even if shortest path travel times with $D_\sigma S720$ are 30% larger than those of $D_{100}S6$, on average. When an instance is $(\sigma, 720)$-feasible, its optimal tour $T^{D_\sigma S720}$ is always $(100, 6)$-consistent as shortest path travel times of $D_{100}S6$ are smaller than those of $D_\sigma S720$: it may happen that the vehicle arrives before the beginning of the time window but, in this case, the vehicle can wait. This explains why $F^{\%}$ is close to 100%

with $l = 720$ when $m \geq 2$.

Finally, we display in the bottom row of Fig. 12 the quality gap, on average for the instances which are $(\sigma, l)$-feasible and for which $T^{D_\sigma Sl}$ is $(100, 6)$-consistent. Whem $m \geq 2$, we observe that the quality gap consistently increases when $l$ increases, for all values of $\sigma$. In particular, with constant cost functions ($l = 720$), the quality gap is often larger than 100%, *i.e.*, the realistic travel time of solutions optimized with constant costs is often more than twice as large as the travel time of of solutions optimized with $D_{100}S6$.

As a conclusion, the answer to Q5 is: yes, the impact of $(\sigma, l)$ on the satisfaction of time windows changes when increasing the number of vehicles because time windows are easier to satisfy when increasing the number of vehicles. When $m \geq 2$, feasibility with constant data is close to 100%. However, solutions optimised with constant data have larger travel times than those optimized with time-dependent data, and the smaller the time-step $l$, the larger the travel time.

## 7. Conclusion

We have reviewed the literature on time-dependent routing problems, with a specific focus on benchmarks and performance measures. In most papers, the main performance criterion is the CPU time needed to compute solutions. Only a few papers have studied the interest of exploiting time-dependent data, and many existing benchmarks are not well suited for this kind of study, either because they have been randomly generated, or because they have rather large time-steps. Furthermore, real-world traffic data used to generate benchmarks are generally incomplete since no actual distribution of sensors fully captures the complex traffic flow dynamics, and the impact of this spatial distribution on solution quality has never been studied.

This motivated us for using a microscopic simulation platform augmented with realistically estimated traffic demand inputs of Lyon to generate more comprehensive and realistic traffic data. This allowed us to generate a realistic benchmark with different spatio-temporal granularity levels: for the spatial dimension, we considered different scenarios for choosing the number and the

position of the sensors, ranging from a complete coverage where every road link is equipped with a sensor to a realistic coverage where sensors are positioned as in the Lyon traffic network; for the temporal dimension, we considered different time-step lengths, ranging from 6 to 720 minutes.

We have used this benchmark to study the impact of the spatio-temporal granularity of traffic data on the quality of solutions of several classical routing problems. We have started our study with the TD-TSP, without any additional constraint, and we have shown that, in case of complete spatial information, optimizing tours with time-dependent costs is highly relevant, and smaller time-steps lead to better tours. For instance, the average temporal gain is equal to 9% with 6 minute time-steps when the number of points to visit is equal to 30, and this gain is larger than 15% for several instances. However, when decreasing the percentage of road links covered by sensors, it becomes less interesting to exploit time-dependent data and when less than 50% of the road links are covered by sensors, better tours are computed with constant cost functions. Also, when considering a realistic spatial distribution where sensors are positioned as in the Lyon network, the gain is null.

Then, we have considered routing problems with additional constraints, *i.e.*, precedence, capacity and time window constraints. These constraints occur, for example, in pickup and delivery problems and in dial-a-ride problems. We have shown that exploiting time-dependent data becomes less profitable when adding precedence and/or capacity constraints, because these constraints reduce the number of valid tours while their satisfaction does not depend on travel times. However, this is not the case for time window constraints as the satisfaction of these constraints depends on arrival times, and we have shown that exploiting time-dependent data has a strong impact on the satisfaction of these constraints when they are tight.

Finally, we have considered the case where several vehicles are available and shown that, if most solutions computed with constant data satisfy time window constraints, they are often more than twice as long compared to solutions computed with 6 minute time steps.

Our study has substantial implications for transportation planning. First, it shows that logistics providers can still rely on classical (constant) routing problems to reduce their operational costs when there is no time window constraints since traffic data is usually obtained from sparse sensor networks: in this case, there is no evidence that exploiting time-dependent data allows to compute better solutions. However, when there are time window constraints, logistics providers should exploit time-dependent data as this both increases feasibility and decreases travel times.

Second, time-dependent cost functions should have small time steps. In particular, solutions computed with time steps of one hour are not better than solutions computed with constant data, even when there are time windows. Hence, new approaches for solving TD-GPDPs should be evaluated on benchmarks with small time steps. We hope that our public benchmark will be useful for this kind of evaluation and become a reference benchmark.

Finally, our study has shown that the number and the position of the sensors have a strong impact on the quality of the results. In particular, we have shown that the current sensor infrastructure in Lyon does not allow to obtain reliable time-dependent data when using interpolation to compute travel times of road links not equipped with sensors. Hence, local authorities should improve this infrastructure. We should study other policies for choosing the position of the sensors, and other approaches than interpolation for completing missing data.

[1] R. Prud'homme, C.-W. Lee, Size, sprawl, speed and the efficiency of cities, Urban Studies 36 (11) (1999) 1849–1858.

[2] P. Christidis, J. N. I. Rivas, et al., Measuring road congestion, Institute for Prospective and Technological Studies, Joint Research Centre, Brussels (2012).

[3] M. Gendreau, G. Ghiani, E. Guerriero, Time-dependent routing problems: A review, Computers & Operations Research 64 (2015) 189–197.

[4] C. Malandraki, M. S. Daskin, Time dependent vehicle routing problems:

Formulations, properties and heuristic algorithms, Transportation Science 26 (3) (1992) 185–200.

[5] C. Malandraki, R. B. Dial, A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem, European Journal of Operational Research 90 (1) (1996) 45–55.

[6] J. Schneider, The time-dependent traveling salesman problem, Physica A: Statistical Mechanics and its Applications 314 (1) (2002) 151–155.

[7] S. Ichoua, M. Gendreau, J.-Y. Potvin, Vehicle dispatching with time-dependent travel times, European Journal of Operational Research 144 (2) (2003) 379–396.

[8] B. Fleischmann, M. Gietz, S. Gnutzmann, Time-varying travel times in vehicle routing, Transportation Science 38 (2) (2004) 160–173.

[9] A. Haghani, S. Jung, A dynamic vehicle routing problem with time-dependent travel times, Computers & operations research 32 (11) (2005) 2959–2986.

[10] F. Li, B. Golden, E. Wasil, Solving the time dependent traveling salesman problem, in: The Next Wave in Computing, Optimization, and Decision Technologies, Springer, 2005, pp. 163–182.

[11] R. Eglese, W. Maden, A. Slater, A road timetabletm to aid vehicle routing and scheduling, Computers & operations research 33 (12) (2006) 3508–3519.

[12] T. Van Woensel, L. Kerbache, H. Peremans, N. Vandaele, Vehicle routing with dynamic travel times: A queueing approach, European journal of operational research 186 (3) (2008) 990–1007.

[13] A. V. Donati, R. Montemanni, N. Casagrande, A. E. Rizzoli, L. M. Gambardella, Time dependent vehicle routing problem with a multi ant colony system, European Journal of Operational Research 185 (3) (2008) 1174–1191.

[14] J. F. Ehmke, A. Steinert, D. C. Mattfeld, Advanced routing for city logistics service providers based on time-dependent travel times, Journal of computational science 3 (4) (2012) 193–205.

[15] H. Kanoh, J. Ochiai, Solving time-dependent traveling salesman problems using ant colony optimization based on predicted traffic, in: Distributed Computing and Artificial Intelligence (DCAI), Springer, 2012, pp. 25–32.

[16] M. A. Figliozzi, The time dependent vehicle routing problem with time windows: Benchmark problems, an efficient solution algorithm, and solution characteristics, Transportation Research Part E: Logistics and Transportation Review 48 (3) (2012) 616–636.

[17] J. Cordeau, G. Ghiani, E. Guerriero, Analysis and branch-and-cut algorithm for the time-dependent travelling salesman problem, Transportation Science 48 (1) (2014) 46–58.

[18] P. A. Melgarejo, P. Laborie, C. Solnon, A time-dependent no-overlap constraint: Application to urban delivery problems, in: Proceedings of Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR), Springer, 2015, pp. 1–17.

[19] A. Montero, I. Méndez-Díaz, J. J. Miranda-Bront, An integer programming approach for the time-dependent traveling salesman problem with time windows, Computers & Operations Research 88 (2017) 280–289.

[20] D. M. Vu, N. Boland, M. Hewitt, M. Savelsbergh, Solving time dependent traveling salesman problems with time windows, Optimization Online 6640 (2018).

[21] A. Arigliano, G. Ghiani, A. Grieco, E. Guerriero, I. Plana, Time-dependent asymmetric traveling salesman problem with time windows: Properties and an exact algorithm, Discrete Applied Mathematics 261 (2019) 28–39.

[22] P. Sun, L. P. Veelenturf, M. Hewitt, T. Van Woensel, Adaptive large neighborhood search for the time-dependent profitable pickup and delivery prob-

lem with time windows, Transportation Research Part E: Logistics and Transportation Review 138 (2020) 101942.

[23] R. M. Karp, Reducibility among combinatorial problems, in: Complexity of Computer Computations, Springer, 1972, pp. 85–103.

[24] M. Held, R. M. Karp, A dynamic programming approach to sequencing problems, Journal of the Society for Industrial and Applied mathematics 10 (1) (1962) 196–210.

[25] J. Desrosiers, Y. Dumas, F. Soumis, A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows, American Journal of Mathematical and Management Sciences 6 (3-4) (1986) 301–325.

[26] N. Chiabaut, M. Küng, M. Menendez, L. Leclercq, Perimeter control as an alternative to dedicated bus lanes: A case study, Transportation Research Record 2672 (20) (2018) 110–120.

[27] G. F. Newell, A simplified car-following theory: a lower order model, Transportation Research Part B: Methodological 36 (3) (2002) 195–205.

[28] J. A. Laval, L. Leclercq, Microscopic modeling of the relaxation phenomenon using a macroscopic lane-changing model, Transportation Research Part B: Methodological 42 (6) (2008) 511–522.

[29] L. Leclercq, Bounded acceleration close to fixed and moving bottlenecks, Transportation Research Part B: Methodological 41 (3) (2007) 309–319.

[30] INSEE, Population census results of France of 2016, `https://www.insee.fr/fr/information/2008354`, accessed: 18 June 2020 (2016).

[31] L. C. Edie, Discussion of traffic stream measurements and definitions, in: Proceedings of the 2nd International Symposium on the Theory of Traffic Flow, OECD, 1963, pp. 139–154.

[32] L. Leclercq, N. Chiabaut, B. Trinquier, Macroscopic fundamental diagrams: A cross-comparison of estimation methods, Transportation Research Part B: Methodological 62 (2014) 1–12.

[33] J. Salotti, S. Fenet, R. Billot, N.-E. E. Faouzi, C. Solnon, Comparison of traffic forecasting methods in urban and suburban context, in: Proceedings of International Conference on Tools with Artificial Intelligence (ICTAI), IEEE, 2018.

[34] J. Coldefy, Optimodlyon ou la mise en synergie des acteurs publics et privés pour une gestion optimale de lintermodalité, Pollution Atmosphérique (231) (2016) 217–235.

[35] J. F. Ehmke, A. M. Campbell, B. W. Thomas, Optimizing for total costs in vehicle routing in urban areas, Transportation Research Part E: Logistics and Transportation Review 116 (2018) 242–265.

[36] D. E. Kaufman, R. L. Smith, Fastest paths in time-dependent networks for intelligent vehicle-highway systems application, Journal of Intelligent Transportation Systems 1 (1993) 1–11.