European Journal of Operational Research 313 (2024) 465–477

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Mathematical formulations for consistent travelling salesman problems

Daniel Díaz-Ríos, Juan-José Salazar-González*

IMAULL, University of La Laguna, La Laguna 38200, Tenerife, Spain

ARTICLE INFO

Article history:
Received 25 October 2022
Accepted 12 August 2023

Keywords:
Travelling salesman
Time consistency
Benders' decomposition
Branch and cut

ABSTRACT

The consistent travelling salesman problem looks for a minimum-cost set of Hamiltonian routes, one for every day of a given time period. When a customer requires service in several days, the service times on different days must differ by no more than a given threshold (for example, one hour). We analyze two variants of the problem, depending on whether the vehicle is allowed to wait or not at a customer location before its service starts. There are three mathematical models in the literature for the problem without waiting times, and this paper describes a new model appropriated to be solved with a branch-and-cut algorithm. The new model is a multi-commodity flow formulation on which Benders' Decomposition helps manage a large number of flow variables. There were no mathematical models in the literature for the variant with waiting times, and this paper adapts the four mathematical models to it. We analyze the computational results of the formulations on instances from the literature with up to 100 customers and three days.

1. Introduction

Given a set of customers, the Travelling Salesman Problem (TSP) aims to find a minimum-cost circuit for a vehicle to visit each customer exactly once. In a sense, the TSP looks for a single route to visit the customers requiring service on a given day. It is assumed that each customer accepts the service at any time of the day. When several days are considered and a customer requires service on different days, solving a TSP for each day independently could have very different arrival times for that customer. For example, the vehicle could serve a customer at 09:30 on Tuesdays and at 16:45 on Thursdays. In some applications this dissimilarity is undesired, thus motivating the problem studied in this article: the Consistent Travelling Salesman Problem (CTSP). In the CTSP we assume to be given a threshold T representing the maximum allowed dissimilarity between the arrival times to a customer on different days. For example, T may be one hour, meaning that when a customer requires service on two days and is visited at 09:30 on one of them, then it must be visited between 08:30 and 10:30 on the other day.

This variant of the TSP is also applicable to the multi-vehicle extension generally known as the Vehicle Routing Problem (VRP); see, e.g., Toth & Vigo (2002). It has arisen in the face of market evolution where companies are trying to provide better and better customer service as explained by Groër, Golden, & Wasil (2009) on the small package shipping industry, or other contexts such as the transport of people with disabilities (see, e.g., Feillet, Garaix, Lehuédé, Péton, & Quadri, 2014) or service level agreements in the pharmaceutical industry (see, e.g., Campelo, Neves-Moreira, Amorim, & Almada-Lobo, 2019).

The literature includes articles dealing with consistency in other different contexts and meanings. For example, when the service is executed with a fleet of vehicles, a customer may be desired to be served by the same driver; see, e.g., Rodríguez-Martín, Salazar-González, & Yaman (2019). This is called *driver consistency*. Another example occurs when the demand of the customer is split and served in different visits, and it is desired to receive roughly the same amount at each visit; see, e.g., Gulczynski, Golden, & Wasil (2010). We refer to this as *demand consistency*. There are articles dealing with consistency in other contexts, such as Inventory Routing (see, e.g., Coelho, Cordeau, & Laporte, 2012), where visits to customers should ideally be spread out evenly over the planning horizon to ensure smoother operations, and the quantities delivered should be controlled to avoid large variations over time, which are negatively perceived by customers. Yao, Van Woensel, Veelenturf, & Mo (2021) focus on *path consistency*, which encourages vehicles to perform similar paths every day by providing a discount for the travel cost on some sections of the road network.

* Corresponding author.
E-mail addresses: ddiazrio@ull.edu.es (D. Díaz-Ríos), jjsalaza@ull.es (J.-J. Salazar-González).

https://doi.org/10.1016/j.ejor.2023.08.021

Recently, Alvarez, Cordeau, & Jans (2022) address a production-routing problem where they also consider *source consistency* (regarding the different plants from which a customer is served), *product consistency* (for the different number of product types manufactured in a plant), and *plant consistency* (on the number of plants manufacturing a product). Our paper does not consider these other meanings for consistency, but the one on the arrival times of a vehicle to a customer. We call this *time consistency*.

The Consistent VRP (ConVRP) with time consistency and driver consistency at the same time has been approached from different perspectives. On the one hand, Tarantilis, Stavropoulou, & Repoussis (2012) and Kovacs, Parragh, & Hartl (2014c) proposed a template-based meta-heuristic, while Goeke, Roberti, & Schneider (2019) was the only study to address the ConVRP with exact algorithms. The Generalized ConVRP has been approached by Kovacs, Golden, Hartl, & Parragh (2014a), where the number of drivers visiting each customer is constrained and the variation in arrival times is penalized in the objective function. A multi-period dial-a-ride problem with driver consistency is introduced and heuristically solved in Braekers & Kovacs (2016). Other variants of the ConVRP have also emerged, such as the Collaborative ConVRP with workload balance (see, e.g., Mancini, Gansterer, & Hartl, 2021), the Multi-objective ConVRP (see, e.g., Kovacs, Parragh, & Hartl, 2015 and Lian, Milburn, & Rardin, 2016), the Multi-period ConVRP (see, e.g., Luo, Qin, Che, & Lim, 2015), the ConVRP with Profits (see, e.g., Stavropoulou, Repoussis, & Tarantilis, 2019), the ConVRP with simultaneous distribution and collection (see, e.g., Zhen, Lv, Wang, Ma, & Xu, 2020), the ConVRP with a heterogeneous fleet (see, e.g., Stavropoulou, 2022), and the ConVRP with limited charging resources (see, e.g., Nolz, Absi, Feillet, & Seragiotto, 2022).

Even in the single vehicle case, the CTSP admits several variants depending on whether the vehicle is allowed to idle (or wait) along the route, whether the potential idle time is included in the cost of the route, and whether the time inconsistencies are limited as hard constraints, or minimized in the objective function. For our purpose, there is no difference between idling at a customer before or after the service, or even idling along the route between two consecutive customers. In this paper, the *duration of a route* includes the travel times, service times and idle times if they exist, while the *cost of a route* is the sum of travel costs, ignoring the impact of idle times when they exist. For brevity and clarity, our paper describes two CTSP variants.

The first variant is the CTSP with no idle time, referred to here as CTSP1. The objective function is to minimize the cost of the routes, and the worst time inconsistency in a customer is limited by a given threshold T. CTSP1 is introduced in Kovacs, Golden, Hartl, & Parragh (2014b) and mathematically formulated in Subramanyam & Gounaris (2016) through three models solving instances with 50 customers. To follow the convention in the literature, all routes leave the depot at the same time (e.g., 07:00), which can be interpreted as idle time being forbidden in the CTSP1 even at the depot. Additionally, for the convention, the travel costs and the travel times coincide, and the service time at customers is assumed to be zero. CTSP1 is appropriate for those companies where drivers want to finish as soon as possible, so they do not wait along the route.

The second variant is the CTSP with idle times, with the worst-case time inconsistency limited by a given threshold T. It is referred to as CTSP2. We assume in CTSP2 that all routes depart at the same time (e.g., 07:00), although this is not a major constraint because any potential delay in the departure from the depot is equivalent to an idle time before serving the first customer along the route. Subramanyam & Gounaris (2018) describe a dynamic programming approach for the CTSP2 with also a route duration constraint, i.e. the total time elapsed until the vehicle returns to the depot is upper limited. They solve instances with up to 100 customers to proven optimality. In their settings, travel costs and travel times coincide, service times and service costs are zero, and the objective function does not include idle time. In our paper, CTSP2 has the same assumptions but does not include the route duration constraint for coherence with CTSP1. CTSP2 is appropriate for those companies where customer satisfaction is the priority; thus, drivers are forced to wait when convenient.

Making use of the models in this article, one could also deal with other variants of CTSP. For example, one could consider T as a decision variable to be minimized. It could lead to a bicriteria optimization problem, minimizing both the routing cost and the time inconsistency for all customers; it could lead to a single-criterium optimization problem, minimizing the time inconsistency for all customers subject to a routing cost upper limited by a given threshold (as in the Orienteering Problem; see Vansteenwegen, Souffriau, & Oudheusden, 2011). To avoid dispersing the reader, our paper does not analyze these other variants, although some models and algorithms in this paper can be adapted easily.

This article contributes to the literature in four respects. First, it describes a new mathematical formulation for CTSP1. This formulation is based on flow variables to determine the time when a customer is visited. There are a large number of variables, making the formulation impractical if it needs to be solved in a compact format. However, the flow structure makes the formulation quite suitable for solving large instances within a Bender's Decomposition approach. A second contribution of the article is the characterization of the Bender's cuts when separating invalid integer solutions. Indeed, two lemmas in the article describe families of inequalities to be used in a branch-and-cut algorithm, and there is no need to solve linear subproblems for constructing these inequalities. A third contribution is on the CTSP2, where the literature did not contain any mathematical formulation. Our article instead presents fourth formulations adapted from the ones described for the CTSP1. Finally, the fourth main contribution of this article is a computational analysis of our computer implementations solving some benchmark CTSP1 instances. Our computer codes are available as supplementary material of this article to help future researchers.

Section 2 describes CTSP1 and CTSP2, emphasizing their special characteristics through a numerical example with 13 customers. Section 3 shows the three models in the literature and presents a novel model based on Benders' Decomposition for CTSP1. The new model is based on a family of inequalities that can be generated in polynomial time to eliminate invalid integer or fractional solutions. Section 4 adapts the four models to CTSP2. No mathematical formulations existed before for CTSP2, so this is another major contribution of this manuscript. Finally Section 5 analyses the performances of our implementations to solve some CTSP1 instances from the literature, and we report optimal solutions to CTSP1 instances with up to 100 customers and three days.

2. Problem description, notation and example

Let K be a set of days of a time period (e.g., week). Let V be a set of $n+1$ locations, where 0 represents a depot and $1, \dots, n$ represent the customers. Let V^k be the subset of locations to be visited on day k. We assume $0 \in V^k$ for $k \in K$, meaning that the vehicle route starts and ends at the depot every day. We use $G^k = (V^k, A^k)$ for the graph where a Hamiltonian circuit must be generated to compose a CTSP solution. Let us also consider the standard notation $\delta_k^+(S) = \{(u, v) \in A^k : u \in S, v \notin S\}$ and $\delta_k^-(S) = \{(u, v) \in A^k : u \notin S, v \in S\}$ for all $S \subset V^k$, writing $\delta_k^+(i)$ and $\delta_k^-(i)$ instead of $\delta_k^+(\{i\})$ and $\delta_k^-(\{i\})$ for all $i \in V^k$. As in the literature, each arc $a \in A^k$ is associated with a travel cost c_a and a travel time t_a which are independent of day k. Let the *service time of a*

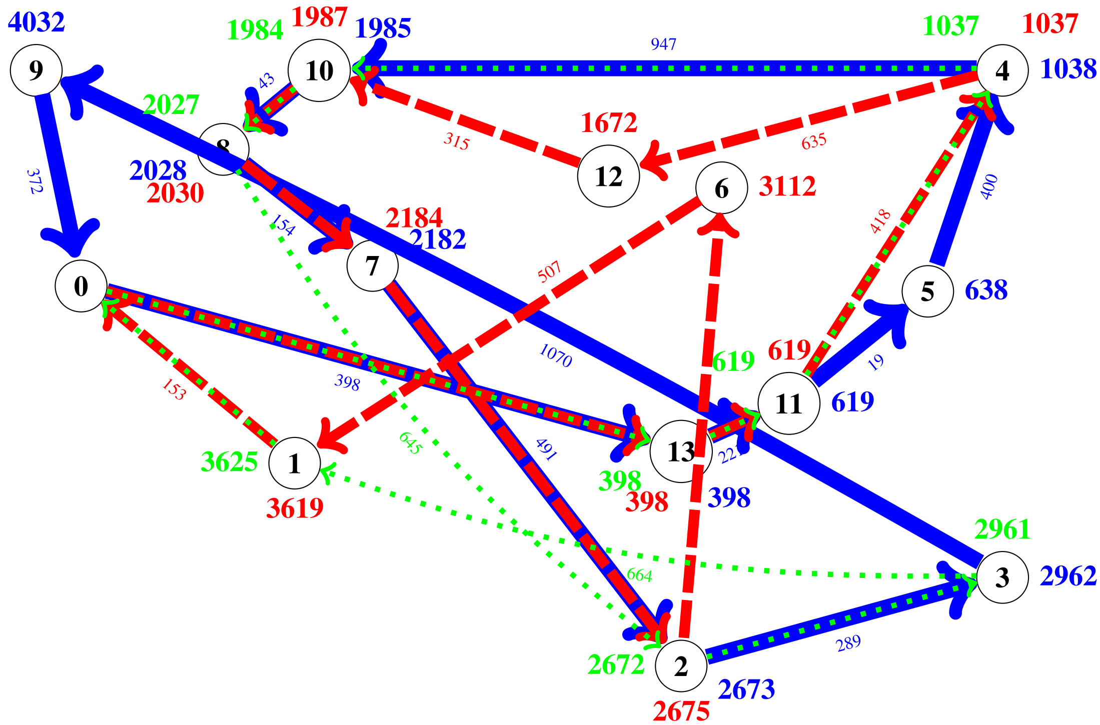


Fig. 1. Optimal CTSP1 solution with $6 \leq T \leq 10$. Optimal Value: 11954.

customer in a day be the moment in that day when the vehicle starts serving the customer. As in the literature, we assume that serving a customer consumes zero time. Let T be the maximum allowed difference between the service times to a customer on two different days. Both CTSP1 and CTSP2 look for Hamiltonian tours minimizing the total sum of travel costs, and are subject to a maximum time inconsistency of T units for all customers. In CTSP1, the vehicle serves a customer immediately when it arrives, and it travels to the next customer when it finishes a service. In other words, idle times are not considered in CTSP1. Instead, the vehicle is allowed to wait along the route in CTSP2. Following assumptions in the literature for these problems, the vehicle is assumed to leave the depot at the same time for all days in CTSP1, while this assumption is relaxed in CTSP2.

To clarify the impact of these characteristics of the problem variants, we now show and discuss some CTSP solutions on the benchmark TSP instance `burma14`. This TSP instance is based on geographical distances between 14 locations and the data can be obtained from the TSPLIB. We consider a time period of three days and generate a CTSP instance with a probability of 0.7 that a customer requires service on each day. In our scenario, depot 0 and customers 2, 4, 8, 10, 11 and 13 require three services, customers 1, 3 and 7 require two services, and customers 5, 6, 9 and 12 require one service.

Figures 1–2 show CTSP1 solutions for different values of T. Each solution includes three routes, each one identified with a different line style (blue continuous, red dashed, green dotted) and associated with a day. The number next to a customer location is the arrival time, and the tiny number near an arrow is the travel cost (also travel time) of the arc connecting the two linked locations. Note that all routes leave the depot at the same time. From Fig. 1, customers 13 and 11 require service every day, and they are the first and second visited customers, respectively, on each day; the vehicle leaves the depot at time 0, arrives at customer 13 at time 398, and arrives at customer 11 at time 398+221. Customer 5 requires service on the blue day, and customer 4 requires service on the three days; for that reason, the blue route arrives at customer 4 at time 398+221+19+400, while the green and red routes arrive at customer 4 at time 398+221+418. The blue, red and green routes return to the depot at times 4032+372, 3619+153 and 3625+153, respectively. The CTSP1 instance with $T < 6$ is infeasible. The worst time inconsistency when $T = 6$ (Fig. 1) occurs at customer 1, where the service starts at time 3619 on the red day and at time 3625 on the green day. Fig. 2 shows the other extreme case, where customers 2 and 13 have a time inconsistency of (at least) 1897 units of time. This CTSP1 solution is composed of the three optimal TSP routes, one for each day. Although the cost matrix is symmetric for this instance, the routes in the opposite direction have the same travel costs but may have different time inconsistencies.

Figure 3 shows a feasible solution when allowing different departure times from the depot, which is a relaxed variant of CTSP1. This problem variant is equivalent to CTSP1 with all routes leaving from the depot at the same time, with potential waiting times at the first customers, and without waiting times at the other customers. The vehicle on the blue route leaves the depot at time 1185. The solution in the figure is optimal for this intermediate variant when $T = 0$.

Figure 4 shows the optimal CTSP2 solution when $T = 0$, which has a smaller total travel cost with respect to the costs of the routes in Fig. 3 because waiting times are allowed at all customer locations. The customer locations in Fig. 4 show two numbers, the first being the waiting time and the second being the time when service starts. For example, the vehicle on the red route arrives at customer 4, waits 36 units of time, starts the service at 1634, travels to customer 11, waits 3599 units of time, starts the service at 5651, travels to customer 6, etc.

It is worth noting that the definition of CTSP2 does not constrain either the waiting times at a customer or the arrival time of the vehicle back to the depot. Similarly, CTSP1 also has no constraint on the length of the route on each day. Adding these constraints to the problem definition may be of practical interest, but out of the scope of our paper. Observe also that, when all the

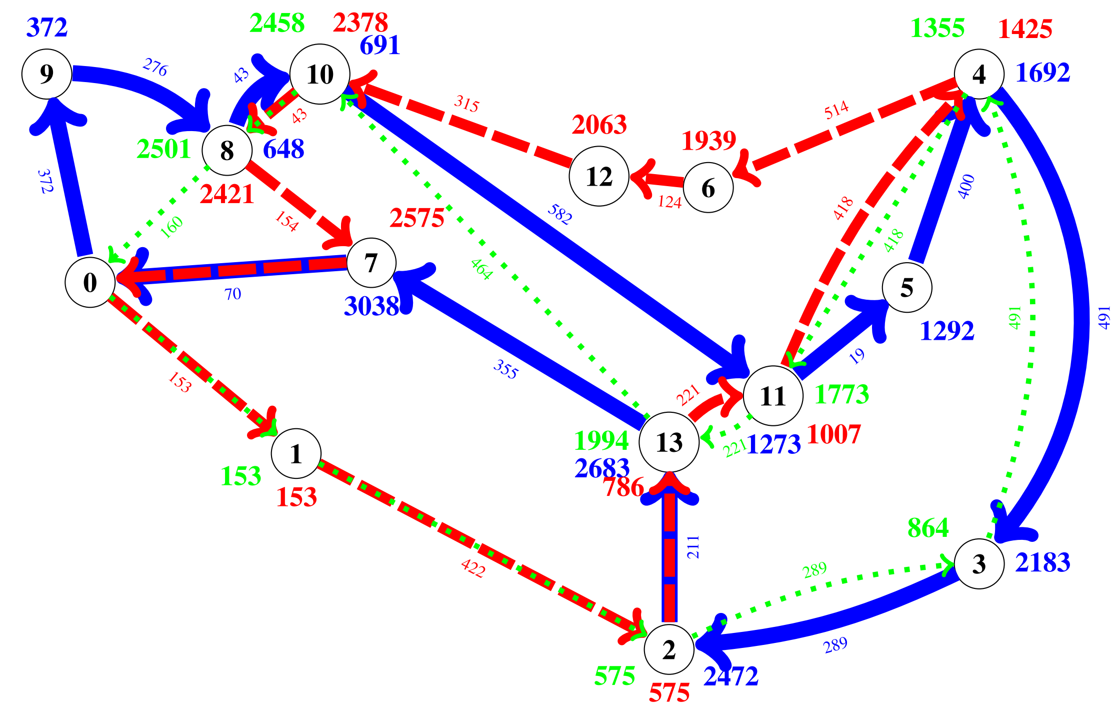


Fig. 2. Optimal CTSP1 solution with $T \geq 1897$. Optimal Value: 8414.

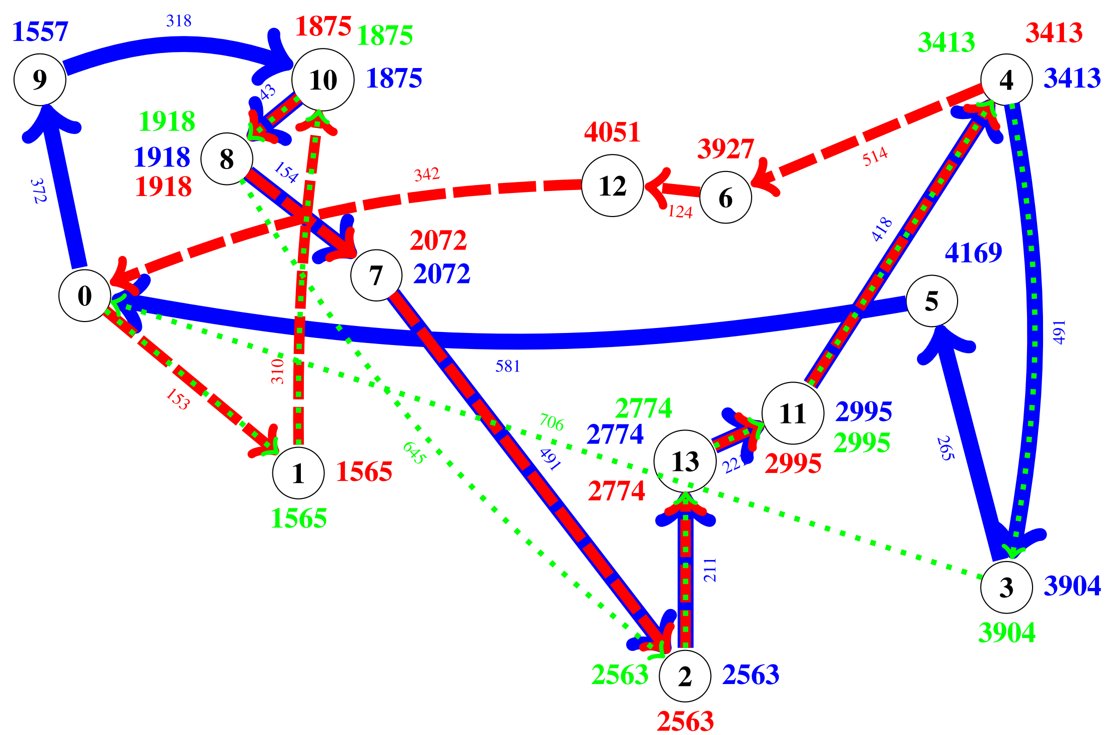


Fig. 3. Optimal solution with idle time at the depot before leaving, no idle time at customers, and $T = 0$. Optimal Value: 9744.

routes in a CTSP2 solution are reversed (i.e., the vehicle follows the route from the last to the first customer), we obtain another CTSP2 solution. This does not occur with a CTSP1 solution due to the requirement that all routes must leave the depot at the same time (time zero). Indeed, reversing all the arcs in Fig. 3 would need $T \geq 2300$ to be a feasible CTSP1 solution.

Table 1 shows the travel cost of the routes of the CTSP1 and CTSP2 solutions for different values of T. Not surprisingly, the total travel cost decreases when T increases. The optimal CTSP1 solution when $T \geq 1897$ and the optimal CTSP2 solution when $T \geq 1667$ coincide and are determined by the optimal TSP route on each day.

3. Consistent TSP without waiting time

This section describes CTSP1. In this problem variant, the vehicle is not allowed to idle at any place along the route. This means that every day, the vehicle departs from the depot at time zero, and it keeps either travelling from one customer to another,

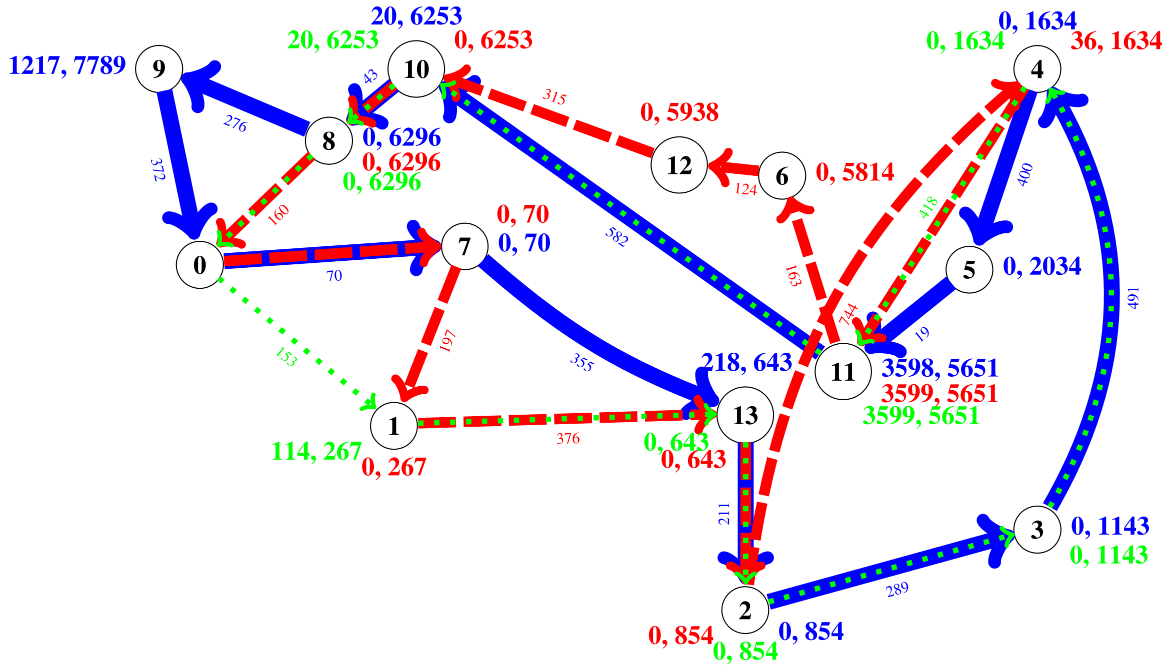


Fig. 4. Optimal CTSP2 solution with $T = 0$. Optimal Value: 8652.

Table 1
Optimal objective values for burma14.

CTSP1						CTSP2	
T	OptVal	T	OptVal	T	OptVal	T	OptVal
0–5	infeasible	65–83	8897	817–882	8526	0–164	8652
6–10	11,954	84–127	8886	883–1014	8508	165–210	8634
11	10,107	128–130	8814	1015–1160	8477	211–417	8572
12–13	10,072	131–145	8693	1161–1191	8446	418	8540
14	9878	146–217	8661	1192–1896	8415	419–637	8508
15–29	9459	218–384	8652	1897– ∞	8414	638–639	8479
30–36	9390	385–387	8634			640–1666	8415
37–64	8911	388–816	8572			1667– ∞	8414

or serving a customer. For simplicity of notation, we assume that serving a customer takes zero time.

3.1. Known mathematical models

We are aware of only one article in the literature with mathematical formulations for CTSP1. Subramanyam & Gounaris (2016) describe three formulations that are summarized in this section. The formulations are based on a 0–1 variable x_a^k equal to 1 if the vehicle goes along arc a on day k for each $k \in K$ and $a \in A^k$. For abbreviation, we write $x^k(B)$ instead of $\sum_{a \in B} x_a^k$ for any $B \subseteq A^k$. With these variables, the objective function of a CTSP1 formulation is

$$\min \sum_{k \in K} \sum_{a \in A^k} c_a x_a^k \tag{1}$$

and basic constraints to guarantee a TSP route for each $k \in K$ are

$$x^k(\delta_k^+(i)) = x^k(\delta_k^-(i)) = 1 \qquad i \in V^k \tag{2}$$

$$x^k(\delta_k^+(S)) \geq 1 \qquad S \subset V^k \setminus \{0\} \tag{3}$$

$$x_a^k \in \{0, 1\} \qquad a \in A^k. \tag{4}$$

The missing part to complete a CTSP1 formulation is the requirement that the service times for each customer served on different days differ in at most T units.

Given a solution x^* of (1)–(4), checking the time consistency requires a very simple computation. Indeed, for each day $k \in K$ and each $i \in V^k$, there is one path P^{ik} from 0 to i in the subgraph defined by the arcs a with $x_a^{k*} = 1$. Hence, the computation consists of evaluating

$$t(P^{ip}) - t(P^{iq}) \leq T \qquad p, q \in K, i \in V^p \cap V^q \setminus \{0\}, \tag{5}$$

where $t(P^{ik})$ denotes $\sum_{a \in P^{ik}} t_a$ for a path P^{ik} in G^k. We now show three alternatives to model this computation in a CTSP1 formulation.

Formulation 1 imposes the time consistency directly on the x_a variables, without additional variables, through the following family of inequalities:

$$x^p(\Delta(P^{ip})) + x^q(\Delta(P^{iq})) \leq |P^{ip}| + |P^{iq}| - 1 \tag{6}$$

for each path P^{ip} from 0 to i in G^p and each path P^{iq} from 0 to i in G^q such that inequality (5) does not hold. $\Delta(P^{ik})$ represents the set of arcs $(u, v) \in A^k$ such that u precedes v in the path P^{ik} for $k \in \{p, q\}$. These inequalities discard two incompatible paths from 0 to i on different days in the same CTSP1 solution. They are called *Tournament Constraints* and are standard to avoid infeasible routes in vehicle routing problems (see, e.g., Dalmeijer & Spliet, 2018).

Formulation 2 models the time consistency by introducing an additional continuous variable z_i^k representing the service time of customer i on day k. Then CTSP1 can be modelled with (1)–(4) and

$$z_j^k \geq z_i^k + t_{(i,j)} \cdot x_{(i,j)}^k - M^k \cdot (1 - x_{(i,j)}^k) + (M^k - t_{(j,i)}) \cdot x_{(j,i)}^k \quad k \in K,\ i,j \in V^k \setminus \{0\},\ i \neq j \tag{7}$$

$$z_j^k \leq z_i^k + t_{(i,j)} \cdot x_{(i,j)}^k + M^k \cdot (1 - x_{(i,j)}^k) - (M^k + t_{(j,i)}) \cdot x_{(j,i)}^k \quad k \in K,\ i,j \in V^k \setminus \{0\},\ i \neq j \tag{8}$$

$$z_j^k \geq t_{(0,j)} \cdot x_{(0,j)}^k \qquad k \in K,\ j \in V^k \setminus \{0\} \tag{9}$$

$$z_j^k \leq t_{(0,j)} \cdot x_{(0,j)}^k + M^k \cdot (1 - x_{(0,j)}^k) \qquad k \in K,\ j \in V^k \setminus \{0\}, \tag{10}$$

where M^k is an upper bound on the duration of the route on day k. Inequalities (7)–(8) imply that $z_j^k = z_i^k + t_{(i,j)}$ when $i,j \in V^k \setminus \{0\}$ and $x_{(i,j)}^k = 1$, and guarantee no idle time at any customer. Inequalities (9)–(10) imply that $z_j^k = t_{(0,j)}$ when $x_{(0,j)}^k = 1$, and they force all routes to leave the depot at time zero.

Constraints (7)–(8) are called *Miller-Tucker-Zemlin inequalities*, and they are standard in vehicle routing problems with time constraints (see, e.g., Bektas & Gouveia, 2014). With the z_i^k variables, the time consistency of a solution (x, z) is ensured by the inequalities

$$z_i^p - z_i^q \leq T \qquad p,q \in K,\ i \in V^p \cap V^q \setminus \{0\}. \tag{11}$$

As usual, large values harm MILP solvers, and for that reason, it is convenient to decrease the M^k values as much as possible without losing the validity of the formulation.

Formulation 3 models the time consistency with a different variable $g_{(i,j)}^k$ for each $k \in K$ and $(i,j) \in A^k$ representing the arrival time at customer i on day k if j is the next location to visit. For the abbreviation, we write $g^k(B)$ instead of $\sum_{a \in B} g_a^k$ for any $B \subseteq A^k$. Then CTSP1 can be modelled with (1)–(4) and

$$g^k(\delta_k^+(j)) - g^k(\delta_k^-(j)) = \sum_{i \in V^k \setminus \{j\}} t_{(i,j)} x_{(i,j)}^k \quad k \in K,\ j \in V^k \setminus \{0\} \tag{12}$$

$$g_a^k = 0 \qquad k \in K,\ a \in \delta_k^+(0) \tag{13}$$

$$0 \leq g_a^k \leq M^k \cdot x_a^k \qquad k \in K,\ a \in A^k \setminus \delta_k^+(0) \tag{14}$$

$$g^p(\delta_p^+(i)) - g^q(\delta_q^+(i)) \leq T \qquad p,q \in K,\ i \in V^p \cap V^q \setminus \{0\}. \tag{15}$$

Eq. (12) increase the arrival time at j with the travel cost $t_{(i,j)}$ with respect to the arrival time at i when the vehicle goes from i to j. Eq. (13) force all the routes to leave the depot at time zero. Inequalities (14) guarantee that time increases only along the route, and Inequalities (15) impose time consistency.

Observe that the constraints on z_j^k and g_a^k variables eliminate subtours disconnected from the depot on each day. Thus, (3) can be eliminated from Formulations 2 and 3 to generate compact models for CTSP1.

Subramanyam & Gounaris (2016) analyze the computational results of the three formulations and conclude that the best formulation is Formulation 1. This is explained by the negative impact of using large values M^k on MILP solvers. Nevertheless, violated inequalities (6) are quite difficult to detect (i.e., separate) on non-integer solutions. The next section proposes a different family of polynomial-time separable inequalities to replace (6) in Formulation 1.

3.2. New mathematical model

The variables g_a^k in Formulation 3 represent a single-commodity flow on G^k for each day k. An alternative formulation arises by considering a multi-commodity flow on each day, with one commodity per customer on that day. More precisely, given a CTSP1 solution x, there is one path P^{ik} from 0 to i on day k if $i \in V^k$. This path can be mathematically represented by continuous variables f_a^{ik} satisfying

$$f^{ik}(\delta_k^+(0)) - f^{ik}(\delta_k^-(0)) = \ 1 \tag{16}$$

$$f^{ik}(\delta_k^+(i)) - f^{ik}(\delta_k^-(i)) = -1 \tag{17}$$

$$f^{ik}(\delta_k^+(j)) - f^{ik}(\delta_k^-(j)) = 0 \qquad j \in V^k \setminus \{0, i\} \tag{18}$$

$$0 \leq f_a^{ik} \leq x_a^k \qquad a \in A^k, \tag{19}$$

and the duration of the path is given by $\sum_{a \in A^k} t_a f_a^{ik}$. Then a new valid CTSP1 formulation, called Formulation 4, is given by (1)–(4) together with (16)–(19) for each $k \in K$ and $i \in V^k \setminus \{0\}$, and

$$\sum_{a \in A^p} t_a f_a^{ip} - \sum_{a \in A^q} t_a f_a^{iq} \leq T \qquad p,q \in K, i \in V^p \cap V^q \setminus \{0\}. \tag{20}$$

This formulation has a larger number of continuous variables than the three formulations in the previous section. However, as this section shows, it also has a special structure that suggests applying Benders' decomposition (see, e.g., Rahmaniani, Crainic, Gendreau, & Rei, 2017) to solve CTSP1 with a cutting-plane or branch-and-cut framework.

First, regardless of whether x is an integer or fractional vector, the duration of any path (i.e., one unit of flow) from 0 to i can be estimated by solving the following two linear programs:

$$\underline{s}^{ik} := \min \sum_{a \in A^k} t_a f_a^{ik} \tag{21}$$

and

$$\bar{s}^{ik} := \max \sum_{a \in A^k} t_a f_a^{ik}, \tag{22}$$

each subject to (16)–(19). Intuitively, $\underline{s}^{ik}$ and $\bar{s}^{ik}$ represent the earliest and latest arrival times, respectively, to serve customer i on day k when the TSP routes are defined by x. The two values coincide when x is an integer vector; they may be different otherwise. Indeed, Fig. 5 shows a fractional solution for the CTSP1 instance `burma14` mentioned in Section 2 with $T = 6$. The (fractional) TSP route for each day is depicted separately. The small number near each arc a is the value of x_a^k on the fractional solution. For customer $i = 13$ and day $k = 1$, one obtains $\underline{s}^{ik} = 971.65$ and $\bar{s}^{ik} = 1683.41$.

Now, each inequality (5) for the time consistency can be replaced by $\underline{s}^{ip} - \bar{s}^{iq} \leq T$. Using Duality Theory, this constraint is equivalent to the following family of linear inequalities:

$$\left(\beta_0' - \beta_i' + \sum_{a \in A^p} \alpha_a' x_a^p\right) - \left(\beta_0'' - \beta_i'' + \sum_{a \in A^q} \alpha_a'' x_a^q\right) \leq T \tag{23}$$

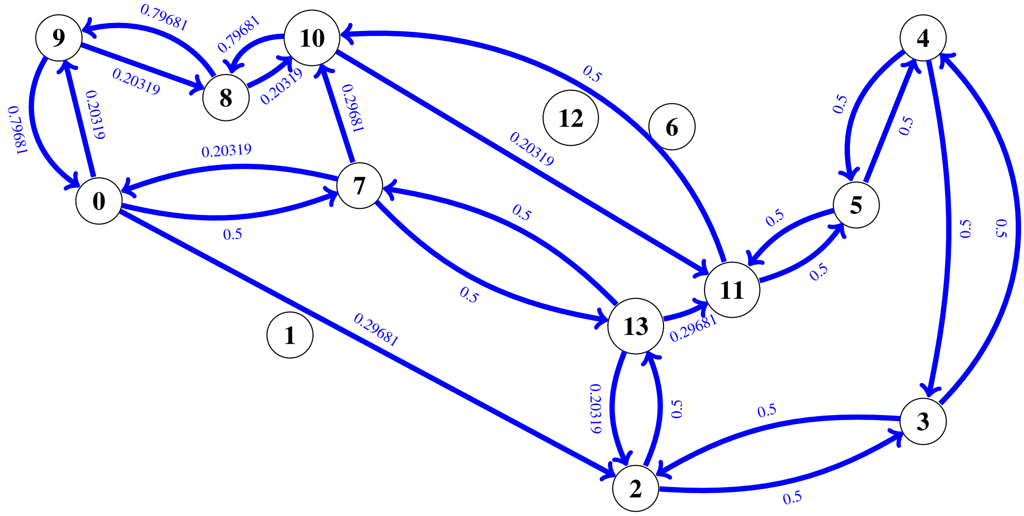
for each extreme point (α', β') of the polyhedron

$$\beta_u' - \beta_v' + \alpha_{(u,v)}' \leq t_{(u,v)} \quad (u,v) \in A^p \tag{24}$$

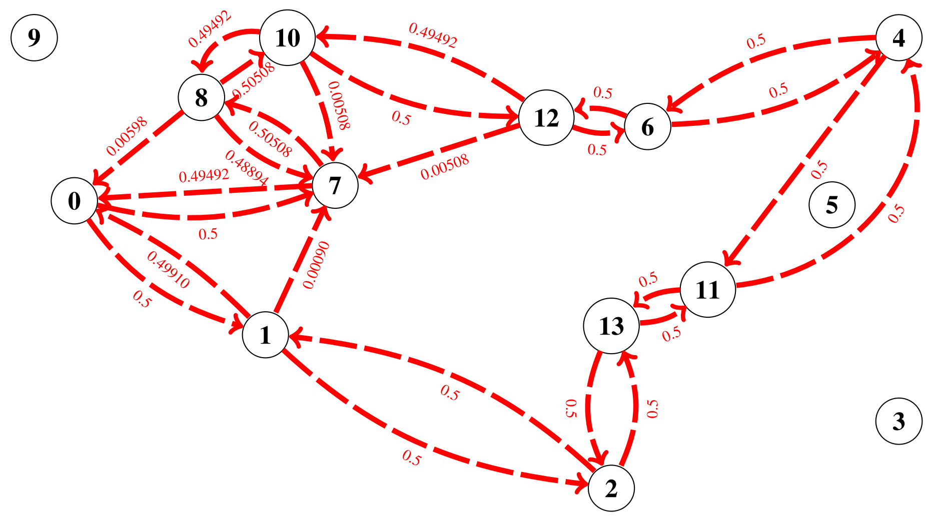
$$\alpha_{(u,v)}' \leq 0 \qquad (u,v) \in A^p, \tag{25}$$

and for each extreme point (α'', β'') of the polyhedron

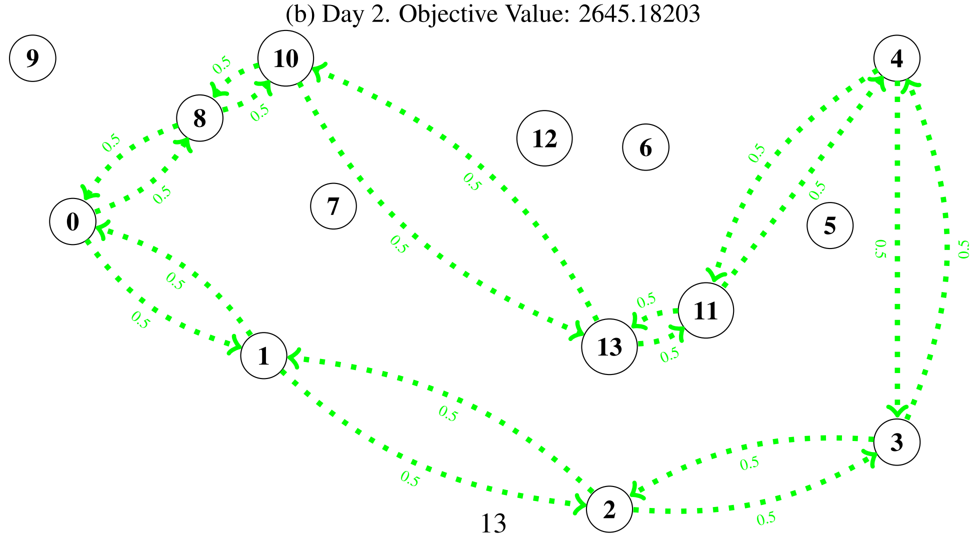
$$\beta_u'' - \beta_v'' + \alpha_{(u,v)}'' \geq t_{(u,v)} \quad (u,v) \in A^q \tag{26}$$



(a) Day 1. Objective Value: 3108.29681



(b) Day 2. Objective Value: 2645.18203



(c) Day 3. Objective Value: 2661.00000

Fig. 5. Fractional solution from Formulation 4 for CTSP1 with $T = 6$. Optimal Value: 8414.47884.

$$\alpha''_{(u,v)} \geq 0 \qquad (u,v) \in A^q. \tag{27}$$

While the number of inequalities (23) is quite large, a violated inequality by x^* can be easily detected (separated) as detailed in the next section. In summary, Formulation 4 is equivalent to the model determined by (1)–(4) and (23), which has the advantage of having a reduced number of variables (only x), with constraints that are easy to manage in a dynamic way (i.e. polynomially separable even from fractional solutions). It is a Benders' Decomposition on Formulation 4.

Formulation 4 can be strengthened by fixing $f_a^{ik} = x_a^k$ when $a \in \delta^+(0) \cup \delta^-(i)$, and $f_a^{ik} = 0$ when $a \in \delta^-(0) \cup \delta^+(i)$. This strengthening also applies to the Benders' reformulation with equations in (24) and (26) for $(u,v) \in \delta^-(0) \cup \delta^+(i)$ In addition, Formulations 4 and 5 are MILP models even if T is a mathematical variable, perhaps in the objective function, on some problem variants. This advantage is also shared by Formulations 2 and 3, but not by Formulation 1 since constraints (6) depend non-linearly on T.

3.3. Algorithm

Benders' Decomposition on Formulation 4 requires the use of a branch-and-cut approach to managing the large number of inequalities (3) and (23). This means solving a master problem defined by (1), (2) and (4), and adding dynamically (3) and (23) through a callback function on a solution x^* as follows:

Step 1: For each day $k \in K$ and each customer $i \in V^k \setminus \{0\}$, compute the maximum-capacity flow in G^k from 0 to i, using x_a^{*k} as the capacity of each $a \in A^k$. In other words, solve:

$$\overline{w}^{ik} := \max \quad f^{ik}(\delta_k^+(0)) - f^{ik}(\delta_k^-(0)) \quad \text{subject to}$$
$$(18) \text{ and } (19) \text{ with } x = x^*.$$

Intuitively, $\overline{w}^{ik}$ represents whether the vehicle arrives at customer i on day k in the current master solution x^*, and it coincides with the capacity of the minimum-capacity cut S^*, i.e. $\overline{w}^{ik} = x^{*k}(\delta_k^+(S^*))$. If $\overline{w}^{ik} < 1$ then S^* determines a subtour elimination constraint (3) to add to the master problem.
Exit the callback if an inequality was added to the master problem, and go to Step 2 only with a solution x^* where (at least) one unit of flow goes from 0 to each customer $i \in V^k$ for each day $k \in K$.

Step 2: For each day $k \in K$ and each customer $i \in V^k \setminus \{0\}$, compute the minimum and maximum cost flows sending one unit from 0 to i in G^k using x_a^{*k} as the capacity and t_a as the cost of each arc $a \in A^k$. In other words, solve the two linear programs with the objective functions (21) and (22), respectively, under the constraints (16)–(19) with $x = x^*$.
Save the optimal values $\underline{s}^{ik}$ and $\overline{s}^{ik}$, and the associated optimal dual solutions $(\underline{\alpha}^{ik}, \underline{\beta}^{ik})$ and $(\overline{\alpha}^{ik}, \overline{\beta}^{ik})$ for each k and i.
Check whether there exists (p,q,i) with $p \neq q$ such that $\underline{s}^{ip} - \overline{s}^{iq} > T$. If so, the dual solutions $(\underline{\alpha}^{ip}, \underline{\beta}^{ip})$ and $(\overline{\alpha}^{iq}, \overline{\beta}^{iq})$ determine an inequality (23) to add to the master problem. Otherwise, the current optimal master solution x^* satisfies all inequalities (3) and (23).

3.4. Practical issue

Eqs. (16)–(18) can be replaced by greater-or-equal inequalities without changing the optimal value of the programs. Then we may assume optimal dual solutions with $\underline{\beta}_j^{ik} \geq 0$ and $\overline{\beta}_j^{ik} \leq 0$.

Assuming $t_a \geq 0$, the optimal dual solutions satisfy

$$\underline{\alpha}_{(u,v)}^{ik} = \min\left\{0, t_{(u,v)} - \underline{\beta}_u^{ik} + \underline{\beta}_v^{ik}\right\};$$
$$\overline{\alpha}_{(u,v)}^{ik} = \max\left\{0, t_{(u,v)} - \overline{\beta}_u^{ik} + \overline{\beta}_v^{ik}\right\}.$$

Hence, the inequalities (23) are equivalent to

$$\sum_{(u,v)\in A^q} \max\left\{0, t_{(u,v)} - \beta''_u + \beta''_v\right\} \cdot x^q_{(u,v)} - \sum_{(u,v)\in A^p} \min\left\{0, t_{(u,v)} - \beta'_u + \beta'_v\right\} \cdot x^p_{(u,v)} \geq \left(\beta'_0 - \beta'_i\right) - \left(\beta''_0 - \beta''_i\right) - T$$

for all $\beta'_j \geq 0$ with $j \in V^p$, and for all $\beta''_j \leq 0$ with $j \in V^q$.

Step 1 is the well-known exact separation procedure for the subtour elimination constraints of the TSP, solvable in $O(|A^k||V^k|^2)$ using Orlin's algorithm for each $k \in K$. Step 2 is also polynomial-time solvable using a minimum-cost flow algorithm, with complexity $O(|A^k||V^k|^2 log^2 |V^k|)$ for each $k \in K$. See, e.g., Orlin (2013) for the complexity of network simplex algorithms. When x^* is an integer vector, then the complexity of Steps 1 and 2 is $O(|V^k|)$ for each k because there is only one path in x^{*k} from 0 to each customer in V^k. Indeed, it is unnecessary to solve the linear programs in Step 1 when x^* is an integer vector because it is enough to follow the route starting from 0 each day. Similarly, solving the linear programs in Step 2 is also unnecessary when x^* is an integer vector due to the following results.

Lemma 1. *Let x^* satisfy (2)–(4) and, for each $k \in K$, let C^k be the TSP route in x^* on day k. Let s^{0k} be the time duration of C^k and s^{jk} be the time duration of the path from 0 to j for all $j \in V^k \setminus \{0\}$. Define $\underline{\beta}_j = s^{0k} - s^{jk}$ and $\overline{\beta}_j = -s^{jk}$ for all $j \in V^k \setminus \{0\}$, $\underline{\beta}_0 = s^{0k}$ and $\overline{\beta}_0 = -s^{0k}$. Then, the points $\underline{\beta}$ and $\overline{\beta}$ are optimal dual solutions of the linear programs defined by (21) and (22), respectively, subject to (16)–(19).*

Proof. The points are dual solutions because $\underline{\beta}_j \geq 0$ and $\overline{\beta}_j \leq 0$ for each $j \in V^k$. They are optimal because their objective values are s^{ik}. Indeed, note that

$$t_{(u,v)} - \underline{\beta}_u + \underline{\beta}_v = \begin{cases} 0 & (u,v) \in C^k \ : \ v \neq 0 \\ s^{0k} & (u,v) \in C^k \ : \ v = 0 \end{cases},$$

$$t_{(u,v)} - \overline{\beta}_u + \overline{\beta}_v = \begin{cases} 0 & (u,v) \in C^k \ : \ u \neq 0 \\ s^{0k} & (u,v) \in C^k \ : \ u = 0 \end{cases}$$

Additionally, $\underline{\beta}_0 - \underline{\beta}_i = s^{ik}$ and $\overline{\beta}_0 - \overline{\beta}_i = s^{ik} - s^{0k}$. Hence,

$$\underline{\beta}_0 - \underline{\beta}_i + \sum_{(u,v)\in A^k} \min\left\{0, t_{(u,v)} - \underline{\beta}_u + \underline{\beta}_v\right\} \cdot x^{*k}_{(u,v)} = s^{ik} = \overline{\beta}_0 - \overline{\beta}_i + \sum_{(u,v)\in A^k} \max\left\{0, t_{(u,v)} - \overline{\beta}_u + \overline{\beta}_v\right\} \cdot x^{*k}_{(u,v)}.$$

□

The points in the lemma determine a linear inequality to separate an integer x^* when the time consistency at a customer is violated. It is worth noting that these points depend on k but not on i. From that observation, one should only consider the inequality associated with customer i with the largest time inconsistency $s^{ip} - s^{iq}$ in a given pair of days p, q. However, the linear programs defined by (21) and (22) subject to (16)–(19) have many alternative optimal dual solutions, as illustrated in the next result.

Lemma 2. *Let x^* satisfy (2)–(4) and, for each $k \in K$, let C^k be the TSP route in x^* on day k. Let s^{0k} be the time duration of C^k and s^{jk} be the* *time duration of the path from 0 to j for all $j \in V^k \setminus \{0\}$. Define*

$$\underline{\beta}_j = \begin{cases} s^{ik} - s^{jk} + s^{0k} & \text{if } i \text{ precedes } j \text{ in } C^k \\ s^{ik} - s^{jk} & \text{otherwise} \end{cases},$$

$$\overline{\beta}_j = \begin{cases} s^{ik} - s^{jk} - s^{0k} & \text{if } j \text{ precedes } i \text{ in } C^k \\ s^{ik} - s^{jk} & \text{otherwise} \end{cases}$$

for all $j \in V^k \setminus \{0\}$, $\underline{\beta}_0 = s^{ik}$ and $\overline{\beta}_0 = s^{ik} - s^{0k}$. Then, the points $\underline{\beta}$ and $\overline{\beta}$ are optimal dual solutions of the linear programs defined by (21) and (22), respectively, subject to (16)–(19).

Proof. The points are dual solutions because $\underline{\beta}_j \geq 0$ and $\overline{\beta}_j \leq 0$ for each $j \in V^k$. They are optimal because their objective values are s^{ik}. Indeed, note that

$$t_{(u,v)} - \underline{\beta}_u + \underline{\beta}_v = \begin{cases} 0 & (u,v) \in C^k \ : \ u \neq i \\ s^{0k} & (u,v) \in C^k \ : \ u = i \end{cases},$$

$$t_{(u,v)} - \overline{\beta}_u + \overline{\beta}_v = \begin{cases} 0 & (u,v) \in C^k \ : \ v \neq i \\ s^{0k} & (u,v) \in C^k \ : \ v = i \end{cases}$$

Also $\underline{\beta}_0 - \underline{\beta}_i = s^{ik}$ and $\overline{\beta}_0 - \overline{\beta}_i = s^{ik} - s^{0k}$. Hence

$$\underline{\beta}_0 - \underline{\beta}_i + \sum_{(u,v)\in A^k} \min\left\{0, t_{(u,v)} - \underline{\beta}_u + \underline{\beta}_v\right\} \cdot x^{*k}_{(u,v)}$$
$$= s^{ik} = \overline{\beta}_0 - \overline{\beta}_i + \sum_{(u,v)\in A^k} \max\left\{0, t_{(u,v)} - \overline{\beta}_u + \overline{\beta}_v\right\} \cdot x^{*k}_{(u,v)}.$$

□

Different dual solutions $\underline{\beta}$ and $\overline{\beta}$ may generate different cuts. Hence, several cuts could be added when the time consistency at customer i is violated. Selecting "good" optimal dual solutions to have "the best" Benders' cuts is an old and difficult research topic on MILP techniques (see e.g. Magnanti & Wong, 1981 and Costa, 2005).

4. Consistent TSP with waiting time

In this section, the vehicle is allowed to idle along the route. In other words, the vehicle can arrive at a customer and wait before starting to serve. Without loss of generality, we will not consider waiting times in a customer after its service, or along the road connecting two consecutive customers. As in the previous section, we assume that the vehicle leaves the depot at time zero every day; it is not a restrictive assumption because a potential delay in the departure could be identified with an idle time just before serving the first customer. The objective function in CTSP2 is assumed to minimize the total travel cost, as in CTSP1. CTSP2 also shares with CTSP1 the need for a TSP route for each day k. Thus, when using the mathematical variables x^k_a introduced for CTSP1, a CTSP2 formulation may also include (1)–(4). The novelty when dealing with CTSP2 is that checking the time consistency cannot be done with (5), but it requires finding continuous values y^i_k representing the idle time of the vehicle at customer i on day k if $i \in V^k$. These values must satisfy the linear system

$$\sum_{(u,v)\in P^{ip}} (t_{(u,v)} + y^p_v) - \sum_{(u,v)\in P^{iq}} (t_{(u,v)} + y^q_v) \leq T$$
$$p, q \in K, i \in V^p \cap V^q \setminus \{0\}, \tag{28}$$

$$y^k_i \geq 0 \quad k \in K, i \in V^k \setminus \{0\}, \tag{29}$$

where P^{ik} represents the path from 0 to i on day k in solution x. Since there may be many solutions when the system is feasible, it could be desired to change the objective function to minimize also the largest or average idle time, for example. In the same spirit, it could make sense to limit each idle time with a parameter N^k_i, and therefore add $y^k_i \leq N^k_i$ in (29). The objective function (1) could also include the idle times y^k_i if desired. For clarity in the exposition, we do not consider these features in this article.

To our knowledge, Subramanyam & Gounaris (2018) is the only article in the literature on CTSP2. It deals with an extension of the CTSP2 where the vehicle must return to the depot within a maximum time limitation, thus the problem coincides with CTSP2 when the limitation is a sufficiently large value. The authors proposed an exact algorithm that decomposes the problem into a sequence of TSPs with time windows within a branch-and-bound search procedure, and no mathematical formulation is presented. We now introduce four CTSP2 formulations motivated by the CTSP1 formulations presented in the previous section.

4.1. Formulation 1

Formulation 1 for CTSP1 suggests modelling CTSP2 with (1)–(4) and the following infeasible solution elimination constraint:

$$\sum_{k\in K} x^k(\Delta(C^k)) \leq \sum_{k\in K} |C^k| - 1 \tag{30}$$

for each TSP route C^k in G^k such that the linear system (28)–(29) for these TSP routes is infeasible. Inequalities (30) are known to be weak when using general-purpose MILP solvers and a strengthening procedure arises by considering an Irreducible Inconsistent Subsystem (IIS) of a given infeasible system (see, e.g., Chinneck, 1997). An IIS is an infeasible subsystem such that, if any of the constraints or bounds are removed, the new subsystem becomes feasible. Finding an IIS with the minimum number of inequalities is an $\mathcal{NP}$-hard problem in general, but many MILP solvers include heuristic approaches to provide users with minimal (not necessarily minimum) IIS. An IIS from system (28)–(29) is determined by a subset L of customers and days such that the restricted subsystem is still infeasible, and then inequality (30) is dominated by inequality

$$\sum_{(i,k)\in L} x^k(\Delta(P^{ik})) \leq \sum_{(i,k)\in L} |P^{ik}| - 1. \tag{31}$$

Note that inequality (31) is (6) when $L = \{(i,p), (i,q)\}$. However, for CTSP1, the time consistency can be checked independently on the customers, while for CTSP2, the idle time y^k_i must take one value on all checks; hence, L may contain more than two pairs. As stated in the previous section regarding constraints (6), inequalities (31) are also known to be weak and very difficult to separate on fractional solutions.

An alternative approach to not using the IIS provided by the MILP solver is to replace T with a variable in (28), solve the linear program that minimizes such a variable, and compare its optimal value with T. If the optimal value is not larger than T, the solution x is CTSP2 feasible; otherwise, the indices (i,p,q) violating (28) determine a set L associated with an inequality (31) violated by the current x.

4.2. Formulation 2

Formulation 2 for CTSP1 suggests modelling the CTSP2 with a variable z^k_i representing the time when service starts at customer i on day k, including the travel time from the depot to that customer and the idle times of all customers along that path (also the extreme i). Then, CTSP2 can be modelled with (1)–(4), (11) and

$$z^k_j \geq z^k_i + t_{(i,j)} \cdot x^k_{(i,j)} - M^k \cdot \left(1 - x^k_{(i,j)}\right)$$
$$k \in K,\ i \in V^k \setminus \{0\},\ j \in V^k \setminus \{0, i\}. \tag{32}$$

Note that we cannot strengthen the left-hand side of (32) as in (7) because z^k_j may be strictly larger than $z^k_i - t_{(j,i)}$ when $x^k_{(j,i)} = 1$ due to a potential idle time at customer i on day k. Note also that the subtour elimination constraints (3) can be removed to properly obtain a compact model.

Additionally, with the help of the z_i^k variables, one could easily adapt the formulation to minimize the duration of the routes rather than the total travel cost. To this end, one can introduce a new variable z^k representing the time duration of the route on day k, and then the model would minimize $\sum_{k\in K} z^k$ subject to (2)–(4), (11), (32) and $z_i^k + t_{(i,0)} \le z^k$ for all $k \in K$ and $i \in V^k \setminus \{0\}$. However, this would be a model for a variant of CTSP2.

4.3. Formulation 3

Another CTSP2 model is suggested by Formulation 3 for CTSP1 with Eq. (12) replaced by greater-or-equal inequalities. Again, the subtour elimination constraints (3) can be ignored to obtain a compact model.

4.4. Formulation 4

Another model for CTSP2 is inspired by Formulation 4 for CTSP1. Let y_i^k be the idle time of the vehicle at customer i on day k. If we replace $t_{(u,v)}$ by $t_{(u,v)} + y_v^k$ in (20), we obtain the non-linear inequalities

$$\left(\sum_{a\in A^p} t_a f_a^{ip} + \sum_{j\in V^p\setminus\{0\}} f^{ip}(\delta_p^-(j)) \cdot y_j^p\right) - \left(\sum_{a\in A^q} t_a f_a^{iq} + \sum_{j\in V^q\setminus\{0\}} f^{iq}(\delta_q^-(j)) \cdot y_j^q\right) \le T \qquad p, q \in K, i \in V^p \cap V^q \setminus \{0\}.$$

To linearize $f^{ik}(\delta_k^-(j)) \cdot y_j^k$ when $i \ne j$, we include another variable w_j^{ik} equal to the idle time y_j^k if j precedes i on day k, and 0 otherwise. We set $w_i^{ik} = y_i^k$ for convenience of notation. Then, a linear formulation for CTSP2 is given by the variables x satisfying (1)–(4), the variables f_a^{ik} satisfying (16)–(19) for each $k \in K$ and $i \in V^k$, the variables w_j^{ik} satisfying

$$y_j^k - N_j^k \cdot f^{jk}(\delta_k^-(i)) \le w_j^{ik} \le N_j^k \cdot f^{ik}(\delta_k^-(j))$$
$$0 \le w_j^{ik} \le y_j^k$$

for each $k \in K$ and $i, j \in V^k \setminus \{0\}$, and the time consistency constraints

$$\left(\sum_{a\in A^p} t_a f_a^{ip} + \sum_{j\in V^p\setminus\{0\}} w_j^{ip}\right) - \left(\sum_{a\in A^q} t_a f_a^{iq} + \sum_{j\in V^q\setminus\{0\}} w_j^{iq}\right) \le T \qquad p, q \in K,\ i \in V^p \cap V^q \setminus \{0\}.$$

We assume N_i^k to be an upper bound on the maximum idle time at customer i on day k. It is a large value, but not as large as M^k. Since $y_i^k = w_i^{ik}$, the variables y_i^k are unnecessary in the formulation. Similarly, since $x_{(i,j)}^k = f_{(i,j)}^{jk}$ when $j \ne 0$, the variables x_a^k are also unnecessary, and the formulation can be rewritten using only the binary variables f_a^{jk} and the continuous variables w_j^{ik}.

The LP relaxation of this formulation can be strengthened by observing that, for each $k \in K$, the values $f^{ik}(\delta_k^-(j))$ define a precedence relation between the customers in V^k with respect to the depot. Indeed, the following inequalities (see Sarin, Sherali, & Bhootra, 2005) are valid:

$$f^{jk}(\delta_k^-(i)) \ge x_{(i,j)}^k \qquad i, j \in V^k \setminus \{0\} : i \ne j$$
$$f^{ik}(\delta_k^-(j)) + f^{jk}(\delta_k^-(i)) = 1 \qquad i, j \in V^k \setminus \{0\} : i \ne j$$
$$x_{(i,j)}^k + f^{ik}(\delta_k^-(j)) + f^{jk}(\delta_k^-(l)) + f^{lk}(\delta_k^-(i)) \le 2 \qquad i, j, l \in V^k \setminus \{0\} : i \ne j, i \ne l, j \ne l.$$

The last family of inequalities, although of polynomial size, is quite large. Thus, a dynamic generation of these inequalities may be necessary for an efficient implementation.

While it is reasonable to apply Benders' Decomposition to deal with the large number of continuous variables f_a^{ik} for CTSP2, we did not succeed in identifying any particular combinatorial structure on the dual solutions or cuts as we did for CTSP1.

5. Computational results

To measure the practical implications of the formulations described in this paper, we coded computer programs to solve them using Gurobi 9.1.2 through its Python API on a personal computer with Intel(R) Core(TM) i7-10700 CPU @ 2.90 GHz, 24 GB RAM, and Windows 10 Pro. We report here the results of our experiments on three families of instances taken from Subramanyam & Gounaris (2016): `ulysses22` with 21 customers, `bays29` with 28 customers, and five instances with 100 customers (`kroA100`, `kroB100`, `kroC100`, `kroD100`, `kroE100`). The first and second families include 18 instances considering two numbers of days (i.e., $|K| = 3, 5$), three values for the service frequency of a customer in a given day (i.e. f=50%, 70%, 90%), and three values for the maximum time inconsistency (T). The third family includes 50 instances considering three days, a service frequency of 50% for each customer, and 10 different values of T. Each instance has been solved as a CTSP1 case, i.e. without waiting times.

We applied ten computer implementations on each CTSP1 instance of the first and second families. There is one implementation for each of the four formulations. These implementations are referred to as F1, F2, F3 and F4. F1 dynamically generates the tournament constraints (6), while F2, F3 and F4 solve the compact models. To measure the importance of the Subtour Elimination Constraints (3), these four implementations do not generate them, and we created three other implementations dynamically adding (3) to the implementations F1, F2 and F3; these three implementations are referred to as F1+SEC, F2+SEC and F3+SEC, respectively. Additionally, we produced three other implementations by applying Benders' Decomposition to Formulation 4 in different ways. Note that our Benders' cuts are constraints (3) to project out (16)–(19) and inequalities (23) to project out (20), and they are added to the relaxed model (1), (2) and (4). F4B generates (3) and (23) only from integer solutions, F4C also generates (3) from fractional solutions, and F4D generates both (3) and (23) from fractional solutions.

Subramanyam & Gounaris (2016) also implemented computer codes to solve CTSP1 instances using the three mathematical formulations in Section 3.1. From their reported results, for each of these formulations, our computer implementation is slower than their computer implementation. Our computer codes are available as supplementary material to this article. This material includes our implementations of the approaches for solving CTSP1 and CTSP2, and all instances used in this section for the computational analysis. We report details only to solve CTSP1 instances because the performances of the CTSP2 implementations are quite similar and add nothing interesting to the analysis.

Tables 2 and 3 show the computational results. Column *OptVal* is the objective value of the best solution found. Column *Time* is the time (in seconds) taken by our code implementation on the formulation, with a time limit of one hour (1h). Column *%-gap* is the difference between *OptVal* and linear programming bound at the root node, divided by *OptVal* and multiplied by 100. Bottom lines show average results computed over the instances solved to optimality, i.e., before the time limit.

Our experiments confirm that the three formulations already in the literature need the Subtour Elimination Constraints (3) to have stronger linear-programming relaxations and hopefully close the

Table 2
Results on `ulysses22`.

$\lvert K\rvert$	f	T	OptVal	F1		F1 + SEC		F2		F2 + SEC		F3		F3 + SEC		F4		F4B		F4C		F4D	
				Time	%-gap	Time	%-gap	Time	%-gap	Time	%-gap	Time	%-gap	Time	%-gap	Time	%-gap	Time	%-gap	Time	%-gap	Time	%-gap
3	50	1239	18,453	0.0	22.5	0.1	0.0	46.3	13.6	0.1	0.0	1598.2	18.8	0.2	0.0	0.1	0.0	0.0	22.5	0.0	0.0	0.0	0.0
3	50	929	18,453	0.0	22.5	0.0	0.0	48.5	13.6	0.2	0.0	868.9	18.8	0.2	0.0	0.1	0.0	0.0	22.5	0.0	0.0	0.0	0.0
3	50	619	18,453	0.0	22.5	0.0	0.0	29.2	13.6	0.1	0.0	1624.5	18.8	0.2	0.0	0.1	0.0	0.0	22.5	0.0	0.0	0.0	0.0
3	70	1330	19,213	0.0	20.0	0.0	0.1	1966.5	12.4	0.2	0.1	2891.1	17.0	0.4	0.1	0.5	0.1	0.0	20.0	0.0	0.1	0.2	0.1
3	70	997	19,213	0.0	20.0	0.0	0.1	1368.1	12.4	0.2	0.1	1h	17.0	0.4	0.1	0.5	0.1	0.0	20.0	0.0	0.1	0.2	0.1
3	70	665	19,213	0.0	20.0	0.0	0.1	389.8	12.4	0.2	0.1	1850.9	17.0	0.4	0.1	0.7	0.1	0.0	20.0	0.0	0.1	0.2	0.1
3	90	1393	20,718	1.0	25.6	1.9	0.3	1h	14.9	1.1	0.3	1h	23.3	1.4	0.3	5.9	0.3	0.2	25.6	0.5	0.3	15.9	0.3
3	90	1044	20,718	0.5	25.6	2.7	0.3	1h	14.9	1.1	0.3	1h	23.3	1.4	0.3	5.0	0.3	0.2	25.6	7.6	0.3	3.7	0.3
3	90	696	20,718	0.5	25.6	2.5	0.3	1h	14.9	0.7	0.3	1h	23.3	1.2	0.3	4.7	0.3	0.8	25.6	9.2	0.3	2.1	0.3
5	50	1239	30,433	0.0	19.9	0.1	0.1	1h	11.7	0.3	0.1	1h	16.2	0.6	0.1	0.7	0.1	0.0	19.9	0.1	0.1	0.5	0.1
5	50	929	30,433	0.0	19.9	0.1	0.1	1h	11.7	0.3	0.1	1h	16.2	0.6	0.1	0.7	0.1	0.1	19.9	0.1	0.1	0.5	0.1
5	50	619	30,433	0.2	19.9	0.1	0.1	2781.9	11.7	0.5	0.1	1h	16.2	1.1	0.1	0.6	0.1	0.1	19.9	0.1	0.1	0.3	0.1
5	70	1330	32,405	0.1	22.9	0.1	0.0	1h	13.9	0.4	0.0	1h	19.7	0.8	0.0	1.4	0.0	0.1	22.9	0.2	0.0	1.1	0.0
5	70	997	32,405	0.1	22.9	0.1	0.0	1h	13.9	0.3	0.0	1h	19.7	0.7	0.0	1.5	0.0	0.1	22.9	0.2	0.0	0.9	0.0
5	70	665	32,405	0.0	22.9	0.1	0.0	1h	13.9	0.5	0.0	1h	19.7	0.9	0.0	3.6	0.0	0.1	22.9	0.2	0.0	1.0	0.0
5	90	1393	33,904	1h	23.7	1912.7	0.2	1h	14.3	7.6	0.2	1h	21.4	34.5	0.2	24.2	0.2	22.5	23.7	15.1	0.2	135.5	0.2
5	90	1044	33,904	7.3	23.7	54.6	0.2	1h	14.3	4.1	0.2	1h	21.4	9.8	0.2	29.5	0.2	12.5	23.7	12.8	0.2	46.6	0.2
5	90	696	33,930	1h	23.7	1h	0.3	1h	14.4	9.8	0.3	1h	21.5	29.1	0.3	145.8	0.3	18.8	23.7	34.3	0.3	1314.8	0.3
$\lvert K\rvert = 3$ average				0.2	22.7	0.8	0.1	641.4	13.0	0.4	0.1	1766.7	18.1	0.6	0.1	2.0	0.1	0.1	22.7	1.9	0.1	2.5	0.1
$\lvert K\rvert = 5$ average				1.1	21.7	246.0	0.1	2781.9	11.7	2.6	0.1	-	-	8.7	0.1	23.1	0.1	6.0	22.2	7.0	0.1	166.8	0.1
$f = 50$ average				0.0	21.2	0.1	0.1	726.5	13.1	0.3	0.1	1363.9	18.8	0.5	0.1	0.4	0.1	0.0	21.2	0.1	0.1	0.2	0.1
$f = 70$ average				0.0	21.5	0.1	0.1	1241.5	12.4	0.3	0.1	2371.0	17.0	0.6	0.1	1.4	0.1	0.1	21.5	0.1	0.1	0.6	0.1
$f = 90$ average				2.3	25.1	394.9	0.3	-	-	4.1	0.3	-	-	12.9	0.3	35.9	0.3	9.2	24.7	13.3	0.3	253.1	0.3
Average				0.6	22.3	116.2	0.1	947.2	12.8	1.5	0.1	1766.7	18.1	4.7	0.1	12.5	0.1	3.1	22.4	4.5	0.1	84.6	0.1

Table 3
Results on `bays29`.

$\lvert K\rvert$	f	T	OptVal	F1		F1 + SEC		F2		F2 + SEC		F3		F3 + SEC		F4		F4B		F4C		F5D	
				Time	%-gap	Time	%-gap	Time	%-gap	Time	%-gap	Time	%-gap	Time	%-gap	Time	%-gap	Time	%-gap	Time	%-gap	Time	%-gap
3	50	294	4274	0.0	17.6	0.1	0.1	6.2	5.3	1.5	0.1	3153.8	16.7	0.9	0.1	1.2	0.1	0.1	17.6	0.1	0.1	0.5	0.1
3	50	220	4293	0.1	18.0	0.5	0.5	13.4	5.7	1.6	0.5	1h	17.0	0.6	0.5	1.7	0.5	0.3	18.0	0.5	0.5	1.5	0.5
3	50	147	4301	0.1	18.1	1.1	0.7	17.2	5.9	5.6	0.7	1h	17.2	3.4	0.7	2.3	0.7	0.1	18.1	0.4	0.7	3.0	0.7
3	70	361	5075	0.0	15.2	0.1	0.0	15.4	2.0	0.3	0.0	1h	14.4	0.5	0.0	1.4	0.0	0.1	15.2	0.1	0.0	0.8	0.0
3	70	271	5075	0.0	15.2	0.1	0.0	85.5	2.0	0.3	0.0	1h	14.4	0.5	0.0	1.4	0.0	0.1	15.2	0.1	0.0	0.8	0.0
3	70	180	5122	7.3	16.0	16.6	0.9	84.3	2.9	8.1	0.9	1h	15.1	7.9	0.9	20.1	0.9	15.1	16.0	15.7	0.9	189.1	0.9
3	90	386	5690	71.0	11.3	37.9	0.9	1h	6.1	65.8	0.9	1h	10.8	125.8	0.9	162.7	0.9	17.1	11.3	18.1	0.9	147.2	0.9
3	90	289	5690	23.1	11.3	465.3	0.9	1h	6.1	1008.8	0.9	1h	10.8	30.3	0.9	57.6	0.9	737.7	11.3	59.9	0.9	268.2	0.9
3	90	193	5690	49.7	11.3	39.3	0.9	1h	6.1	732.5	0.9	1h	10.8	688.5	0.9	105.2	0.9	188.6	11.3	1424.2	0.9	180.9	0.9
5	50	294	7130	0.1	15.0	0.2	0.1	322.0	4.7	1.7	0.1	1h	14.2	2.0	0.1	2.6	0.1	0.1	15.0	0.2	0.1	1.1	0.1
5	50	220	7149	0.2	15.3	1.3	0.3	1h	5.0	3.5	0.3	1h	14.4	1.5	0.3	3.5	0.3	0.1	15.3	0.3	0.3	2.6	0.3
5	50	147	7183	3.7	15.7	3.0	0.8	2656.7	5.4	617.5	0.8	1h	14.8	289.0	0.8	121.1	0.8	48.9	15.7	344.5	0.8	278.9	0.8
5	70	361	8542	1.9	14.4	10.6	0.2	1h	3.0	4.6	0.2	1h	13.8	18.0	0.2	13.8	0.2	2.0	14.4	7.9	0.2	19.0	0.2
5	70	271	8542	1.5	14.4	20.9	0.2	1h	3.0	5.8	0.2	1h	13.8	18.7	0.2	13.6	0.2	3.9	14.4	5.9	0.2	665.8	0.2
5	70	180	8596	1h	15.0	1h	0.9	1h	3.7	1h	0.9	1h	14.3	1h	0.9	148.0	0.9	1h	15.0	1h	0.9	1h	0.9
5	90	386	9534	1h	12.0	1h	0.8	1h	5.6	1h	0.8	1h	11.4	1h	0.8	1336.1	0.8	1h	12.0	1h	0.8	1h	0.8
5	90	289	9534	1h	12.0	1h	0.8	1h	5.6	1h	0.8	1h	11.4	1h	0.8	2833.4	0.8	1h	12.0	1521.2	0.8	1h	0.8
5	90	193	9534	1h	12.0	1h	0.8	1h	5.6	1h	0.8	1h	11.4	1h	0.8	1731.7	0.8	1h	12.0	1713.5	0.8	2705.3	0.8
$\lvert K\rvert = 3$ average				16.8	14.9	62.3	0.5	37.0	4.0	202.7	0.5	3153.8	16.7	95.4	0.5	39.3	0.5	106.6	14.9	168.8	0.5	88.0	0.5
$\lvert K\rvert = 5$ average				1.5	15.0	7.2	0.3	1489.4	5.1	126.6	0.3	-	-	65.8	0.3	689.3	0.5	11.0	15.0	513.4	0.5	612.1	0.4
$f = 50$ average				0.7	16.6	1.0	0.4	603.1	5.4	105.2	0.4	3153.8	16.7	49.6	0.4	22.1	0.4	8.3	16.6	57.7	0.4	47.9	0.4
$f = 70$ average				2.1	15.0	9.7	0.3	61.7	2.3	3.8	0.3	-	-	9.1	0.3	33.1	0.4	4.2	15.0	5.9	0.3	175.1	0.3
$f = 90$ average				47.9	11.3	180.8	0.9	-	-	602.4	0.9	-	-	281.5	0.9	1037.8	0.9	314.5	11.3	947.4	0.9	825.4	0.9
Average				11.3	14.9	42.6	0.5	400.1	4.2	175.5	0.5	3153.8	16.7	84.8	0.5	364.3	0.5	72.4	14.9	319.5	0.5	297.6	0.5

integrability gap in reasonable times. There are some weird cases where the implementation F1 is faster than F1+SEC; we believe they are due to internal procedures in the MILP solver (Gurobi); indeed, F1 loses efficiency when deactivating the preprocessing and internal cuts of the MILP solver. This is not the case for the new formulation F4 proposed in this paper, where the Subtour Elimination Constraints (3) are already implicit, and the internal cuts of the MILP solver seem to have no effect. When comparing the ten implementations, although there is not a clear winner, the implementations based on the new formulation (F4, F4B, F4C and F4D) have better performances than the implementations based on the formulations in the literature, at least on most of the instances in our experiments. We have also performed experiments on larger instances, increasing the computational difficulties on all the implementations. Note that the linear programs solved at the root node in F1 and in F4B coincide with the continuous relaxation of model (1), (2) and (4). For that reason, the two implementations show the same values in Column *%-gap*. The values in column *Time* are different for F1 and F4B because the implementations differ in the inequalities generated to separate invalid integer solutions and, clearly, F4B is faster than F1.

Although the number of customers $\lvert V^k\rvert$ has a negative impact when solving a CTSP1 instance, the number of days $\lvert K\rvert$ has a more negative impact. Indeed, many implementations went to the time limit when $\lvert K\rvert = 5$, especially when f is large (i.e., when $\lvert V^k\rvert$ approaches $\lvert V\rvert$). The combination $\lvert K\rvert = 5$ and $f = 90$ tends to create

Table 4
Results on instances with 100 customers and three days.

kroA100				kroB100				kroC100				kroD100				kroE100			
T	OptVal	Time	%-gap	T	OptVal	Time	%-gap	T	OptVal	Time	%-gap	T	OptVal	Time	%-gap	T	OptVal	Time	%-gap
4505	50,418	8.0	24.6	15,009	52,986	2.8	23.5	5847	50,189	248.6	25.6	14,513	50,183	24.0	20.8	14,481	50,048	4.4	21.8
4483	50,439	11.2	24.6	14,900	52,986	7.9	23.5	5843	50,190	91.6	25.6	14,362	50,219	42.9	20.8	14,452	50,079	13.3	21.8
4468	50,453	15.1	24.6	14,549	53,054	14.5	23.5	5838	50,195	318.0	25.6	12,928	50,225	9.8	20.8	14,421	50,084	11.9	21.8
4456	50,466	20.4	24.7	8367	53,085	46.5	23.6	5833	50,204	275.5	25.7	10,128	50,237	18.7	20.9	14,401	50,115	33.4	21.9
4441	50,479	22.2	24.7	8336	53,097	73.6	23.6	5828	50,208	143.3	25.7	10,116	50,323	377.0	21.0	14,370	50,202	85.1	22.0
4427	50,493	44.4	24.7	8324	53,108	18.9	23.6	5823	50,213	253.4	25.7	9800	50,323	49.5	21.0	14,364	50,204	72.9	22.0
3940	50,536	42.8	24.8	8012	53,139	589.3	23.7	5818	50,218	134.4	25.7	9739	50,335	203.1	21.0	8469	50,204	774.9	22.0
3924	50,550	75.5	24.8	7981	53,151	109.9	23.7	5814	50,228	118.3	25.7	9727	50,363	392.1	21.1	6122	50,236	94.8	22.1
3916	50,558	91.5	24.8	7964	53,182	533.9	23.7	5799	50,235	20.8	25.7	6563	50,363	39.8	21.1	6090	50,305	334.1	22.2
1609	50,566	787.4	24.8	7933	53,187	99.8	23.7	5797	50,237	21.9	25.7	6506	50,399	471.3	21.1	4870	50,305	285.22	22.2
Average		111.9	24.7	Average		149.7	23.6	Average		162.6	25.7	Average		162.8	21.0	Average		171.0	22.0

quite difficult CTSP1 instances for our implementations even when $|V|$ is small.

Tables 2 and 3 show a minor impact of T on the performance for each code, but it is due to the values selected in Subramanyam & Gounaris (2016) (10%, 15% and 20% of the maximum of the optimal TSP values on V^k). As reported in Table 1, smaller values of T can generate more desirable routes in practice. Our experiments revealed that very small values of T also contribute negatively to the complexity of solving the problem. Solving CTSP2 instances is slightly easier than solving the same CTSP1 instances, but still dealing with a larger number of customers is a challenging research task. Implementation F4, based on the new formulation introduced in this paper, is the only one that solves all instances in Table 3 before the time limit. When solving larger instances, the number of variables harms the performance, and then Benders' Decomposition becomes necessary.

Table 4 shows the performance of Implementation F4B on instances with 100 customers, $f = 50\%$ and three days. The values in %-gap are close to 25 because constraints (3) and (23) are generated only from integer solutions. Hence, the root-node solution can often be eliminated without branching. In particular, the Benders' constraints are easily identified with Lemmas 1 and 2, and with no need to solve the linear problems that other Benders' Decomposition approaches need. This explains the low computational times to prove optimality in the table. However, these instances are taken from the literature and have relatively large T values. Solving to optimality these instances with smaller values of T is still quite challenging.

6. Conclusions

We have presented and analyzed four mathematical formulations for the time-consistent TSP without waiting times. Three of the formulations already existed in the literature, and the other one is new. The new one is a multi-commodity flow formulation based on modelling the path from the depot to each customer as a flow, and it has the advantage that the subproblem in a Benders' Decomposition approach consists of two well-known combinatorial problems: the max flow problem and the min-cost flow problem. This allows an easy generation of the Benders' cuts through network flow procedures to separate invalid integer and fractional solutions of the master problem. Moreover, the two lemmas in this paper describe Benders' cuts to eliminate invalid integer solutions with no need to solve any subproblem. Computational results show that the new approach can solve instances that other approaches (based on other formulations) cannot solve within our time limit. The four formulations are also adapted to model the time-consistent TSP with waiting times. This is another major contribution since there was no formulation for such a variant in the literature.

Although this paper contributes to a better understanding of the time-consistent TSP, a very difficult vehicle routing problem especially when T is a tight threshold. The Benders' cuts are fundamental to solve instances with approximately 100 customers. Strengthening the coefficients of the Benders' cuts should help solve larger instances. Another interesting question would be to adapt these cuts to remain valid when T is a decision variable rather than a given input parameter. A challenging problem variant would be to find the smallest T value for which an instance is feasible. When waiting times are allowed, there is also the research question of finding Benders' cuts to separate integer solutions without the need to solve a linear program. In addition, this paper provides new directions to afford other problem variants (limiting the route duration for each day, limiting the waiting times, managing a fleet of vehicles, etc.)

Acknowledgment

This research was supported by the Spanish research project PID2019-104928RB-I00.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ejor.2023.08.021.

References

Alvarez, A., Cordeau, J. F., & Jans, R. (2022). The consistent production routing problem. *Networks, 80*, 356–381. https://doi.org/10.1002/net.22112.

Bektas, T., & Gouveia, L. (2014). Requiem for the Miller-Tucker-Zemlin subtour elimination constraints. *European Journal of Operational Research, 236*, 820–832. https://doi.org/10.1016/j.ejor.2013.07.038. Vehicle Routing and Distribution Logistics

Braekers, K., & Kovacs, A. A. (2016). A multi-period dial-a-ride problem with driver consistency. *Transportation Research Part B: Methodological, 94*, 355–377. https://doi.org/10.1016/j.trb.2016.09.010.

Campelo, P., Neves-Moreira, F., Amorim, P., & Almada-Lobo, B. (2019). Consistent vehicle routing problem with service level agreements: A case study in the pharmaceutical distribution sector. *European Journal of Operational Research, 273*, 131–145. https://doi.org/10.1016/j.ejor.2018.07.030.

Chinneck, J. W. (1997). Finding a useful subset of constraints for analysis in an infeasible linear program. *INFORMS Journal on Computing, 9*, 164–174. https://doi.org/10.1287/ijoc.9.2.164.

Coelho, L. C., Cordeau, J. F., & Laporte, G. (2012). Consistency in multi-vehicle inventory-routing. *Transportation Research Part C: Emerging Technologies, 24*, 270–287. https://doi.org/10.1016/j.trc.2012.03.007.

Costa, A. M. (2005). A survey on benders decomposition applied to fixed-charge network design problems. *Computers & Operations Research, 32*, 1429–1450. https://doi.org/10.1016/j.cor.2003.11.012.

Dalmeijer, K., & Spliet, R. (2018). A branch-and-cut algorithm for the time window assignment vehicle routing problem. *Computers & Operations Research, 89*, 140–152. https://doi.org/10.1016/j.cor.2017.08.015.

Feillet, D., Garaix, T., Lehuédé, F., Péton, O., & Quadri, D. (2014). A new consistent vehicle routing problem for the transportation of people with disabilities. *Networks, 63*, 211–224. https://doi.org/10.1002/net.21538.

Goeke, D., Roberti, R., & Schneider, M. (2019). Exact and heuristic solution of the consistent vehicle-routing problem. *Transportation Science, 53*, 1023–1042. https://doi.org/10.1287/trsc.2018.0864.

Groër, C., Golden, B., & Wasil, E. (2009). The consistent vehicle routing problem. *Manufacturing & Service Operations Management, 11*, 630–643. https://doi.org/10.1287/msom.1080.0243.

Gulczynski, D., Golden, B., & Wasil, E. (2010). The split delivery vehicle routing problem with minimum delivery amounts. *Transportation Research Part E: Logistics and Transportation Review, 46*, 612–626. https://doi.org/10.1016/j.tre.2009.12.007.

Kovacs, A., Golden, B., Hartl, R., & Parragh, S. (2014). The generalized consistent vehicle routing problem. *Transportation Science, 24*. https://doi.org/10.1287/trsc.2014.0529.

Kovacs, A., Parragh, S., & Hartl, R. (2015). The multi-objective generalized consistent vehicle routing problem. *European Journal of Operational Research, 247*, 441–458. https://doi.org/10.1016/j.ejor.2015.06.030.

Kovacs, A. A., Golden, B. L., Hartl, R. F., & Parragh, S. N. (2014). Vehicle routing problems in which consistency considerations are important: A survey. *Networks, 64*, 192–213. https://doi.org/10.1002/net.21565.

Kovacs, A. A., Parragh, S. N., & Hartl, R. F. (2014). A template-based adaptive large neighborhood search for the consistent vehicle routing problem. *Networks, 63*, 60–81. https://doi.org/10.1002/net.21522.

Lian, K., Milburn, A. B., & Rardin, R. L. (2016). An improved multi-directional local search algorithm for the multi-objective consistent vehicle routing problem. *IIE Transactions, 48*, 975–992. https://doi.org/10.1080/0740817X.2016.1167288.

Luo, Z., Qin, H., Che, C., & Lim, A. (2015). On service consistency in multi-period vehicle routing. *European Journal of Operational Research, 243*, 731–744. https://doi.org/10.1016/j.ejor.2014.12.019.

Magnanti, T. L., & Wong, R. T. (1981). Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research, 29*, 464–484. http://www.jstor.org/stable/170108

Mancini, S., Gansterer, M., & Hartl, R. (2021). The collaborative consistent vehicle routing problem with workload balance. *European Journal of Operational Research, 293*. https://doi.org/10.1016/j.ejor.2020.12.064.

Nolz, P. C., Absi, N., Feillet, D., & Seragiotto, C. (2022). The consistent electric-vehicle routing problem with backhauls and charging management. *European Journal of Operational Research, 302*, 700–716. https://doi.org/10.1016/j.ejor.2022.01.024.

Orlin, J. B. (2013). Max flows in *O(nm)* time, or better. In *Proceedings of the forty-fifth annual ACM symposium on theory of computing* (pp. 765–774). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2488608.2488705.

Rahmaniani, R., Crainic, T. G., Gendreau, M., & Rei, W. (2017). The benders decomposition algorithm: A literature review. *European Journal of Operational Research, 259*, 801–817. https://doi.org/10.1016/j.ejor.2016.12.005.

Rodríguez-Martín, I., Salazar-González, J. J., & Yaman, H. (2019). The periodic vehicle routing problem with driver consistency. *European Journal of Operational Research, 273*, 575–584. https://doi.org/10.1016/j.ejor.2018.08.032.

Sarin, S. C., Sherali, H. D., & Bhootra, A. (2005). New tighter polynomial length formulations for the asymmetric traveling salesman problem with and without precedence constraints. *Operations Research Letters, 33*, 62–70. https://doi.org/10.1016/j.orl.2004.03.007.

Stavropoulou, F. (2022). The consistent vehicle routing problem with heterogeneous fleet. *Computers & Operations Research, 140*, 105644. https://doi.org/10.1016/j.cor.2021.105644.

Stavropoulou, F., Repoussis, P., & Tarantilis, C. (2019). The vehicle routing problem with profits and consistency constraints. *European Journal of Operational Research, 274*, 340–356. https://doi.org/10.1016/j.ejor.2018.09.046.

Subramanyam, A., & Gounaris, C. E. (2016). A branch-and-cut framework for the consistent traveling salesman problem. *European Journal of Operational Research, 248*, 384–395. https://doi.org/10.1016/j.ejor.2015.07.030.

Subramanyam, A., & Gounaris, C. E. (2018). A decomposition algorithm for the consistent traveling salesman problem with vehicle idling. *Transportation Science, 52*, 386–401. https://doi.org/10.1287/trsc.2017.0741.

Tarantilis, C., Stavropoulou, F., & Repoussis, P. (2012). A template-based tabu search algorithm for the consistent vehicle routing problem. *Expert Systems with Applications, 39*, 4233–4239. https://doi.org/10.1016/j.eswa.2011.09.111.

(2002). *The vehicle routing problem*. In P. Toth, & D. Vigo (Eds.). Society for Industrial and Applied Mathematics. 10.1137/1.9780898718515

Vansteenwegen, P., Souffriau, W., & Oudheusden, D. V. (2011). The orienteering problem: A survey. *European Journal of Operational Research, 209*, 1–10. https://doi.org/10.1016/j.ejor.2010.03.045.

Yao, Y., Van Woensel, T., Veelenturf, L. P., & Mo, P. (2021). The consistent vehicle routing problem considering path consistency in a road network. *Transportation Research Part B: Methodological, 153*, 21–44. https://doi.org/10.1016/j.trb.2021.09.005.

Zhen, L., Lv, W., Wang, K., Ma, C., & Xu, Z. (2020). Consistent vehicle routing problem with simultaneous distribution and collection. *Journal of the Operational Research Society, 71*, 813–830. https://doi.org/10.1080/01605682.2019.1590134.