The Biobjective Consistent Traveling Salesman Problem

Daniel Díaz-Ríos and Juan-José Salazar-González(✉)

IMAULL, University of La Laguna, 38200 La Laguna, Tenerife, Spain
ddiazrio@ull.edu.es, jjsalaza@ull.es

Abstract. This article deals with the problem of designing a route for each day of a time period to minimize the total travel cost and the discrepancy in the service time to customers visited on different days. In this paper we allow waiting times for the vehicle at customer locations. The literature already includes exact and heuristic approaches for the variant where the first objective is minimized and the second objective is constrained by a given threshold. The variant do not allow waiting times at customer locations, and it is known as Consistent Travelling Salesman Problem. We are not aware of any previous algorithm in the literature to tackle the biobjective problem, and this article describes three compact formulations for it. Each formulation is suitable for approaching the problem through the well-known weighted-sum method for multiobjective optimization, where some Pareto optimal solutions are sequentially determined by systematically changing the weights among the objective functions. We perform a computational study applying the formulations to tackle instances adapted from the TSPLIB library.

Keywords: Travelling Salesman · Time consistency · Branch and cut

1 Introduction

The Traveling Salesman Problem (TSP) is an NP-Hard problem that has been extensively studied in the literature. Given a depot and a set of customers, it consists of finding a minimum-cost circuit so that a vehicle visits each customer exactly once. The Consistent Travelling Salesman Problem (CTSP) is a variant of the TSP introduced by [2] where a time period is given (say, a week), each customer requires service on some (known in advance) days, and one TSP must be solved for each day. The novel requirement in the CTSP is that customers desire to be served consistently in time when visited in different days, meaning that the service times should be similar for each customer. For example, giving service to a customer in the early morning on Monday and in the late evening on Thursday is undesirable. In the literature this requirement has been modelled as a hard constraint (see e.g. [7,8]). Indeed, an input threshold T is given in advance to limit the maximum allowed time inconsistency for each customer visited in several days, and then the routes should be determined to ensure it. While

J. de Armas et al. (Eds.): ICCL 2022, LNCS 13557, pp. 120–131, 2022.
https://doi.org/10.1007/978-3-031-16579-5_9

there are articles where waiting times of the vehicle at the customer locations are forbidden before the services start, better solutions are found when waiting times are allowed. For that reason, in this paper, we assume that the vehicle is allowed to wait along the route. Under this assumption the mathematical problem is more complex since a solution is not only a route for each day, but also the waiting time at each customer location served each day.

Given a route for each day and the waiting times at each customer location, we consider two characteristics. One characteristic is the total travel cost of the routes, not including waiting times. The other feature is the maximum of the time difference between the visits in different days to a customer, over all the customers. The CTSP looks for a solution, minimizing the first feature, while it limits the second feature to T with a hard constraint. However the real-world problem has a multicriteria nature and a CTSP solution also admits other characteristics, like the travel cost of the route in each day (perhaps even including the waiting times), or the maximum waiting time between when a vehicle arrives to a customer and when the service starts at the customer. To reduce the complexity of the exposition in this article, we restrict our analysis to two characteristic, thus leading to a bicriteria optimization problem. The models and implementations can be easily adapted to also include the other characteristics.

Since the two characteristics (the total travel cost and the time inconsistency at customers) evolve in different directions, there does not exist a feasible solution that minimizes the two objective functions simultaneously. In this context, solving the bicriteria problem is finding a Pareto optimal solution, which is a set of routes that cannot be improved in any of the objectives without degrading the other objective. Since there may be a huge number of Pareto optimal solutions, we only aim at generating supporting non-dominated Pareto-optimal solutions, i.e., solutions with objective values on the convex hull of the Pareto frontier. As typically followed when approaching a bicriteria optimization problem, our article models and solves a single-objective problem where the two selected characteristics are linearly combined with a generic weight α, the standard weighted-sum method in multicriteria optimization to generate some Pareto-optimal solutions (see e.g. [5]). We propose three mathematical programming formulations for the single-objective problem and analyse computational experiments using a modern solver to find optimal solutions. The formulations can also be adapted to work on more sophisticated approaches to generate all the Pareto-optimal solutions (see e.g. [3,4]).

2 Problem Definition

Let $G = (V, A)$ be a complete directed graph where $V = \{0, 1, ..., n\}$ is the set of nodes, with 0 representing the depot and $1, ..., n$ representing the customers. Each arc $(i, j) \in A$ is associated with two variables: c_{ij} and t_{ij} representing the travel cost and travel time, respectively, to go from node i to node j. We use $K = \{1, ..., m\}$ to represent the time period (for example, the days of a

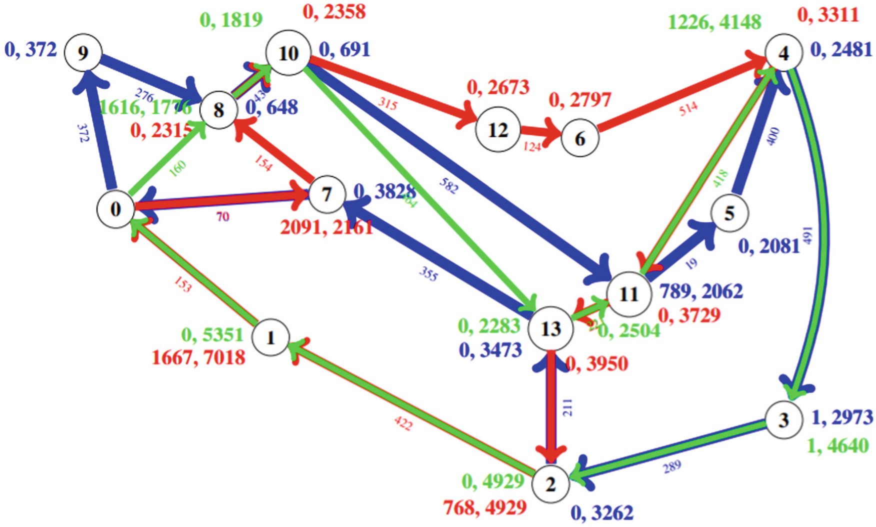

Fig. 1. Optimal solution with $\alpha = 0$. Travel cost: 8414. Time inconsistency: 1667

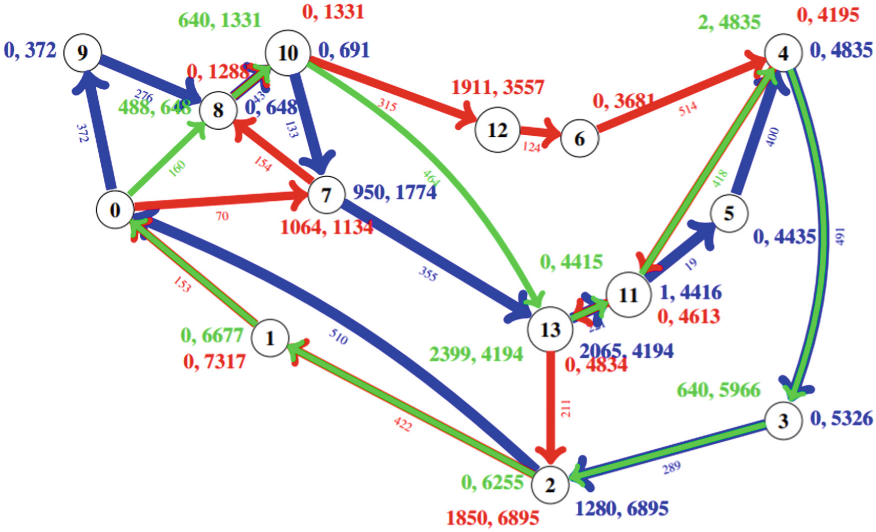

Fig. 2. Optimal solution with $\alpha = 0.1$. Travel cost: 8415. Time inconsistency: 640

week). Each day $k \in K$ is associated with a subset of nodes V_k, representing the customers to be visited in that day, with $0 \in V_k$. The day k is also associated with a set of arcs $A_k = \{(i,j) : i,j \in V_k, i \neq j\} \subseteq A$. Let $G^k = (V^k, A^k)$ be the graph on which a Hamiltonian circuit must be computed to define the route on day k. We assume that waiting times are allowed between the vehicle arrives at

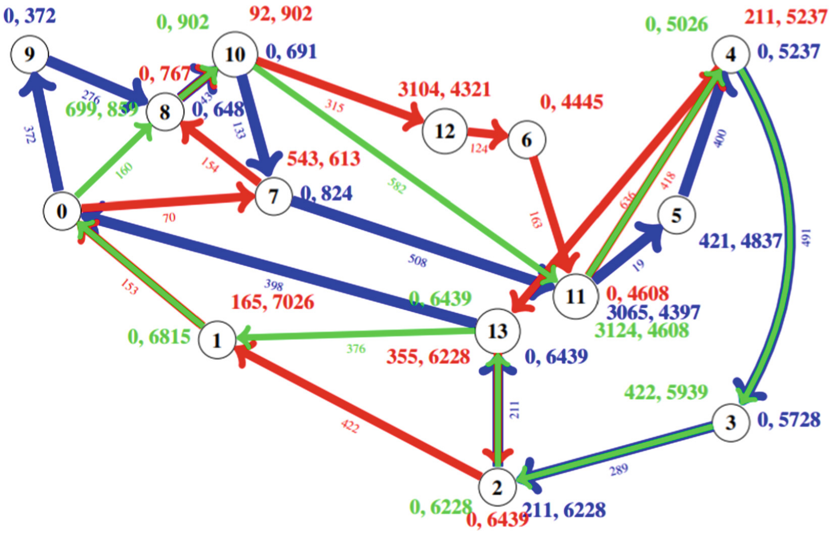

Fig. 3. Optimal solution with $\alpha = 0.27$. Travel Cost: 8572. Time inconsistency: 211

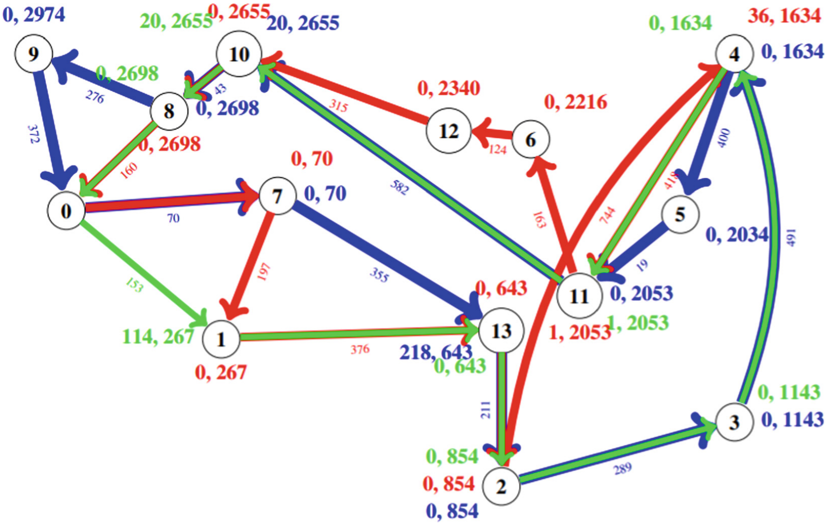

Fig. 4. Optimal solution with $\alpha = 0.3$. Travel Cost: 8652. Time inconsistency: 0

a customer location and before the service starts. To evaluate the two selected characteristics of a problem solution into a single objective function, let α be a given weight in the interval $[0, 1]$ so the cost of the solution is $1 - \alpha$ times the travel cost plus α times the maximum time discrepancy over all the customers.

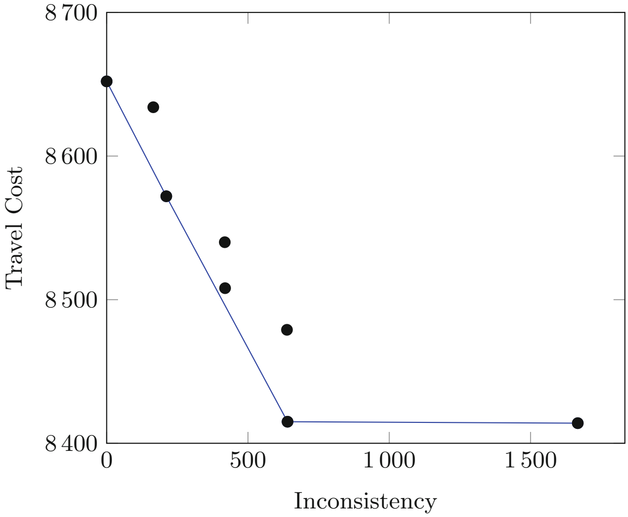

Fig. 5. Pareto frontier

Figures 1, 2, 3 and 4 show the optimal solutions for different values of α using an instance introduced in [7]. It is the `burma14` instance with 13 customers, with travel costs computed from geographic coordinates. Three days are considered (i.e., $|K| = 3$) and every customer requires a visit on each day with probability 0.7. In each figure, we use a different colour for each day, and show a pair of numbers near each customer. The left number is the waiting time of the vehicle at the customer before service starts, which is the right number. The tiny number on an arc shows the travel cost of that arc. Figure 1 corresponds to the solution when $\alpha = 0$, hence the time inconsistencies are not penalized and the result is equivalent to solving three independent TSPs. Figure 2 shows the optimal solution of the problem when $\alpha = 0.1$; the maximum time difference is reduced from 1667 to 640 increasing the travel cost of the routes by only one unit. It is worth mentioning that a solution with the same travel cost could be obtained with the approach in [8] and with any threshold T in the interval $[640, 1666]$. As expected, the more importance we place on time consistency in the objective function, the smaller the maximum time difference and the higher the total path cost. Figure 3 shows the solution when $\alpha = 0.27$, with travel cost equals to 8572 and the maximum time difference reduced to 211. Figure 4 shows the solution for a decision maker accepting a cost of 8652 to reduce the maximum time difference to zero, i.e. there is no inconsistency for any customer. Although α could still be increased to the value 1, no better solution is generated. Figure 5 shows the eight Pareto-optimal points for this bicriteria problem `burma14`. The four points along the blue line (frontier) correspond to the optimal solutions depicted in the previous figures, which are non-dominated Pareto solutions.

3 Mathematical Formulations

In this section we describe three formulations. The three formulations are based on a binary variable x_a^k which takes the value 1 if the vehicle in day $k \in K$ traverses the arc $a \in A^k$ and 0 otherwise. As standard in the literature, with $S \subseteq V^k$ we define $\delta_k^+(S) := \{(i,j) \in A^k : i \in S, j \notin S\}$ and $\delta_k^-(S) := \{(i,j) \in A^k : j \in S, i \notin S\}$. We denote the successors of a node i in day k as $\delta_k^+(i)$ instead of $\delta_k^+(\{i\})$ and the predecessors of node i in day k as $\delta_k^-(i)$ instead of $\delta_k^-(\{i\})$, in addition to $x^k(B)$ instead of $\sum_{a \in B} x_a^k$ where $B \subseteq A^k$. We also define a continuous variable T to represent the maximum time inconsistency of any customer. Therefore the objective function is:

$$\text{minimize} \qquad (1-\alpha) \cdot \sum_{k \in K} \sum_{a \in A^k} c_a x_a^k + \alpha \cdot T \tag{1}$$

Constraints shared by the three formulations are:

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

$$x_a^k \in \{0,1\} \qquad a \in A^k. \tag{3}$$

Formulation 1 uses an additional continuous variable z_i^k to represent the visit time on day k to customer i, and a big value M^k to upper bound the duration of the route in day k. Consequently the problem is formulated as (1)–(3) and

$$z_j^k \geq z_i^k + t_{(i,j)} \cdot x_{(i,j)}^k - M^k \cdot (1 - x_{(i,j)}^k) \qquad k \in K,\ i,j \in V^k \setminus \{0\},\ i \neq j \tag{4}$$

$$z_j^k \geq t_{(0,j)} \cdot x_{(0,j)}^k \qquad k \in K,\ j \in V^k \setminus \{0\} \tag{5}$$

$$z_i^p - z_i^q \leq T \qquad p,q \in K,\ i \in V^p \cap V^q \setminus \{0\}. \tag{6}$$

The constraints (4) are commonly used in vehicle routing problems when dealing with time constraints, they are known as *Miller-Tucker-Zemlin inequalities* (see, e.g. [1]), and they imply that $z_j^k \geq z_i^k + t_{(i,j)}$ when $x_{(i,j)}^k = 1$ with $i,j \in V^k \setminus \{0\}$. Constraints (5) force the same to the depot. Finally time consistency is established with the inequality (6).

Formulation 2 uses a different variable $g_{(i,j)}^k$ with $k \in K$, $(i,j) \in A^k$ to represent the arrival time at customer i on day k only if j is the next location to visit. The expression $\sum_{a \in B} g_a^k$ is abbreviated as $g^k(B)$ for all $B \subseteq A^k$.

$$g^k(\delta_k^+(j)) - g^k(\delta_k^-(j)) \geq \sum_{i \in V^k \setminus \{j\}} t_{(i,j)} x_{(i,j)}^k \qquad k \in K,\ j \in V^k \setminus \{0\} \tag{7}$$

$$0 \leq g_a^k \leq M^k \cdot x_a^k \qquad k \in K,\ a \in A^k \setminus \delta_k^+(0) \tag{8}$$

$$g^p(\delta_p^+(i)) - g^q(\delta_q^+(i)) \leq T \qquad p,q \in K,\ i \in V^p \cap V^q \setminus \{0\} \tag{9}$$

Constraints (7) imply the duration of the route to customer j to be greater than or equal to the duration of the route to its predecessor i plus the time that the vehicle needs to make the journey from i to j. Equations (8) set the value of g_a^k to zero whenever $x_a^k = 0$. The remaining constraints (9) establish just like constraints (6) in the previous formulation, the time consistency. As in the previous formulation, there are big values M^k for all $k \in K$.

Formulation 3 is based on three families of mathematical variables. A first family consider a multi-commodity flow on each day with f_a^{ik} representing the path from the depot 0 to each customer $i \in V_k$ on each day $k \in K$. A second set includes y_i^k representing the idle time of the vehicle at customer i on day k. A third family includes w_j^{ik} representing the idle time on the variable w_j^{ik} if j precedes i. Each feasible solution must verify:

$$\begin{aligned}
f^{ik}(\delta_k^+(0)) - f^{ik}(\delta_k^-(0)) &= 1 && && (10)\\
f^{ik}(\delta_k^+(i)) - f^{ik}(\delta_k^-(i)) &= -1 && && (11)\\
f^{ik}(\delta_k^+(j)) - f^{ik}(\delta_k^-(j)) &= 0 && j \in V^k \setminus \{0, i\} && (12)\\
0 \leq f_a^{ik} &\leq x_a^k && a \in A^k && (13)\\
y_j^k - N_j^k \cdot f^{jk}(\delta_k^-(i)) \leq w_j^{ik} &\leq N_j^k \cdot f^{ik}(\delta_k^-(j)) && i, j \in V^k \setminus \{0\} && (14)\\
0 \leq w_j^{ik} &\leq y_j^k && i, j \in V^k \setminus \{0\} && (15)\\
\left(\sum_{a \in A^p} t_a f_a^{ip} + \sum_{j \in V^p \setminus \{0\}} w_j^{ip}\right) &- \\
\left(\sum_{a \in A^q} t_a f_a^{iq} + \sum_{j \in V^q \setminus \{0\}} w_j^{iq}\right) &\leq T && \begin{array}{l} p, q \in K \\ i \in V^p \cap V^q \setminus \{0\}. \end{array} && (16)
\end{aligned}$$

We take N_j^k as an upper bound on the maximum idle time at customer i in day k. The Eqs. (10)–(13) guarantee the existence of a path from the depot to each customer requiring a visit in each day. Inequalities (14)–(15) force w_j^{ik} to take the value y_j^k if j precedes i on day k, and it is zero otherwise. Constraints (16) establish the time consistency.

Formulations 1 and 2 have a weak linear programming relaxation, not only due to the big values involved in the constraint definitions, but also because the well-known subtour elimination constraints are poorly imposed, as [6] demonstrate. Indeed, the two compact formulations can be strengthened by including the following exponential set of inequalities:

$$x^k(\delta_k^+(S)) \geq 1 \qquad S \subset V^k \setminus \{0\}. \qquad (17)$$

These inequalities are already implicit in Formulation 3, which has a better linear-programming relaxation when compared to the ones of Formulations 1 and 2, as computationally confirmed in the next section.

4 Preliminary Computational Results

We have implemented computer codes to solve the formulations 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 tested the computer codes on benchmark instances from [7]. These instances were generated with $|K| = 3$ and 5 days, and with a probability for each client to require service on each day of $freq = 0.5$, 0.7 or 0.9. For example, an instance generated with $freq = 0.5$ means that a customer requires service in about half of the days in the time period K. The instances also include three options for the threshold T to limit the time inconsistency in the CTSP, but it is no longer meaningful in our bicriteria problem. Recall that T is unknown in our problem. Instead, we created three instances by using $\alpha = 0$, 0.1 and 0.3. The magnitude of the inconsistency makes useless to solve the problem with larger values for α on these benchmark instances.

We created five computer codes. Codes F1, F2 and F3 are associated with Formulations 1, 2 and 3, respectively. Formulations 1 and 2 were extended with the use of the Subtour Elimination Constraints (17), leading to codes F1+SEC and F2+SEC. Recall that Formulation 3 has these inequalities implicitly. We report computational results on tables with results from the five codes on each instance. The tables report three columns with characteristics of the instances. Column *Incons* shows the value of the maximum time difference obtained in the solution (i.e., the value of T in the best solution found). Column *Travel* shows the total travel cost of the routes. Column *Obj* shows the value of the obtained objective function 1 where the penalty α affects. In addition, for each code, each table reports two columns. Column *Time* shows the number of seconds taken by the MILP solver to solve the formulation with a time limit of one hour. Column *%-gap* is the difference between *Obj* and linear programming bound at the root node, divided by *Obj* and multiplied by 100.

Tables 1, 2 and 3 show how Formulations 1 and 2 benefit significantly from the use of the Subtour Elimination Constraints (17), it can be observed that Formulation 1 reaches the time limit in 21 of the 54 instances, however, applying the subtour elimination in the linear relaxation, the amount of reached time limits is reduced to only 1 case, in addition to reducing the execution time in the remaining cases. Similarly occurs with Formulation 2, where by applying the linear relaxation the number decreases from 31 to 3 time limits reached. In both cases the *%-gap* is significantly reduced. On the other hand Formulation 3 performs better than Formulations 1 and 2 since it reaches the time limit in 18 instances. However, it does not behave better than Formulations 1 and 2 when the subtour elimination constraints are added (i.e., codes F1+SEC and F2+SEC). Quite interestingly, the codes F1+SEC, F2+SEC and F3 produced the same linear programming bounds, never larger than 2%. In terms of computational times F1+SEC is the clear winner followed by F2+SEC. In overall terms, the problem becomes more difficult to solve regardless of the formulation as $|K|$, $freq$ or α increases. When the value of days or the number of nodes to visit each day increases, it is simply a bigger problem. On the other hand, increasing the weight α of the inconsistency T in the objective function makes the instance harder to be solved.

Table 1. Results on `ulysses22`

$\|K\|$	$freq$	α	Incons	Travel	Obj	F1		F1+SEC		F2		F2+SEC		F3	
						Time	%-gap	Time	%-gap	Time	%-gap	Time	%-gap	Time	%-gap
3	50	0	113	18453	18453.0	0.3	21.6	0.1	0.0	0.3	18.8	0.1	0.0	0.1	0.0
3	50	0.1	0	18463	16616.7	156.7	21.7	0.2	0.1	1793.3	18.8	0.2	0.1	2.5	0.1
3	50	0.3	0	18463	12924.1	28.5	21.7	0.1	0.1	2893.7	18.8	0.2	0.1	2.6	0.1
3	70	0	113	19213	19213.0	1.0	19.4	0.2	0.1	0.9	17.0	0.4	0.1	0.4	0.1
3	70	0.1	113	19213	17303.0	2148.7	19.5	0.2	0.1	351.6	17.0	0.5	0.1	13.8	0.1
3	70	0.3	113	19213	13483.0	221.7	19.6	0.4	0.3	294.4	17.2	0.6	0.3	18.5	0.3
3	90	0	1995	20657	20657.0	56.3	24.9	0.4	0.0	20.5	23.1	0.4	0.0	1.1	0.0
3	90	0.1	52	20718	18651.4	1 h	-	1.8	0.3	1 h	-	1.5	0.3	834.6	0.3
3	90	0.3	0	20724	14506.8	1 h	-	3.2	0.3	1 h	-	1.7	0.3	2395.8	0.3
5	50	0	3191	30412	30412.0	0.3	18.8	0.2	0.0	0.4	16.1	0.2	0.0	0.2	0.0
5	50	0.1	113	20433	27401.0	1 h	-	1.2	0.1	1 h	-	0.7	0.1	8.5	0.1
5	50	0.3	0	30449	21314.3	2323.4	18.9	1.1	0.1	1 h	-	0.5	0.1	13.0	0.1
5	70	0	113	32405	32405.0	2.1	22.1	0.6	0.0	3.5	19.7	0.7	0.0	0.8	0.0
5	70	0.1	113	32405	29175.8	1 h	-	0.5	0.1	1 h	-	1.0	0.1	75.0	0.1
5	70	0.3	113	32405	22717.4	1 h	-	1.4	0.2	1 h	-	2.1	0.2	73.2	0.2
5	90	0	3627	33826	33826.0	53.3	23.1	0.5	0.0	23.0	21.2	0.9	0.0	1.9	0.0
5	90	0.1	148	33904	30528.4	1 h	-	6.6	0.3	1 h	-	10.4	0.3	1 h	-
5	90	0.3	148	33904	23777.2	1 h	-	9.1	0.4	1 h	-	21.8	0.4	1 h	-

Table 2. Results on `bays29`

$\lvert K\rvert$	$freq$	α	Incons	Travel	Obj	F1		F1+SEC		F2		F2 + SEC		F3	
						Time	%-gap	Time	%-gap	Time	%-gap	Time	%-gap	Time	%-gap
3	50	0	319	4270	4270.0	0.3	17.1	0.2	0.0	0.5	16.6	0.2	0.0	0.4	0.0
3	50	0.1	184	4274	3865.0	4.2	17.6	0.6	0.6	208.3	17.1	0.8	0.6	21.0	0.6
3	50	0.3	0	4300	3010.0	4.6	17.7	1.4	0.7	488.8	17.2	1.0	0.7	84.1	0.7
3	70	0	57	5075	5075.0	0.4	14.8	0.3	0.0	1.3	14.4	0.5	0.0	2.0	0.0
3	70	0.1	57	5075	4573.2	11.5	15.0	1.2	0.1	2960.2	14.5	0.9	0.1	53.9	0.1
3	70	0.3	0	5083	3558.1	13.4	15.0	0.4	0.2	1 h	-	1.1	0.2	35.0	0.2
3	90	0	744	5644	5644.0	1.3	10.5	0.7	0.1	7.1	10.0	1.4	0.1	5.4	0.1
3	90	0.1	0	5690	5121.0	3350.5	11.2	8.8	0.9	1 h	-	16.6	0.9	1 h	-
3	90	0.3	0	5690	3983.0	2758.6	11.2	11.9	0.9	1 h	-	39.7	0.9	1 h	-
5	50	0	319	7126	7126.0	0.4	14.6	0.3	0.0	0.7	14.1	0.5	0.0	0.7	0.0
5	50	0.1	184	7130	6435.4	43.8	14.9	2.3	0.3	1 h	-	1.9	0.3	56.7	0.3
5	50	0.3	0	7197	5037.9	84.8	15.4	7.5	1.0	1 h	-	9.1	1.0	659.0	1.0
5	70	0	283	8527	8527.0	0.7	14.0	0.9	0.1	2.6	13.6	1.7	0.1	3.0	0.1
5	70	0.1	57	8542	7693.5	1504.1	14.2	8.6	0.3	1 h	-	12.3	0.3	1047.2	0.3
5	70	0.3	57	8542	5996.5	558.7	14.4	10.9	0.5	1 h	-	20.3	0.5	3447.7	0.5
5	90	0	775	9477	9477.0	1.9	11.3	6.1	0.2	23.2	10.9	14.1	0.2	31.1	0.2
5	90	0.1	240	9512	8584.8	1 h	-	39.6	0.9	1 h	-	112.4	0.9	1 h	-
5	90	0.3	0	9551	6685.7	1 h	-	115.1	1.0	1 h	-	878.4	1.0	1 h	-

Table 3. Results on `dantzig42`

$\|K\|$	$freq$	α	Incons	Travel	Obj	F1		F1+SEC		F2		F2 + SEC		F3	
						Time	%-gap	Time	%-gap	Time	%-gap	Time	%-gap	Time	%-gap
3	50	0	247	1599	1599.0	2.2	22.8	0.4	0.0	1.9	21.4	0.6	0.0	2.2	0.0
3	50	0.1	42	1619	1461.3	1 h	-	9.2	1.5	1 h	-	12.5	1.5	1 h	-
3	50	0.3	0	1631	1141.7	1 h	-	18.4	2.0	1 h	-	23.3	2.0	1 h	-
3	70	0	409	1758	1758.0	134.2	24.5	1.0	0.4	15.6	22.7	1.9	0.4	17.0	0.4
3	70	0.1	0	1787	1608.3	1 h	-	35.4	2.0	1 h	-	135.1	2.0	1 h	-
3	70	0.3	0	1787	1250.9	1 h	-	82.4	2.0	1 h	-	226.7	2.0	1 h	-
3	90	0	487	2041	2041.0	2001.7	23.2	4.4	0.4	1 h	-	13.0	0.4	324.7	0.4
3	90	0.1	30	2046	1844.4	1 h	-	41.9	0.8	1 h	-	130.1	0.8	1 h	-
3	90	0.3	12	2049	1437.9	1 h	-	88.5	1.1	1 h	-	418.0	1.1	1 h	-
5	50	0	452	2678	2678.0	3.5	22.8	0.8	0.0	13.0	21.6	1.1	0.0	3.2	0.0
5	50	0.1	288	2678	2439.0	1 h	-	111.7	1.2	1 h	-	144.2	1.2	1 h	-
5	50	0.3	23	2719	1910.2	1 h	-	1028.9	1.9	1 h	-	1 h	-	1 h	-
5	70	0	494	2936	2936.0	288.7	26.1	5.0	0.2	268.5	24.3	6.6	0.2	45.9	0.2
5	70	0.1	187	2952	2675.5	1 h	-	1699.0	1.5	1 h	-	3238.1	1.5	1 h	-
5	70	0.3	48	2968	2092.0	1 h	-	2664.6	2.0	1 h	-	1 h	-	1 h	-
5	90	0	521	3337	3337.0	2351.9	23.0	28.1	0.3	1 h	-	58.2	0.3	258.0	0.3
5	90	0.1	118	3344	3021.4	1 h	-	454.0	0.9	1 h	-	2083.1	0.9	1 h	-
5	90	0.3	-	-	-	1 h	-	1 h	-	1 h	-	1 h	-	1 h	-

5 Conclusions

This article introduces and addresses a bicriteria optimization variant of the Consistent Travelling Salesman Problem. While waiting times are forbidden in the Consistent Travelling Salesman Problem, our variant allows waiting times for the vehicle at customer locations. This paper describes three Integer Linear Programming models that are suitable to enumerate some Pareto solutions using the weighted-sum method known in Multiobjective Optimization. The first and second formulations outperform the third formulation when they are strengthened with the subtour elimination constraints. As a future direction of research, we plan to implement a Benders' Decomposition Approach for the third formulation. Although in principle generating inequalities also consume time, the advantage of working with a multicriteria problem is that the valid inequalities generated when using a weight α are also valid for a different weight α'. This advantage can be used both for the subtour elimination constraints and for the Benders' cuts, thus potentially saving computational time when solving a sequence of problems.

Acknowledgements. This research has been supported by the Spanish research project PID2019-104928RB-I00.

References

1. Bektas, T., Gouveia, L.: Requiem for the Miller-Tucker-Zemlin subtour elimination constraints. Eur. J. Oper. Res. **236**(3), 820–832 (2014). https://doi.org/10.1016/j.ejor.2013.07.038
2. Groër, C., Golden, B., Wasil, E.: The consistent vehicle routing problem. Manuf. Serv. Oper. Manag. **11**(4), 630–643 (2009). https://doi.org/10.1287/msom.1080.0243
3. Gunantara, N.: A review of multi-objective optimization: methods and its applications. Cogent Eng. **5**(1), 1502242 (2018). https://doi.org/10.1080/23311916.2018.1502242
4. Kirlik, G., Sayın, S.: A new algorithm for generating all nondominated solutions of multiobjective discrete optimization problems. Eur. J. Oper. Res. **232**(3), 479–488 (2014). https://doi.org/10.1016/j.ejor.2013.08.001
5. Marler, R.T., Arora, J.S.: The weighted sum method for multi-objective optimization: new insights. Struct. Multidiscip. Optim. **41**, 853–862 (2010). https://doi.org/10.1007/s00158-009-0460-7
6. Orman, A.J., Williams, H.P.: A survey of different integer programming formulations of the travelling salesman problem (2007)
7. Subramanyam, A., Gounaris, C.E.: A branch-and-cut framework for the consistent traveling salesman problem. Eur. J. Oper. Res. **248**(2), 384–395 (2016). https://doi.org/10.1016/j.ejor.2015.07.030
8. Subramanyam, A., Gounaris, C.E.: A decomposition algorithm for the consistent traveling salesman problem with vehicle idling. Transp. Sci. **52**(2), 386–401 (2018). https://doi.org/10.1287/trsc.2017.0741