Efficient formulations for the Traveling Car Renter Problem and its Quota variant

Mathieu Lacroix

Université Sorbonne Paris Nord, LIPN, CNRS, UMR 7030, F93430, Villetaneuse, France

Yasmín A Ríos-Solís[1]

Escuela de Ingeniería y Ciencias, Tecnológico de Monterrey, Mexico

Roberto Wolfler Calvo

Université Sorbonne Paris Nord, LIPN, CNRS, UMR 7030, F93430, Villetaneuse, France

Abstract

In this paper, we study the recently introduced Traveling Car Renter Problem. This latter is a generalization of the well-known Traveling Salesman Problem (TSP), where a solution is a set of paths of different colors as well as an orientation of each path in such a way that the union forms a directed Hamiltonian circuit. Considering costs associated with all edges and all ordered pairs of nodes for each color, the cost of a solution is the sum of the costs of its colored oriented paths, the cost of these later being the sum of the edge costs plus the costs of the arcs from their destination to their origin. We also consider the Quota version of this problem where a weight is associated with every node and the circuit formed by a solution may not be Hamiltonian but must cover a subset of nodes whose sum of weights should be greater than or equal to a fixed value. We propose integer linear programming formulations for these problems. We also propose some valid inequalities for strengthening the models and we devise branch-and-cut algorithms for solving these formulations. The computational results show the efficiency of our formulations as we solve to optimality almost all the instances of the literature, and outperform by an order of magnitude all published approaches.

October 16, 2023

1 Introduction

In this paper, we consider the *Traveling Car Renter Problem (CARS)*. An instance of the CARS is given by an undirected graph $G = (V, E)$ where $V = \{1, \ldots, n\}$ indicates a set of nodes and $ij \in E$ is an edge between nodes i and j. A set $K = \{1, \ldots, o\}$ of colors is also given. Moreover, for every color $k \in K$, a cost d_e^k is associated with each edge $e \in E$ and a cost $c_{i,j}^k$ is associated with each ordered pair $(i, j) \in V \times V$ (note that c is asymmetric whereas d is symmetric). A solution to the CARS is a set of paths of different colors as well as an orientation of each path in such a way that the union forms a directed Hamiltonian circuit. The cost of a solution is defined as the sum of the costs of the colored oriented paths, where the cost of a st-path P with color k oriented from s to t is defined as $\sum_{e \in P} d_e^k + c_{s,t}^k$. The CARS consists of finding a solution of minimum cost.

The CARS was introduced by Goldbarg et al. (2012), and it is directed to the viewpoint of vehicle rental customers. In general, a user of rented vehicles aims at traveling a specific route

---

[1]This author carried out part of this research during an academic stay at *Université Sorbonne Paris Nord, LIPN, CNRS, UMR 7030, F93430, Villetaneuse, France*

by minimizing the rental cost. Nowadays, there are several available vehicles, which means that a user may choose the most attractive vehicle for traveling different parts of the intended route. Nevertheless, an extra fee must be paid whenever a vehicle is delivered to a city different from the one it was rented. Hence, this problem corresponds to the CARS: each vehicle corresponds to a color $k \in K$ and a st-path P of color k oriented from s to t in a solution corresponds to renting the vehicle k at city s, traveling with this vehicle through P and delivering it at city t. The total rental cost of a trip is given by the travel cost $\sum_{e \in P} d_e^k$ plus the extra fee $c_{s,t}^k$ of returning the vehicle from t to s. Note that in this context, a specific node corresponding to the initial position of the customer is generally set, and an initial rental is required at this node.

The *Quota CARS (QCARS)* is a variant of the CARS where a weight q_i is associated with each node $i \in V$, and a quota Q is given, representing the minimum amount of weight that should be collected. In the QCARS, it is not anymore mandatory to visit all the nodes. Indeed, the algorithm can select the subset of nodes it visits under the constraint to reach at least the threshold Q. Therefore, the QCARS aims at finding at minimum cost a set of paths of different colors with orientations in such a way that the union forms a directed circuit covering a subset of nodes U satisfying $\sum_{i \in U} q_i \geq Q$.

The QCARS was introduced in (da Silva Menezes et al., 2014) to model the variant of the CARS where the user, a tourist, wants to visit a subset of touristic places to reach a level of satisfaction. The QCARS also models a specific case of the Rapid Transit Networks Design problems that consist of embedding a set of interconnected transit lines within an undirected network. The nodes correspond to population centroids in a city, while the edges correspond to potential connections to be built between node pairs. Let q_i be the population associated with node i, usually defined as the population living within a reasonable walking distance of the node. Let d_{ij}^k be the cost of constructing a link of type k between i and j and let $c_{i,j}^k$ indicate the cost of constructing a depot at the beginning (i.e., node i) and a depot at the ending (i.e., node j) of the transit line of type k. This cost is asymmetric due to the difference in real-estate price in i and j and the different size of the starting and ending depots. Moreover, the cost could be different since the lines could be of different types, such as: buses, metros, light metros, tramways, or fully overground light-rail systems. An objective of a rapid transit system is to minimize the cost while deserving a certain amount of population (Laporte et al., 2011). In this case, the model locates the stations chosen among a subset of nodes, determines the connections between the stations, and partitions them into different lines. Under the hypotheses that there is only one line of each type and the lines should create a circuit, this is precisely the QCARS.

The TSP is a particular case of the CARS where only one car is considered, and since the TSP is NP-hard, so is the CARS. However, from a computational point of view, the TSP can be solved up to thousands of nodes (Applegate et al., 2006), whereas the exact existing methods can only solve instances of the CARS with up to 20 nodes and five colors. In this paper, we provide a new integer linear programming formulation for the CARS, and devise a branch-and-cut algorithm able to solve in an exact way almost all the instances of Goldbarg et al. (2012), even those containing hundreds of nodes and more than four colors.

Moreover, the proposed formulation and the branch-and-cut algorithm have been adapted to the QCARS, solving to optimality all the instances generated in (da Silva Menezes et al., 2014), the vast majority being solved in less than a second.

2 Literature review

Mathematical programming models and approaches for solving the CARS have been scarcely explored. The first formulations for modeling the CARS have been proposed in (Goldbarg et al., 2016) and in (da Silva and Ochi, 2016). As pointed out by Rios et al. (2017), the formulation in (da Silva and Ochi, 2016) actually models the CARS without an initial rental required on a specific node. Nevertheless, an incomplete integer programming formulation containing nonlinear constraints was previously proposed in the literature (Goldbarg et al., 2013). In (Goldbarg et al., 2018), three different mathematical formulations are proposed and computationally compared. The first model has a quadratic objective function and is based on the TSP formulation viewed as a particular case of the quadratic assignment problem. The assignment variables represent the positions of the nodes in the Hamiltonian circuit. The second model also has a quadratic objective function, but it is based on the Gavish-Grave's formulation for the TSP. The last proposed model is based on the Dantzig-Fulkerson-Johnson's formulation for the TSP (Applegate et al., 2006) and has quadratic constraints. An experimental comparison of the three linearized formulations is performed. Instances with up to 52 nodes and three colors are solved to optimality. However, when the number of colors increases, the performance worsens dramatically: with five colors, only instances with less than 20 nodes are solved to optimality.

Pedrosa et al. (2019) study the CARS for which the rental costs $c_{s,t}^k$ are all equal for any $s, t \in V$ and any color $k \in K$. They propose an $O(log\ n)-$approximation algorithm for the problem based on the randomized rounding of an exponentially large linear relaxation.

Different metaheuristics have been devised for the CARS. Goldbarg et al. (2012) devise two greedy randomized adaptive search procedures with variable neighborhood descent. Several evolutionary algorithms have also been developed: memetic algorithms (Goldbarg et al., 2012; da Silva Menezes et al., 2014), transgenetic algorithms (Asconavieta et al., 2011; Goldbarg et al., 2013) and scientific algorithm (Felipe et al., 2014). Two hybrid algorithms have also been proposed (da Silva and Ochi, 2016; Rios et al., 2017).

The QCARS has also been considered in the literature. Two integer linear programming formulations for the QCARS are presented in (da Silva Menezes et al., 2014; Goldbarg et al., 2016). Both formulations are based on the same nonlinear constraints, but differ by the linearization techniques applied. By using these formulations, instances with up to 16 nodes are solved to optimality. da Silva Menezes et al. (2014) and Goldbarg et al. (2016) also devise evolutionary algorithms to solve the QCARS. Computational experiments based on instances with up to 100 nodes are reported.

A related problem to the CARS is the *colorful traveling salesman problem* (Xiong et al., 2007; Ismkhan, 2017), where edges are colored, and the objective is to find a Hamiltonian circuit with the minimum number of distinct colors. The different colors must not be contiguous, and there are no rental costs as in the CARS. Another problem close to the CARS is the *traveling salesman problem with flexible coloring* (Roemer et al., 2012). In this problem, nodes belong to various color classes, and the aim is to assign a color to each node to find the shortest Hamiltonian circuit with the particularity that all nodes of the same color must be visited consecutively.

3 Integer linear programming formulations

In this section, we introduce new formulations for the CARS and the QCARS. Moreover, we present valid inequalities to strengthen the formulations and devise branch-and-cut algorithms to solve these formulations. We also discuss some variants of the problems.

To compare our approach with those in the literature (except the one proposed by da Silva and Ochi (2016)), we take into account the same assumption: node one must be the extremity of a path. This assumption stems from the fact that a tourist arrives at a specific city (node one) and starts its trip by renting a vehicle.

3.1 CARS Formulation

In our formulation the orientation of a st-path of color k is represented by the arc from s to t with color k. Then a solution with several colors consists of a union of undirected colored paths and by a directed circuit composed of the orientations of the paths. Figure 1 represents a solution on 14 nodes with three colored paths: one with extremities 1 and 8, one with extremities 8 and 11, and the last one with extremities 11 and 1. The first path is oriented from 1 to 8, the second from 8 to 11, and the last one from 11 to 1.
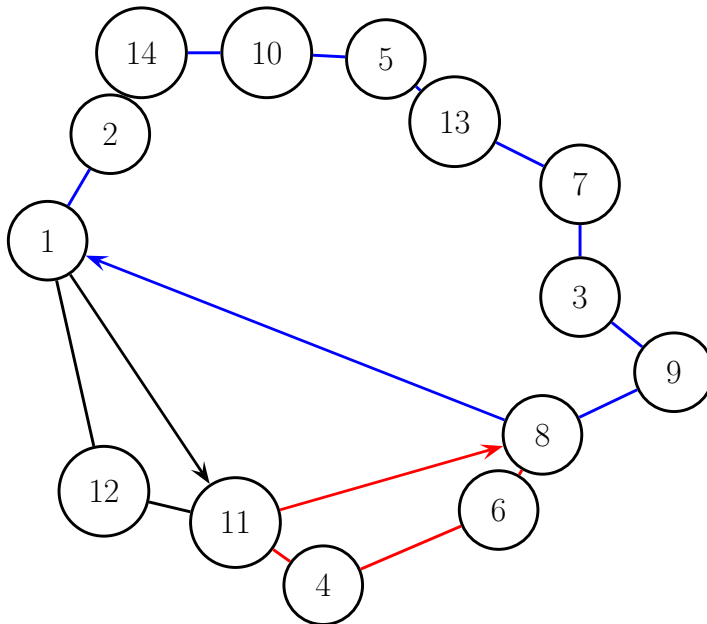
Figure 1: An example of a solution of an instance of 14 nodes which uses three colors: red, blue and black

The proposed formulation contains four types of variables. The binary variable x_e^k equals one when the edge $e \in E$ belongs to the path having color $k \in K$. The binary variable $y_{i,j}^k$ equals one when the solution contains a path of color $k \in K$ that has i and j as extremities and that is oriented from i to j, for all $i \neq j \in V \times V$. The binary variable z_i^k equals 1 if node $i \in V$ is

adjacent to an edge of color $k \in K$. The binary variable w^k equals 1 if color $k \in K$ is the only color used in the solution. Therefore the CARS can be formulated as follows where $\delta(i)$ is the set of edges of E incident to a node $i \in V$, $E[S]$ is the set of edges having both extremities in a node set $S \subseteq V$ and $x^k(F) = \sum_{e \in F} x_e^k$ for $F \subseteq E$.

$$(CARS) \quad \min \sum_{k \in K} \sum_{e \in E} d_e^k x_e^k + \sum_{k \in K} \sum_{i \neq j \in V} c_{i,j}^k y_{i,j}^k + \sum_{k \in K} c_{1,1}^k w^k \tag{1}$$

$$s.t. \sum_{k \in K} x^k(\delta(i)) = 2 \qquad\qquad\qquad i \in V \tag{2}$$

$$\sum_{k \in K} x^k(E[S]) \leq |S| - 1 \qquad\qquad S \subseteq V, 2 \leq |S| \leq \lfloor \frac{n}{2} \rfloor \tag{3}$$

$$x^k(\delta(i)) + \sum_{j \in V \setminus \{i\}} \left(y_{j,i}^k + y_{i,j}^k \right) = 2z_i^k \qquad k \in K, i \in V \tag{4}$$

$$\sum_{i \neq j \in V} y_{i,j}^k + \sum_{\ell \in K} w^\ell \leq 1 \qquad\qquad k \in K \tag{5}$$

$$x^k(\delta(1)) = z_1^k + w^k \qquad\qquad\qquad k \in K \tag{6}$$

$$\sum_{k \in K} \sum_{j \in V \setminus \{i\}} y_{i,j}^k - \sum_{k \in K} \sum_{j \in V \setminus \{i\}} y_{j,i}^k = 0 \qquad i \in V \tag{7}$$

$$w^k, x_e^k, y_{i,j}^k, z_i^k \in \{0,1\}$$

The objective function (1) minimizes the travel cost of the Hamiltonian circuit and the sum of the rental costs. Equations (2) are the degree constraints that force the tour to visit each node once. Constraints (3) are the well-known subtour elimination constraints stating that for each node subset $S \subsetneq V$ the number of edges in the solution having both extremities in S must be less than $|S|$. By (4), if a node i is colored with $k \in K$ (that is $z_i^k = 1$) then it is incident to two edges of the path of color k or it is one of its extremities. Inequalities (5) imply that there is at most one path of each color if the solution contains several colors (i.e., $\sum_{k \in K} w^k = 0$) and if the solution contains only one color ($w^k = 1$ for some $k \in K$), its colored path is a circuit (i.e., $y_{i,j}^k = 0$ for all $i \neq j \in V$, $k \in K$). Equation (6) specifies that node 1 is adjacent to two edges of a same color $k \in K$ only if k is the only color in the solution. Equations (7) are the flow conservation constraints associated with variables y. Consider a binary point $\bar{w}, \bar{x}, \bar{y}, \bar{z}$ satisfying constraints (2)-(7). By (2) and (3), the set of edges satisfying $\sum_{k \in K} \bar{x}_e^k = 1$ corresponds to a Hamiltonian circuit, say C. For a color $k \in K$, let S^k be the set of nodes $i \in V$ with $\bar{z}_i^k = 1$, E^k be the set of edges $e \in E$ with $\bar{x}_e^k = 1$ and A^k be the set of arcs (i,j) such that $\bar{y}_{i,j}^k = 1$. By (4), $E^k \subseteq E[S^k]$ and $i, j \in S^k$ for all $(i,j) \in A^k$. Moreover, $|E^k| + |A^k| = |S^k|$. If $w^k = 1$ for some $k \in K$, then $A^k = \emptyset$ by (5) and $E^k \neq \emptyset$ by (6). Since $|E^k| = |S^k|$, constraints (3) imply that $S^k = V$ and the solution if a Hamiltonian circuit of color k. Suppose now that $w^k = 0$ for all $k \in K$. For each $k \in K$, E^k is not a Hamiltonian circuit by (6). Hence, by (3), $|E^k| \leq |S^k| - 1$. This implies that $A^k = \{(s_k, t_k)\}$ and $|E^k| = |S^k| - 1$ by (5). Constraints (2), (3) and (4) ensure that E^k is a $s_k t_k$-path covering S^k which means that C is a union of different colored paths. By (7), their orientation forms a directed Hamiltonian circuit, proving the validity of the formulation.

3.2 QCARS formulation

Recall that for this version of the CARS, some nodes may not be visited. Indeed, the model determines the nodes to visit to reach the quota Q at the least cost. Thus, we modify the previous model by introducing additional binary variables: the variable u_i equals one if the node $i \in V$ is visited. To ensure that the collected profit reaches the quota, we consider the following constraint:

$$\sum_{i \in V} q_i u_i \geq Q \tag{8}$$

We also need to modify inequalities (2) and (3). Equations (2) are replaced by the following:

$$\sum_{k \in K} x^k(\delta(i)) = 2u_i \quad i \in V \tag{9}$$

These equations impose that the circuit formed by $\{e \in E : \sum_{k \in K} x_e^k = 1\}$ covers i if and only if i is visited (i.e., $u_i = 1$).

The colored paths may form a circuit that is not Hamiltonian but covers node 1 since some of the other nodes may not be visited. Hence, inequalities (3) are no more valid. We must either restrict inequalities (3) to node subsets not containing node 1 or replace these inequalities by the following generalized subtour inequalities introduced by Wolsey (1998):

$$\sum_{k \in K} x^k(E[S]) \leq \sum_{j \in S \setminus \{i\}} u_j, \quad S \subseteq V \text{ such that } 1 \notin S, i \in S. \tag{10}$$

3.3 Strengthening the linear programming relaxations

In this section, we reinforce the linear relaxations of the CARS and QCARS formulations by strengthening the lower and upper bounds of each variable z_i^k with three inequalities. The efficiency of these new inequalities is experimentally showed in Section 4.

The first family gives an upper bound on variables z by ensuring that a node i is adjacent to an edge of color k (i.e., $z_i^k = 1$) only if there exists a path or a circuit of color k (i.e., $\sum_{u \neq v \in V} y_{u,v}^k + w^k = 1$):

$$z_i^k \leq \sum_{u \neq v \in V} y_{u,v}^k + w^k \quad i \in V, k \in K \tag{11}$$

Note that inequalities (5) imply that inequalities (11) dominate $z \leq 1$.

The other two families of constraints give a lower bound on variables z. The second family imposes that an edge $e = ij$ is of color k (i.e., $x_e^k = 1$) only if its extremities are incident to an edge of color k (i.e., $z_i^k = 1$ and $z_j^k = 1$).

$$x_e^k \leq z_i^k \quad i \in V, e \in \delta(i), k \in K. \tag{12}$$

The last family implies that a node $i \in V \setminus \{1\}$ is the extremity of the path of color k (i.e., $\sum_{j \in V \setminus \{i\}} (y_{i,j}^k + y_{j,i}^k) = 1$) only if it is incident to an edge of color k (i.e., $z_i^k = 1$).

$$w^k + \sum_{j \in V \setminus \{i\}} (y_{i,j}^k + y_{j,i}^k) \leq z_i^k \quad i \in V \setminus \{1\}, k \in K. \tag{13}$$

Note that inequalities (13) are not defined for node 1 since it is redundant with respect to the equation (4) associated with node 1 and equation (6).

3.4   Branch-and-cut algorithms

In the previous sections, we proposed integer linear formulations for the CARS and its quota variant. Both formulations contain an exponential number of inequalities in order to prevent the formation of subtours: the subtour elimination constraints (3) for the CARS and the generalized subtour elimination constraints (10) for the QCARS. Note that the other constraints, even those introduced in Section 3.3 for strengthening the linear relaxations, are in polynomial number. These formulations cannot be directly solved outside very small instances due to this exponential number of constraints. Hence, we solve these formulations using branch-and-cut algorithms where the (generalized) subtour elimination constraints (3) and (10) are added in a lazy way.

CARS   The formulation obtained by adding the strengthening inequalities of Section 3.3, but removing the subtour elimination constraints (3), that is, the formulation given by inequalities (2), (4)-(7), (11), (12) and (13) is solved using a branch-and-bound procedure. Each time an integer solution is found, the algorithm checks whether it is feasible for the CARS, that is, whether it satisfies all the subtour elimination constraints (3). If not, the integer point is discarded and a constraint (3) violated by this point is added to the current linear relaxation.

This cκeck procedure, called *separation problem associated with the subtour elimination constraints*, is performed in linear time using a breadth-first search. When subtours are encountered, only the inequality associated with the smallest subtour is added to the current linear relaxation[2].

QCARS   The formulation solved using a branch-and-bound is given by inequalities (4)-(9), (11), (12) and (13). An integer solution of this formulation is feasible for the QCARS if it satisfies all the generalized subtour elimination constraints (10).

The separation problem associated with the generalized subtour elimination constraints is performed in linear time using a breadth-first search. If the solution contains subtours, let S be the smallest subset of nodes inducing a subtour and not containing node 1. In this case, the $|S|$ violated inequalities (10) associated with S are added to the current linear relaxation.

3.5   Variants

The formulations introduced below can be slightly modified to tackle different variants of the CARS and QCARS.

Depot.   The proposed formulations are based on the assumption that node one is the extremity of a path or the starting node of the Hamiltonian circuit with only one color. However, this assumption may be relaxed by modifying the cost $c_{1,1}^k$ to $\min_{i \in V} c_{i,i}^k$ and by replacing constraints (6) by inequalities (11). Indeed, the modification of the cost ensures that the associated rental cost is

---

[2]We tested the variant where instead of adding one inequality, we added the inequalities associated with all the encountered subtours, but it is less efficient probably because many more constraints have to be handled at each node.

minimum when a solution contains only one color. Constraints (11) ensure that if color $k \in K$ is the only used color (*i.e.*, $w^k = 1$), no other color is used (*i.e.*, $z_i^\ell = 0$ for all $i \in V$, $\ell \in K$).

Remark that for the QCARS, node one must still be covered by the circuit.

Symmetric vs. asymmetric costs. Accordingly to the instances proposed in the literature, we consider in this paper that the cost matrix c is asymmetric, whereas d is symmetric. However, the proposed formulations can be adapted to all the other cases. If c is symmetric, the formulations may be changed by only using variables $y_{i,j}^k$ with $i < j$ and removing equations (7). If d is asymmetric, one needs to consider directed variables $x_{i,j}^k$ and $x_{j,i}^k$ for all $ij \in E$ and $k \in K$ and add flow conservation constraints similar to (7) for the x variables.

Paths with the same color. In a solution to the CARS and QCARS, the paths have different colors. This hypothesis is quite restrictive, and one may consider a variant where several paths with the same color are allowed. This variant can be tackled by removing inequalities (5) and by adding the following inequalities:

$$\sum_{i \in W} \sum_{j \notin W} (y_{i,j}^k + y_{j,i}^k) \leq x^k(\delta(W)) \quad k \in K, \ \emptyset \neq W \subsetneq V. \tag{14}$$

and replacing (5) by

$$y_{i,j}^k + \sum_{\ell \in K} w^\ell \leq 1 \quad k \in K, i \neq j \in V. \tag{15}$$

Inequalities (14) impose that the number of paths of a color $k \in K$ having one extremity in W is no more than the number of edges of color k in the cut $\delta(W)$, for every $\emptyset \neq W \subsetneq V$. The validity of these inequalities stems from the fact that the paths a the same color are disjoint since the union of all the colored paths forms a Hamiltonian circuit. Note that the complexity of the separation problem associated with inequalities (14) is unknown, but such a model may be solved within a branch-and-cut algorithm by separating these inequalities in a lazy way (that is, for binary points only).

4 Computational study

In this section, we present the computational results we obtain for the CARS and the QCARS by using the branch-and-cut algorithms given in Section 3.4. We describe the instances of the literature on which our algorithms have been applied and then, we discuss the experimental results we obtain. The instances, the codes and the detailed results can be found in https://doi.org/10.6084/m9.figshare.11959287.

Implementation details The branch-and-cut algorithms presented in Sections 3.4 were implemented in C++, solved with Gurobi 8.01 with the default parameters, and executed on a MacPro with 3.5 GHz 6-Core Intel Xeon E5 and 32 GB of RAM. The time limit for solving each instance is of 10000 seconds, as proposed by Goldbarg et al. (2018).

4.1 Instances

CARS We use the benchmark proposed in Goldbarg et al. (2018), which is composed of 100 different instances based on complete graphs, divided into two sets of 50 instances each: the *Euclidean instances* and the *non-Euclidean instances*. These two sets differ by the way the distance matrices d^k, $k \in K$, are obtained.

The symmetric cost matrices d^k are computed from an initial $n \times n$ symmetric matrix $\bar{d}$. This matrix $\bar{d}$ stems from a TSP instance, or it is generated either by considering real distances between some arbitrarily chosen cities or by taking random values.

In the Euclidean instances, $\bar{d}$ is Euclidean and a positive integer L_i^k is randomly generated for all $i \in V$ and for all $k \in K$. The matrix d^k, $k \in K$, is then given by $d_{ij}^k = \frac{2L_i^k + 3L_j^k}{3} + \bar{d}_{ij}$ for all $i < j$. In the non-Euclidean instances, for each $k \in K$, the matrix d^k is given by $d_{ij}^k = \omega_{ij}^k \bar{d}_{ij}$ for all $i < j$, where ω_{ij}^k is a random value between $[1.4, 2]$. In all instances, the asymmetric cost matrices c^k are generated as follows. A positive integer α_i^k is randomly chosen for all $i \in V$ and for all $k \in K$ and $c_{i,j}^k = 6\alpha_i^k + 2\alpha_j^k$ for all $i \neq j$.

QCARS We use the benchmark proposed in Goldbarg et al. (2016). These instances are some CARS instances adapted to the QCARS. For this, an integer weight q_i is generated uniformly from $[0,100]$ for all $i \in V$ and the quota is $Q = 0.8 \sum_{i \in V} q_i$. There are 34 Euclidean instances and 33 non-Euclidean.

4.2 Discussion of Results

In Tables 1-3, we compare the results we obtain with our algorithms with the best exact and heuristic approaches of the literature[3]. All tables contain four sets of columns. The first one describes the instances, and the second reports the results we obtain with our formulation. The last two sets report the best results among the exact and heuristic approaches proposed in the literature, respectively. The first set contains the instance name, the number of nodes $|V|$, and the number of different colors $|K|$. For all methods, the column entitled "Sol." reports the best solution value found, and the column entitled "Time" denotes the computational time in seconds. For exact methods, the column entitled "Gap"[4] reports the percentage relative gap between the upper and lower bounds. Column "Col." indicates the number of colors in the best solution found by our algorithm. Finally, we always report to which paper of the literature we refer to. When it is clear since there is only one paper, we report it as a label over the set of columns. When there are several of them, we add a column entitled "Model" for the exact approach or "Algo" for the heuristic one.

Empty values for a row in any of the last two sets of columns indicate that we could not find computational results for the instance with the corresponding approach. Note that for both CARS and QCARS, there are many instances for which there are neither exact nor heuristic computational results in the literature, to the best of our knowledge.

In all the tables solution values in bold font correspond to the best found solution.

---

[3]The comparisons about running times in Tables 1-3 give an order of magnitude, but it is not precise since the algorithms run on different computers.

[4]The gap is computed as 100(best solution found - best lower bound)/best lower bound.

4.2.1   CARS Results

We compare our CARS model with the best approaches found in the literature. For exact approaches, DFJ and GG refer to the formulations given in (Goldbarg et al., 2018) based on the Dantzig-Fulkerson-Johnson's and Gavish-Grave's formulations for the TSP, respectively. Since in their paper, the authors run experiments with both Cplex 12.6.3.0 and Gurobi 6.5.2 on a PC with an Intel Core i7 3.45GHz x 8 and 32 Gb of RAM, we precise the solver by adding the letter C (DFJ-C and GG-C) or G (DFJ-G and GG-G) when the result is obtained using Cplex or Gurobi, respectively. Concerning the heuristics, TA and MA refer to the transgenetic heuristic and memetic algorithm proposed by Goldbarg et al. (2013) and implemented on an Intel Xeon QuadCore W3520 2.8 GHz, 8 GB RAM, running Scientific Linux 5.5 with 64 bits and coded in C++. EALSP reports the result achieved by the evolutionary algorithm hybridized with an adaptive local search procedure presented in (da Silva and Ochi, 2016) on a notebook with i7 3630-QM 2.4 GHz processor, 8GB RAM and Windows 8.0 64-bits using CPLEX 12.6.1. Finally, ScS+ALSP gives the result obtained with the scientific heuristic also hybridized with the adaptive local search procedure proposed by Rios et al. (2017) and tested on a PC with an Intel Core i5-2450M, CPU 2.50GHz x4, 3.8 Gb of RAM which ran Ubuntu.

Table 1 reports the computational results for the CARS on Euclidean instances. Our approach is able to solve to optimality all the instances except three. Moreover, 86% of the instances are solved to optimality in less than a minute. Note that the proposed approach outperforms the best exact ones by at least one order of magnitude of computational time. Moreover, it is competitive with heuristics in terms of computational time. Our algorithm is outperformed by the heuristics for only two instances. However, it improves the best known solution for five instances.

Table 2 reports the results obtained for the non-Euclidean instances for the CARS. Our algorithm solves all the instances except the one with 300 nodes and five colors. Moreover, 79% of the instances are solved to optimality in less than a minute. Note that our exact approach is in general at least one order of magnitude faster than the algorithms proposed in the literature including the heuristics. For 14 instances, our algorithm finds a solution better than the best solution reported in the literature.

4.2.2   QCARS results

In tables 4 and 3 we compare our QCARS model with the exact formulation and the evolutionary algorithm proposed in (Goldbarg et al., 2016). One instance presents inconsistencies (reported by I in the table) in the solution (the heuristic finds a solution whose value is better than the optimum). Table 4 summarizes the results obtained for the QCARS on the Euclidean instances. All instances except one are solved in less than nine seconds. The latter one is solved in 182 seconds, but it seems to be a hard instance since the model proposed in Goldbarg et al. (2016) finds a gap of 38.12% after 80000 seconds while for all the others the gap is less than 0.3%. Table 3 summarizes the results obtained for the QCARS on the non-Euclidean instances. All instances are solved in less than four seconds except one which is solved in less than eight seconds.

The proposed algorithm is really more efficient than the exact approach proposed in (Goldbarg et al., 2016). Some instances that were unsolved in 80000 seconds are now solved within few seconds. Moreover, our algorithm is faster than their heuristic approach. However, note that it

			CARS				Exact Approaches				Heuristic Approaches		
Instance	$|V|$	$|K|$	Sol.	Gap	Time	K	Sol.	Gap	Time	Model	Sol.	Time	Algo.
Mauritania10e	10	2	**540**	0	0.01	2					**540**	1	TA, MA
Colombia11e	11	2	**620**	0	0.01	2					**620**	1	TA, MA
Angola12e	12	2	**719**	0	0.01	2					**719**	1	TA, MA
Peru13e	13	2	**672**	0	0.02	2					**672**	1	TA, MA
BrasilRJ14e	14	2	**294**	0	0.02	2	**294**	0	0.41	DFJ- G	**294**	1	TA, MA
Libia14e	14	2	**730**	0	0.02	2	**730**	0	0.50	GG-G	**730**	1	TA, MA
Congo15e	15	2	**756**	0	0.02	2	**756**	0	2.50	GLPK	**756**	1	TA, MA
Argentina16e	16	2	**955**	0	0.05	2	**955**	0	0.81	DFJ-G	**955**	1	TA, MA
BrasilRN16e	16	2	**375**	0	0.02	2	**375**	0	31.8	GLPK	**375**	1	TA, MA
EUA17e	17	2	**912**	0	0.02	2	**912**	0	0.85	DFJ-G			
Bolivia10e	10	3	**592**	0	0.03	3							
AfricaSul11e	11	3	**567**	0	0.03	3							
Niger12e	12	3	**743**	0	0.03	3							
Mongolia13e	13	3	**760**	0	0.30	3							
Indonesia14e	14	3	**799**	0	0.05	3	**799**	0	0.36	GG - G			
Argelia15e	15	3	**840**	0	0.28	3	**840**	0	6.95	GG - G			
India16e	16	3	**1035**	0	0.11	3	**1035**	0	1.00	GG - G			
China17e	17	3	**1003**	0	0.39	3	**1003**	0	24.40	DFJ-C			
BrasilPR25e	25	3	**508**	0	0.22	3					**508**	12	TA
BrasilAM26e	26	3	**467**	0	0.38	2	**467**	0	9.44	DFJ-G	**467**	13	TA
att48eA	48	3	**34571**	0	21.39	3	**34571**	0	198.70	GG-G	**34571**	14	TA
berlin52eA	52	3	**8948**	0	55.37	3	**8948**	0	4458.30	GG-G	**8948**	52	TA
Betim100e	100	3	**1394**	0	24.39	3					**1394**	247	TA
PortoVelho200e	200	3	**2274**	0	81.91	3					2312	3165	TA
Egito9e	9	4	**644**	0	0.06	4							
Etiopia10e	10	4	**631**	0	0.03	3							
Mali11e	11	4	**645**	0	0.07	2							
Chade12e	12	4	**840**	0	0.08	3							
Ira13e	13	4	**764**	0	0.12	3							
Mexico14e	14	4	**789**	0	0.09	3	**789**	0	2.46	GG-G			
Sudao15e	15	4	**823**	0	0.06	4	**823**	0	1.55	DFJ-G			
Australia16e	16	4	**1051**	0	0.42	4	**1051**	0	44.14	GG-G			
Canada17e	17	4	**1251**	0	0.16	4	**1251**	0	30.84	GG-C			
BrasilMG30e	30	4	**529**	0	0.58	4	**529**	0	72.13	DFJ-G	**529**	26	TA
BrasilRS32e	32	4	**491**	0	1.46	2					**491**	24	TA
BrasilSP32e	32	4	**588**	0	8.74	4					**588**	27	TA
st70eB	70	4	**1706**	2.05	10000.00	4					1777	310	TA
eil76eB	76	4	**1638**	0	3650.13	4					1703	420	TA
rd100eB	100	4	10062	4.44	10000.10	3					**9909**	255	TA
Cuiaba200e	200	4	**2214**	0	95.63	4					2275	3039	TA
Belem300e	300	4	**2963**	0	683.83	4					2985	9649	TA
Arabia14e	14	5	**851**	0	0.06	5	**851**	0	1.93	DFJ-G			
Cazaquistao15e	15	5	**904**	0	0.11	5	**904**	0	14.4	GG-C			
Brasil16e	16	5	**1136**	0	0.46	5							
Russia17e	17	5	**1061**	0	0.40	4	**1061**	0	51.20	DFJ-C			
BrasilCO40e	40	5	**668**	0	2.03	5					**668**	51	TA
BrasilNO45e	45	5	**829**	0	2.47	5					**829**	30	TA
BrasilNE50e	50	5	**756**	0	1.57	5					**756**	35	TA
rat99eB	99	5	4157	35.65	10000.10	5					**3042**	309	TA
Vitoria100e	100	5	**1354**	0	15.69	5					**1354**	292	TA

Table 1: The CARS formulation tested on the Euclidean instances and compared with other mathematical formulations and heuristic approaches of the literature.

			CARS				Exact Approaches				Heuristic Approaches		
Instance	$|V|$	$|K|$	Sol.	Gap	Time	K	Sol.	Gap	Time	Model	Sol.	Time	Algo.
Mauritania10n	10	2	**571**	0	0.01	2					**571**	1	TA, MA
Colombia11n	11	2	**639**	0	0.02	2					**639**	1	TA, MA
Angola12n	12	2	**656**	0	0.01	2					**656**	1	TA, MA
Peru13n	13	2	**693**	0	0.01	2					**693**	1	TA, MA
BrasilRJ14n	14	2	**167**	0	0.02	2	**167**	0	0.53	DFJ-G	**167**	0	MA
Libia14n	14	2	**760**	0	0.02	2	**760**	0	0.39	DFJ-C	**760**	1	TA, MA
Congo15n	15	2	**886**	0	0.02	2					**886**	1	TA, MA
Argentina16n	16	2	**894**	0	0.02	2	**894**	0	0.32	DFJ-C	**894**	1	TA, MA
BrasilRN16n	16	2	**188**	0	0.02	2					**188**	0.54	ScA, ScA+ALSP
EUA17n	17	2	**822**	0	0.02	2	**822**	**0**	0.88	DFJ-G			
Bolivia10n	10	3	**681**	0	0.03	3							
AfricaSul11n	11	3	**714**	0	0.06	3							
Niger12n	12	3	**869**	0	0.09	3							
Mongolia13n	13	3	**740**	0	0.06	3							
Indonesia14n	14	3	**796**	0	0.08	3	**796**	0	1.48	DFJ-G			
Argelia15n	15	3	**863**	0	0.12	3	**863**	0	6.53	DFJ-G			
India16n	16	3	**985**	0	0.06	3	**985**	0	1.69	DFJ-G			
China17n	17	3	**918**	0	0.10	3	**918**	0	11.99	DFJ-G			
BrasilPR25n	25	3	**226**	0	0.26	3					226	9.54	ScA, ScA+ALSP
BrasilAM26n	26	3	**202**	0	0.18	2	**202**	0	6.17	DFJ-G	202	3.50	ScA, ScA+ALSP
att48nA	48	3	**987**	0	2.12	3	**987**	0	9695	DFJ-G	988	89.3	EALSP
berlin52nA	52	3	**1303**	0	2.03	3	1303	1.3	10000	DFJ-G	**1303**	120.6	EALSP
Londrina100n	100	3	**1146**	0	35.21	3					**1146**	738.0	EALSP
kroB150n	150	3	**2839**	0	103.63	3					2845	1200.0	EALSP
PortoVelho160n	160	3	**1382**	0	43.50	3							
Aracaju200n	200	3	**1836**	0	236.72	3					1839	2400.0	EALSP
Egito9n	9	4	**610**	0	0.10	3							
Etiopia10n	10	4	**666**	0	0.03	3							
Mali11n	11	4	**777**	0	0.07	2							
Chade12n	12	4	**930**	0	0.11	3							
Ira13n	13	4	**909**	0	0.11	4							
Mexico14n	14	4	**902**	0	0.20	3	**902**	0	58.93	DFJ-G			
Sudao15n	15	4	**1020**	0	0.30	4	**1020**	0	19.48	DFJ-G			
Australia16n	16	4	**1061**	0	0.07	3	**1061**	0	26.6	DFJ-G			
Canada17n	17	4	**1136**	0	0.15	4	**1136**	0	12.42	DFJ-G			
Canoas30n	30	4	**376**	0	0.43	3					**376**	24.6	TA
BrasilMG30n	30	4	**271**	0	0.54	4	**271**	0	652.54	DFJ-G	**271**	19.17	ScA, ScA+ALSP
BrasilRS32n	32	4	**269**	0	3.49	4					269	23.94	ScA, ScA+ALSP
BrasilSP32n	32	4	**254**	0	0.77	4					**254**	23.40	ScA, ScA+ALSP
st70nB	70	4	**879**	0	10.16	4					**879**	276.6	EALSP
w100nB	100	4	**1615**	0	73.80	4					**1615**	777.3	EALSP
rd100nB	100	4	**1356**	**0**	69.47	4					1357	834.0	EALSP
Osasco100n	100	4	**964**	0	18.12	4					**964**	666.0	EALSP
Cuiaba140n	140	4	**1292**	0	129.19	4					1505	441	TA
d198n	198	4	**3024**	0	503.75	4					3036	2400.0	EALSP
Arabia14n	14	5	**1026**	0	0.10	4	**1026**	0	29.38	DFJ-G			
Cazaquistao15n	15	5	**1043**	0	0.07	3	**1043**	0	76.13	DFJ-C			
Brasil16n	16	5	**1164**	0	0.29	4							
Russia17n	17	5	**1094**	0	2.94	4	**1094**	0	116.77	DFJ-G			
BrasilCO40n	40	5	**574**	0	3.29	5					575	50.4	ScA, ScA+ALSP
BrasilNO45n	45	5	**539**	0	2.26	5					541	62.37	ScA+ALSP
Santos50n	50	5	**382**	0	7.06	5					382	102.0	EALSP
BrasilNE50n	50	5	**608**	0	3.03	5					611	98.0	EALSP
Macapa80n	80	5	**596**	0	61.94	5					599	416.0	EALSP
rat99nB	99	5	**1336**	0	86.42	5					1349	873.3	EALSP
pr107n	107	5	**1627**	0	95.98	5					1631	1054.3	EALSP
ch130n	130	5	**1630**	0	75.91	5					1632	1200.0	EALSP
Teresina200n	200	5	**1337**	0	606.51	5					1343	2400.0	EALSP
Curitiba300n	300	5	NoInt	NoInt	10000.80						**2100**	3600.0	EALSP

Table 2: The CARS formulation tested on the non-Euclidean instances and compared with other mathematical formulations and heuristic approaches of the literature. NoInt: no feasible solution was found within the time-limit.

| | | | QCARS | | | | Goldbarg et al. (2016) | | | | | |
| | | | | | | | MILP | | | | Heuristic | |
Instance	$|V|$	$|K|$	Sol.	Gap	Time	Col.	Sol.	Gap	Time	Model	Sol.	Time
Mauritania10n	10	2	**306**	0.00	0.04	1						
Colombia11n	11	2	**461**	0.00	0.07	2						
Angola12n	12	2	**409**	0.00	0.05	1						
Peru13n	13	2	**502**	0.00	0.11	2						
Libia14n	14	2	**504**	0.00	0.16	1						
BrasilRJ14n	14	2	**101**	0.00	0.14	2						
Congo15n	15	2	**573**	0.00	0.15	2						
Argentina16n	16	2	**642**	0.00	0.24	2						
EUA17n	17	2	**579**	0.00	0.19	2						
Bolivia10n	10	3	**448**	0.00	0.19	2	**448**	0.00	19	RLT	**448**	2.5
AfricaSul11n	11	3	**537**	0.00	0.34	3	**537**	0.00	4999	RLT	**537**	3.5
Niger12n	12	3	**607**	0.00	0.29	2	**607**	0.00	804	RLT	**607**	6.5
Mongolia13n	13	3	**551**	0.00	0.21	2	**551**	2.00	80000	Std	**551**	7.1
Indonesia14n	14	3	**522**	0.00	0.43	2	**522**	0.00	11616	RLT	**522**	5.9
Argelia15n	15	3	**616**	0.00	0.34	2	619	17.30	48859	Std	**616**	9.2
India16n	16	3	**723**	0.00	0.73	2	727	23.30	80000	RLT	**723**	9.4
China17n	17	3	**638**	0.00	0.61	3	651	20.10	62162	Std	**638**	8.3
BrasilAM26n	26	3	**107**	0.00	0.36	2	**107**	17.80	80000	Std	**107**	27.3
att48nA	48	3	**524**	0.00	7.88	3	579	24.00	80000	RLT	526	270.2
Egito9n	9	4	**520**	0.00	0.09	3						
Etiopia10n	10	4	**403**	0.00	0.10	1	**403**	0.00	73	RLT	**403**	3.6
Mali11n	11	4	**494**	0.00	0.07	1	**494**	0.00	262	Std	**494**	5.9
Chade12n	12	4	**649**	0.00	0.29	2	654	10.40	69111	Std	**649**	6.1
Ira13n	13	4	**693**	0.00	0.95	3	697	20.90	54936	Std	625[I]	5.4
Mexico14n	14	4	**610**	0.00	0.51	2	620	18.70	61358	Std	**610**	5.5
Sudao15n	15	4	**769**	0.00	1.21	3	793	29.50	80000	Std	**769**	8.4
Australia16n	16	4	**525**	0.00	0.22	2	**525**	12.60	80000	RLT	**525**	8.5
Canada17n	17	4	**824**	0.00	0.39	3	827	23.80	77542	Std	**824**	13.0
BrasilMG30n	30	4	**160**	0.00	1.78	3	179	29.10	80000	RLT	**160**	36.3
Arabia14n	14	5	**688**	0.00	0.74	3	701	27.80	37832	Std	**688**	8.9
Cazaquistao15n	15	5	**830**	0.00	0.83	3	843	28.80	67421	Std	**830**	9.3
Brasil16n	16	5	**742**	0.00	1.52	2	743	26.90	32052	RLT	**742**	7.8
Russia17n	17	5	**778**	0.00	3.33	3	820	37.60	52468	Std	**778**	9.8

Table 3: The QCARS formulation compared on the non-Euclidean instances with the mathematical formulations and heuristic approach of Goldbarg et al. (2016). I: incomparable solutions (probably different instances with the same names).

is difficult to compare our running times with those of the literature since the algorithms have been executed on different environments. In particular, the mathematical solver used in (Goldbarg et al., 2016) is not competitive with the state of the art.

It is worth notice that the percentage of used colors by the QCARS solutions is around 55% of the available ones, which is less than for the CARS instances. This result is probably due to the fact that not all the nodes are chosen in the optimal solution.

4.2.3 Further Analysis of the Results

In Tables 5 and 6 we evaluate the impact of inequalities (11), (12), and (13) in our formulations for the CARS and the QCARS, respectively. We have solved all the instances of the CARS and the QCARS with every combination of these families of inequalities.

The first column of Tables 5 and 6 indicates whether the results are reported for Euclidean instances (first eight lines marked with "E") or non-Euclidean instances (last eight lines marked

| | | | QCARS | | | | Goldbarg et al. (2016) | | | | | |
| | | | | | | | MILP | | | | Heuristic | |
Instance	$|V|$	$|K|$	Sol.	Gap	Time	Col.	Sol.	Gap	Time	Model	Sol.	Time
Mauritania10e	10	2	**422**	0.00	0.07	2						
Colombia11e	11	2	**326**	0.00	0.06	1						
Angola12e	12	2	**490**	0.00	0.11	1						
Peru13e	13	2	**556**	0.00	0.34	2						
Libia14e	14	2	**521**	0.00	0.11	2						
BrasilRJ14e	14	2	**230**	0.00	0.43	2						
Congo15e	15	2	**513**	0.00	0.10	2						
BrasilRN16e	16	2	**289**	0.00	0.29	2						
Argentina16e	16	2	**719**	0.00	0.43	2						
EUA17e	17	2	**602**	0.00	0.15	2						
Bolivia10e	10	3	**384**	0.00	0.06	3	**384**	0.00	1	RLT	**384**	3.2
AfricaSul11e	11	3	**402**	0.00	0.14	1	**402**	0.00	7	RLT	**402**	4.1
Niger12e	12	3	**564**	0.00	0.31	3	**564**	0.00	35	RLT	**564**	5.7
Mongolia13e	13	3	**543**	0.00	0.92	2	**543**	0.00	550	RLT	**543**	1
Indonesia14e	14	3	**504**	0.00	0.11	3	**504**	0.00	22	Std	**504**	6.5
Argelia15e	15	3	**487**	0.00	0.36	3	**487**	0.00	351	Std	**487**	6.3
India16e	16	3	**705**	0.00	0.18	2	**705**	0.00	30	RLT	**705**	8.4
China17e	17	3	**728**	0.00	0.76	2	735	3.70	80000	RLT	**728**	11.8
BrasilPR25e	25	3	**328**	0.00	4.06	1						
BrasilAM26e	26	3	**338**	0.00	2.98	1	371	20.50	80000	Std	**338**	32.5
att48eA	48	3	**19672**	0.00	182.03	3	25234	38.12	80000	Std	19676	207
Etiopia10e	10	4	**283**	0.00	0.07	1	**283**	0.00	2	Std	**283**	2.9
Mali11e	11	4	**428**	0.00	0.16	1	**428**	0.00	8	RLT	**428**	4.7
Chade12e	12	4	**655**	0.00	0.39	3	**655**	0.00	861	Std	**655**	7.6
Ira13e	13	4	**532**	0.00	0.33	4	**532**	0.00	40	RLT	**532**	7.9
Mexico14e	14	4	**492**	0.00	0.49	2	**492**	0.00	144	Std	**492**	5.3
Sudao15e	15	4	**422**	0.00	0.21	2	**422**	0.00	33	RLT	**422**	4.4
Australia16e	16	4	**682**	0.00	0.18	4	**682**	0.00	453	Std	**682**	12.9
Canada17e	17	4	**783**	0.00	0.69	3	**783**	0.00	1599	Std	**783**	12.9
BrasilMG30e	30	4	**368**	0.00	8.81	1	412	31.30	80000	RLT	**368**	47.6
Arabia14e	14	5	**482**	0.00	0.27	4	**482**	0.00	48	RLT	**482**	4.8
Cazaquistao15e	15	5	**574**	0.00	0.57	5	**574**	0.00	823	RLT	**574**	9.32
Brasil16e	16	5	**619**	0.00	0.49	4	**619**	0.00	718	Std	637	7.9
Russia17e	17	5	**750**	0.00	1.78	3	760	10.40	80000	Std	**750**	12.3

Table 4: The QCARS formulation compared on the Euclidean instances with the mathematical formulations and heuristic approach of Goldbarg et al. (2016).

with "NE"). Thus, each line of the table shows the average results over the 50 instances for the CARS and over 34 for the QCARS. The three following columns labeled with (11), (12), and (13) are check-marked if the corresponding family of inequalities is considered in the model. Columns entitled "Gap %", "Nodes", and "Time" (in seconds) are the average values of those retrieved at the end of the branch-and-cut algorithm of the integer linear solver Gurobi. The "Lazy callbacks" column reports the average number of subtour elimination constraints added to the model. The "Root" column reports the average value of the continuous relaxations without subtour elimination constraints. The column "Int. S. %", gives the percentage of instances for which a feasible solution could be found by the solver. Note that the instances for which the solver could not find an integer solution within the time limit are not included in the computation of the gap. The last column indicates the percentage of instances solved to optimality within the time limit. Bold values are the best ones.

The results reported in Table 5 confirm the quality of the model without valid inequalities presented in Section 3.1 which correspond to the rows of the table without check-marks. Indeed, we are able to solve to optimality 88% of the instances, and the running time is around 1200 seconds for the Euclidean instances and 3000 seconds for the Non-Euclidean. However, when inequalities (11), (12), and (13) are added for strengthening the model, the number of instances solved to optimality increases up to 98 % and the computational time is almost divided by two for the Euclidean instances and by 10 for the Non-Euclidean. These results show, experimentally, the usefulness of these strengthening inequalities. More precisely, inequalities (11) have the most significant impact, since alone, they are able to divide the computational time by at least 1.7 for the Euclidean instances and around 10 times for the non-Euclidean instances. Similarly, the number of nodes of the branch-and-cut tree is divided by 2 and by more than 300, respectively. Note that for the Euclidean instances considering inequalities (11) and (13) to strengthen the model is the best configuration since the computational time is the smallest, the number of instances solved to optimality is the greatest and last, but not least, the solver is able to find at least a feasible solution for all instances. The best configuration for the non-Euclidean instances is when the three families of inequalities are added. Even if there is one instance in this configuration for which no feasible solution is found, the running time is better than in any other configuration as well as the linear relaxation. Remark that if inequalities (11) have the most significant impact, it is not this family that increases the most the value of the linear relaxation. Indeed, inequalities (12) and (13) are the families impacting the most the value of the linear relaxation for the Euclidean and non-Euclidean instances, respectively. Finally, note that the number of added lazy violated subtour inequalities behaves similarly to the number of nodes in the branch-and-cut tree. Without the three families of valid inequalities, several instances could not be solved. There are ten instances where the optimum cannot be found without the three families of valid inequalities in 10000 seconds. When the valid inequalities enforce the model, nine out of ten are solved to optimality in less than 600 seconds. For example, the Euclidean instance Belem300e reaches the time limit without having found a feasible solution when solved with the model without the strengthening inequalities, but it is solved to optimality in ten minutes with them.

The results reported in Table 6 confirm that the model reported in Section 3.2 is useful for solving the QCARS, since it is able to solve to optimality 58 instances over 68 with an average running time of 3000 seconds. Moreover, when inequalities (11), (12), and (13) are added for strengthening

	(11)	(12)	(13)	Gap %	Nodes	Time (s)	Lazy	Root	Int.S. %	Opt. %
				2.85	107412.75	1197.55	288.57	1768.14	98	88
	✓			0.87	49389.90	720.07	61.08	1780.89	**100**	**94**
		✓		2.65	62406.98	1188.95	214.75	1815.03	98	88
E			✓	2.88	87086.37	1087.69	325.24	1769.99	98	90
	✓	✓		2.06	22033.63	669.74	62.12	1824.37	98	**94**
	✓		✓	**0.61**	42084.73	**641.45**	57.29	1784.92	**100**	**94**
		✓	✓	2.59	66807.04	1190.99	240.16	1816.35	98	88
	✓	✓	✓	0.83	**19491.90**	679.54	**52.94**	**1827.25**	**100**	**94**
				2.48	185144.19	2806.15	3328.72	864.98	**98**	72
	✓			3.45	2419.93	383.95	34.66	886.43	97	97
		✓		4.27	170367.47	2863.52	2505.97	869.98	97	72
NE			✓	5.75	170698.57	2788.13	3525.48	869.14	95	72
	✓	✓		**1.72**	1541.10	299.07	53.26	890.96	**98**	**98**
	✓		✓	**1.72**	763.76	213.58	32.05	897.31	**98**	**98**
		✓	✓	6.01	167289.74	2852.50	2537.47	873.90	95	72
	✓	✓	✓	**1.72**	**542.69**	**209.91**	**30.17**	**901.21**	**98**	**98**

Table 5: Performance of the CARS integer linear model enhanced by inequalities (11), (12), and (13).

the model, the number of instances solved to optimality increases up 100%, the computational time is divided by 500, and the number of nodes by 350. These results show, experimentally, the usefulness of these strengthening inequalities. The results reported in Table 6 confirms the importance of valid inequality (11) also for the QCARS instances. Indeed the computational time for non-Euclidean instances is four orders of magnitudes higher when inequalities (11) are not included. Notice that for the Euclidean instances, the combination of (11), (13) and (12) is the most effective, while for non-Euclidean instances the most effective combination is of (11) with (13).

5   Conclusions

In this paper, we have proposed two integer linear programming formulations for the CARS and the QCARS. We have reinforced these formulations using a polynomial number of inequalities and have devised branch-and-cut algorithms for solving these formulations. The experimental results show the effectiveness of our approach to solve these problems. We were able to solve to optimality all the instances found in the literature but four, and our algorithms clearly outperform the existing approaches. Moreover, the experimental results also show the usefulness of reinforcing the models. The efficiency of our model and its strengthening inequalities deeply relies on the assumption that there exists at most one path of each color in a solution. The variant without this assumption seems far more difficult. However, it is more realistic from an application point of view, especially in the rapid transit network design field, and deserves to be studied. A starting point could be to evaluate the quality of the formulation we propose for this variant.

References

David L. Applegate, Robert E. Bixby, Vaek Chvatl, and William J. Cook. *The Traveling Salesman Problem: A Computational Study.* Princeton University Press, 2006.

	(11)	(12)	(13)	Gap %	Nodes	Time (s)	Lazy	Root	Int.S. %	Opt. %
				0.30	30910.59	355.29	4165.94	842.23	**100**	97
	✓			**0.00**	922.18	9.50	112.03	861.44	**100**	**100**
		✓		0.30	31755.62	379.58	4453.24	910.39	**100**	97
E			✓	0.28	34414.32	363.37	4241.88	842.23	**100**	97
	✓	✓		**0.00**	810.85	14.61	139.18	925.03	**100**	**100**
	✓		✓	**0.00**	773.21	10.16	111.35	864.56	**100**	**100**
		✓	✓	0.28	33258.53	387.01	4309.26	910.77	**100**	97
	✓	✓	✓	**0.00**	**449.82**	**6.13**	111.24	**926.87**	**100**	**100**
				3.57	153930.67	2697.65	79475.15	357.61	**100**	76
	✓			**0.00**	208.42	0.72	40.42	412.51	**100**	**100**
		✓		3.63	134807.42	2695.66	81716.03	366.31	**100**	76
NE			✓	3.45	142703.06	2572.01	84935.33	357.61	**100**	76
	✓	✓		**0.00**	197.39	0.79	**39.24**	417.78	**100**	**100**
	✓		✓	**0.00**	**116.85**	**0.50**	46.03	421.82	**100**	**100**
		✓	✓	3.64	139784.91	2725.89	83398.39	366.31	**100**	76
	✓	✓	✓	**0.00**	132.52	0.74	44.06	**426.17**	**100**	**100**

Table 6: Performance of the QCARS integer linear model enhanced by inequalities (11), (12), and (13).

P. H. S. Asconavieta, M. C. Goldbarg, and E. F. G. Goldbarg. Evolutionary algorithm for the car renter salesman. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 593–600, June 2011.

André Renato Villela da Silva and Luiz Satoru Ochi. An efficient hybrid algorithm for the traveling car renter problem. *Expert Systems with Applications*, 64:132–140, 2016.

M. da Silva Menezes, M. C. Goldbarg, and E. F. G. Goldbarg. A memetic algorithm for the prize-collecting traveling car renter problem. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 3258–3265, July 2014.

D. Felipe, E. F. G. Goldbarg, and M. C. Goldbarg. Scientific algorithms for the car renter salesman problem. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 873–879, July 2014.

Marco C. Goldbarg, Elizabeth F.G. Goldbarg, Paulo H. Asconavieta, Matheus da S. Menezes, and Henrique P.L. Luna. A transgenetic algorithm applied to the traveling car renter problem. *Expert Systems with Applications*, 40(16):6298 – 6310, 2013.

Marco C. Goldbarg, Elizabeth F.G. Goldbarg, Matheus da S. Menezes, and Henrique P.L. Luna. Quota traveling car renter problem: Model and evolutionary algorithm. *Information Sciences*, 367-368:232 – 245, 2016.

Marco C Goldbarg, Elizabeth FG Goldbarg, Henrique PL Luna, Matheus S Menezes, and Lucas Corrales. Integer programming models and linearizations for the traveling car renter problem. *Optimization Letters*, 12(4):743–761, 2018.

Marco César Goldbarg, Paulo Henrique Asconavieta, and Elizabeth Ferreira Gouvêa Goldbarg. Memetic algorithm for the traveling car renter problem: an experimental investigation. *Memetic Computing*, 4(2):89–108, Jun 2012.

Hassan Ismkhan. Effective three-phase evolutionary algorithm to handle the large-scale colorful traveling salesman problem. *Expert Syst. Appl.*, 67(C):148–162, January 2017.

G. Laporte, J.A. Mesa, F.A. Ortega, and F. Perea. Planning rapid transit networks. *Socio-Economic Planning Sciences*, 45(3):95 – 104, 2011.

Lehilton L. C. Pedrosa, Greis Y. O. Quesquén, and Rafael C. S. Schouery. An Asymptotically Optimal Approximation Algorithm for the Travelling Car Renter Problem. In Valentina Cacchiani and Alberto Marchetti-Spaccamela, editors, *19th Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2019)*, volume 75 of *OpenAccess Series in Informatics (OASIcs)*, pages 14:1–14:15, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

B. H. O. Rios, E. F. G. Goldbarg, and G. Y. O. Quesquén. A hybrid metaheuristic using a corrected formulation for the traveling car renter salesman problem. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 2308–2314, June 2017.

Thomas A. Roemer, Reza Ahmadi, and Sriram Dasu. The traveling salesman problem with flexible coloring. *Discrete Applied Mathematics*, 160(12):1798 – 1814, 2012.

L.A. Wolsey. *Integer Programming*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 1998.

Yupei Xiong, Bruce Golden, and Edward Wasil. *The Colorful Traveling Salesman Problem*, pages 115–123. Springer US, Boston, MA, 2007.