



# Efficient formulations for the traveling car renter problem and its quota variant

Mathieu Lacroix<sup>1</sup> · Yasmín A. Ríos-Solís<sup>2</sup> · Roberto Wolfler Calvo<sup>1,3</sup> 

Received: 25 April 2020 / Accepted: 5 January 2021 / Published online: 22 January 2021

© The Author(s), under exclusive licence to Springer-Verlag GmbH, DE part of Springer Nature 2021

## 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, 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.

**Keywords** Vehicle routing · Traveling salesman problem · Integer linear programming formulation · Branch-and-cut

---

Yasmín A. Ríos-Solís carried out part of this research during an academic stay at LIPN, CNRS, UMR 7030, Université Sorbonne Paris Nord, 93430, Villetaneuse, France. She also wishes to acknowledge grants 710289 and FC-2016/1948 from CONACYT, Mexico.

---

✉ Roberto Wolfler Calvo  
wolfler@lipn.fr

<sup>1</sup> LIPN, CNRS, UMR 7030, Université Sorbonne Paris Nord, 93430 Villetaneuse, France

<sup>2</sup> Escuela de Ingeniería y Ciencias, Tecnológico de Monterrey, Monterrey, Mexico

<sup>3</sup> Department of Mathematics and Computer Science, University of Cagliari, Cagliari, Italy

## 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  $V = \{1, \dots, n\}$  indicates a set of nodes and  $ij \in E$  is an edge between nodes  $i$  and  $j$ . A set  $K = \{1, \dots, o\}$  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. [9], and it is directed to the viewpoint of vehicle rental customers. In general, a user of rented vehicles aims at traveling a specific route 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 [4] 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 deservng a certain amount of population [11]. 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 traveling salesman problem (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 [1], 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 Goldberg et al. [9], 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 [4], 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 [7] and in [3]. As pointed out by Rios et al. [13], the formulation in [3] 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 [6]. In [8], 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 [1] 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. [12] 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. Goldberg et al. [9] devise two greedy randomized adaptive search procedures with variable neighborhood descent. Several evolutionary algorithms have also been developed: memetic

algorithms [4,9], transgenetic algorithms [2,6] and scientific algorithm [5]. Two hybrid algorithms have also been proposed [3,13].

The QCARS has also been considered in the literature. Two integer linear programming formulations for the QCARS are presented in [4,7]. 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. [4] and Goldbarg et al. [7] 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* [10,16], 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* [14]. 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.

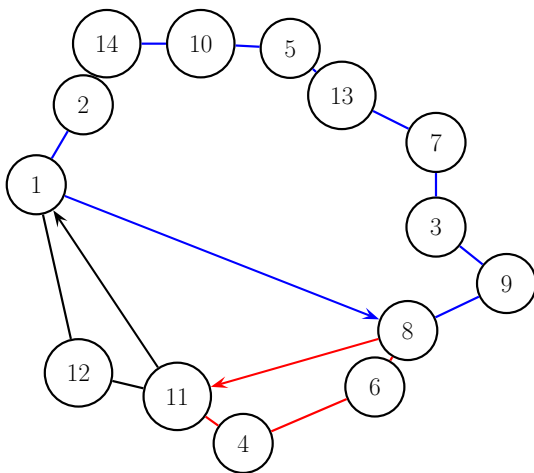
To compare our approach with those in the literature (except the one proposed by da Silva and Ochi [3]), we take into account the same assumption: node one must be the extremity of a path. This assumption stems from the fact that a tourist arrives at a specific city (node one) and starts its trip by 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. Note that if the solution contains only one color, then we do not have an orientation of the associated circuit. 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.

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

**Fig. 1** An example of a solution of an instance of 14 nodes which uses three colors: red, blue and black (color figure online)



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  $F \subseteq E$ .

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

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

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

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

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

$$x^k(\delta(1)) = z_1^k + w^k \quad k \in K \quad (6)$$

$$\sum_{k \in K} \sum_{j \in V \setminus \{i\}} y_{i,j}^k - \sum_{k \in K} \sum_{j \in V \setminus \{i\}} y_{j,i}^k = 0 \quad i \in V \quad (7)$$

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

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 is 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 [15]:

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

### 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 Sect. 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 Eq. (4) associated with node 1 and Eq. (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 Sect. 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 Sect. 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 check 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<sup>1</sup>.

**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 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  $i \in V$ ,  $\ell \in K \setminus \{k\}$ ).

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 adding the following inequalities:

<sup>1</sup> 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.



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

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 Sect. 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 Sect. 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 10,000 s, as proposed by Goldberg et al. [8].

#### 4.1 Instances

**CARS** We use the benchmark proposed in Goldberg et al. [8], 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  $\omega_{ij}^k$  is a random value between [1.4, 2]. In all instances, the asymmetric cost matrices  $c^k$  are generated as

follows. A positive integer  $\alpha_i^k$  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 Goldberg et al. [7]. 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, 2 and 4, we compare the results we obtain with our algorithms with the best exact and heuristic approaches of the literature<sup>2</sup>. 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”<sup>3</sup> 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.

### 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 [8] 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 Goldberg et al. [6] 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

<sup>2</sup> The comparisons about running times in Tables 1, 2 and 4 give an order of magnitude, but it is not precise since the algorithms run on different computers.

<sup>3</sup> The gap is computed as  $100(\text{best solution found} - \text{best lower bound})/\text{best lower bound}$ .

hybridized with an adaptive local search procedure presented in [3] 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. [13] 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 3 and 4 we compare our QCARS model with the exact formulation and the evolutionary algorithm proposed in [7]. 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 3 summarizes the results obtained for the QCARS on the Euclidean instances. All instances except one are solved in less than 9 s. The latter one is solved in 182 s, but it seems to be a hard instance since the model proposed in Goldberg et al. [7] finds a gap of 38.12% after 80,000 s while for all the others the gap is less than 0.3%. Table 4 summarizes the results obtained for the QCARS on the non-Euclidean instances. All instances are solved in less than 4 s except one which is solved in less than 8 s.

The proposed algorithm is really more efficient than the exact approach proposed in [7]. Some instances that were unsolved in 80,000 s are now solved within few seconds. Moreover, our algorithm is faster than their heuristic approach. However, note that it 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 [7] 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.

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

Instance	V	K	CARS			Exact approaches			Heuristic approaches				
			Sol.	Gap	Time	K	Sol.	Gap	Time	Model	Sol.	Time	Algo.
Mauritania10e	10	2	<b>540</b>	0	0.01	2					<b>540</b>	1	TA, MA
Colombia11e	11	2	<b>620</b>	0	0.01	2					<b>620</b>	1	TA, MA
Angola12e	12	2	<b>719</b>	0	0.01	2					<b>719</b>	1	TA, MA
Peru13e	13	2	<b>672</b>	0	0.02	2					<b>672</b>	1	TA, MA
BrasilRJ14e	14	2	<b>294</b>	0	0.02	2	<b>294</b>	0	0.41	DFJ-G	<b>294</b>	1	TA, MA
Libia14e	14	2	<b>730</b>	0	0.02	2	<b>730</b>	0	0.50	GG-G	<b>730</b>	1	TA, MA
Congo15e	15	2	<b>756</b>	0	0.02	2	<b>756</b>	0	2.50	GLPK	<b>756</b>	1	TA, MA
Argentina16e	16	2	<b>955</b>	0	0.05	2	<b>955</b>	0	0.81	DFJ-G	<b>955</b>	1	TA, MA
BrasilRN16e	16	2	<b>375</b>	0	0.02	2	<b>375</b>	0	31.8	GLPK	<b>375</b>	1	TA, MA
EUA17e	17	2	<b>912</b>	0	0.02	2	<b>912</b>	0	0.85	DFJ-G			
Bolivia10e	10	3	<b>592</b>	0	0.03	3							
AfricaSul11e	11	3	<b>567</b>	0	0.03	3							
Niger12e	12	3	<b>743</b>	0	0.03	3							
Mongolia13e	13	3	<b>760</b>	0	0.30	3							
Indonesia14e	14	3	<b>799</b>	0	0.05	3	<b>799</b>	0	0.36	GG-G	<b>799</b>		GG-G
Argelia15e	15	3	<b>840</b>	0	0.28	3	<b>840</b>	0	6.95	GG-G	<b>840</b>		GG-G
India16e	16	3	<b>1035</b>	0	0.11	3	<b>1035</b>	0	1.00	GG-G	<b>1035</b>		GG-G
China17e	17	3	<b>1003</b>	0	0.39	3	<b>1003</b>	0	24.40	DFJ-C	<b>1003</b>		DFJ-C
BrasilPR25e	25	3	<b>508</b>	0	0.22	3					<b>508</b>	12	TA

Table 1 continued

Instance	V	K	CARS			Exact approaches			Heuristic approaches				
			Sol.	Gap	Time	K	Sol.	Gap	Time	Model	Sol.	Time	Algo.
BrasiIAM26e	26	3	467	0	0.38	2	467	0	9.44	DFJ-G	467	13	TA
att48eA	48	3	34,571	0	21.39	3	34,571	0	198.70	GG-G	34,571	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		26	TA
Sudaol5e	15	4	823	0	0.06	4	823	0	1.55	DFJ-G	491	24	TA
Australia16e	16	4	1051	0	0.42	4	1051	0	44.14	GG-G	588	27	TA
Canada17e	17	4	1251	0	0.16	4	1251	0	30.84	GG-C	1777	310	TA
BrasiIMG30e	30	4	529	0	0.58	4	529	0	72.13	DFJ-G	529	26	TA
BrasiIRS32e	32	4	491	0	1.46	2					491	24	TA
BrasiISP32e	32	4	588	0	8.74	4					588	27	TA
st70eB	70	4	1706	2.05	10,000.00	4					1777	310	TA
eil76eB	76	4	1638	0	3650.13	4					1703	420	TA
rd100eB	100	4	10,062	4.44	10,000.10	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			

Table 1 continued

Instance	V	K	CARS			Exact approaches			Heuristic approaches				
			Sol.	Gap	Time	K	Sol.	Gap	Time	Model	Sol.	Time	Algo.
Cazaquistao15e	15	5	<b>904</b>	0	0.11	5	<b>904</b>	0	14.4	GG-C			
Brasil16e	16	5	<b>1136</b>	0	0.46	5							
Russia17e	17	5	<b>1061</b>	0	0.40	4	<b>1061</b>	0	51.20	DFJ-C			
BrasilCO40e	40	5	<b>668</b>	0	2.03	5					<b>668</b>	51	TA
BrasilNO45e	45	5	<b>829</b>	0	2.47	5					<b>829</b>	30	TA
BrasilNE50e	50	5	<b>756</b>	0	1.57	5					<b>756</b>	35	TA
rat99eB	99	5	4157	35.65	10,000.10	5					<b>3042</b>	309	TA
Vitoria100e	100	5	<b>1354</b>	0	15.69	5					<b>1354</b>	292	TA

**Table 2** The CARS formulation tested on the non-Euclidean instances and compared with other mathematical formulations and heuristic approaches of the literature

Instance	V	K	CARS			Exact approaches			Heuristic approaches				
			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	

Table 2 continued

Instance	V	K	CARS			Exact approaches			Heuristic approaches				
			Sol.	Gap	Time	K	Sol.	Gap	Time	Model	Sol.	Time	Algo.
BrasiAM26n	26	3	<b>202</b>	0	0.18	2	<b>202</b>	0	6.17	DFJ-G	<b>202</b>	3.50	ScA, ScA+ALSP
att48nA	48	3	<b>987</b>	0	2.12	3	<b>987</b>	0	9695	DFJ-G	988	89.3	EALSP
berlin52nA	52	3	<b>1303</b>	0	2.03	3	<b>1303</b>	1.3	10,000	DFJ-G	<b>1303</b>	120.6	EALSP
Londrina100n	100	3	<b>1146</b>	0	35.21	3					<b>1146</b>	738.0	EALSP
krb150n	150	3	<b>2839</b>	0	103.63	3					2845	1200.0	EALSP
PortoVelho160n	160	3	<b>1382</b>	0	43.50	3					1839	2400.0	EALSP
Aracaju200n	200	3	<b>1836</b>	0	236.72	3							
Egito9n	9	4	<b>610</b>	0	0.10	3							
Etiopia10n	10	4	<b>666</b>	0	0.03	3							
Mali11n	11	4	<b>777</b>	0	0.07	2							
Chade12n	12	4	<b>930</b>	0	0.11	3							
Ira13n	13	4	<b>909</b>	0	0.11	4							
Mexico14n	14	4	<b>902</b>	0	0.20	3	<b>902</b>	0	58.93	DFJ-G			
Sudao15n	15	4	<b>1020</b>	0	0.30	4	<b>1020</b>	0	19.48	DFJ-G			
Australia16n	16	4	<b>1061</b>	0	0.07	3	<b>1061</b>	0	26.6	DFJ-G			
Canada17n	17	4	<b>1136</b>	0	0.15	4	<b>1136</b>	0	12.42	DFJ-G			
Canoas30n	30	4	<b>376</b>	0	0.43	3					<b>376</b>	24.6	TA
BrasiMG30n	30	4	<b>271</b>	0	0.54	4	<b>271</b>	0	652.54	DFJ-G	<b>271</b>	19.17	ScA, ScA+ALSP
BrasiRS32n	32	4	<b>269</b>	0	3.49	4					<b>269</b>	23.94	ScA, ScA+ALSP
BrasiSP32n	32	4	<b>254</b>	0	0.77	4					<b>254</b>	23.40	ScA, ScA+ALSP
st70nB	70	4	<b>879</b>	0	10.16	4					<b>879</b>	276.6	EALSP
w100nB	100	4	<b>1615</b>	0	73.80	4					<b>1615</b>	777.3	EALSP



Table 2 continued

Instance	V	K	CARS			Exact approaches			Heuristic approaches			
			Sol.	Gap	Time	K	Sol.	Gap	Time	Model	Sol.	Time
rd100nB	100	4	<b>1356</b>	<b>0</b>	69.47	4				1357	834.0	EALSP
Osasco100n	100	4	<b>964</b>	0	18.12	4				<b>964</b>	666.0	EALSP
Cuiaba140n	140	4	<b>1292</b>	0	129.19	4				1505	441	TA
d198n	198	4	<b>3024</b>	0	503.75	4				3036	2400.0	EALSP
Arabial4n	14	5	<b>1026</b>	0	0.10	4	<b>1026</b>	0	29.38			DFI-G
Cazaquistao15n	15	5	<b>1043</b>	0	0.07	3	<b>1043</b>	0	76.13			DFI-C
Brasil16n	16	5	<b>1164</b>	0	0.29	4						
Russia17n	17	5	<b>1094</b>	0	2.94	4	<b>1094</b>	0	116.77			DFI-G
BrasilCO40n	40	5	<b>574</b>	0	3.29	5				575	50.4	ScA, ScA+ALSP
BrasiliNO45n	45	5	<b>539</b>	0	2.26	5				541	62.37	ScA+ALSP
Santos50n	50	5	<b>382</b>	0	7.06	5				<b>382</b>	102.0	EALSP
BrasiliNE50n	50	5	<b>608</b>	0	3.03	5				611	98.0	EALSP
Macapa80n	80	5	<b>596</b>	0	61.94	5				599	416.0	EALSP
rat99nB	99	5	<b>1336</b>	0	86.42	5				1349	873.3	EALSP
pr107n	107	5	<b>1627</b>	0	95.98	5				1631	1054.3	EALSP
chl130n	130	5	<b>1630</b>	0	75.91	5				1632	1200.0	EALSP
Teresina200n	200	5	<b>1337</b>	0	606.51	5				1343	2400.0	EALSP
Curitiba300n	300	5	NoInt	NoInt	10,000.80	5				<b>2100</b>	3600.0	EALSP

NoInt, no feasible solution was found within the time-limit

**Table 3** The QCARS formulation compared on the Euclidean instances with the mathematical formulations and heuristic approach of Goldberg et al. [7]

Instance	V	K	Goldberg et al. [7]														
			QCARS					MILP					Heuristic				
			Sol.	Gap	Time	Col.	Time	Sol.	Gap	Time	Model	Sol.	Time				
Mauritania10e	10	2	<b>422</b>	0.00	0.07	2											
Colombia11e	11	2	<b>326</b>	0.00	0.06	1											
Angola12e	12	2	<b>490</b>	0.00	0.11	1											
Peru13e	13	2	<b>556</b>	0.00	0.34	2											
Libia14e	14	2	<b>521</b>	0.00	0.11	2											
BrasilRJ14e	14	2	<b>230</b>	0.00	0.43	2											
Congo15e	15	2	<b>513</b>	0.00	0.10	2											
BrasilRN16e	16	2	<b>289</b>	0.00	0.29	2											
Argentina16e	16	2	<b>719</b>	0.00	0.43	2											
EUAI7e	17	2	<b>602</b>	0.00	0.15	2											
Bolivia10e	10	3	<b>384</b>	0.00	0.06	3					<b>384</b>	0.00	1	RLT	<b>384</b>	3.2	
AfricaSul11e	11	3	<b>402</b>	0.00	0.14	1					<b>402</b>	0.00	7	RLT	<b>402</b>	4.1	
Niger12e	12	3	<b>564</b>	0.00	0.31	3					<b>564</b>	0.00	35	RLT	<b>564</b>	5.7	
Mongolia13e	13	3	<b>543</b>	0.00	0.92	2					<b>543</b>	0.00	550	RLT	<b>543</b>	1	
Indonesia14e	14	3	<b>504</b>	0.00	0.11	3					<b>504</b>	0.00	22	Std	<b>504</b>	6.5	
Argelia15e	15	3	<b>487</b>	0.00	0.36	3					<b>487</b>	0.00	351	Std	<b>487</b>	6.3	
India16e	16	3	<b>705</b>	0.00	0.18	2					<b>705</b>	0.00	30	RLT	<b>705</b>	8.4	

Table 3 continued

Instance	V	K	Goldberg et al. [7]						Heuristic			
			QCARS			MILP			Sol.	Time		
			Sol.	Gap	Time	Col.	Sol.	Gap			Time	Model
China17e	17	3	728	0.00	0.76	2	735	3.70	80,000	RLT	728	11.8
BrasiIPR25e	25	3	328	0.00	4.06	1	371	20.50	80,000	Std	338	32.5
BrasiIAM26e	26	3	338	0.00	2.98	1	25,234	38.12	80,000	Std	19,676	207
att48eA	48	3	19,672	0.00	182.03	3	283	0.00	2	Std	283	2.9
Etiopia10e	10	4	283	0.00	0.07	1	428	0.00	8	RLT	428	4.7
Mali11e	11	4	428	0.00	0.16	1	655	0.00	861	Std	655	7.6
Chadel12e	12	4	655	0.00	0.39	3	532	0.00	40	RLT	532	7.9
Ira13e	13	4	532	0.00	0.33	4	492	0.00	144	Std	492	5.3
Mexico14e	14	4	492	0.00	0.49	2	422	0.00	33	RLT	422	4.4
Sudao15e	15	4	422	0.00	0.21	2	682	0.00	453	Std	682	12.9
Australia16e	16	4	682	0.00	0.18	4	783	0.00	1599	Std	783	12.9
Canada17e	17	4	783	0.00	0.69	3	412	31.30	80,000	RLT	368	47.6
BrasiIMC30e	30	4	368	0.00	8.81	1	482	0.00	48	RLT	482	4.8
Arabial4e	14	5	482	0.00	0.27	4	574	0.00	823	RLT	574	9.32
Cazaquistao15e	15	5	574	0.00	0.57	5	619	0.00	718	Std	637	7.9
Brasi16e	16	5	619	0.00	0.49	4	760	10.40	80,000	Std	750	12.3
Russia17e	17	5	750	0.00	1.78	3						

**Table 4** The QCARS formulation compared on the non-Euclidean instances with the mathematical formulations and heuristic approach of Goldberg et al. [7]

Instance	V	K	Goldberg et al. [7]														
			QCARS					MILP					Heuristic				
			Sol.	Gap	Time	Col.	Sol.	Gap	Time	Model	Sol.	Time					
Mauritania10n	10	2	<b>306</b>	0.00	0.04	1											
Colombia11n	11	2	<b>461</b>	0.00	0.07	2											
Angola12n	12	2	<b>409</b>	0.00	0.05	1											
Peru13n	13	2	<b>502</b>	0.00	0.11	2											
Libia14n	14	2	<b>504</b>	0.00	0.16	1											
BrasilRJ14n	14	2	<b>101</b>	0.00	0.14	2											
Congo15n	15	2	<b>573</b>	0.00	0.15	2											
Argentina16n	16	2	<b>642</b>	0.00	0.24	2											
EU17n	17	2	<b>579</b>	0.00	0.19	2											
Bolivia10n	10	3	<b>448</b>	0.00	0.19	2		<b>448</b>	0.00	19	RLT	<b>448</b>	2.5				
AfricaSul11n	11	3	<b>537</b>	0.00	0.34	3		<b>537</b>	0.00	4999	RLT	<b>537</b>	3.5				
Niger12n	12	3	<b>607</b>	0.00	0.29	2		<b>607</b>	0.00	804	RLT	<b>607</b>	6.5				
Mongolia13n	13	3	<b>551</b>	0.00	0.21	2		<b>551</b>	2.00	80,000	Std	<b>551</b>	7.1				
Indonesia14n	14	3	<b>522</b>	0.00	0.43	2		<b>522</b>	0.00	11,616	RLT	<b>522</b>	5.9				
Argelia15n	15	3	<b>616</b>	0.00	0.34	2		619	17.30	48,859	Std	<b>616</b>	9.2				
India16n	16	3	<b>723</b>	0.00	0.73	2		727	23.30	80,000	RLT	<b>723</b>	9.4				
China17n	17	3	<b>638</b>	0.00	0.61	3		651	20.10	62,162	Std	<b>638</b>	8.3				

Table 4 continued

Instance	V	K	Goldberg et al. [7]									
			QCARS			MILP			Heuristic			
			Sol.	Gap	Time	Col.	Sol.	Gap	Time	Model	Sol.	Time
BrasiAM26n	26	3	<b>107</b>	0.00	0.36	2	<b>107</b>	17.80	80,000	Std	<b>107</b>	27.3
att48nA	48	3	<b>524</b>	0.00	7.88	3	579	24.00	80,000	RLT	526	270.2
Egito9n	9	4	<b>520</b>	0.00	0.09	3						
Etiopia10n	10	4	<b>403</b>	0.00	0.10	1	<b>403</b>	0.00	73	RLT	<b>403</b>	3.6
Mali11n	11	4	<b>494</b>	0.00	0.07	1	<b>494</b>	0.00	262	Std	<b>494</b>	5.9
Chade12n	12	4	<b>649</b>	0.00	0.29	2	654	10.40	69,111	Std	<b>649</b>	6.1
Ira13n	13	4	<b>693</b>	0.00	0.95	3	697	20.90	54,936	Std	625 <sup>l</sup>	5.4
Mexico14n	14	4	<b>610</b>	0.00	0.51	2	620	18.70	61,358	Std	<b>610</b>	5.5
Sudao15n	15	4	<b>769</b>	0.00	1.21	3	793	29.50	80,000	Std	<b>769</b>	8.4
Australia16n	16	4	<b>525</b>	0.00	0.22	2	<b>525</b>	12.60	80,000	RLT	<b>525</b>	8.5
Canada17n	17	4	<b>824</b>	0.00	0.39	3	827	23.80	77,542	Std	<b>824</b>	13.0
BrasiMCG30n	30	4	<b>160</b>	0.00	1.78	3	179	29.10	80,000	RLT	<b>160</b>	36.3
Arabia14n	14	5	<b>688</b>	0.00	0.74	3	701	27.80	37,832	Std	<b>688</b>	8.9
Cazaquistao15n	15	5	<b>830</b>	0.00	0.83	3	843	28.80	67,421	Std	<b>830</b>	9.3
Brasi16n	16	5	<b>742</b>	0.00	1.52	2	743	26.90	32,052	RLT	<b>742</b>	7.8
Russia17n	17	5	<b>778</b>	0.00	3.33	3	820	37.60	52,468	Std	<b>778</b>	9.8

l, incomparable solutions (probably different instances with the same names)

### 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 with “NE”). Thus, each line of the table shows the average results over the 50 instances for the CARS and over 34 or 33 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 s) 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 Sect. 3.1 which correspond to the rows of the table without check-marks. Indeed, we are able to solve to optimality 80% of the instances, and the running time is around 1200s for the Euclidean instances and 3000s 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 96 % 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 for the Euclidean instances. Indeed, inequalities (12) is the family impacting the most the value of the linear relaxation for these instances. Finally, note that the number of added lazy violated subtour inequalities behaves similarly to the number of nodes

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

	(11)	(12)	(13)	Gap %	Nodes	Time (s)	Lazy	Root	Int.S. %	Opt. %
E	✓			2.85	107,412.75	1197.55	288.57	1768.14	98	88
		✓		0.87	49,389.90	720.07	61.08	1780.89	100	94
			✓	2.65	62,406.98	1188.95	214.75	1815.03	98	88
	✓	✓		2.88	87,086.37	1087.69	325.24	1769.99	98	90
	✓		✓	2.06	22,033.63	669.74	62.12	1824.37	98	94
	✓		✓	<b>0.61</b>	42,084.73	<b>641.45</b>	57.29	1784.92	100	94
		✓	✓	2.59	66,807.04	1190.99	240.16	1816.35	98	88
	✓	✓	✓	0.83	<b>19,491.90</b>	679.54	<b>52.94</b>	<b>1827.25</b>	100	94
	✓			2.48	185,144.19	2806.15	3328.72	864.98	98	72
		✓		3.45	2419.93	383.95	34.66	886.43	97	97
NE		✓		4.27	170,367.47	2863.52	2505.97	869.98	97	72
	✓		✓	5.75	170,698.57	2788.13	3525.48	869.14	95	72
	✓	✓		<b>1.72</b>	1541.10	299.07	53.26	890.96	98	98
	✓		✓	<b>1.72</b>	763.76	213.58	32.05	897.31	98	98
		✓	✓	6.01	167,289.74	2852.50	2537.47	873.90	95	72
	✓	✓	✓	<b>1.72</b>	<b>542.69</b>	<b>209.91</b>	<b>30.17</b>	<b>901.21</b>	98	98

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

	(11)	(12)	(13)	Gap %	Nodes	Time (s)	Lazy	Root	Int.S. %	Opt. %
E	✓			0.30	30,910.59	355.29	4165.94	842.23	100	97
				<b>0.00</b>	922.18	9.50	112.03	861.44	100	100
		✓		0.30	31,755.62	379.58	4453.24	910.39	100	97
			✓	0.28	34,414.32	363.37	4241.88	842.23	100	97
	✓			<b>0.00</b>	810.85	14.61	139.18	925.03	100	100
	✓		✓	<b>0.00</b>	773.21	10.16	111.35	864.56	100	100
			✓	0.28	33,258.53	387.01	4309.26	910.77	100	97
	✓		✓	<b>0.00</b>	<b>449.82</b>	<b>6.13</b>	<b>111.24</b>	<b>926.87</b>	100	100
				3.57	153,930.67	2697.65	79,475.15	357.61	100	76
	✓			<b>0.00</b>	208.42	0.72	40.42	412.51	100	100
NE		✓		3.63	134,807.42	2695.66	81,716.03	366.31	100	76
			✓	3.45	142,703.06	2572.01	84,935.33	357.61	100	76
	✓			<b>0.00</b>	197.39	0.79	<b>39.24</b>	417.78	100	100
	✓		✓	<b>0.00</b>	<b>116.85</b>	<b>0.50</b>	46.03	421.82	100	100
		✓	✓	3.64	139,784.91	2725.89	83,398.39	366.31	100	76
	✓		✓	<b>0.00</b>	132.52	0.74	44.06	<b>426.17</b>	100	100



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 10,000s. When the valid inequalities enforce the model, nine out of ten are solved to optimality in less than 600s. 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 Sect. 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 3000s. Moreover, when inequalities (11), (12), and (13) are added for strengthening 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 more than 300. 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), (12) and (13) 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

1. Applegate, David L., Bixby, Robert E., Chvátal, Vašek, Cook, William J.: *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, Princeton (2006)
2. Asconavieta, P.H.S., Goldberg, M.C., Goldberg, E.F.G.: Evolutionary algorithm for the car renter salesman. In: 2011 IEEE Congress of Evolutionary Computation (CEC), pp. 593–600 (2011)
3. da Silva, A.R.V., Ochi, L.S.: An efficient hybrid algorithm for the traveling car renter problem. *Expert Syst. Appl.* **64**, 132–140 (2016)

4. da Silva Menezes, M., Goldberg, M.C., Goldberg, E.F.G.: A memetic algorithm for the prize-collecting traveling car renter problem. In: 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 3258–3265 (2014)
5. Felipe, D., Goldberg, E.F.G., Goldberg, M.C.: Scientific algorithms for the car renter salesman problem. In: 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 873–879 (2014)
6. Goldberg, M.C., Goldberg, E.F.G., Asconavieta, P.H., Menezes, M.S., Luna, H.P.L.: A transgenetic algorithm applied to the traveling car renter problem. *Expert Syst. Appl.* **40**(16), 6298–6310 (2013)
7. Goldberg, M.C., Goldberg, E.F.G., Menezes, M.S., Luna, H.P.L.: Quota traveling car renter problem: model and evolutionary algorithm. *Inf. Sci.* **367–368**, 232–245 (2016)
8. Goldberg, M.C., Goldberg, E.F.G., Luna, H.P.L., Menezes, M.S., Corrales, L.: Integer programming models and linearizations for the traveling car renter problem. *Optim. Lett.* **12**(4), 743–761 (2018)
9. Goldberg, M.C., Asconavieta, P.H., Goldberg, E.F.G.: Memetic algorithm for the traveling car renter problem: an experimental investigation. *Memet. Comput.* **4**(2), 89–108 (2012)
10. Ismkhan, H.: Effective three-phase evolutionary algorithm to handle the large-scale colorful traveling salesman problem. *Expert Syst. Appl.* **67**(C), 148–162 (2017)
11. Laporte, G., Mesa, J.A., Ortega, F.A., Perea, F.: Planning rapid transit networks. *Socio-Econ. Plan. Sci.* **45**(3), 95–104 (2011)
12. Pedrosa, L.L.C., Quesquén, G.Y.O., Schouery, R.C.S.: An asymptotically optimal approximation algorithm for the travelling car renter problem. In: Cacchiani, V., Marchetti-Spaccamela, A. (eds) 19th Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2019), volume 75 of OpenAccess Series in Informatics (OASIs), pp. 14:1–14:15. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2019)
13. Rios, B.H.O., Goldberg, E.F.G., Quesquén, G.Y.O.: A hybrid metaheuristic using a corrected formulation for the traveling car renter salesman problem. In: 2017 IEEE Congress on Evolutionary Computation (CEC), pp. 2308–2314 (2017)
14. Roemer, Thomas A., Ahmadi, Reza, Dasu, Sriram: The traveling salesman problem with flexible coloring. *Discrete Appl. Math.* **160**(12), 1798–1814 (2012)
15. Wolsey, L.A.: *Integer Programming*. Wiley Series in Discrete Mathematics and Optimization. Wiley, Hoboken (1998)
16. Xiong, Y., Golden, B., Wasil, E.: *The Colorful Traveling Salesman Problem*, pp. 115–123. Springer, Boston (2007)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.