The Rainbow Cycle Cover Problem

Selene Silvestri
Gilbert Laporte
Raffaele Cerulli

August 2015

CIRRELT-2015-40

The Rainbow Cycle Cover Problem

Selene Silvestri[1,2], Gilbert Laporte[1,3,*], Raffaele Cerulli[4]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Department of Computer Science, University of Salerno, Via Giovanni Paolo II 138, 84084, Fisciano, Italy

[3] Department of Management Sciences, HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

[4] Department of Mathematics, University of Salerno, Via Giovanni Paolo II 138, 84084, Fisciano, Italy

Abstract. We model and solve the Rainbow Cycle Cover Problem (RCCP). Given a connected and undirected graph $G = (V;E;L)$ and a coloring function ℓ that assigns a color to each edge of G from the finite color set L, a cycle whose edges have all different colors is called a rainbow cycle. The RCCP consists of finding the minimum number of disjoint rainbow cycles covering G. The RCCP on general graphs is known to be NP-complete. We model the RCCP as an integer linear program, we derive valid inequalities and we solve it by branch-and-cut. Computational results are reported on randomly generated instances.

Keywords. Rainbow Cycle Cover Problem (RCCP), edge-colored graph, rainbow cycles, branch-and-cut.

Acknowledgements: This research was partly supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant 2015-06189. This support is gratefully acknowledged.

---

* Corresponding author: Gilbert.Laporte@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2015

1 Introduction and Problem Description

The purpose of this paper is to present a mathematical model and a branch-and-cut algorithm for the Rainbow Cycle Cover Problem (RCCP). The RCCP is defined on a connected, undirected and edge-colored graph $G = (V, E, L)$ where V is the set of n vertices, E is the set of m edges, and L is a set of l colors. Let $\ell : E \to L$ be a coloring function assigning to each edge a color from the set L. A *rainbow cycle* of G is a cycle $C = (V_C, E_C)$, where $V_C \subseteq V$ and $E_C \subseteq E$, in which all edges have different colors, i.e. $|\ell(E_C)| = |E_C|$. A *rainbow cycle cover* (RCC) of G is a collection of rainbow cycles such that each vertex of the graph G belongs to exactly one cycle. Note that in this context a single vertex is considered as a degenerate rainbow cycle. From now on, we will refer to these as the *trivial* rainbow cycles. The aim of the RCCP is to determine a RCC with the least number of rainbow cycles.

Edge-colored graphs are used to represent many real-world situations in which is necessary to distinguish between different types of connections. For example, colors can represent types of transportations, telecommunication fibers, etc. The Minimum Labeled Spanning Tree Problem (MLSTP) was the first problem introduced in this area by Chang and Leu [5]. These authors proved that the problem is NP-hard and since then it has been studied by numerous reserchers [1], [3], [10], [15]. Many other colored problems have been studied in the literature, like the Colorful Traveling Salesman Problem [2], [9], [12], [16] and the Minimum Labeling Steiner Problem [4], [7], [6]. Li and Zhang [11] investigated the complexity of the rainbow tree, cycle and path partition problems and proved that identifying a RCC with the minimum number of cycles is NP-hard. To the best of our knowledge, no mathematical formulation for this problem has ever been put forward.

The main contribution of the paper is to propose an integer mathematical formulation and valid inequalities that will be used within a branch-and-cut algorithm. We will also consider some properties that a rainbow cycle cover must satisfy. These will allow us to preprocess the instances and add some ad hoc constraints that will help to solve the problems, sometimes in a very effective way.

The remainder of this paper is organized as follows. Section 2 contains the mathematical formulation and the description of the properties that a *RCC* must satisfy. The valid inequalities are introduced in Section 3. The branch-and-cut algorithm is described in Section 4. Computational results and conclusions are presented in Section 5 and 6, respectively.

2 Rainbow Cycle Cover Problem: Mathematical model and Properties

In this section we present an integer linear mathematical formulation for the RCCP. We introduce the set of binary variables γ_c, for $c = 1, \ldots, \bar{c}$ associated with each non-trivial cycle c of a *RCC*, whose value is equal to 1 if and only if c contains at least three vertices. It is easy to see that the number of variables γ_c depends on the number of possible non-trivial cycles, so it is useful to compute a good upper bound on this number. Note that, according to the definition, the variables γ_c are equal to one if and only if c contains at least three vertices, which means that $\lfloor n/3 \rfloor$ is an obvious upper bound on the maximum number of non-trivial rainbow cycles that a *RCC* of a graph can contain. We define binary variables y_v^c equal to 1 if and only if vertex v belongs to cycle c, and binary variables x_e^c equal to 1 if and only if edge e belongs to cycle c. In order to introduce constraints that can help prevent equivalent solutions, it is useful to define an index set $I_q = \{1, \ldots, q\}$, for an integer q, and to define the undirected graph $G = (V, E, L)$, with vertex set $V = I_n$. The formulation is then as follows:

$$\text{minimize } z = \sum_{c=1}^{\bar{c}} \gamma_c + \sum_{v \in V} M\Big(1 - \sum_{c=1}^{\bar{c}} y_v^c\Big) \tag{1}$$

subject to

$$\sum_{v\in V} y_v^c \le l\,\gamma_c \qquad c=1,\dots,\bar{c} \tag{2}$$

$$3\,\gamma_c \le \sum_{v\in V} y_v^c \qquad c=1,\dots,\bar{c} \tag{3}$$

$$\sum_{c=1}^{\bar{c}} y_v^c \le 1 \qquad v\in V \tag{4}$$

$$\sum_{e\in\delta(v)} x_e^c = 2\,y_v^c \qquad v\in V,\ c=1,\dots,\bar{c} \tag{5}$$

$$x_e^c \le y_v^c \qquad v\in V,\ e\in\delta(v) \tag{6}$$

$$\sum_{e\in\delta(S)} x_e^c \ge 2(y_v^c + y_u^c - 1) \qquad S\subset V,\ v\in S,\ \{v,u\}\in\{S,V\setminus S\},\ c=1,\dots,\bar{c} \tag{7}$$

$$\sum_{e\in E_k} x_e^c \le 1 \qquad c=1,\dots,\bar{c},\ k\in L \tag{8}$$

$$\gamma_{c+1} \le \gamma_c \qquad c=1,\dots,\bar{c}-1 \tag{9}$$

$$\sum_{c=v+1}^{\bar{c}} y_v^c = 0 \qquad v\in V : v<\bar{c} \tag{10}$$

$$y_v^c \le \sum_{w<v} y_w^{c-1} \qquad v\in V\setminus\{1\},\ c=3,\dots,\bar{c} \tag{11}$$

$$\gamma_c \in \{0,1\} \qquad c=1,\dots,\bar{c} \tag{12}$$

$$y_v^c \in \{0,1\} \qquad v\in V,\ c=1,\dots,\bar{c} \tag{13}$$

$$x_e^c \in \{0,1\} \qquad e\in E,\ c=1,\dots,\bar{c}, \tag{14}$$

where $\delta(v)$ denotes the set of edges incident to v in G, $\delta(S)=\{e=(v,u)\in E : v\in S\ u\in V\setminus S\}$, $\{S,V\setminus S\}=\{\{v,u\} : v\in S,\ u\in V\setminus S\}$ and $E_k=\{e\in E : \ell(e)=k\}$. Note that the difference between the set $\delta(S)$ and $\{S,V\setminus S\}$ is that the first one contains only edges of the graph, whereas the second set contains all the possible pairs between s and $V\setminus S$. The objective function (1) requires the minimization of the number of rainbow cycles. To this end, we need an objective function with two terms. The first part minimizes the sum of the variables γ_c, that is, the number of non-trivial cycles. The second part forces the vertices to belong to a cycle, whenever possible, giving a weight M to each isolated vertex. Without the second part of the objective function the optimal solution would be zero. Constraints (2) and (3) are logical constraints linking the binary variables γ_c with the binary variables y_v^c. Note that the maximum number of vertices that can belong to the same cycle is l since this is the number of different colors of the graph. Constraints (4) and (5) ensure that each vertex belongs to at most one cycle and, if it belongs to a cycle, it has a degree equal to 2 in that cycle. It is easy to see that to ensure the validity of the constraints (5), if a variable γ_c is equal to one, then there must be at least three vertices in the corresponding rainbow cycle. Constraints (6) impose that if a vertex is not in the cycle c, the edge incident on such vertex cannot belong to that cycle. Constraints (7) guarantee solutions with not more than one cycle associated to each variable γ_c. Constraints (8) impose that a cycle cannot contain two edges having the same color. These constraints ensure the rainbow property. Constraints (9), (10) and (11) are not necessary for the model, but they help eliminate symmetries. Constraints (9), mean that there will never be a variable γ_{c+1} equal to one if γ_c is equal to zero, for any c. Constraints (10) impose that vertices with and index $v < \bar{c}$ cannot belong to a cycle c such that $c > v$. Moreover, constraints (11) mean that a vertex v can belong to a cycle of index c if and only if at least one vertex w with a lower index belongs to the cycle of index $c - 1$. The constraints (10) and (11) are a generalization of the symmetry constraints introduced by Fischetti et al. [8] in the context of the Vehicle Routing Problem.

2.1 Properties of a Rainbow Cycle Cover

In this section we present some properties that a RCC must satisfy. Let ζ_v be the colored degree of vertex v, that is, the number of different colors incidents to v. It is easy to see that if the colored degree of a vertex v is equal to one, i.e. $\zeta_v = 1$, then vertex v will be a trivial rainbow cycle, that is, it will be isolated:

$$\sum_{c=1}^{\bar{c}} y_v^c = 0 \qquad\qquad v \in V : \ \zeta_v = 1. \tag{15}$$

Note that in such a case, all edges incident to that vertex will be equal to 0 in the optimal solution. In view of these observations, if we denote by n_1 the number of vertices whose colored degree is equal to one, then the upper bound on the number of variables γ reduces to $\lfloor (n - n_1)/3 \rfloor$.

We can extend this observation to the edges. Suppose that given an edge $e = (v, u)$, the total number of colors incident to v and u is equal to two, i.e. $|\{\ell(\delta(v)) \cup \ell(\delta(u))\}| = 2$, then edge e cannot belong to a rainbow cycle. Note that, if $|\{\ell(\delta(v)) \cup \ell(\delta(u))\}| = 2$, then one of these two colors must be the color of edge e, therefore there will be only one more color available to connect e with others edges of a RCC which is not possible. We can state this property through the constraints

$$\sum_{c=1}^{\bar{c}} x_e^c = 0 \qquad\qquad e = (v, u) \in E : \ |\{\ell(\delta(v)) \cup \ell(\delta(u))\}| = 2. \tag{16}$$

We can use this two observations in a preprocessing phase to reduce the size and the difficulty of the instances. Note that, the last property is included in the following most general one. Pairs of vertices having a colored degree equal to 2 and the same set of incident colors cannot belong to the same cycle:

$$y_v^c + y_u^c \leq 1 \qquad\qquad v, u \in V : \ell(\delta(v)) = \ell(\delta(u)), \ \zeta_v = \zeta_u = 2, \ c = 1, \ldots, \bar{c}. \tag{17}$$

Obviously, if two of these vertices are adjacent, the edge linking them cannot belong to a RCC, which means that all variables associated to the edge will be equal to zero and then constraints (16) are satisfied. Moreover, for each vertex v such that $\zeta_v = 2$, the following constraints are valid for the RCCP:

$$\sum_{e \in \delta_k(v)} x_e^c - \sum_{e \in \delta_h(v)} x_e^c = 0 \qquad v \in V : \ \zeta_v = 2, \ \ell(\delta(v)) = \{k, h\}, \ c = 1, \ldots, \bar{c}, \tag{18}$$

where $\delta_k(v) = \{e \in E : e \in \{\delta(v) \cap E_k\}\}$, i.e. the set of all the edges incident to v and having color k.

Another simple observation regarding the properties that a RCC must satisfy is that if the shortest cycle that includes two fixed vertices v and u contains at least $l+1$ edges, then the two vertices cannot belong to the same rainbow cycle. Note that a rainbow cycle cannot contain more than l edges. Identifying shortest cycles containing two fixed vertices can be efficiently achieved by means of Suurballe's algorithm ([13], [14]). This algorithm identifies two disjoint paths in a non-negative weighted directed graph so that both paths connect the same pair of vertices and have a minimum total length. Suurballe's algorithm uses Dijkstra's algorithm to find the first path. It then modifies the weights of the edges preserving their non-negativity. After this modification Suurballe's algorithm uses Dijkstra's algorithm a second time. Procedure 1 shows the details of Suurballe's algorithm.

Procedure 1: Suurballe's algorithm

```
Input: $G(V,A)$, $v \in V$, $u \in V$, $w(A)$
Output: $SC(v,u)$ the shortest cycle containing $v, u$
1 $T_1 \leftarrow$ DijkstraAlgorithm$(v, G, w(A))$
2 $P_1 \leftarrow$ IdentifyPath$(v, u, T_1)$
3 $w(A) \leftarrow$ updateWeightEdge$(T_1, G)$
4 $G_{P_1} \leftarrow$ CreateResidualGraph$(P_1, G)$
5 $T_2 \leftarrow$ DijkstraAlgorithm$(v, G_{P_1}, w(A))$
6 $P_2 \leftarrow$ IdentifyPath$(v, u, T_2)$
7 $SC(v,u) \leftarrow$ mergePaths$(P_1, P_2)$
8 return $SC(v,u)$
```

A key point of the algorithm is line 3. The weights of the arcs are modified according to the following formula

$$w(i,j) = w(i,j) - d(v,j) + d(v,i) \qquad (i,j) \in A, \tag{19}$$

where v is the root vertex of the shortest path tree T_1, and $d(v,j)$, $d(v,i)$ are the distances from v to j and i in T_1. Thanks to Suurballe's algorithm we can impose the constraints

$$y_v^c + y_u^c \leq 1 \qquad v, u \in V : |SC(v,u)| \geq l+1,\ c = 1, \dots, \bar{c}, \tag{20}$$

where $SC(v,u)$ represents the shortest cycle containing the vertices v and u. It is easy to see that when two vertices v and u are incompatible, that is, when two vertices cannot belong to the same cycle because of (17) and (20), then the following inequalities are valid for the RCCP:

$$\sum_{e \in \{\delta(v) \cup \delta(u)\}} x_e^c \leq 2 \qquad v, u \in V :\ y_v^c + y_u^c \leq 1,\ c = 1, \dots, \bar{c} \tag{21}$$

$$\sum_{e \in \{\delta_k(v) \cup \delta_k(u)\}} x_e^c \leq 1 \qquad v, u \in V :\ y_v^c + y_u^c \leq 1,\ k \in L,\ c = 1, \dots, \bar{c}. \tag{22}$$

Our last observation about the properties of a RCC is the following: for each color k, if the edges of color k are incident to f_k distinct vertices, i.e. if $|\{v \in V : \delta(v) \cap E_k \neq \emptyset\}| = f_k$, then only $\lfloor f_k/2 \rfloor$ edges of color k can belong to a RCC. If only one more edge is selected, then there will be at least two edges having the same color and incident to a same vertex. We can therefore impose the following constraints:

$$\sum_{c=1}^{\bar{c}} \sum_{e \in E_k} x_e^c \leq \lfloor f_k/2 \rfloor \qquad k \in L. \tag{23}$$

3 Valid inequalities

In this section we present some valid inequalities for the RCCP.

Proposition 1. *The constraints*

$$y_v^c - \gamma_c \leq 0 \qquad v \in V,\ c = 1, \dots, \bar{c} \tag{24}$$

$$x_e^c - \gamma_c \leq 0 \qquad e \in E,\ c = 1, \dots, \bar{c} \tag{25}$$

are satisfied by all optimal RCCP solutions.

Proof. These constraints state that if a vertex or an edge belongs to a cycle, then the variable representing that cycle must be used. □

An example of a solution violating constraints (24) but satisfying constraints (2) is the following: $\gamma_1 = 1$, $\gamma_2 = 0.34$, $y_1^1 = 1$, $y_2^1 = 1$, $y_3^1 = 1$, $y_4^1 = 0.53$, $y_5^1 = 0.24$, $y_6^1 = 0.71$, $y_7^1 = 0.41$, $y_4^2 = 0.47$, $y_5^2 = 0.76$, $y_6^2 = 0.29$, $y_7^2 = 0.06$, $x^1_{(1,2)} = 1$, $x^1_{(1,3)} = 0.35$, $x^1_{(1,5)} = 0.06$, $x_{(1,6)} = 0.59$, $x^1_{(2,3)} = 0.76$, $x^1_{(2,6)} = 0.24$, $x^1_{(3,4)} = 0.88$, $x^1_{(4,6)} = 0.18$, $x^1_{(5,7)} = 0.42$, $x^1_{(6,7)} = 0.41$, $x^2_{(4,5)} = 0.93$, $x^2_{(5,6)} = 0.53$, $x^2_{(5,7)} = 0.06$, $x^2_{(6,7)} = 0.06$. It is depicted in Figure 1:

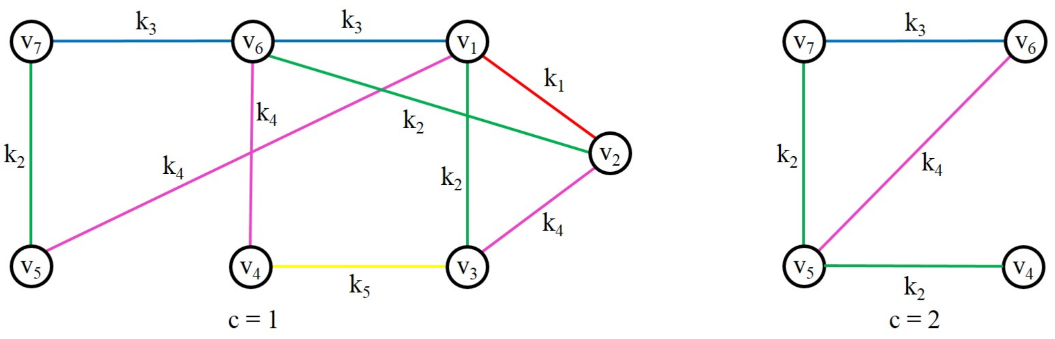

Figure 1: Solution where $\gamma_1 = 1$ and $\gamma_2 = 0.34$ and constraints (24) are violated by v_4^2 and v_5^2 in $c = 2$.

Proposition 2. *The constraints*

$$\sum_{c=1}^{\bar{c}} \left\{ x_e^c + \sum_{f \in \{\delta_h(u) \cup \delta_h(v)\}} x_f^c \right\} \leq 2 \qquad e = (v,u) \in E : \ell(e) = k,\ h \in L \setminus \ell(e) \qquad (26)$$

are valid for the RCCP.

Proof. These constraints impose that if an edge $e = (v, u)$ having color $\ell(e) = k$ is selected, then at most one edge having color $h \neq k$ and belonging to the set $\{\delta_h(v) \cup \delta_h(u)\}$ can be selected. □

Proposition 3. *The constraints*

$$\sum_{e\in\delta_k(v)} x_e^c \le y_v^c \qquad k\in L,\ v\in V,\ c=1,\dots,\bar{c} \tag{27}$$

are valid for the RCCP.

Proof. If a vertex v belongs to a cycle c, then at most one edge of color k and incident on v can be selected. Note that for fixed a vertex v, a cycle c and a color k, the inequality

$$\sum_{e\in\delta_k(v)} x_e^c \le |\delta_k(v)| y_v^c \qquad k\in L,\ v\in V,\ c=1,\dots,\bar{c} \tag{28}$$

represents an aggregate version of constraints (6), when the color k is fixed. Moreover, due to (8) the left-hand side of (28) results in

$$\sum_{e\in\delta_k(v)} x_e^c \le \sum_{e\in E_k} x_e^c \le 1 \qquad k\in L,\ v\in V,\ c=1,\dots,\bar{c}, \tag{29}$$

and due to (5) all the edges x_e^c, i.e. $e\in\delta(v)$, are equal to 0 if y_v^c is equal to 0. Thanks to these two observations the right-hand side of (28) can be reduced to y_v^c, i.e. exactly constraints (27). □

Proposition 4. *The constraints*

$$\sum_{e\in\delta_k(v)} x_e^c - \sum_{e\in\{\delta(v)\setminus\delta_k(v)\}} x_e^c \le 0 \qquad v\in V,\ k\in L,\ c=1,\dots,\bar{c} \tag{30}$$

are satisfied by all the optimal RCCP solutions.

Proof. These constraints impose that for each vertex v, for each color k and for each possible cycle $c=1,\dots,\bar{c}$, if an edge incident to v and having color k is selected, then an edge incident to v and having a different color must be selected. □

Proposition 5. *The valid inequalities (27) and (30) are equivalent.*

Proof. The valid inequalities (27) state that, fixed $k \in L$ and $c = \{1, \dots, \bar{c}\}$,

$$\sum_{e \in \delta_k(v)} x_e^c \leq y_v^c$$

adding and subtracting $\frac{1}{2} \sum_{e \in \delta(v)} x_e^c$ to the left-hand side, we obtain

$$\sum_{e \in \delta_k(v)} x_e^c + \frac{1}{2} \sum_{e \in \delta(v)} x_e^c - \frac{1}{2} \sum_{e \in \delta(v)} x_e^c \leq y_v^c$$

which, due to (5), is equivalent to

$$\sum_{e \in \delta_k(v)} x_e^c - \frac{1}{2} \sum_{e \in \delta(v)} x_e^c \leq 0. \tag{31}$$

Note that, with easy mathematical operations, one can see that (31) are exactly the valid inequalities (30). One also observes that without constraints (5), constraints (27) are stronger than constraints (30). □

Proposition 6. *The constraints*

$$x_e^c - \sum_{h \in L \setminus \ell(\delta(v))} \sum_{f \in \delta_h(u)} x_f^c \leq 0 \qquad v : \zeta_v = 2,\ c = 1, \dots, \bar{c} \tag{32}$$

are valid for the RCCP.

Proof. These constraints state that if an edge $e = (v, u)$, incident to a vertex v having colored degree $\zeta_v = 2$, is selected, then at least one edge that is incident on the vertex u and having color $h \in \{L \setminus \ell(\delta(v))\}$ must be selected. □

Proposition 7. *The constraints*

$$\sum_{\{v \in V : \delta_k(v) \neq \emptyset\}} \left\{ y_v^c - \sum_{e \in \delta_k(v)} x_e^c \right\} \geq 0 \qquad k \in L,\ c = 1, \dots, \bar{c} \tag{33}$$

are valid for the RCCP.

Proof. These constraints impose that for each color k and for each cycle c, twice the number of edges having color k and belonging to c must be less than or equal to the number of vertices belonging to c on which those edges are incident, since on a single vertex cannot incide twice the same color. $\square$

An example of a solution violating constraints (33) when $k = k_2$ but satisfying the constraints (2 - 11) is depicted in Figure (2):

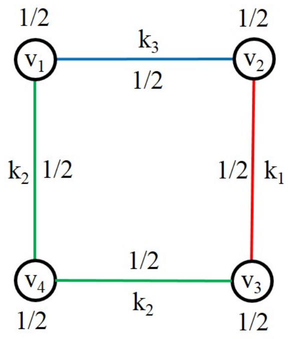

Figure 2: A constraint (33) is violated for $k = k_2$

Proposition 8. *Let* $F = \{(v, u) \in E : |\ell(\delta(v)) \cup \ell(\delta(u))| = 3\}\}$. *Moreover, for each edge* $e = (u, v)$, *let* $\Delta(e) = \{\delta(u) \cup \delta(v)\}$ *and let* $\Delta_k(e) = \{\delta_k(v) \cup \delta_k(u)\}$. *Then the constraints*

$$x^c_{(e)} - \sum_{f \in \Delta_k(e)} x^c_f \leq 0 \qquad e = (u, v) \in F,\ k \in \{\ell(\Delta(e)) \setminus \ell(e)\},\ c = 1, \dots, \bar{c} \tag{34}$$

are valid for the RCCP.

Proof. These constraints impose that each edge $e = (u, v)$ such that $\big\{\{\ell(\delta(u)) \cup \ell(\delta(v))\} \setminus \ell(e)\big\} = \{h, k\}$ can belong to a RCC if and only if at least one edge of the set $\{\delta_h(u) \cup \delta_h(v)\}$ and one of the set $\{\delta_k(u) \cup \delta_k(v)\}$ is selected. It is clear that, since we are looking for a RCC, if an edge of the set F will be selected, then exactly one edge for each set will belong to the solution. $\square$

We can extend the observation made for constraints (34) to the set of edges $\bar{F} = \{(u, v) \in E : |\ell(\delta(u)) \cup \ell(\delta(v))| = 4\}$.

Proposition 9. *The constraints*

$$x_e^c - \sum_{f \in \Delta_t(e) \cup \Delta_s(e)} x_f^c \leq 0 \qquad e = (u, v) \in \bar{F},\ t, s \in \{\ell(\Delta(e)) \setminus \ell(e)\},\ c = 1, \ldots, \bar{c} \tag{35}$$

are valid for the RCCP.

The valid inequalities that follow, differently from all the sets described until now, are not easy to identify. Let $P(v, u)$ be a path between the vertices v and u, where $v \neq u$, and let $\mathcal{P}$ be the set of all the rainbow paths of the graph G, i.e. $\mathcal{P} = \{P(u, v) :\ P(u, v) \text{ is rainbow}\}$.

Proposition 10. *The constraints*

$$\sum_{c=1}^{\bar{c}} \left\{ \sum_{e \in \delta_k(v) \cup \delta_k(u)} x_e^c + \sum_{e \in P(u,v)} x_e^c \right\} \leq |P(u, v)| + 1 \quad k \in L \setminus \ell(P(u, v)),\ P(u, v) \in \mathcal{P} \tag{36}$$

are valid for the RCCP.

Proof. These constraints state that if all the edges of the rainbow path $P(v, u)$ belong to the solution, then at most one edge belonging to the set $\{\delta_k(v) \cup \delta_k(u)\}$, where $k \in L \setminus \ell(P(u, v))$, can be selected. □

In the Figure (3) is depicted an example of the structure described above for the valid inequalities (36). Note that the valid inequalities (26) are a particular case of this set of inequalities, in which the rainbow path is a single edge.

4 Branch-and-cut algorithm

We solve the RCCP by means of a branch-and-cut algorithm. The description of the steps is summarized in Procedure 2. The first step of our algorithm consists in a preprocessing

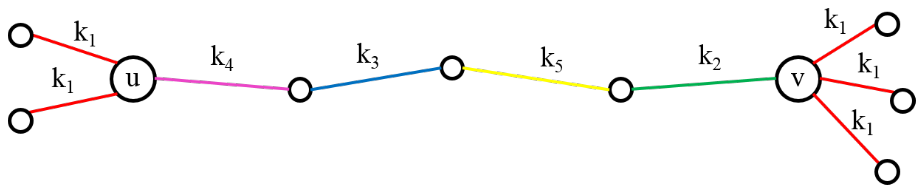

Figure 3: An example of valid inequalities (36)

phase in which we identify all vertices and edges that satisfy the properties (15) and (16). We then set the corresponding variables to 0, which allows us to reduce the instance size. The initial subproblem is obtained by relaxing constraints (6) and (7) as well as the integrality constraints (12), (13) and (14). Note that, thanks to constraints (5), constraints (6) are redundant in an integer solution, but we add them as valid cuts. We also add constraints (17), (18), (20), (21), (22) and (23) to the initial subproblem. From this point, a search for violated constraints (6) and (7) and violated valid inequalities (24), (25), (26), (27) and (32) is performed. Valid inequalities (33), (34) and (35) turned out to be ineffective and were not considered. A subset of the most violated inequalities of each type is added to the cut-pool. Moreover, since identifing all rainbow paths between all pairs of vertices in a graph is not possible, a search for violated inequalities (36) is performed only among the set $\mathcal{SP} = \{P(u,v) \in \mathcal{P} :\ P(u,v) \text{ is a shortest path}\}$. However, except for (7) and (36), in order to identify violated inequalities, we consider all of them and verify which are violated by the current relaxed solution. The algorithm for the identification of the most violated constraint (7) is a simple max-flow separation problem. Branching is performed in priority on the y_v^c and x_v^c variables with the lower index c and having the fractional value closest to 0.5.

Procedure 2: Branch-and-cut algorithm

```
Input: an integer program $P$.
Output: an optimal solution of $P$, if exists.

$ub \leftarrow \infty$, $L = \emptyset$
Define a first subproblem $S_0$
$L \leftarrow S_0$
while $L \neq \emptyset$ do
    $S_i \leftarrow$ chooseSubproblem($L$)
    $L \leftarrow L \setminus S_i$
    $z \leftarrow$ solveSubproblem($S_i$)
    if $z < ub$ then
        if the solution is integer then
            $ub \leftarrow z$
        else
            cuts $\leftarrow$ generateCuts
            if cuts violated constraints are identified then
                addCuts (cuts)
            else
                $\{S_{i1}, S_{i2}\} \leftarrow$ branching($S_i$)
                $L \leftarrow L \cup \{S_{i1}, S_{i2}\}$
```

5 Computational results

The algorithm was coded in C and solved using IBM ILOG CPLEX 12.5. The computational experiments were performed on a 64-bit GNU/Linux operating system, 96 GB of RAM and one processor Intel Xeon X5675 running at 3.07 GHz.

Experiments for the RCCP were conducted on randomly generated instances and their results are reported in Table 1. Each instance is characterized by the number of vertices n (size), the number of edges m and the number of colors l. For each instance with n vertices, the number of edges is set to $m = \lceil n(n-1)/2 \times d + n \rceil$, with $d \in \{0.1, 0.2, 0.3\}$, and the number of colors is set to $\lceil \log(m)/2 \rceil$, $\lceil \log(m) \rceil$ and $\lceil 2\log(m) \rceil$. The total number of different scenarios is nine for each size. We have generated five instances for each scenario, with the same number of nodes, edges and colors and the results reported in each line of our tables are average values over these five instances. In Table 1 the first four columns report the characteristics of each scenario: scenario ID, the number of vertices (n), the number of edges (m) and the number of colors (l), respectively. This table also provides the number of rainbow cycles (#cycles), the number of non-trivial cycles and the number of trivial cycles (#non-trivial and #trivial, respectively) the value of the optimal solution (Obj), the computing time (t(s)) in seconds and the number of nodes in the search tree (#nodes). We have imposed a time limit equal to $10,800$ seconds. Whenever at least one instance of the scenario was not solved to optimality within the time limit, we indicate with the symbol * that the value reported is an upper bound on the optimal solution. We also refer to the solutions with the symbol * as the *best known* solutions. Note that in the objective function, the part that minimizes the number of non-trivial cycles is less then or equal to $\bar{c}$ in the worst case. Thanks to this observation a value equal to $2\,\bar{c}$ is given to the weight M.

The results show that when the number of vertices increases, instances with a large number of edges and a small number of colors are the hardest to solve. We also note that the computation time and the number of nodes in the search tree seem to increase in these cases. This is due to the symmetry of the problem. Indeed several cycles can have the same number of colors and several equivalent solutions can be identified. The number of colors also affects the solution in terms of the presence of trivial cycles, mainly for the instances with a small number of edges.

Table 1: Summary of computational results for the RCCP

ID	**Instance**			**RCCP**						
	n	m	l	**#cycles**	**#non-trivial**	**#trivial**	**Obj**		**t(s)**	**#nodes**
1	20	39	3	18.00	1.00	17.00	120.00		0.01	0.00
2	20	39	6	13.60	2.00	11.60	83.20		0.14	0.00
3	20	39	11	10.60	2.20	8.40	61.00		0.18	0.00
4	20	58	3	14.80	2.60	12.20	88.00		0.13	0.00
5	20	58	6	9.80	3.40	6.40	48.20		1.76	5.60
6	20	58	12	6.80	2.20	4.60	34.40		1.71	12.00
7	20	77	4	10.00	3.80	6.20	47.20		2.41	21.20
8	20	77	7	6.80	3.60	3.20	26.00		5.86	77.00
9	20	77	13	4.80	2.60	2.20	18.00		5.71	65.60
10	30	74	4	21.80	3.20	18.60	207.80		2.59	4.00
11	30	74	7	19.40	2.80	16.60	185.40		15.16	16.80
12	30	74	13	15.40	2.60	12.80	143.40		9.33	8.60
13	30	117	4	18.20	4.80	13.40	152.20		21.38	38.60
14	30	117	7	13.00	4.40	8.60	99.00		56.52	114.80
15	30	117	14	9.60	3.20	6.40	73.60		54.07	119.20
16	30	161	4	15.20	6.00	9.20	107.20		180.50	541.20
17	30	161	8	8.00	5.20	2.80	36.00		210.39	344.00
18	30	161	15	5.40	3.60	1.80	23.40		55.79	67.80
19	40	118	4	30.60	3.60	27.00	381.60		10.67	22.00
20	40	118	7	24.80	3.80	21.00	297.80		129.83	49.80
21	40	118	14	20.40	3.40	17.00	241.40		34.08	10.40
22	40	196	4	25.00	6.20	18.80	269.40		314.22	191.20
23	40	196	8	15.80	6.00	9.80	143.20		361.94	281.60
24	40	196	16	11.00	4.00	7.00	102.00		480.33	457.60
25	40	274	5	14.60	8.60	6.00	92.60	*	5136.55	2894.00
26	40	274	9	8.80	6.20	2.60	42.60	*	6420.92	2141.20
27	40	274	17	5.20	3.80	1.40	23.40	*	3185.14	1119.20
28	50	173	4	35.80	5.60	30.20	519.00		526.10	42.00
29	50	173	8	30.40	5.40	25.00	430.40		1185.35	368.40
30	50	173	15	23.40	5.20	18.20	314.60		694.63	357.40
31	50	295	5	25.40	8.00	17.40	303.80	*	7354.71	2027.80
32	50	295	9	18.40	7.20	11.20	197.60	*	5558.08	1477.40
33	50	295	17	12.80	5.60	7.20	128.00		2695.05	756.20
34	50	418	5	19.80	10.20	9.60	173.40	*	10800.00	1194.20
35	50	418	9	12.80	8.20	4.60	86.40	*	10214.81	1130.00
36	50	418	18	6.60	4.40	2.20	41.80	*	8954.48	1247.00

5.1 LP lower bounds and duality gaps

We present the LP lower bounds and the duality gaps for the RCCP obtained by adding one valid inequality each time, respectively in Table 2 and Table 3. The first two columns of the Table 2 provide the instance ID and the objective value. Again, the symbol * appears in the table to indicate that the value reported is an upper bound of the optimal solution value. The next columns provide lower bounds $w(P)$, $w(P1)$, $w(P2)$, $w(P3)$, $w(P4)$ and $w(P5)$, where P denotes the polytope obtained by relaxing the integrality constraints, while $P1$, $P2$, $P3$, $P4$ and $P5$ denote the intersection of P with (27), (26), (24) – (25), (36) and (32), respectively. Table 3 provide the duality gap obtained on the six polytopes comparing with the optimal solution. From the tables we can see that the valid inequalities often help to improve the lower bound $w(P)$. It is also easy to see that the best lower bounds are provided by $w(P1)$ and $w(P2)$. However, it is interesting to observe that for the hardest instances, the values of the gap are significantly high. The presence of the constant M in the objective function affects the results, but symmetry seems to remain the main problem.

6 Conclusions

We have proposed a mathematical formulation of the Rainbow Cycle Cover Problem, some properties that a RCC must satisfied and some valid inequalities used to solve the RCCP within a branch-and-cut algorithm. Computational experiments were conducted on randomly generated instances. Results show that the branch-and-cut algorithm is able to solve instances having between 20 and 50 vertices, and between 3 and 18 colors. The presence of a constant M in the objective function and the symmetry of the problem affect the final results and the effectiveness of the algorithm, mainly on instances with a large number of edges and a small number of colors when the number of vertices increases.

Table 2: Linear programming lower bounds for RCCP

ID	**RCCP**							
	Obj		**w(P)**	**w(P1)**	**w(P2)**	**w(P3)**	**w(P4)**	**w(P5)**
1	120.00		120.00	120.00	120.00	120.00	120.00	120.00
2	83.20		79.62	81.14	79.71	79.70	79.62	79.62
3	61.00		56.33	56.99	56.38	56.45	56.38	56.51
4	88.00		85.38	86.17	85.98	85.38	85.38	85.79
5	48.20		29.76	32.87	30.50	29.82	30.12	30.86
6	34.40		26.05	27.06	26.84	26.36	26.12	26.16
7	47.20		28.86	32.37	30.99	28.88	28.99	29.69
8	26.00		16.92	18.10	17.43	17.06	17.02	16.92
9	18.00		11.63	13.08	12.92	12.04	11.63	11.63
10	207.80		191.20	194.43	193.09	191.20	191.84	192.98
11	185.40		131.39	142.10	134.91	131.43	131.91	132.50
12	143.40		122.14	123.89	122.67	122.31	124.18	122.43
13	152.20		94.83	105.95	105.21	94.83	96.71	97.65
14	99.00		62.67	71.05	67.90	62.73	64.81	65.31
15	73.60		53.60	58.47	54.51	53.93	53.78	54.16
16	107.20		44.66	52.90	52.95	44.66	46.04	50.17
17	36.00		17.58	18.93	22.11	17.71	17.67	17.70
18	23.40		16.01	20.59	18.10	16.50	16.59	16.01
19	381.60		320.46	326.28	326.17	320.46	322.99	324.15
20	297.80		235.51	245.40	244.61	235.59	236.46	240.75
21	241.40		202.22	211.60	207.59	202.57	203.82	206.17
22	269.40		133.77	164.04	155.76	133.77	138.69	144.44
23	143.20		77.03	88.87	91.21	77.06	82.55	78.44
24	102.00		69.79	75.49	74.25	70.00	69.87	69.79
25	92.60	*	31.74	39.58	36.03	31.78	34.37	31.74
26	42.60	*	23.89	26.67	26.67	23.89	26.67	23.89
27	23.40	*	19.08	21.87	19.08	19.08	19.08	19.08
28	519.00		388.47	412.81	410.57	388.47	393.28	407.81
29	430.40		277.13	308.62	296.86	277.33	282.80	288.49
30	314.60		233.08	251.27	243.86	233.56	236.17	233.20
31	303.80	*	148.93	176.66	164.85	148.95	153.38	155.18
32	197.60	*	114.41	132.04	124.42	114.45	116.63	114.41
33	128.00		97.06	108.35	101.71	97.40	97.64	97.91
34	173.40	*	63.98	78.23	79.88	64.02	67.21	67.07
35	86.40	*	51.16	55.10	52.84	51.16	51.16	52.84
36	41.80	*	40.06	40.06	40.06	40.06	40.06	40.06

Table 3: Linear programming duality gap for RCCP

ID	RCCP					
	gapP(%)	gapP1(%)	gapP2(%)	gapP3(%)	gapP4(%)	gapP5(%)
1	0.0	0.0	0.0	0.0	0.0	0.0
2	4.3	2.5	4.2	4.2	4.3	4.3
3	7.7	6.6	7.6	7.5	7.6	7.4
4	3.0	2.1	2.3	3.0	3.0	2.5
5	38.3	31.8	36.7	38.1	37.5	36.0
6	24.3	21.3	22.0	23.4	24.1	24.0
7	38.9	31.4	34.3	38.8	38.6	37.1
8	34.9	30.4	32.9	34.4	34.6	34.9
9	35.4	27.4	28.2	33.1	35.4	35.4
10	8.0	6.4	7.1	8.0	7.7	7.1
11	29.1	23.4	27.2	29.1	28.9	28.5
12	14.8	13.6	14.5	14.7	13.4	14.6
13	37.7	30.4	30.9	37.7	36.5	35.8
14	36.7	28.2	31.4	36.6	34.5	34.0
15	27.2	20.6	25.9	26.7	26.9	26.4
16	58.3	50.7	50.6	58.3	57.1	53.2
17	51.2	47.4	38.6	50.8	50.9	50.8
18	31.6	12.0	22.6	29.5	29.1	31.6
19	16.0	14.5	14.5	16.0	15.4	15.1
20	20.9	17.6	17.9	20.9	20.6	19.2
21	16.2	12.3	14.0	16.1	15.6	14.6
22	50.3	39.1	42.2	50.3	48.5	46.4
23	46.2	37.9	36.3	46.2	42.4	45.2
24	31.6	26.0	27.2	31.4	31.5	31.6
25	65.7	57.3	61.1	65.7	62.9	65.7
26	43.9	37.4	37.4	43.9	37.4	43.9
27	18.5	6.5	18.5	18.5	18.5	18.5
28	25.2	20.5	20.9	25.2	24.2	21.4
29	35.6	28.3	31.0	35.6	34.3	33.0
30	25.9	20.1	22.5	25.8	24.9	25.9
31	51.0	41.9	45.7	51.0	49.5	48.9
32	42.1	33.2	37.0	42.1	41.0	42.1
33	24.2	15.3	20.5	23.9	23.7	23.5
34	63.1	54.9	53.9	63.1	61.2	61.3
35	40.8	36.2	38.8	40.8	40.8	38.8
36	4.2	4.2	4.2	4.2	4.2	4.2

Acknowledgements

This research was partly supported by the Canadian Natural Sciences and Engineering Research Council under grant 2015-06189. This support is gratefully acknowledged.

References

[1] T. Brüggemann, J. Monnot, and G. J. Woeginger. Local search for the minimum label spanning tree problem with bounded color classes. *Operations Research Letters*, 31:195–201, 2003.

[2] R. Cerulli, P. Dell'Olmo, M. Gentili, and A. Raiconi. Heuristic approaches for the minimum labelling hamiltonian cycle problem. *Electronic notes in discrete mathematics*, 25:131–138, 2006.

[3] R. Cerulli, A. Fink, M. Gentili, and S. Voß. Metaheuristics comparison for the minimum labelling spanning tree problem. In *The Next Wave on Computing, Optimization, and Decision Technologies*, pages 93–106. Springer US, 2005.

[4] R. Cerulli, A. Fink, M. Gentili, and S. Voß. Extensions of the minimum labelling spanning tree problem. *Journal of Telecommunications and Information Technology*, 4:39–45, 2006.

[5] R.S. Chang and S.J. Leu. The minimum labeling spanning trees. *Information Processing Letters*, 63:277–282, 1997.

[6] S. Consoli, K. Darby-Dowman, N. Mladenović, and J. A. Moreno-Pérez. Variable neighbourhood search for the minimum labelling steiner tree problem. *Annals of Operations Research*, 172:71–96, 2009.

[7] S. Consoli, J. A. Moreno-Pérez, K. Darby-Dowman, and N. Mladenović. Discrete particle swarm optimization for the minimum labelling steiner tree problem. *Natural Computing*, 9:29–46, 2010.

[8] M. Fischetti, J. J. Salazar González, and P. Toth. Experiments with a multi-commodity formulation for the symmetric capacitated vehicle routing problem. In *Proceedings of the 3rd Meeting of the EURO Working Group on Transportation*, pages 169–173, 1995.

[9] N. Jozefowiez, G. Laporte, and F. Semet. A branch-and-cut algorithm for the minimum labeling hamiltonian cycle problem and two variants. *Computers & Operations Research*, 38:1534–1542, 2011.

[10] S. O. Krumke and H.C. Wirth. On the minimum label spanning tree problem. *Information Processing Letters*, 66:81–85, 1998.

[11] X. Li and X.Y. Zhang. On the minimum monochromatic or multicolored subgraph partition problems. *Theoretical Computer Science*, 385:1–10, 2007.

[12] J. Silberholz, A. Raiconi, R. Cerulli, M. Gentili, B.L. Golden, and S. Chen. Comparison of heuristics for the colourful travelling salesman problem. *International Journal of Metaheuristics*, 2:141–173, 2013.

[13] J. W Suurballe and R. E. Tarjan. A quick method for finding shortest pairs of disjoint paths. *Networks*, 14:325–336, 1984.

[14] J.W. Suurballe. Disjoint paths in a network. *Networks*, 4:125–145, 1974.

[15] Y. Xiong, B.L. Golden, and E.A. Wasil. Improved heuristics for the minimum label spanning tree problem. *IEEE Transactions on Evolutionary Computation*, 10:700–703, 2006.

[16] Y. Xiong, B.L. Golden, and E.A. Wasil. The colorful traveling salesman problem. In *Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies*, pages 115–123. Boston: Springer, 2007.