Discrete Applied Mathematics 164 (2014) 112–120

Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

# On Minimum Reload Cost Cycle Cover

Giulia Galbiati [a], Stefano Gualandi [b,*], Francesco Maffioli [c]

[a] Dipartimento di Informatica e Sistemistica, Università degli Studi di Pavia, Via Ferrata 1, 27100 Pavia, Italy
[b] Dipartimento di Matematica, Università degli Studi di Pavia, Via Ferrata 1, 27100 Pavia, Italy
[c] Politecnico di Milano, Dipartimento di Elettronica e Informazione, Piazza Leonardo da Vinci 32, 20133 Milano, Italy

ARTICLE INFO

*Article history:*
Received 23 July 2010
Received in revised form 26 October 2011
Accepted 8 December 2011
Available online 13 January 2012

*Keywords:*
Network optimization
Reload cost model
Computational complexity

ABSTRACT

We consider the problem of spanning the nodes of a given colored graph $G = (N, A)$ by a set of node-disjoint cycles at minimum reload cost, where a non-negative reload cost is paid whenever passing through a node where the two consecutive arcs have different colors. We call this problem Minimum Reload Cost Cycle Cover (MinRC3 for short). We prove that it is strongly NP-hard and not approximable within $\frac{1}{\epsilon}$ for any $\epsilon > 0$ even when the number of colors is 2, the reload costs are symmetric and satisfy the triangle inequality. Some IP models for MinRC3 are then presented, one well suited for a Column Generation approach. The corresponding pricing subproblem is also proved strongly NP-hard. Primal bounds for MinRC3 are obtained via local search based heuristics exploiting 2-opt and 3-opt neighborhoods. Computational results are presented comparing lower and upper bounds obtained by the above mentioned approaches.



## 1. Introduction

We consider an optimization problem defined on a graph $G = (N, A)$, either directed or undirected, where all arcs have received a color from a set $L$ of cardinality $h$, and where it is known that all nodes can be spanned by a set of node-disjoint cycles. A *reload cost* is paid along a path or cycle in $G$ whenever passing through a node where the two consecutive arcs have different colors. All possible values of reload costs are given by a *reload cost function* $r : L \times L \to \mathbb{Z}^+$, where $r(l, l')$ denotes the reload cost that arises at a node when passing from color $l$ of one incident arc to color $l'$ of the other. We define the reload cost of a cycle in $G$ as the sum of the reload costs that arise at its nodes. We are interested in spanning all nodes of $G$ by a set $S$ of node-disjoint cycles so as to minimize the sum of the reload costs of the cycles in $S$. We call this problem Minimum Reload Cost Cycle Cover (MinRC3).[1]

The concept of reload costs was introduced in the seminal paper [11] where various applications are mentioned. This very natural concept, which models a kind of cost incurred during a transportation/transmission activity, has been, amazingly, considered only recently, so that the list of relevant references is comparatively small [1,5,6,9].

We will be interested in asymmetric reload costs and in reload costs that satisfy the *triangle inequality*, where, for undirected graphs, we say that the reload costs satisfy the triangle inequality if, for any three edges $e, e', e''$ incident in a node of the graph, colored, respectively, with colors $l, l', l''$, we have that $r(l, l') \leq r(l, l'') + r(l'', l')$. We will denote the color of an arc $e = (i, j)$ by $c_e$ or $c_{i,j}$.

MinRC3 is solvable in polynomial time when the number $h$ of colors is equal to 1, since in this case we are left with the well known problems of finding a spanning 2-matching of $G$ if the graph is undirected, or a set of directed cycles spanning $N$ if the graph is directed.

* Corresponding author. Fax: +39 0223993412.
*E-mail addresses:* giulia.galbiati@unipv.it (G. Galbiati), stefano.gualandi@unipv.it (S. Gualandi), maffioli@elet.polimi.it (F. Maffioli).

1 For standard definitions on graph related concepts, see for instance [3].

0166-218X/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2011.12.006

In this work we analyze the complexity of MinRC3, we present some integer programming formulations, and we report on preliminary computational results. In Section 2 we prove that, even when $h = 2$, the reload costs are symmetric and satisfy the triangle inequality, MinRC3 is strongly NP-hard and it is not approximable within $\frac{1}{\epsilon}$, for any $\epsilon > 0$. Moreover we show that even in this case, if we additionally require that $r(l, l) \geq 1$ for any $l \in L$, then the problem is not approximable within $O(2^{p(n)})$, for any polynomial $p()$. In Section 3 we present a bilinear formulation, its standard linearized version, and a formulation suitable for a Column Generation approach. In Section 4 we describe a local search approach for obtaining primal bounds. In Section 5 some preliminary computational results are reported. Ongoing work and directions for further research are also discussed.

## 2. Complexity

In this section we analyze the complexity and the approximability of problem MinRC3 formulated on undirected graphs. The results are presented in Theorem 1, Corollaries 1 and 2. It is straightforward to show that these results also hold for MinRC3 formulated on directed graphs.

The first result of this section derives from a reduction of the well known Vertex Cover problem to the recognition form of MinRC3, which we call Reload Cost Cycle Cover(RC3). We begin by giving the definitions of these problems and by presenting the reduction.

Vertex Cover: An *instance I* of Vertex-Cover consists of an undirected graph $G = (V, E)$ and a positive integer $k$. The *question* is whether there exists a subset $S$ of $V$ having $|S| \leq k$ and covering all edges in $E$, i.e. such that for each edge $e \in E$ one of its end vertices belongs to $S$.

Reload Cost Cycle Cover: An *instance* $I'$ of RC3 consists of an undirected graph $G' = (N, A)$, a finite set $L$ of colors, a function $c : A \to L$ that assigns a color to each edge, a reload cost function $r : L \times L \to \mathbb{Z}^+$, and a positive integer $k'$. The *question* is whether there exists a set $C$ of node-disjoint cycles spanning the set $N$ of nodes and having a reload cost at most equal to $k'$.

The reduction that we now propose for building in polynomial time, given an instance $I$ of Vertex Cover, a corresponding instance $I'$ of RC3, is a modification of the reduction described in [7] that reduces Vertex Cover to Hamiltonian Circuit. We do not describe here the entire reduction, since it is practically identical to the very well known one presented in [7], to which we refer and to which we invite the reader to refer also for the notations, but we focus our attention only on the differences in the two reductions that allow us to conclude that Lemma 1 holds. The difference among the reductions lies in the so called "cover-testing component" for an edge $e = \{u, v\}$ of $G$ and, of course, in the coloring of the edges of $G'$.

In Fig. 1, for sake of clarity, we illustrate the cover-testing component used in [7] and in our reduction. In our reduction, for each $e = \{u, v\}$ of $G$ there are in $G'$ five new vertices $(u, e, 0)$, $(v, e, 0)$, $uv_1$, $uv_2$, $uv_3$ connected by new edges as illustrated. Notice that in Fig. 1(b) the colors of the edges are also depicted, with edges drawn with light lines having color 1 and edges drawn with heavy lines having color 2. To conclude the presentation of our reduction we specify that all other edges of $G'$ have color 1 so that $L = \{1, 2\}$, the reload cost function $r$ is such that $r(l_1, l_2) = 0$ if $l_1 = l_2$ and $r(l_1, l_2) = 1$ if $l_1 \neq l_2$, and finally the integer $k'$ is set equal to 0.

**Lemma 1.** *Let I be an instance of* Vertex Cover *and $I'$ be the corresponding instance of* Reload Cost Cycle Cover. *There exists in $G = (V, E)$ a subset $S$ of $V$ having $|S| \leq k$ and covering all edges in $E$ if and only if there exists in $G' = (N, A)$ a set of node-disjoint cycles spanning the set $N$ of nodes and having a reload cost equal to* 0.

**Proof.** Suppose there exists in $G$ a set $S$ of $k$ vertices covering all edges in $E$. If we disregard for a while all vertices $uv_i$, $i = 1, 2, 3$, for each $\{u, v\} = e \in E$, the same reasoning in [7] allows us to assert that there is a cycle, made with edges of color 1, that goes thorough the $k$ "selector" vertices $a_1, \ldots, a_k$, (so called in [7]), and the remaining vertices of $G'$ (except possibly some $(u, e, 0)$ or $(v, e, 0)$); this cycle, within the cover testing component for edge $e = \{u, v\}$, has one of the three possible configurations illustrated in Fig. 2, where (a)–(c) correspond to the cases in which $u$ belongs to $S$ but $v$ does not, both $u$ and $v$ belong to $S$, $v$ belongs to $S$ but $u$ does not.

From this observation it is straightforward to see that in the three cases it is possible to span all vertices $uv_i$, $i = 1, 2, 3$, $\{u, v\} = e \in E$ (and the unspanned $(u, e, 0)$ or $(v, e, 0)$) with a node-disjoint cycle of color 2 (also illustrated in Fig. 2), therefore obtaining a set of node-disjoint cycles spanning the nodes of $G'$ and having a reload cost equal to 0. Suppose on the contrary that $G'$ has a set $C$ of node-disjoint cycles spanning the set $N$ of nodes and having reload cost equal to 0. In this case each cycle in $C$ cannot have edges of both colors. Therefore, for each edge $e = \{u, v\}$, the vertices $uv_i$, $i = 1, 2, 3$ must be spanned by a cycle of the three types illustrated in Fig. 2 and this implies that the remaining vertices $(u, e, i)$, $(v, e, i)$, $i = 1, \ldots, 6$ must be spanned by edges of color 1, again in one of the tree types illustrated in Fig. 2. At this point it is easy to conclude, as done in [7], that any portion of the cycles with edges of color 1 that begins at one selector vertex and ends at a selector vertex without passing through any other selector vertex, corresponds to those edges from $E$ that are incident to some particular vertex in $V$. The cycles passing through the $k$ selector vertices identify $k$ such portions that identify the $k$ vertices of $G$ that cover all the edges in $E$. □

Since in the reduction presented in the proof of Lemma 1 the number of colors of the edges of $G'$ is 2, the reload costs are symmetric, and satisfy the triangle inequality, we may conclude that the following theorem is true.

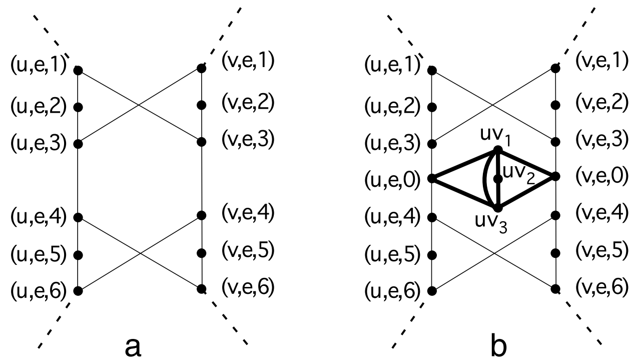


**Fig. 1.** Cover-testing component for edge $e = \{u, v\}$ used in transforming Vertex Cover to Hamiltonian Circuit (a) and to RC3 (b).

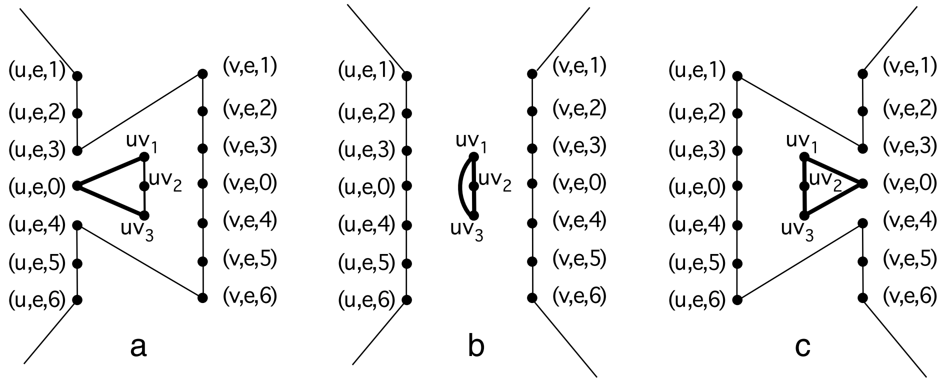


**Fig. 2.** The three possible configurations of a cycle within the cover-testing component for edge $e = \{u, v\}$, corresponding to (a) $u$ belongs to the cover but $v$ does not, ($b$) both $u$ and $v$ belong to the cover, (c) $v$ belongs to the cover but $u$ does not.

**Theorem 1.** MinRC3 *is strongly* NP-*hard even if the number of colors is* 2, *the reload costs are symmetric, and satisfy the triangle inequality.*

**Corollary 1.** MinRC3 *is not approximable within* $\frac{1}{\epsilon}$, *for any* $\epsilon > 0$, *even if the number of colors is* 2, *the reload costs are symmetric, and satisfy the triangle inequality.*

**Proof.** We refer to the reduction used in the proof of Lemma 1. If we denote by $\mathrm{OPT}(G')$ the reload cost of an optimum solution of $G'$, this reduction shows that $\mathrm{OPT}(G') = 0$ if and only if $I$ is a satisfiable instance of Vertex Cover and also shows that it is NP-complete to distinguish between $\mathrm{OPT}(G') = 0$ and $\mathrm{OPT}(G') \geq 1$. Therefore the reduction introduces a gap and by the gap reduction technique the problem MinRC3 is not approximable within $\frac{1}{\epsilon}$, for any $\epsilon > 0$. □

In the more general case, where a reload cost is paid even when there is not a change of color, we have the following negative result.

**Corollary 2.** *If* $r(l, l') \geq 1$ *for every* $l, l' \in L$, *then* MinRC3 *is not approximable within* $O(2^{p(n)})$ *for every polynomial* $p()$, *even if the number of colors is* 2, *the reload costs are symmetric, and satisfy the triangle inequality.*

**Proof.** If in the reduction in Lemma 1 we modify the reload cost matrix $r$ so that $r(1, 1) = r(2, 2) = 1$ and $r(1, 2) = r(2, 1) = Mn'$, with $M = O(2^{p(n')})$ and $n'$ denoting the number of vertices in $G'$, then we deduce that it is NP-complete to distinguish between $\mathrm{OPT}(G') = n'$ and $\mathrm{OPT}(G') > Mn'$. Using again the gap reduction technique the problem is not approximable within $O(2^{p(n)})$. □

## 3. Integer programming models

In this section we deal with MinRC3 formulated on directed graphs. Let $G = (N, A)$ be a directed graph, $L$ a set of $h$ colors, $r : L \times L \to \mathbb{Z}^+$ a reload cost function, and $c : A \to L$ an arc coloring. We denote by $\delta^+(i)$ and by $\delta^-(i)$ the set of outgoing and incoming arcs at node $i$, respectively.

MinRC3 has the following Integer Bilinear formulation:

$$\min \sum_{i \in N} \sum_{\substack{e \in \delta^-(i), \\ f \in \delta^+(i)}} r(c_e, c_f) x_e x_f \tag{1}$$

$$\text{s.t.} \sum_{e \in \delta^+(i)} x_e = \sum_{e \in \delta^-(i)} x_e = 1, \quad \forall i \in N, \tag{2}$$

$$x_e \in \{0, 1\}, \quad \forall e \in A. \tag{3}$$

The binary variable $x_e$ is equal to 1 if arc $e$ belongs to the cycle cover, and 0 otherwise. Constraints (2) force to select a single outgoing arc and a single incoming arc for each node, respectively. Note that the bilinear term $x_e x_f$ in (1) can be linearized with standard techniques, as shown in the formulation below, where we let $z_{ef}$ be a 0–1 variable equal to 1, if both $x_e$ and $x_f$ are equal to 1, and 0 otherwise.

$$\min \sum_{i \in N} \sum_{\substack{e \in \delta^-(i), \\ f \in \delta^+(i)}} r(c_e, c_f) z_{ef} \tag{4}$$

$$\text{s.t. } (2), (3), \tag{5}$$

$$z_{ef} \geq x_e + x_f - 1, \quad \forall i \in N,\ \forall e \in \delta^-(i),\ \forall f \in \delta^+(i), \tag{6}$$

$$z_{ef} \geq 0, \quad \forall i \in N,\ \forall e \in \delta^-(i),\ \forall f \in \delta^+(i). \tag{7}$$

As many covering problems, MinRC3 can be formulated with an exponential number of columns with respect to the number of vertices, and be solved via the Column Generation approach [10]. The idea is to have a set partitioning formulation, where each column $j$ represents a cycle of reload cost $w_j$. Let $\mathcal{C}$ be the collection of every possible directed cycle in $G$. MinRC3 can then be formulated as follows:

$$\min \sum_{c \in \mathcal{C}} w_c \lambda_c \tag{8}$$

$$\text{s.t.} \sum_{c \in \mathcal{C}: i \in c} \lambda_c = 1, \quad \forall i \in N, \tag{9}$$

$$\lambda_c \in \{0, 1\}, \quad \forall c \in \mathcal{C}. \tag{10}$$

## 3.1. Column Generation approach

Since the number of directed cycles in $\mathcal{C}$ is exponential with respect to the number of vertices, we start with a subset $\bar{\mathcal{C}} \subseteq \mathcal{C}$ such that the constraints in (9) are satisfied, and we relax the integrality constraints. We get the following restricted master problem:

$$\min \sum_{c \in \bar{\mathcal{C}}} w_c \lambda_c \tag{11}$$

$$\text{s.t.} \sum_{c \in \bar{\mathcal{C}}: i \in c} \lambda_c = 1, \quad \forall i \in N, \tag{12}$$

$$\lambda_c \geq 0, \quad \forall c \in \bar{\mathcal{C}}. \tag{13}$$

Let $\bar{\pi}$ be the vector of dual multipliers of constraints (12). Then, the pricing subproblem, that is the problem of generating a negative reduced-cost column for the restricted master problem, is as follows:

$$\min \sum_{i \in N} \sum_{\substack{e \in \delta^-(i), \\ f \in \delta^+(i)}} r(c_e, c_f) x_e x_f - \sum_{i \in N} \bar{\pi}_i y_i \tag{14}$$

$$\text{s.t.} \sum_{e \in \delta^+(i)} x_e = \sum_{e \in \delta^-(i)} x_e = y_i, \quad \forall i \in N, \tag{15}$$

$$\sum_{e \in A} x_e \geq 2, \tag{16}$$

$$x_e \in \{0, 1\}, \quad \forall e \in A, \tag{17}$$

$$y_i \in \{0, 1\}, \quad \forall i \in N. \tag{18}$$

Note that the bilinear term in (14) can be linearized as before. The constraints (15)–(16) define a set of vertex disjoint cycles in $G$, and we look for a cycle having a negative minimum cost with respect to the objective (14), since it would correspond to a negative reduced-cost column to be added in (11)–(13). As long as the objective function has a negative value, at least a cycle of negative cost exists.

Looking for a cycle having minimum negative cost with respect to the quadratic objective (14) is itself an interesting combinatorial optimization problem, which we call MINIMUM QUADRATIC CYCLE (MINQCYCLE). Unfortunately only in some restricted cases it turns out to be solvable in polynomial time, but in general it is an NP-hard problem. The negative and positive results for the complexity of this problem are given in Theorem 2 and in Theorem 3 of the next subsection.

#### *3.1.1. Complexity of the pricing*

The negative result is based on a reduction described in [9] from the 2-BALANCED 3-SAT problem ((3, B2)-SAT for short). An instance $I$ of (3, B2)-SAT is a set $C$ of CNF clauses defined over a set $X$ of Boolean variables, where each clause has exactly 3 literals, each of them appearing exactly 4 times in the clauses, twice negated and twice unnegated. Deciding whether an instance of 2-BALANCED 3-SAT is satisfiable is NP-complete [2]. An instance $I'$ of MINQCYCLE is like an instance of RC3, with the addition that to each node $i$ of $G' = (N, A)$ is assigned an integer weight $w(i) \in \mathbb{Z}$.

**Theorem 2.** MINQCYCLE *is strongly* NP-*hard even if the number of colors is* 3*, the reload costs are symmetric, satisfy the triangle inequality and the maximum degree of* $G'$ *is* 4*.*

**Proof** (*Sketch*)**.** We prove this result by slightly modifying the reduction described in Theorem 8 of [9] for proving, under the same hypothesis, that the *Minimum Symmetric Reload s-t path problem* is NP-hard. We do not present here such reduction, which is long and very technical, but we invite the reader to refer to it for a comprehension of the rest of this proof. Here we simply describe the modification that we perform to the reduction in [9], in order to build the bi-connected graph $G'$ (which plays the role of graph $G^c$ in [9]) of instance $I'$ of MINQCYCLE. The modification consists in the addition of an edge of color 3 between nodes $s$ and $t$, and in the assignment of the weights to the nodes of $G'$, which are all set equal to 1 except for node $s$ which receives weight 2. We prove that an instance $I$ of (3, B2)-SAT is satisfiable if and only if graph $G'$ of instance $I'$ has a cycle $C$ with reload cost $r(C)$ less than the sum $w(C)$ of the weights of its nodes. It is not too difficult to see, similarly as in [9], that if a truth assignment $\tau$ satisfies instance $I$ then graph $G'$ admits a cycle $C$ where there is no change from color 1 to color 2; this cycle has reload cost $r(C) = 11|X| + 3|C| + 2$ and weight $w(C) = r(C) + 1$, so that $r(C) < w(C)$. On the other hand any cycle of $G'$ that does not include $s$ has a weight equal to the number $n$ of its nodes, and a reload cost at least equal to this number; hence if a cycle $C$ of $G'$ has $r(C) < w(C)$ it must go through $s$ and $t$: since in this case its weight is equal to $n + 1$, it must be that $r(C) \leq n$ and the $s - t$ path that does not include the edge $(s, t)$ must have a reload cost less then $n - 1$. This is enough to conclude that this path does not allow any change from color 1 to color 2 and therefore must touch $11|X| + 3|C| + 2$ nodes and have reload cost equal to $11|X| + 3|C|$; the conclusion that this path induces a truth assignment that satisfies $I$ follows therefore as in [9]. □

**Theorem 3.** MINQCYCLE *is solvable in polynomial time if the reload costs are symmetric,* $G'$ *is bi-connected and has maximum degree* 3*.*

**Proof** (*Sketch*)**.** We use the construction used to prove Theorem 4 of [9], slightly modified to reach our purposes, and again we invite the reader to refer to it for a comprehension of the rest of this proof.[2] Let us remark the reader that this construction starts from a colored graph $G^c$ with reload costs assigned, where two distinct vertices $s$ and $t$ are chosen, and builds an edge weighted non-colored graph $G$ in such a way that an $s - t$ path of $G^c$ joining vertices $s$ and $t$ there corresponds in $G$ a perfect matching and vice versa; moreover the reload cost of the $s - t$ path is equal to the weight of the perfect matching. The modification to the construction, that we perform here on the bi-connected graph $G'$ of instance $I'$ (that plays the role of graph $G^c$ in [9]), consists in a different assignment of weights to the edges of non-colored $G$, that takes in consideration not only the reload costs but also the weights that are assigned of the vertices of $G'$. Precisely, using the notations in [9], we set:

$$w(\{v'_{i,j}, p^i_{j.k}\}) = w(\{v'_{i,k}, q^i_{j.k}\}) = \frac{1}{2}(r_{c(\{v_i,v_j\}),\ c(\{v_i,v_k\})} - w(i))$$
$$w(\{s', v_{i,j}\} : v_j = s) = r_{c(\{s,t\}),c(\{s,v_i\})} - w(s)$$
$$w(\{t', v_{i,j}\} : v_j = t) = r_{c(\{s,t\}),c(\{t,v_i\})} - w(t)$$

and all other edges have weight 0.

Consider now any edge of $G'$, called $\{s, t\}$ without loss of generality, a cycle $C$ of $G'$ that uses this edge, and the $s - t$ path that does not include this edge. It is not to difficult to see that, using the weights defined above, the weight of the perfect matching of $G$ corresponding to the $s - t$ path of $G'$ is equal to the reload cost $r(C)$ of the cycle minus the sum $w(C)$ of the weights of its vertices. Hence a minimum weight perfect matching of $G$ identifies a cycle $C_{s,t}$ of $G'$ that uses edge $\{s, t\}$ and minimizes $r(C_{s,t}) - w(C_{s,t})$. If this construction is repeated for each edge $\{s, t\}$ of $G'$ it is possible, in polynomial time, to decide if there exists in $G'$ a cycle $C$ having $r(C) < w(C)$. □

[2] Note that a trail in a graph with maximum degree 3 is always a path.

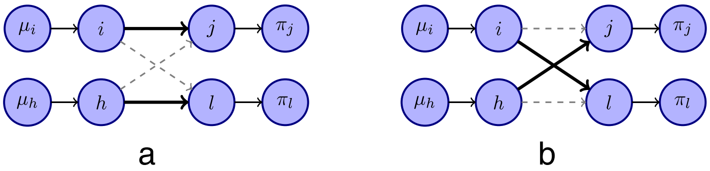


**Fig. 3.** Example of exchanging the two arcs $(i, j)$ and $(h, l)$ with the two arcs $(i, l)$ and $(h, j)$.

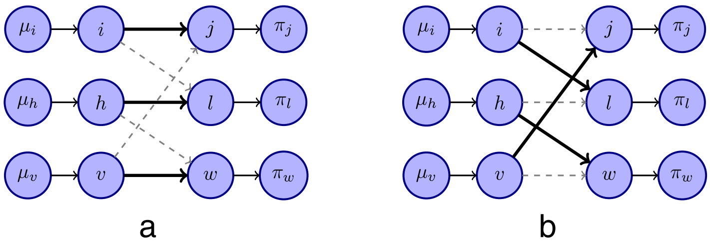


**Fig. 4.** Example of exchanging the three arcs $(i, j)$, $(h, l)$, and $(v, w)$ with the three arcs $(i, l)$, $(h, w)$, and $(v, j)$.

## 4. Primal bounds via local search

In this section we propose local search algorithms based on simple 2-exchange and 3-exchange neighborhoods. These algorithms start with a spanning set of node disjoint cycles, which can be found in polynomial time, as noted in the introduction. The solution space of MinRC3 is represented as a subset of the permutations of the vertices. Let $\pi$ be a permutation of the set $N$, with the additional constraint that $\pi_i = j$ is a valid assignment if and only if $(i, j) \in A$. Note that this implies also $\pi_i \neq i$. Then $\pi$ represents the cycle vertex cover where each vertex $i$ of $N$ has the successor $j = \pi_i$. Let $\mu$ be a vector representing the predecessor of each node, in the solution given by the permutation $\pi$, that is, $\mu_i = j$ if and only if $\pi_j = i$.

Since the vector $\pi$ gives the successor of each node, and the corresponding vector $\mu$ gives the predecessor of each node, the reload cost of the spanning cycle cover given by $\pi$ (and the corresponding vector $\mu$) can be written as:

$$\sum_{i \in N} r(c_{\mu_i,i}, c_{i,\pi_i}). \tag{19}$$

Using this notation is easy to describe a 2-exchange operator: first, select two arcs $(i, j)$ and $(h, l)$ of the current cycle cover, such that the two arcs $(i, l)$ and $(h, j)$ belong to $A$; then, swap the successors of the vertices $i$ and $h$, that is, replace $(i, j)$ and $(h, l)$ with $(i, l)$ and $(h, j)$ (see Fig. 3). In the permutation vector $\pi$, this move is equivalent to swap $\pi_i$ and $\pi_h$. The size of this neighborhood is quadratic, but evaluating the change $\Delta(ij, hl)$ in the objective function (19) due to this operator takes constant time:

$$\begin{aligned} \Delta(ij, hl) = {} & -r(c_{\mu_i,i}, c_{i,j}) - r(c_{i,j}, c_{j,\pi_j}) + r(c_{\mu_i,i}, c_{i,l}) + r(c_{i,l}, c_{l,\pi_l}) \\ & - r(c_{\mu_h,h}, c_{h,l}) - r(c_{h,l}, c_{l,\pi_l}) + r(c_{\mu_h,h}, c_{h,j}) + r(c_{h,j}, c_{j,\pi_j}). \end{aligned} \tag{20}$$

Similarly, it is possible to define a 3-exchange operator: first, select three arcs $(i, j)$, $(h, l)$, and $(v, w)$ of the current cycle cover; second, exchange their successors, that is replace those arcs with arcs $(i, l)$, $(h, w)$, and $(v, j)$ (see Fig. 4). In this case, the neighborhood size is cubic, but evaluating a single move still takes constant time:

$$\begin{aligned} \Delta(ij, hl, vw) = {} & -r(c_{\mu_i,i}, c_{i,j}) - r(c_{i,j}, c_{j,\pi_j}) + r(c_{\mu_i,i}, c_{i,l}) + r(c_{i,l}, c_{l,\pi_l}) \\ & - r(c_{\mu_h,h}, c_{h,l}) - r(c_{h,l}, c_{l,\pi_l}) + r(c_{\mu_h,h}, c_{h,w}) + r(c_{h,w}, c_{w,\pi_w}) \\ & - r(c_{\mu_v,v}, c_{v,w}) - r(c_{v,w}, c_{w,\pi_w}) + r(c_{\mu_v,v}, c_{v,j}) + r(c_{v,j}, c_{j,\pi_j}). \end{aligned} \tag{21}$$

The two neighborhoods are used in the following way: first we perform a best improvement local search [8] using only 2-exchange moves. Then, once we reach a 2-exchange local optimum solution, we perform a few iterations using the more expensive 3-exchange operator, as explained in the next section.

## 5. Computational results

In order to evaluate the approaches presented in the previous sections, we have generated a set of graphs according to the standard $G(n, p)$ Erdös–Rényi model [4], with randomly generated arc colorings. In all instances we generated asymmetric

**Table 1**
Asymmetric reload costs.

| $d$ | $h$ | $n$ | $m$ | Opt. | LB(LP) | LB(Cuts) | LB(CG) |
|---|---|---|---|---|---|---|---|
| 0.3 | 2 | 10 | 28 | 11 | 1.0 | 1.0 | **11.0*** |
| | | 15 | 56 | 11 | 10.3 | 10.3 | **11.0*** |
| | | 20 | 113 | 0 | 0.0 | 0.0 | 0.0 |
| | 3 | 10 | 26 | 30 | 17.0 | 22.0 | **30.0*** |
| | | 15 | 61 | 18 | 0.0 | **18.0*** | 17.0 |
| | | 20 | 116 | 12 | 0.0 | 0.0 | **3.5** |
| | 5 | 10 | 27 | 32 | 23.5 | 27.0 | **32.0*** |
| | | 15 | 48 | 56 | 33.5 | **56.0*** | **56.0*** |
| | | 20 | 113 | 18 | 0.0 | 5.0 | **17.3*** |
| 0.5 | 2 | 10 | 51 | 9 | 0.0 | 0.0 | **1.3** |
| | | 15 | 94 | 0 | 0.0 | 0.0 | 0.0 |
| | | 20 | 189 | 0 | 0.0 | 0.0 | 0.0 |
| | 3 | 10 | 46 | 6 | 0.0 | 0.0 | **6.0*** |
| | | 15 | 101 | 6 | 0.0 | 0.0 | 0.0 |
| | | 20 | 170 | 0 | 0.0 | 0.0 | 0.0 |
| | 5 | 10 | 46 | 18 | 2.3 | 16.7 | **18.0*** |
| | | 15 | 101 | 13 | 0.0 | 0.0 | **10.4** |
| | | 20 | 194 | 0 | 0.0 | 0.0 | 0.0 |

**Table 2**
Symmetric reload costs.

| $d$ | $h$ | $n$ | $m$ | Opt. | LB(LP) | LB(Cuts) | LB(CG) |
|---|---|---|---|---|---|---|---|
| 0.3 | 2 | 10 | 31 | 8 | **8*** | **8*** | **8*** |
| | | 15 | 59 | 20 | 10 | **20*** | **20*** |
| | | 20 | 95 | 0 | 0 | 0 | 0 |
| | 3 | 10 | 30 | 20 | 9.5 | **20*** | **20*** |
| | | 15 | 59 | 42 | 34 | **42*** | **42*** |
| | | 20 | 95 | 20 | 3 | 16.2 | **20*** |
| | 5 | 10 | 30 | 43 | 31 | **43*** | 41 |
| | | 15 | 61 | 29 | 0 | **29*** | **29*** |
| | | 20 | 95 | 28 | 11.7 | 24 | **28*** |
| 0.5 | 2 | 10 | 51 | 0 | 0 | 0 | 0 |
| | | 15 | 105 | 0 | 0 | 0 | 0 |
| | | 20 | 175 | 0 | 0 | 0 | 0 |
| | 3 | 10 | 51 | 8 | 0 | 0 | **8*** |
| | | 15 | 104 | 0 | 0 | 0 | 0 |
| | | 20 | 174 | 0 | 0 | 0 | 0 |
| | 5 | 10 | 50 | 7 | 0 | 4 | **7*** |
| | | 15 | 103 | 4 | 0 | 0 | **3.7*** |
| | | 20 | 176 | 2 | 0 | 0 | **1.2*** |

reload costs that do not necessarily satisfy the triangle inequality, with $r(l, l) = 0$ for all $l \in L$. Note that the focus of the computational results is on the quality of the lower bounds and not on the computational times.

In the following, let $n = |N(G)|$, $m = |A(G)|$, and $d = \frac{m}{n(n-1)}$ denote the density of the graph. All tests are run on a standard desktop with 2Gb of ram, a i686 CPU at 1.8 GHz, using CPLEX12.2 as Integer Linear Programming solver.

## 5.1. Computing lower bounds

Preliminary experiments showed that only very small instances of MinRC3 can be solved to optimality. Although the bilinear formulation (1)–(3) could be solved using CPLEX, only digraphs up to 15 nodes and 3 colors are solved within a time limit of an hour. On the other hand, the linearized version (4)–(7) is more efficient, since CPLEX is able to automatically generate a number of cuts that improve the formulation, and digraphs up to 20 nodes and 5 colors are solved to optimality. Nevertheless, the lower bounds at the root node, after the automatic generation of CPLEX, are still quite loose. Therefore, in order to obtain tighter lower bounds, we have experimented with the Column Generation formulation (11)–(13).

Tables 1 and 2 show the comparison of three different lower bounds on two small set of instances, with asymmetric and symmetric costs, respectively. The first lower bound LB(LP) is the lower bound obtained with the linear relaxation of problem (4)–(7); the second lower bound LB(Cuts) is the lower bound obtained after the automatic generation of cuts by CPLEX; the third lower bound LB(CG) is obtained via the Column Generation approach. A bold entry indicates the best bound, while a '$*$' denotes that the bound equals the optimum value. This notation is not used when the optimum value is zero.

The results show that the Column Generation approach provides, in general, tighter lower bounds. The lower bounds are stronger for sparse digraphs and for high number of colors. In Table 1 the lower bounds are loose in two of the cases

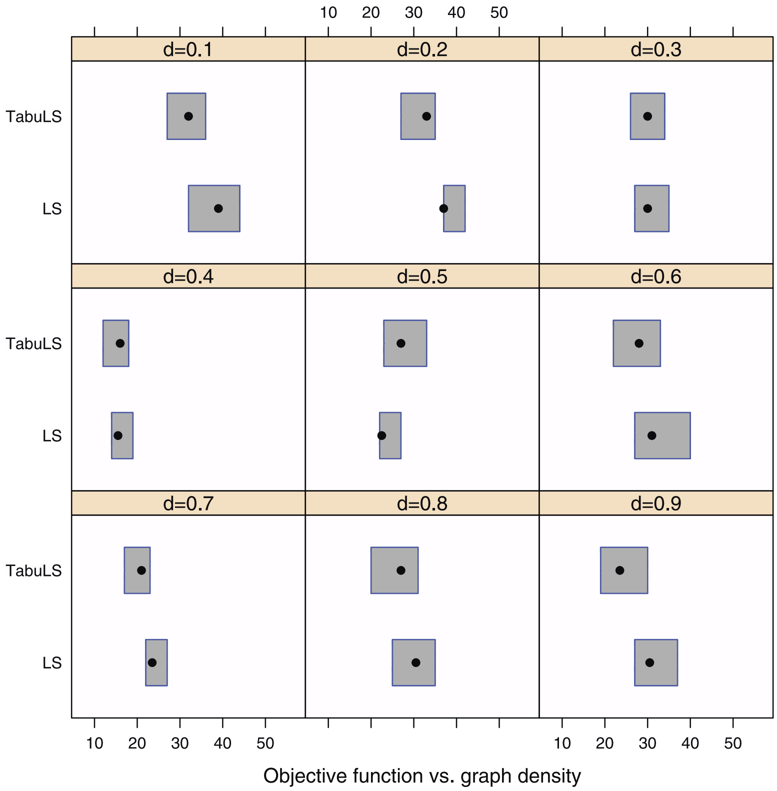


**Fig. 5.** Comparison of the results of the local search with the tabu mechanism (TabuLS) and the local search (LS) in terms of the objective function values on graphs with different densities and $h = 5$ colors. Reload costs are asymmetric.

considered, i.e. for the instance with density 0.3%, $h = 3$, and $n = 20$, where the optimum reload cost is 12, but the lower bound is 3.5, and for the instance with density 0.5%, $h = 2$, and $n = 10$, where the optimum reload cost is 9, but the lower bound is 1.3. However in most of the other cases the LB(CG) gives the optimum. There are not significant differences between the results with asymmetric and symmetric costs.

### *5.2. Computing upper bounds*

Using the local search algorithms presented in Section 4 we are able to solve instances with a higher number of nodes. We have considered two variants of the local search algorithm. The first variant is a basic local search algorithm that executes 200 iterations using only the 2-exchange neighborhood, and then 20 iterations with the 3-exchange neighborhood, where each iteration selects the best improvement in the neighborhood. The second variant adds a tabu mechanism to the algorithm: once a move is selected, it becomes *tabu*, and it is not repeated in the subsequent $t$ iterations (we set $t = 7$) [8].

In order to assess the two variants of local search algorithm we have performed two set of experiments. In the first set of experiments we have $n = 50$, $h = 5$, and a density ranging from $d = 0.1$ up to $d = 0.9$. Fig. 5 shows the box-plots of the run-time distributions. For each instance we have executed 50 runs. In the box we report the results of the objective function between the 1st and the 3rd quartile, and the point in the box gives the medians. The basic local search algorithm takes in average 0.6 s, while adding the tabu mechanism the algorithm takes in average 1.1 s. Note that with this addition the algorithm gives clearly better results for graph with density $d = \{0.1, 0.2, 0.6, 0.7, 0.8, 0.9\}$, while for density $d = \{0.3, 0.4, 0.5\}$ the two methods give similar results.

The second set of experiments shows the impact of the number of colors on graphs with $n = 50$ and density $d = 0.3$. Fig. 6(a) and (b) show that when $h = 2$ there always exists a spanning cycle cover of zero cost, and the basic version of the local search algorithm is much faster (we can stop the algorithm as soon as we find a zero cost solution). For $h = 3$, the median of the tabu search algorithm is better, but the results are still similar; for $h = 5$ the tabu search gives better results, taking however roughly double computational time.

## 6. Conclusions

We have introduced the new Combinatorial Optimization problem MinRC3, and analyzed its complexity and approximability properties. We have proposed some IP formulations, a Column Generation approach, and Local Search based

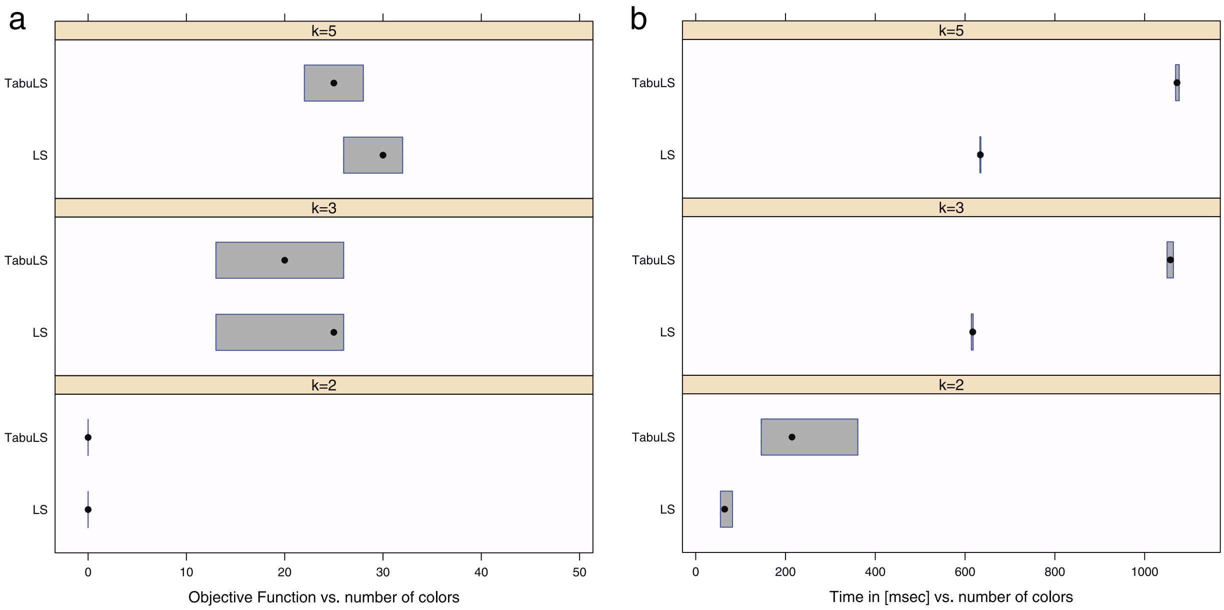


**Fig. 6.** Comparison of the results of the local search with the tabu mechanism (TabuLS) and the local search (LS) in terms of the objective function values (a) and the computation time (b) on graphs with different numbers of colors, i.e., $h = \{2, 3, 5\}$, and density $d = 0.3$. Reload costs are asymmetric.

heuristics. The results of some preliminary computational experiments have been reported. Future research could focus on improving algorithmic performance, extending computational experiments, identifying other approaches for obtaining tighter dual bounds (e.g. via Semidefinite Programming), and evaluating the suitability of a branch-and-price approach.

## References

[1] E. Amaldi, G. Galbiati, F. Maffioli, On minimum reload cost paths, tours and flows, Networks 57 (2011) 254–260.
[2] P. Berman, M. Karpimsky, A.D. Scott, Computational complexity of some restricted instances of 3-SAT, Discrete Appl. Math. 155 (2007) 649–653.
[3] B. Bollobás, Modern Graph Theory, Springer Verlag, 1998.
[4] P. Erdős, A. Rényi, On random graphs, Publ. Math. 6 (2) (1959) 290–297.
[5] G. Galbiati, The complexity of a minimum reload cost diameter problem, Discrete Appl. Math. 156 (2008) 3494–3497.
[6] I. Gamvros, L. Gouveia, S. Raghavan, Reload cost trees and network design, Networks (2011) doi:10.1002/net.20443.
[7] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, 1979.
[8] F. Glover, M. Laguna, Tabu Search Textquotedblright, Kluwer Academic Pub, 1998.
[9] L. Gourvè, A. Lyra, C. Martinhon, J. Monnot, The minimum reload s-t path, trail and walk problems, Discrete Appl. Math. 158 (13) (2010) 1404–1417, Elsevier.
[10] M.E. Lübbecke, J. Desrosiers, Selected topics in column generation, Oper. Res. 53 (6) (2005) 1007–1023. INFORMS.
[11] H. Wirth, J. Steffan, Reload cost problems: minimum diameter spanning tree, Discrete Appl. Math. 113 (2001) 73–85.