Cardinality constrained and multicriteria (multi)cut problems

C. Bentz, M.-C. Costa, N. Derhy *, F. Roupin

CEDRIC-CNAM, 292 Rue Saint-Martin, 75141 Paris Cedex 03, France

Abstract

In this paper, we consider multicriteria and cardinality constrained multicut problems. Let G be a graph where each edge is weighted by R positive costs corresponding to R criteria and consider k source-sink pairs of vertices of G and R integers $B_1, \ldots, B_R$. The problem R-CriMultiCut consists in finding a set of edges whose removal leaves no path between the i^{th} source and the i^{th} sink for each i, and whose cost, with respect to the j^{th} criterion, is at most B_j, for $1 \leq j \leq R$. We prove this problem to be $\mathcal{NP}$-complete in paths and cycles even if $R = 2$. When $R = 2$ and the edge costs of the second criterion are all 1, the problem can be seen as a monocriterion multicut problem subject to a cardinality constraint. In this case, we show that the problem is strongly $\mathcal{NP}$-complete if $k = 1$ and that, for arbitrary k, it remains strongly $\mathcal{NP}$-complete in directed stars but can be solved by (polynomial) dynamic programming algorithms in paths and cycles. For $k = 1$, we also prove that R-CriMultiCut is strongly $\mathcal{NP}$-complete in planar bipartite graphs and remains $\mathcal{NP}$-complete in $K_{2,d}$ even for $R = 2$.

Key words: Multicut, Cardinality Constraint, Multicriteria Optimization, Dynamic Programming, $\mathcal{NP}$-hardness

1 Introduction

In (2), Bruglieri et al. study a generalization of the well-known minimum cut problem where an additional cardinality constraint is considered. They show that the problems of finding a minimum cut of cardinality either equal to or greater than a given value p are both strongly $\mathcal{NP}$-hard. However, they

* Corresponding author. Tel.: +33 (0) 1 58 80 85 50; fax: +33 (0) 1 40 27 22 96.
Email addresses: `cedric.bentz@lri.fr` (C. Bentz), `costa@cnam.fr` (M.-C. Costa), `nicolas.derhy@cnam.fr` (N. Derhy), `roupin@cnam.fr` (F. Roupin).

ask whether the problem MinCutCard, where we look for a minimum cut separating the source s and the sink t, and whose cardinality is at most p, can be solved in polynomial time.

In fact, the decision version of this problem can be seen as a particular case of a multicriteria simple cut problem. In the problem R-CriCut, we are given two vertices s and t, R edge-weight positive functions $w_1, \dots, w_R$, R bounds $B_1, \dots, B_R$ and we look for a cut C which separates s and t such that $w_i(C) \leq B_i \ \forall i \in \{1, \dots, R\}$. For $R = 2$, if we set $w_2(e) = 1 \ \forall e \in E$ and $B_2 = p$, we obtain the decision version of MinCutCard. 2-CriCut has been shown strongly $\mathcal{NP}$-complete for general graphs in (11). Besides, when we look for a global cut of the graph, i.e. a partition of the vertices into two connected components, the problem is polynomial when the number of criteria is bounded (1).

Let MinMultiCutCard and R-CriMultiCut be generalizations of MinCutCard and R-CriCut respectively, defined as the cardinality constrained and the muticriteria versions of the multicut problem. Given a (directed or not) graph $G = (V, E)$ and a set $T = \{(s_1, t_1), \dots, (s_k, t_k)\}$ of k distinct source-sink pairs of terminal vertices, a multicut C is a subset of E whose removal leaves no (directed) path between s_i and t_i for each $i \in \{1, \dots, k\}$. MinMultiCutCard and R-CriMultiCut can then be defined from MinCutCard and R-CriCut respectively, by replacing "cut" by "multicut". For fixed $k > 2$, the minimal multicut problem MinMultiCut (i.e. the optimization version of 1-CriMultiCut) is APX-hard both in undirected and in directed graphs (5). For arbitrary values of k, it is APX-hard in undirected stars (i.e. trees of height 1) (8) but becomes polynomial in directed trees (4).

Obviously, the difficult cases of MinMultiCut are difficult for MinMultiCutCard and R-CriMultiCut. The question is then: do the polynomial cases of MinMultiCut remain polynomial when we add a cardinality constraint or when we consider the multicriteria version?

We study these problems and provide some answers in this paper, which is divided into three sections.

The first one deals with simple cut problems. We show that MinCutCard is strongly $\mathcal{NP}$-hard thus settling one of the open problems of Bruglieri et al. in (2). Then, we prove that, in planar bipartite graphs, 2-CriCut is $\mathcal{NP}$-complete and R-CriCut is strongly $\mathcal{NP}$-complete.

In the second section, we show that MinMultiCutCard is strongly $\mathcal{NP}$-hard in directed stars but remains polynomial in paths (and directed paths) and cycles (and circuits).

In the third section, we study R-CriMultiCut. We show that, in paths, this problem is strongly $\mathcal{NP}$-complete and remains $\mathcal{NP}$-complete for $R = 2$. Finally, when the number of criteria is bounded, we give a sketch of a pseudo-polynomial algorithm for R-CRIMULTICUT in paths.

2 Simple Cut problems

As already mentioned, Bruglieri et al. study in (2) the problem of finding a minimal cut subject to a cardinality constraint. However, MINCUTCARD (i.e. the case where we have an upper bound on the cardinality of the cut) was an open problem. We show the following theorem:

Theorem 1 MINCUTCARD *is strongly* $\mathcal{NP}$*-hard.*

PROOF. We use a reduction from BISECTION (7). Let $G = (V, E)$ be an undirected graph with $2n$ vertices and m edges, and let B be a given value. The problem is to decide if there exists a partition of V into two disjoint sets V_1 and V_2 such that $|V_1| = |V_2| = n$ and such that the number of edges with one endpoint in V_1 and one endpoint in V_2 is less than or equal to B. Let I_{bi} be an instance of BISECTION. We assume that $B < m$, otherwise I_{bi} would obviously have a solution.

We construct an instance I_{cut} of the decision version of MINCUTCARD as follows (see Figure 1): first, let us assign weight 1 to the edges of G. Then, we add a vertex t and $2n$ edges of weight $nm + n^2 + m$ connecting t to each vertex of G. For each vertex v_i of G, we add a path q_i of $m + n$ vertices and we add $m + n$ edges connecting each vertex of q_i to v_i. The edges of q_i and the edges connecting the vertices of q_i to v_i have a weight equal to $(nm+n^2+m)n+(m+n)n+m$. Finally we add a vertex s and $2(m+n)n$ edges of weight 1 connecting s to all the vertices of the paths q_i ($i \in \{1, \ldots, 2n\}$).

We claim that there exists a solution for I_{bi} if and only if there exists a cut separating s from t such that $w(C) \leq (nm + n^2 + m)n + (m + n)n + B$ and $|C| \leq n + (m + n)n + B$

If we have a solution of I_{bi}, we construct a solution for I_{cut} in the following way. For each vertex v_i of V_1, we cut the edge connecting v_i to t. For each vertex v_i of V_2, we cut the edges connecting the vertices of q_i to s. Moreover, we cut the edges of G with one endpoint in V_1 and one endpoint in V_2. So, the cut separates $\{s\} \cup V_1 \cup \{v \in q_i | v_i \in V_1\}$ from $\{t\} \cup V_2 \cup \{v \in q_i | v_i \in V_2\}$. We have $|C| \leq |V_1| + (m + n)|V_2| + B = n + (m + n)n + B$ and
$w(C) \leq (nm+n^2+m)|V_1|+(m+n)|V_2|+B = (nm+n^2+m)n+(m+n)n+B$.

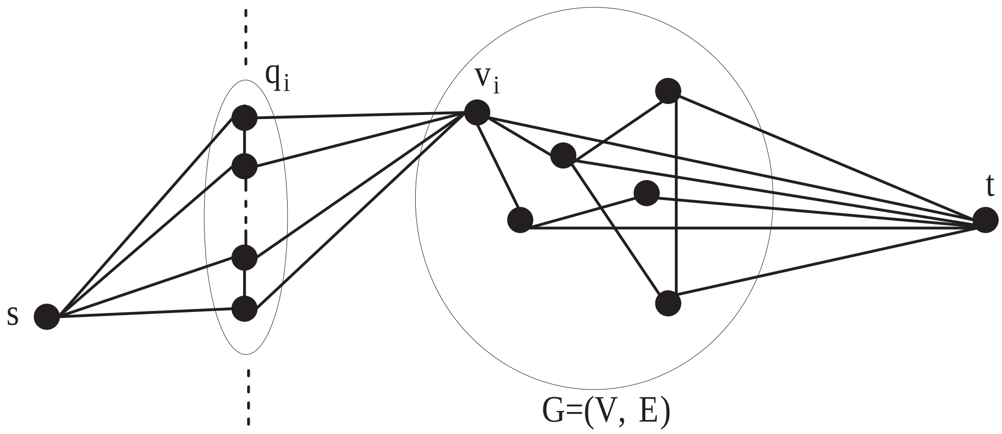

Figure 1. The graph obtained for MinCutCard ($|q_i| = m + n$)

Conversely, if we have a solution C of I_{cut}, we construct a solution of I_{bi} in the following way: V_1 is composed by the vertices of G connected to s and V_2 by the vertices of G connected to t. Note that no edge of weight $(nm + n^2 + m)n + (m + n)n + m$ can be in C since $B < m$ and
$w(C) \leq (nm + n^2 + m)n + (m + n)n + B$.
Let us begin by showing that $|V_1| = |V_2| = n$.
$|V_2| \geq n$, because otherwise, we have to cut at least $n + 1$ edges connecting vertices of G to t, so:
$w(C) \geq (n + 1)(nm + n^2 + m) = (nm + n^2 + m)n + (m + n)n + m > (nm + n^2 + m)n + (m + n)n + B$, which is not possible.
$|V_1| \geq n$ otherwise, we would have to cut at least $(n + 1)(m + n)$ edges connecting vertices of q_i to s $(i/v_i \in V_2)$ so:
$|C| \geq (n + 1)(m + n) = (m + n)n + m + n > (m + n)n + B + n$, and the cardinality constraint would be violated.
Thus, since $|V_1| + |V_2| = 2n$, we necessarily have $|V_1| = |V_2| = n$.
Finally, we have to show that the number of edges with one endpoint in V_1 and one endpoint in V_2 is less than or equal to B. We have cut n edges connecting vertices of G to t and $(m + n)n$ edges connecting vertices of q_i to s $(i|v_i \in V_2)$. Moreover, the total number of edges in the cut is less than or equal to $n + (m + n)n + B$. Thus, the number of edges of G in the cut is less than or equal to B. □

In (11), Papadimitriou and Yannakakis show that 2-CriCut, a problem more general than MinCutCard, is strongly $\mathcal{NP}$-complete in general graphs. We now give some complexity results concerning particular bipartite graphs.

Theorem 2 2-CriCut *is $\mathcal{NP}$-complete in $K_{2,d}$ even when s and t are the two vertices of degree greater than two.*

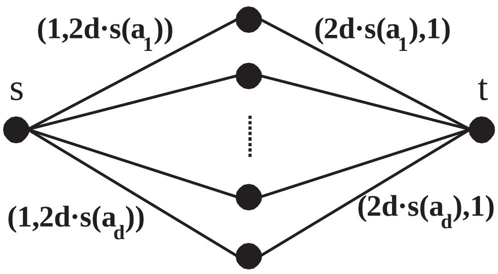

Figure 2. The graph $K_{2,d}$ obtained for 2-CRICUT

PROOF. We use a reduction from PARTITION (6). We are given a finite set A of d elements, a size $s(a) \in \mathbb{N}^*$ for each $a \in A$ and a value S such that $\sum_{a\in A} s(a) = S$. The problem is to decide if there exists a subset A' of A such that $\sum_{a\in A'} s(a) = \sum_{a\notin A'} s(a) = \frac{S}{2}$. Let I_{part} be an instance of PARTITION.

We construct an instance I_{cut} of 2-CRICUT as follows (see Figure 2). Let $G = (V_1, V_2, E)$ be a complete bipartite graph with $V_1 = \{s, t\}$ (so that $|V_1| = 2$) and $|V_2| = d$. Thus, there are d disjoint paths P_a $(a \in A)$ of length 2 linking s to t. For each $a \in A$, let e_a and e'_a be the two edges of P_a. We set $w_1(e_a) = w_2(e'_a) = 1$ and $w_2(e_a) = w_1(e'_a) = 2d \cdot s(a)$. We claim that there exists a solution for I_{part} if and only if there exists a cut C such that $w_1(C) \le d + dS$ and $w_2(C) \le d + dS$.

Indeed, suppose that we have a solution A' for I_{part}. Let us construct a solution C for I_{cut}: for each $a \in A'$ we cut the edge e_a and for each $a \notin A'$ we cut the edge e'_a. Thus, there is exactly one edge of each P_a in C and we have $w_1(C) = \sum_{a\in A'} w_1(e_a) + \sum_{a\notin A'} w_1(e'_a) = \sum_{a\in A'} 1 + 2d\sum_{a\notin A'} s(a) = |A'| + dS \le d + dS$ and $w_2(C) = \sum_{a\in A'} w_2(e_a) + \sum_{a\notin A'} w_2(e'_a) = dS + (d - |A'|) \le dS + d$.

Conversely, suppose that we have a solution C for I_{cut}. We construct a solution for I_{part} as follows: for each path P_a, if e_a is cut then $a \in A'$ else $a \notin A'$. We must now verify that the constructed set A' satisfies $\sum_{a\in A'} s(a) = \sum_{a\notin A'} s(a) = \frac{S}{2}$.
Since C is a solution for I_{cut}, we have $w_1(C) \le d + dS$.
Furthermore, by construction we have $w_1(C) \ge \sum_{a\in A'} w_1(e_a) + \sum_{a\notin A'} w_1(e'_a) = |A'| + 2d\sum_{a\notin A'} s(a)$.
So, $d + dS \ge |A'| + 2d\sum_{a\notin A'} s(a)$ and we have:

$$\sum_{a\notin A'} s(a) \le \frac{S}{2} + \frac{d - |A'|}{2d} < \frac{S}{2} + 1$$

Using the same arguments for w_2 yields:

$$\sum_{a\in A'} s(a) \le \frac{S}{2} + \frac{|A'|}{2d} < \frac{S}{2} + 1$$

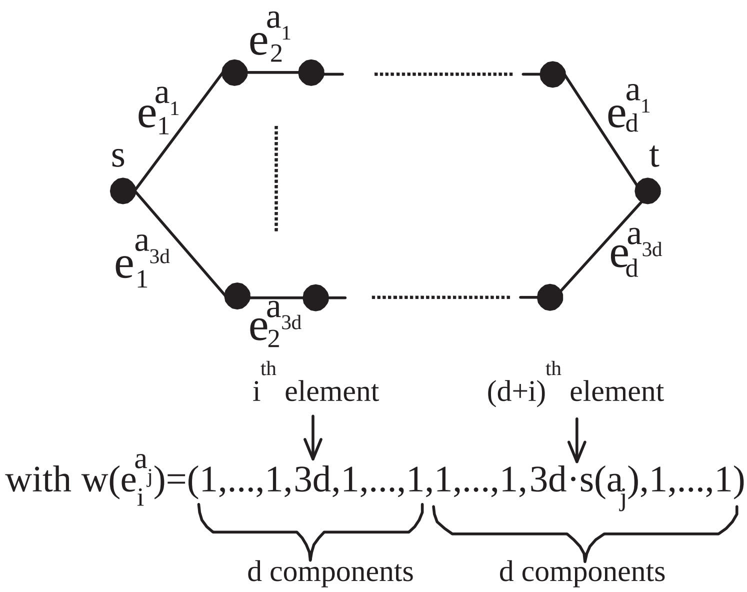

Figure 3. The planar bipartite graph obtained for $2d$-CriCut

Since $\sum_{a\in A} s(a) = S$, we necessarily have $\sum_{a\in A'} s(a) = \sum_{a\notin A'} s(a) = \frac{S}{2}$. □

Let $H_{i,j}$ be the planar bipartite graph composed of two vertices connected by i disjoint paths of length j. For R-CriCut, we show:

Theorem 3 $2d$-CriCut *is strongly $\mathcal{NP}$-complete in $H_{3d,d}$ even when s and t are the two vertices of degree greater than two (thus R-*CriCut *is strongly $\mathcal{NP}$-complete in planar bipartite graphs).*

PROOF. We use a reduction from 3-Partition (6). Given a set A of $3d$ elements, a bound S and a size $s(a) \in \mathbb{N}^*$ for each $a \in A$ such that $\sum_{a\in A} s(a) = dS$, the problem is to decide if there exists a partition of A into d disjoint sets $A_1, \ldots, A_d$ such that, for each $i \in \{1, \ldots, d\}$, $|A_i| = 3$ and $\sum_{a\in A_i} s(a) = S$. Let I_{3-part} be an instance of 3-Partition.

We construct an instance I_{cut} of $2d$-CriCut as follows (see Figure 3). Consider the graph $H_{3d,d}$ and let s and t be the two vertices linked by the $3d$ disjoint paths P_a ($a \in A$) of length d. For each $a \in A$, let $e_1^a, \ldots, e_d^a$ be the d edges of P_a. For each $i \in \{1, \ldots, d\}$, $w_i(e_i^a) = 3d$, $w_{d+i}(e_i^a) = 3ds(a)$ and for $j \neq i$, $w_i(e_j^a) = w_{d+i}(e_j^a) = 1$. The d first criteria will ensure that each set A_i contains exactly three elements whereas the d other criteria will ensure that $\sum_{a\in A_i} s(a) = S$. We claim that there exists a solution for I_{3-part} if and only if there exists a cut C such that $w_i(C) \leq 12d - 3$ $\forall i \in \{1, \ldots, d\}$ and $w_i(C) \leq 3dS + 3d - 3$ $\forall i \in \{d+1, \ldots, 2d\}$.

Suppose that we have a solution for I_{3-part}. We construct a solution C for I_{cut} as follows: if $a \in A_i$ then we cut the edge e_i^a.
Then, $w_i(C) = \sum_{a \in A_i} w_i(e_i^a) + \sum_{a \notin A_i} 1 = |A_i| \cdot 3d + (|A| - |A_i|) = 9d + 3d - 3 = 12d - 3 \ \forall i \in \{1, \ldots, d\}$
and $w_{d+i}(C) = \sum_{a \in A_i} w_{d+i}(e_i^a) + \sum_{a \notin A_i} 1 = 3d \sum_{a \in A_i} s(a) + |A| - |A_i| = 3dS + 3d - 3 \ \forall i \in \{1, \ldots, d\}$.

Conversely, suppose we have a solution C for I_{cut}. For each $a \in A$, there is at least one edge of P_a in C. From the set of edges of P_a which are in C, we select arbitrarily one edge e_i^a and place a in the set A_i. Let us begin by showing that for each $i \in \{1, \ldots, d\}$, $|A_i| = 3$.
Let $i \in \{1, \ldots, d\}$. By construction: $w_i(C) \geq \sum_{a \in A_i} 3d + \sum_{a \notin A_i} 1 = 3d|A_i| + (3d - |A_i|)$. Besides, we necessarily have $w_i(C) \leq 12d - 3$.
So, $3d|A_i| + (3d - |A_i|) \leq 12d - 3$ and thus:

$$|A_i| \leq 3$$

This implies that $|A_i| = 3 \ \forall i \in \{1, \ldots, d\}$, since $\sum_{i \in \{1, \ldots, d\}} |A_i| = 3d$.

Finally, we have to prove that $\sum_{a \in A_i} s(a) = S \ \forall i \in \{1, \ldots, d\}$.
Let $i \in \{1, \ldots, d\}$. By construction, we have $w_{d+i}(C) \geq \sum_{a \in A_i} 3ds(a) + \sum_{a \notin A_i} 1 = 3d \sum_{a \in A_i} s(a) + (3d - 3)$. Besides, we necessarily have $w_{d+i}(C) \leq 3dS + 3d - 3$.
So, $3d \sum_{a \in A_i} s(a) + (3d - 3) \leq 3dS + 3d - 3$ and thus:

$$\sum_{a \in A_i} s(a) \leq S$$

Since $\displaystyle\sum_{i=1}^{d} \sum_{a \in A_i} s(a) = \sum_{a \in A} s(a) = dS$, we have $\displaystyle\sum_{a \in A_i} s(a) = S \ \forall i \in \{1, \ldots, d\}$. □

Before studying the general problem R-CriMultiCut, we deal with MinMultiCutCard which can be seen as a special bicriteria multicut problem.

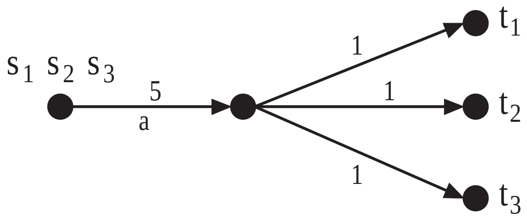

Figure 4. An instance of MinMultiCutCard with $p = 2$ where the constraint matrix is not totally unimodular

3 Minimum multicut with cardinality constraint

Let $(LPtree)$ be the following integral linear program associated with MinMultiCut in a tree:

$$(LPtree)\begin{cases} Min \sum_{e\in E} w(e)z_e \\ \text{s. t.} \sum_{e\in P_i} z_e \geq 1 \ \forall i \in \{1,\ldots,k\} \\ \quad z_e \in \{0,1\} \ \forall e \in E \end{cases}$$

P_i is the path between s_i and t_i ($i \in \{1,\ldots,k\}$) and $z_e \in \{0,1\}$ is the decision variable whose value is 1 if and only if the edge e is in the multicut.

If the tree is directed, the constraint matrix of $(LPtree)$ is totally unimodular and MinMultiCut is polynomial in this case (4). Unfortunately, if we add the cardinality constraint $\sum_{e\in E} z_e \leq p$, generally there is a gap between the integral optimal value and the optimal continuous value (so the constraint matrix is not totally unimodular). In the instance given in Figure 4, the optimal value is equal to 5 and is obtained for $z_a = 1$ and $z_e = 0$ for all $e \neq a$ (i.e. the only arc in the cut is a) while the optimal continuous value is equal to 4 and is obtained for $z_e = 0.5$ for all $e \in E$.

The graph of this instance being both a rooted tree and a directed star, one can wonder if MinMultiCutCard is still $\mathcal{NP}$-hard in graph topologies where the constraint matrix of MinMultiCut is totally unimodular. In this section, we show that MinMultiCutCard becomes strongly $\mathcal{NP}$-hard in directed stars (and thus in directed trees) but remains polynomial in paths and cycles.

3.1 $\mathcal{NP}$-hardness of MinMultiCutCard *in directed stars*

Let $G = (V_1, O, V_2, E)$ be an arc-weighted directed stars where O is the only vertex of degree at least 2, V_1 the set of vertices without predecessors and V_2

the set of vertices without successors. Without loss of generality, we assume that there is at least one terminal on each vertex of V_1 and V_2, that there is no terminal on O and that there is a directed path (of length necessarily equal to 2) between s_i and t_i for each i in $\{1, \ldots, k\}$.

First, we introduce a new problem closely related to MinMultiCutCard in directed stars. Let $G = (V_1, V_2, E)$ be an undirected bipartite graph, w:$(V_1 \cup V_2) \to \mathbb{N}^*$ be a weight function defined on its *vertices* and α and p be two given values. WeightedVCCard consists in finding a vertex cover in G whose weight is less than or equal to α and whose cardinality is at most p.

Proposition 4 WeightedVCCard *is equivalent to the decision version of* MinMultiCutCard *in directed stars.*

PROOF. We use the same kind of transformation as the one used by Garg et al. in (8) where they show that MinMultiCut is APX-hard in undirected stars. For an instance I of the decision problem associated to MinMultiCutCard in an directed star, we consider the demand graph $H = (V_1 \cup V_2, E^H)$: for each source-sink pair (s_i, t_i) with s_i on the vertex v_1 and t_i on the vertex v_2, we connect the vertices v_1 and v_2. The weight of each vertex in H is the one of the arc linking the corresponding vertex and O in G. Since we consider instances of MinMultiCutCard in directed stars without terminals on O, H is necessarily bipartite. Conversely, for an instance of WeightedVCCard, we can easily construct an instance of the decision problem of MinMultiCutCard where the graph is a directed star.

We claim that finding in H a vertex cover whose weight is less than or equal to α and cardinality is at most p, is equivalent to finding a solution of I whose value is less than or equal to α and whose cardinality is at most p. Indeed, cutting an arc e in G corresponds to selecting, in the vertex cover of H, the vertex corresponding to the endpoint of e in $V_1 \cup V_2$. $\square$

Note that without cardinality constraint, finding a minimum vertex cover in a bipartite graph is a well-known polynomial problem (10). In (3), it is proved that:

Theorem 5 (Chen and Kanj) *Let $G = (V_1, V_2, E)$ be an undirected unweighted bipartite graph, let VC_{min} be the size of a minimal vertex cover for G and let p and q be two given values such that $p \leq VC_{min}$, $q \leq VC_{min}$ and $p + q \geq VC_{min}$. The problem of the existence of a minimum vertex cover for G with at most p vertices in V_1 and at most q vertices in V_2, is $\mathcal{NP}$-complete.*

We call this problem ChenVCCards and we modify it to obtain UnweightVCCardsEq, which consists in finding a minimum vertex cover for an undirected bipartite graph $G = (V_1, V_2, E)$ with *exactly* $p_=$ vertices in V_1, *exactly* $q_=$ vertices in V_2 and such that $p_= + q_= = VC_{min}$.

Proposition 6 UNWEIGHTVCCARDSEQ *is* $\mathcal{NP}$*-complete.*

PROOF. Let $G = (V_1, V_2, E)$ be a bipartite graph with n vertices and let VC_{min} be the minimum size of a vertex cover for G. Let I be an instance of CHENVCCARDS composed of G, p and q. Since $p + q \geq VC_{min}$, we have $p \geq VC_{min} - q$. So, we could solve I by solving at most p instances of UNWEIGHTVCCARDSEQ: the i^{th} instance of UNWEIGHTVCCARDSEQ is composed of G, $p_= = i$ and $q_= = VC_{min} - i$ ($i \in \{VC_{min} - q, \dots, p\}$). Clearly, there is a solution for one of the instances of UNWEIGHTVCCARDSEQ if and only if I has a solution. □

Now, we can establish the complexity of MINMULTICUTCARD in directed stars:

Theorem 7 MINMULTICUTCARD *is strongly* $\mathcal{NP}$*-hard in directed stars.*

PROOF. First, we show that UNWEIGHTVCCARDSEQ is polynomial-time reducible to WEIGHTEDVCCARD, which proves that WEIGHTEDVCCARD is strongly $\mathcal{NP}$-complete. Then, using Proposition 4, we obtain that MINMULTICUTCARD is strongly $\mathcal{NP}$-hard in directed stars.

Let I be an instance of UNWEIGHTVCCARDSEQ consisting of $G = (V_1, V_2, E)$, $p_=$ and $q_=$. Let $n = |V_1| + |V_2|$ and let VC_{min} be the minimum size of a vertex cover for G. Recall that $VC_{min} = p_= + q_=$ and that VC_{min} can be computed in polynomial time.

We obtain an instance I' of WEIGHTEDVCCARD in the following way (see Figure 5): let us assign weight 1 to the vertices of V_1 and weight $2n^2+2n+1$ to the vertices of V_2. Then, for each vertex v of V_2, we add $2n+1$ new vertices of weight 1 and we link them to v. Let V' be the set of the $|V_2|(2n+1)$ new vertices and E' be the set of the new edges. Finally, we obtain the vertex-weighted bipartite graph $G' = (V_1 \cup V', V_2, E \cup E')$ and we claim that there exists a vertex cover for G with $p_=$ vertices in V_1 and $q_=$ vertices in V_2 if and only if there exists a vertex cover for G' whose weight is at most $p_= + (2n^2+2n+1)q_= + (2n+1)(|V_2| - q_=)$ and whose cardinality is at most $p_= + q_= + (2n+1)(|V_2| - q_=)$.

From a solution VC_G for I, we build a solution for I': for each vertex v of V_2 not selected in VC_G, we add to VC_G the $2n + 1$ vertices of V' connected to v. This solution is a vertex cover for G' and contains exactly $p_=$ vertices of

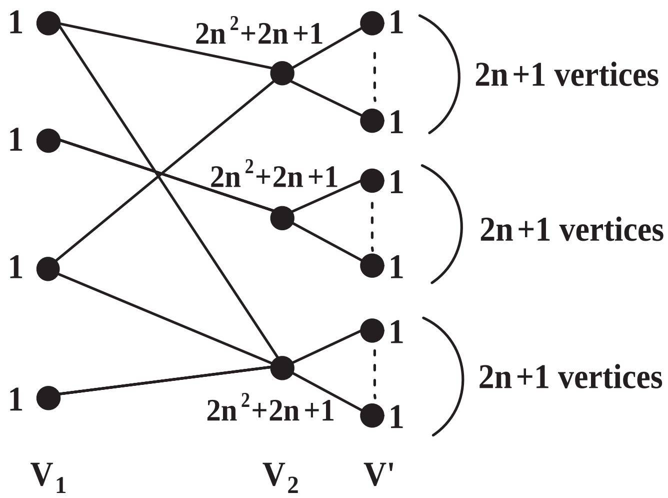

Figure 5. Reduction from UnweightVCCardsEq to WeightedVCCard

V_1, $q_=$ vertices of V_2 and $(2n+1)(|V_2|-q_=)$ vertices of V'. So, its cardinality is equal to $p_= + q_= + (2n+1)(|V_2|-q_=)$ and its weight to $p_= + (2n^2+2n+1)q_= + (2n+1)(|V_2|-q_=)$.

Conversely, suppose that we have a solution $VC_{G'}$ for I'. We get a solution for I by deleting from $VC_{G'}$ the vertices of V'.
$VC_{G'}$ has at least $q_=$ vertices of V_2 otherwise, the number of vertices of V' selected in $VC_{G'}$ would be at least $(2n+1)(|V_2|-q_=+1) = 2n+1+(2n+1)(|V_2|-q_=) > p_= + q_= + (2n+1)(|V_2|-q_=)$ and the cardinality constraint would be violated.
Besides, $VC_{G'}$ has at most $q_=$ vertices of V_2 otherwise, the weight of $VC_{G'}$ would be greater than or equal to $(2n^2+2n+1)(q_=+1) = n + (2n^2+2n+1)q_= + (2n+1)n + 1 > p_= + (2n^2+2n+1)q_= + (2n+1)(|V_2|-q_=)$ which is not possible.
Thus, $VC_{G'}$ contains exactly $q_=$ vertices of V_2. Since for each vertex v of V_2 not selected in $VC_{G'}$, $VC_{G'}$ must contain the $2n+1$ vertices of V' connected to v, $VC_{G'}$ contains at least $(2n+1)(|V_2|-q_=)$ vertices of V'.
Finally, $VC_{G'}$ contains at most $p_=$ vertices of V_1 otherwise, the weight of $VC_{G'}$ would be greater than $(p_=+1)+(2n^2+2n+1)q_=+(2n+1)(|V_2|-q_=)$ which is not possible. In fact, $VC_{G'}$ has exactly $p_=$ vertices of V_1 otherwise, by selecting the vertices of V_1 and V_2 from $VC_{G'}$, we would obtain a vertex cover for G of size strictly less than VC_{min} which is not possible. □

3.2 Some polynomial cases of MinMultiCutCard

We have shown that if we add a cardinality constraint to MinMultiCut, the problem becomes strongly $\mathcal{NP}$-hard in directed stars. However, in this section we prove that MinMultiCutCard remains polynomial in paths (directed paths) and cycles (circuits). We also give dynamic programming algorithms for these problems.

MinMultiCutCard in directed paths is equivalent to MinMultiCutCard in paths because a directed path corresponds to a path where all the edges are orientated in the same direction, inverting source and sink if needed.

In paths, the constraint matrix of (*LPtree*) is an interval matrix. It remains an interval matrix if we add the constraint $\sum_{e\in E} z_e \leq p$ because we add a row of 1's. Since an interval matrix is totally unimodular (12, Chapter 19), MinMultiCutCard is polynomial in paths. To avoid using linear programming, we propose a dynamic programming algorithm which runs in $O(mk^2)$ time.

In this section, we suppose that we have deleted "useless" edges and source-sink pairs:

- if there exist i and j such that P_i is included in P_j, we delete the source-sink pair (s_j, t_j): if P_i is cut by an edge e then P_j is also cut by e. To perform this reduction in $O(n+k)$, we use a queue. Each time we encounter a source, we enqueue the source-sink pair. When a sink is reached, we dequeue all the elements (corresponding to "useless" source-sink pairs) until the source-sink pair corresponding to the sink is reached (which is a "useful" source-sink pair). Hence, each source-sink pair is seen twice.
- then, if consecutive edges intersect the same set of paths P_i, we keep the edge with the smallest weight and delete the others: the deleted edges cannot be selected in an optimal multicut since the edge with the smallest weight is necessarily more interesting. This reduction can be done in $O(n)$ since we only need to consider one time each edge by comparing it to the previous edge and deleting it if necessary.

We also suppose that the source-sink pairs are sorted such that if $i < j$ then s_i is "at the left" of s_j (and thus t_i is "at the left" of t_j).

Let c be the function of two variables α' and α with $0 \leq \alpha' < \alpha \leq k$ such that $c(\alpha', \alpha)$ is equal to the weight of the lowest weight edge belonging to $(P_{\alpha'+1} \cap P_\alpha) \setminus P_{\alpha+1}$ if $(P_{\alpha'+1} \cap P_\alpha) \setminus P_{\alpha+1} \neq \emptyset$ (i.e. s_α is on the left of $t_{\alpha'+1}$) and $c(\alpha', \alpha) = \infty$ otherwise (see Figure 6). Note that $c(\alpha', \alpha)$ is the weight of the "best" edge that has to be added to a multicut which separates exactly the α' first source-sink pairs in order to obtain a multicut which separates exactly the α first source-sink pairs.

We can now define the optimization function g.

Definition 8
g: $\{0, \ldots, k\} \times \{0, \ldots, p\} \rightarrow \mathbb{N}$

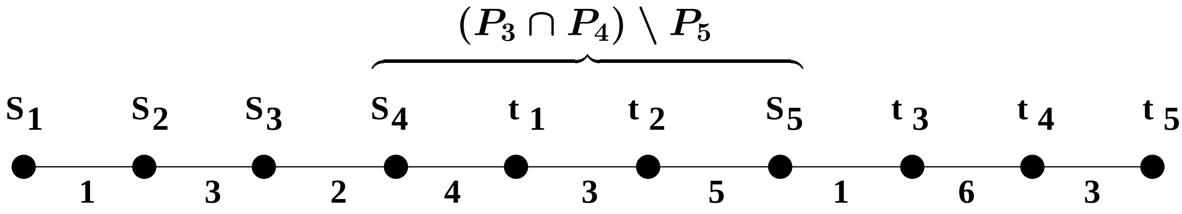

Figure 6. An instance where $c(2,4) = 3$

$g(0,0) = 0$, $g(\alpha,0) = \infty\ \forall\alpha \in \{1,\ldots,k\}$ *and* $g(\alpha,\beta) = \infty\ \forall\beta > \alpha$
$g(\alpha,\beta) = \min_{\alpha' \in \{0,\ldots,\alpha-1\}} \{g(\alpha',\beta-1) + c(\alpha',\alpha)\}$ *for* $\alpha \geq \beta > 0$.

Proposition 9 *Let* $\alpha \in \{1,\ldots,k\}$ *and* $\beta \in \{1,\ldots,p\}$.
$g(\alpha,\beta)$ *is equal to the weight of a minimum multicut of cardinality* β *which separates* s_1 *and* t_1*,…,* s_α *and* t_α *but not* $s_{\alpha+1}$ *and* $t_{\alpha+1}$.
$g(\alpha,\beta) = \infty$ *if and only if such a multicut does not exist.*

PROOF. The proof is obtained by induction on β.
For $\beta = 1$, since for each $\alpha' > 0$ $g(\alpha',0) = \infty$, $g(\alpha,1)$ is equal to $g(0,0) + c(0,\alpha)$. We need an edge which separates exactly the α first source-sink pairs. If the set $(P_1 \cap P_\alpha) \setminus P_{\alpha+1}$ is empty, such an edge does not exist and we have $c(0,\alpha) = \infty$ so $g(\alpha,1) = \infty$. Else, $g(\alpha,1)$ is equal to the weight of a minimum multicut of cardinality 1 which separates exactly the α first source-sink pairs.

Now, assume the proposition is true for a value $\beta \geq 1$. The first case we consider is when $g(\alpha,\beta+1) = \infty$. Suppose that there exists a minimum multicut C of cardinality $\beta+1$ which separates exactly the α first source-sink pairs. Let e be the edge of C which belongs to P_α. e is necessarily unique since C is minimum. Thus, there exists a multicut of cardinality β separating exactly the α' first source-sink pairs where α' is the number of source-sink pairs separated by $C \backslash \{e\}$. So, by induction we have $g(\alpha',\beta) \neq \infty$. Then, we necessarily have $g(\alpha,\beta+1) \leq g(\alpha',\beta) + w(e)$ which leads to a contradiction. Thus, C cannot exist.

Finally, we consider the second case where $g(\alpha,\beta+1) \neq \infty$. Since $g(\alpha,\beta+1)$ is finite, there exists α_1 and $e_1 \in (P_{\alpha_1+1} \cap P_\alpha) \setminus P_{\alpha+1}$ such that $g(\alpha,\beta+1) = g(\alpha_1,\beta) + w(e_1)$. By induction, $g(\alpha_1,\beta)$ is equal to the weight of a multicut of cardinality β which separates exactly the α_1 first source-sink pairs. If we add the edge e_1 to this multicut, we get a multicut of cardinality $\beta+1$ which separates exactly the α first source-sink pairs and whose weight is equal to $g(\alpha,\beta+1)$. Let C be an optimal multicut of cardinality $\beta+1$ which separates exactly the α first source-sink pairs. Since C is optimal, there is exactly one edge e of P_α which belongs to C. Let α' be the number of source-sink pairs separated by $C \backslash \{e\}$. By induction, $g(\alpha',\beta) \leq w(C \backslash \{e\})$ and from Definition

8, we have $g(\alpha, \beta+1) \leq g(\alpha', \beta) + w(e)$. So:

$$g(\alpha, \beta+1) \leq g(\alpha', \beta) + w(e) \leq w(C\backslash\{e\}) + w(e) = w(C)$$

Thus, since C is minimum, we have $g(\alpha, \beta+1) = w(C)$. $\square$

This implies:

Corollary 10 *The weight of a minimal multicut of cardinality at most p is equal to* $\min_{\beta\in\{1,\dots,p\}} g(k, \beta)$.

In practice, the algorithm is divided into two parts:

- The first part is a preprocessing step: for each $\alpha \in \{1, \dots, k\}$ and for each $\alpha' \in \{0, \dots, \alpha-1\}$, we calculate $c(\alpha', \alpha)$ by looking for the edge of lowest weight in the set $(P_{\alpha'+1} \cap P_\alpha) \setminus P_{\alpha+1}$.
- The second part is the computation of all the values of g by using the result of the preprocessing step.

The complexity of the first step is $O(mk^2)$ since we consider $k(k+1)/2 = O(k^2)$ pairs for (α', α). In the second step, it takes $O(k)$ time to compute each value of g and we have to compute $O(pk)$ values of g. Then, the second step takes $O(pk^2)$ time. Since $p = O(m)$, the global complexity of the algorithm is $O(mk^2) + O(pk^2) = O(mk^2)$.

Moreover, this algorithm can easily be extended to cycles and circuits.
Let $G = (V, E)$ be a cycle, $(s_1, t_1), \dots, (s_k, t_k)$ be k source-sink pairs and p be a given value. For each pair (s_i, t_i) ($i \in \{1, \dots, k\}$), there are two paths P_i and P'_i connecting s_i to t_i and we assume that the length $|P_i|$ of P_i is less than or equal to $|P'_i|$. Let i_{min} be such that $|P_{i_{min}}| = \min_{i\in\{1,\dots,k\}} |P_i|$.

The algorithm for cycles consists in deleting successively each edge of $P_{i_{min}}$ and solving the resulting instance (whose graph is now a path and in which we look for a minimum multicut of cardinality at most $p-1$). At the end, we keep the best solution among the $|P_{i_{min}}|$ computed solutions. The complexity of this algorithm is $O(|P_{i_{min}}|(m-1)k^2) = O(m^2k^2)$. The algorithm is similar for circuits (in this case, there is exactly one path P_i between s_i and t_i for each $i \in \{1, \dots, k\}$).

4 Complexity results for the multicriteria version of the multicut problem

We now consider the problem R-CriMultiCut which, given a graph $G = (V, E)$, R weight functions $w_1, \ldots, w_R$ defined on the edges (arcs) of G and R bounds $B_1, \ldots, B_r$ (one for each criterion), consists in finding a multicut C such that $w_r(C) \leq B_r$ $\forall r \in \{1, \ldots, R\}$.

Since the decision problem associated to MinMultiCutCard is a particular case of 2-CriMultiCut, we obtain from Theorem 7 that 2-CriMultiCut is strongly $\mathcal{NP}$-complete in directed stars. However, what is the complexity of R-CriMultiCut in directed paths and circuits (MinMultiCutCard being polynomial in these cases)?

Theorem 11 2-CriMultiCut *is $\mathcal{NP}$-complete in paths.*

PROOF. We use a reduction from 2-CriCut in $K_{2,d}$, where s and t are the two vertices of degree greater than two (see Theorem 2). Let I be an instance of this problem, given by a bipartite graph $G = (V_1, V_2, E)$ with $V_1 = \{s, t\}$ and $|V_2| = d$, two weight functions w_1 and w_2 defined on the edges of G, and two integers B_1 and B_2. There are exactly d paths $P_1, \ldots, P_d$ of length 2 connecting s to t: let e_1^i and e_2^i be the edges of P_i, $i \in \{1, \ldots, d\}$.

We construct an instance I' of 2-CriMultiCut as follows (see Figure 7). We consider a path G' with $2d$ edges $e_1'^1, e_2'^1, \ldots, e_1'^d, e_2'^d$. Let w_1' and w_2' be two weight functions defined on the edges of G' such that, for each $i \in \{1, \ldots, d\}$, $w_1'(e_1'^i) = w_1(e_1^i)$, $w_2'(e_1'^i) = w_2(e_1^i)$, $w_1'(e_2'^i) = w_1(e_2^i)$ and $w_2'(e_2'^i) = w_2(e_2^i)$. We add d source-sink pairs such that, for each $i \in \{1, \ldots, d\}$, the path connecting s_i to t_i is composed by $e_1'^i$ and $e_2'^i$.

A set of edges C (resp. C') is a cut for I (resp. a multicut for I') if and only if, for each $i \in \{1, \ldots, d\}$, $e_1^i \in C$ or $e_2^i \in C$ (resp. $e_1'^i \in C'$ or $e_2'^i \in C'$). Hence, there exists a cut C for I such that $w_1(C) \leq B_1$ and $w_2(C) \leq B_2$ if and only if there exists a multicut C' for I' such that $w_1'(C') \leq B_1$ and $w_2'(C') \leq B_2$. Indeed, simply define C and C' as follows: for each $i \in \{1, \ldots, d\}$, for each $j \in \{1, 2\}$, $e_j^i \in C$ if and only if $e_j'^i \in C'$. □

When R is not bounded, we have:

Theorem 12 R-CriMultiCut *is strongly $\mathcal{NP}$-complete in paths.*

PROOF. We use a reduction from $2d$-CriCut in $H_{3d,d}$ (see Theorem 3). Let

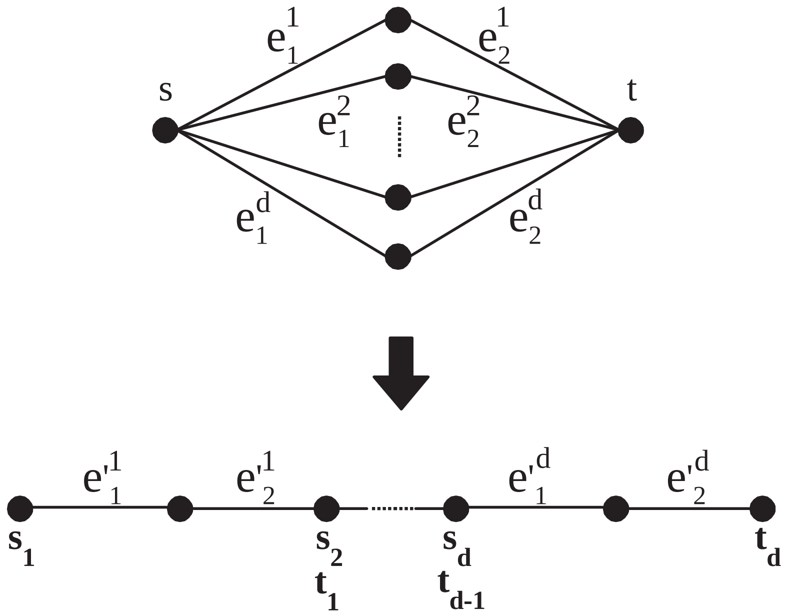

Figure 7. Transformation of an instance of 2-CriCut into an instance of 2-CriMultiCut

I be an instance of this problem, where s and t are the two vertices of degree greater than two. Let $w_1, \ldots, w_{2d}$ be the $2d$ weight functions defined on the edges of $H_{3d,d}$ and let $B_1, \ldots, B_{2d}$ be $2d$ given integers. There are exactly $3d$ paths $P_1, \ldots, P_{3d}$ of length d connecting s to t: let $e_1^i, \ldots, e_d^i$ be the edges of P_i, $i \in \{1, \ldots, 3d\}$.

We construct an instance I' of $2d$-CriMultiCut as follows (see Figure 8). We consider a path G' with $3d^2$ edges $e_1^{'1}, \ldots, e_d^{'1}, \ldots, e_1^{'3d}, \ldots, e_d^{'3d}$. Let $w'_1, \ldots, w'_{2d}$ be $2d$ weight functions defined on the edges of G' such that, for each $i \in \{1, \ldots, 3d\}$, for each $j \in \{1, \ldots, d\}$ and for each $k \in \{1, \ldots, 2d\}$ $w'_k(e_j^{'i}) = w_k(e_j^i)$. We add $3d$ source-sink pairs such that, for each $i \in \{1, \ldots, 3d\}$, the path connecting s_i to t_i is composed by $e_1^{'i}, \ldots, e_d^{'i}$.

A set of edges C (resp. C') is a cut for I (resp. a multicut for I') if and only if, for each $i \in \{1, \ldots, 3d\}$ there exists $j \in \{1, \ldots, d\}$ such that $e_j^i \in C$ (resp. $e_j^{'i} \in C'$). Hence, there exists a cut C for I such that $w_k(C) \leq B_k$ for each $k \in \{1, \ldots, 2d\}$ if and only if there exists a multicut C' for I' such that $w'_k(C') \leq B_k$ for each $k \in \{1, \ldots, 2d\}$. Indeed, simply define C and C' as follows: for each $i \in \{1, \ldots, 3d\}$, for each $j \in \{1, \ldots, d\}$, $e_j^i \in C$ if and only if $e_j^{'i} \in C'$. □

Besides, for R-CriMultiCut, we can easily obtain an instance in a cycle from an instance in a path by adding a source-sink pair, whose source is on one of the extremities of the path and the sink on the other extremity, and by

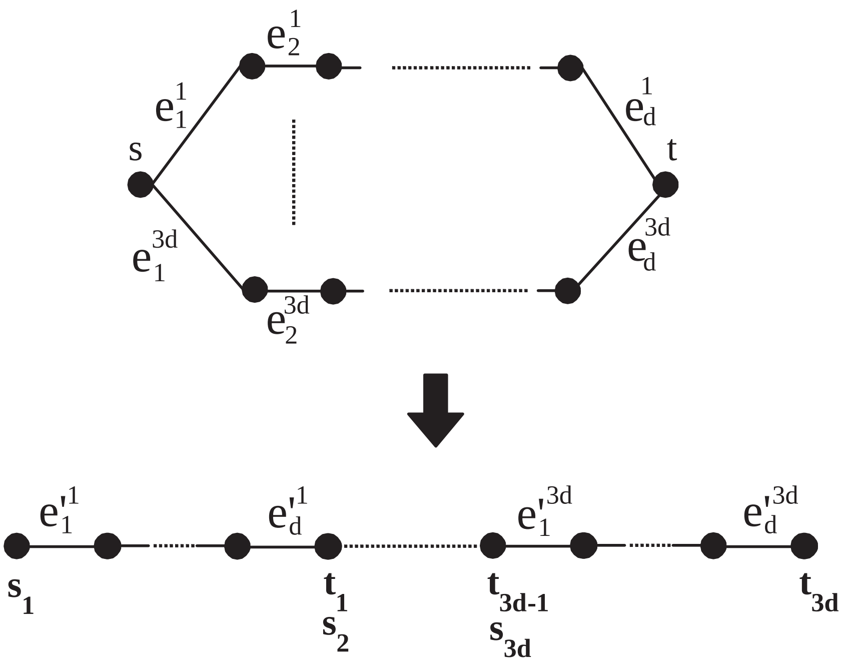

Figure 8. Transformation of an instance of R-CriCut into an instance of R-CriMultiCut

adding an edge connecting the two extremities. So:

Corollary 13 *In cycles,* 2-CriMultiCut *is $\mathcal{NP}$-complete and* R-CriMultiCut *is strongly $\mathcal{NP}$-complete.*

Remark *When R is fixed and G is a path,* R-CriMultiCut *can be solved by a pseudo-polynomial algorithm.*

Let us present a sketch of such an algorithm. Let h be the optimization function defined as follows:

Let $\alpha \in \{1, \dots, k\}$ and $\beta_i \leq B_i$ $(2 \leq i \leq R)$. $h(0, 0, \dots, 0) = 0$

If either ($\alpha = 0$ and there exists $\beta_i \neq 0$) or (there exists $\beta_i < 0$) then $h(\alpha, \beta_2, \dots, \beta_R) = \infty$.

In the general case:

if there exists $\alpha' \in \{0, \dots, \alpha - 1\}$ such that $(P_{\alpha'+1} \cap P_\alpha) \setminus P_{\alpha+1} \neq \emptyset$ then:

$h(\alpha, \beta_2, \dots, \beta_R) =$

$$\min_{0 \leq \alpha' \leq \alpha-1,\ e \in (P_{\alpha'+1} \cap P_\alpha) \setminus P_{\alpha+1}} \{h(\alpha', \beta_2 - w_2(e), \dots, \beta_R - w_R(e)) + w_1(e)\}$$

else $h(\alpha, \beta_2, \dots, \beta_R) = \infty$

Let $\alpha \in \{1, \dots, k\}$, $\beta_i \in \{0, \dots, B_i\}$ $(2 \leq i \leq R)$. $h(\alpha, \beta_2, \dots, \beta_R)$ is equal to the minimal value, with respect to the first criterion, of a multicut C which separates exactly the α first source-sink pairs and such that $w_i(C) = \beta_i$ $\forall i \in \{2, \dots, R\}$.

Then $h(\alpha, \beta_2, \ldots, \beta_R) = \infty$ if and only if such a multicut does not exist.

In fact, the instance has a solution if and only if there exist $\beta_2, \ldots, \beta_R$ such that $h(k, \beta_2, \ldots, \beta_R) \leq B_1$.
In practice, we compute all the values of h, for α from 1 to k. For given values $\alpha, \beta_2, \ldots, \beta_R$, $h(\alpha, \beta_2, \ldots, \beta_R)$ is computed in $O(km)$ since we look for the minimum among $O(km)$ values. Besides, we have to compute $O(kB_2 \ldots B_R)$ values of h, so the global complexity of the algorithm is $O(mk^2 B_2 \ldots B_R)$.

Note that if we consider $R = 2$ and $B_2 = p$, the complexity is worse than the one obtained for MinMultiCutCard. The main reason is the use of the auxiliary function c, specific to MinMultiCutCard, which allows to compute the values of g more efficiently.

Finally, one can remark that B_1 does not contribute to the complexity of the algorithm. Thus, it is better to choose B_1 such that $B_1 \geq B_i$ (for all $i \in \{2, \ldots, R\}$). If we consider the case where R is bounded and $B_i = O(m^\gamma)$ for each $i \in \{2, \ldots, R\}$ and for fixed γ then the problem becomes polynomial.

5 Conclusion

The main purpose of this paper was to study the complexity of cardinality constrained and multicriteria (multi)cut problems in graph topologies where the (multi)cut problem is polynomial.

We have obtained some results about the (strong) $\mathcal{NP}$-hardness of these problems and we have designed dynamic programming algorithms for the (pseudo)polynomial cases. In particular, we have shown that MinCutCard, whose complexity was open until now, is strongly $\mathcal{NP}$-hard.

However, there are still some cases to study: the complexity of MinMultiCutCard in rooted trees is unknown and it would be interesting to determine the complexity of MinCutCard in particular graphs such as planar graphs.

Acknowledgments: We thank the two referees for their valuable remarks and comments.

References

[1] A. Armon, U. Zwick. Multicriteria Global Minimum Cuts. Algorithmica **46**(1) 15–26 (2006)

[2] M. Bruglieri, F. Maffioli, M. Ehrgott. Cardinality constrained minimum cut problems: complexity and algorithms. Discrete Applied Mathematics **137** 311–341 (2004)

[3] J. Chen, I.A. Kanj. Constrained minimum vertex cover in bipartite graphs: complexity and parameterized algorithms. Journal of Computer and System Sciences **67** 833–847 (2003)

[4] M.-C. Costa, L. Létocart, F. Roupin. Minimal multicut and maximal integer multiflow: a survey. European Journal of Operational Research **162**(1) 55–69 (2005)

[5] E. Dahlhaus, D.S. Johnson, C.H. Papadimitriou, P.D. Seymour, M. Yannakakis. The complexity of multiterminal cuts. SIAM Journal on Computing **23** 864–894 (1994)

[6] M.R. Garey, D.S. Johnson. Computers and Intractability, a Guide to the Theory of NP-completeness. ed. Freeman (1979)

[7] M.R. Garey, D.S. Johnson, L. Stockmeyer. Some simplified NP-complete graph problems. Theoretical Computer Science **1** 237–267 (1976)

[8] N. Garg, V.V. Vazirani, M. Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. Algorithmica **18** 3–20 (1997)

[9] N. Garg, V.V. Vazirani, M. Yannakakis. Multiway cuts in node weighted graphs. Journal of algorithms **50** 49–61 (2004)

[10] D. König. Graphok és matrixok. Matematikai és Fizikai Lapok **38** 116–119 (1931)

[11] C.H. Papadimitriou, M. Yannakakis. On the approximability of trade-offs and optimal access of web sources. In Proceedings of the IEEE Symposium on Foundations of Computer Science 86–92 (2000)

[12] A. Schrijver. Theory of Linear and Integer Programming. ed. Wiley (1986)