On the Complexity of the K-way Vertex Cut Problem

Mohammed LALOU
Department of M.CS, Institute of Science & Technology
University Center A. Boussouf of Mila
Mila, Algeria
mohammed.lalou@centre-univ-mila.dz

***Abstract*—The K-way vertex cut problem consists in, given a graph G, finding a subset of vertices of a given size, whose removal partitions G into the maximum number of connected components. This problem has many applications in several areas. It has been proven to be NP-complete on general graphs, as well as on split and planar graphs. In this paper, we enrich its complexity study with two new results. First, we prove that it remains NP-complete even when restricted on the class of bipartite graphs. This is unlike what it is expected, given that the K-way vertex cut problem is a generalization of the Maximum Independent set problem which is polynomially solvable on bipartite graphs. We also provide its equivalence to the well-known problem, namely the *Critical Node Problem* (*CNP*), On split graphs. Therefore, any solving algorithm for the *CNP* on split graphs is a solving algorithm for the *K-way vertex cut problem* and vice versa.**

***Index Terms*—Vertex separator, critical nodes, graph connectivity, bipartite graphs, split graphs, NP-completeness.**

I. Introduction

Given an undirected graph $G = (V, E)$, where V is the set of vertices and E the set of edges, and an integer k, we ask for a subset $S \subseteq V$ of k vertices whose deletion maximizes the number of connected components in the induced subgraph $G[V \setminus S]$. Note that $G[V \setminus S]$ denotes the subgraph induced by $V \setminus S$. This problem is known as the *K-way vertex cut problem* [7], [16]. Its recognition version can be stated as follows. Let $c(G)$ denotes the number of connected components in G.

K-way Vertex Cut Problem (KVCP)

Instance: A graph $G = (V, E)$, and an integer k.

Question: Is there a subset of vertices $S \subseteq V$, where $|S| \leq k$, the deletion of which satisfies $c(G, [V \setminus S]) \geq K$? where K is an integer.

The objective is to find a subset $S \subseteq V$ of at most k vertices, the deletion of which partitions the graph into at least K connected components. This problem is the vertex-version of the well-known *Minimum k-cut problem* [1]–[5], where we ask for deleting a set of edges instead of vertices, with the purpose of maximizing the number of connected components in the induced graph. Note that the number of connected components in a graph can be computed in linear time using either breadth-first search or depth-first search algorithm [6].

The *K-way vertex cut problem* has been proven to be NP-complete on general graphs [7] through a reduction from the *Maximum Independent Set problem* (*MIS*). Indeed, we can easily see that any subset of vertices whose deletion separates the graph into at least K components identifies an independent set of size at least K. Accordingly, the *K-way vertex cut problem* on G is a natural generalization of the *MIS* on G. Conversely, the *MIS* is the particular case of the *K-way vertex cut problem* where the connected component size has to be equal to one. The two problems are equivalent if $k \geq |V \setminus I|$ where I is the maximum independent set of G.

One can hope that the *K-way vertex cut problem* becomes polynomial on classes of graphs for which *MIS* is polynomially solvable. However, this is not the case for the class of bipartite graphs. In this paper, we prove that the *K-way vertex cut problem* remains NP-complete even on this class of graphs. While for the class of split graphs, we provide an equivalence between the *KVCP* and *CNP*. This allows the *KVCP* to be solved using any algorithms for solving the *CNP*.

Figure 1 reviews the *KVCP* complexity on different classes of graphs considered in the literature and highlights the contributions of this paper.

The rest of the paper is organized as follows. We complete this section by a state-of-the-art of the *K-way vertex cut problem*, where we review different works handled this problem in the literature. Also, we give some definitions we need in the rest of the paper. In section II, we provide the NP-completeness proof of the problem on bipartite graphs. In section III, we deduce its equivalence to the *CNP*, while in section IV we deduce its resolvability in polynomial time on weighted graphs of bounded treewidth. We close up the paper by some future works in section V.

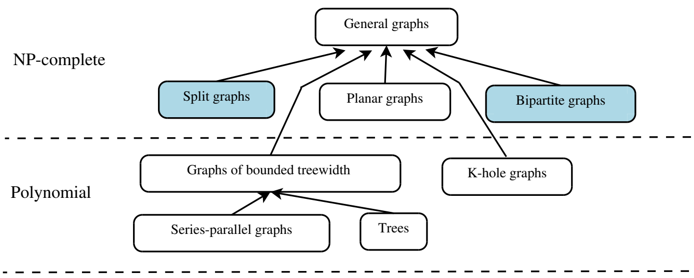

Fig. 1. The complexity of the K-way vertex cut problem on different classes of graphs. Contributions of this paper concern the colored classes.

A. Related works

The *K-way vertex cut problem* can be considered as a parametrized version of the graph separation problem [8], where we ask for the vertex-separator set that partitions the graph into the maximum number of connected components. As well, it can be considered as a variant of the *Critical Nodes Detection Problem (CNDP)* [9]. This problem (*CNDP*) consists in finding the subset of vertices whose removal significantly degrades the graph connectivity according to some predefined connectivity metrics, such as: minimizing the pairwise connectivity in the network [10]–[13], minimizing (or limiting to a given bound) the largest component size [7], [14], [15], etc. In the case of the *K-way vertex cut problem* the metric considered is maximizing the number of connected components.

Although its importance, the *K-way vertex cut problem* has received a little attention, in the literature, as expected for such an important problem. On general graphs, the problem has been shown to be NP-complete [7], [16], and NP-hard to be approximated within a factor of $n^{(1-\epsilon)}$, for any $\epsilon > 0$ [16]. Also, it is W[1]-hard, *i.e.* not fixed-parameter tractable, with respect to the two parameters, namely the number of deleted vertices k, and the number of connected component in the induced graph K [8]. We recall that when we deal with parametrized problems, the input instance has an additional part called *parameters*. The complexity of the problem is then measured as a function of those parameters, and the problem is said to be fixed-parameter tractable if it can be solved using algorithms that are exponential only in the size of these parameters, while polynomial in the size of the problem instance.

For solving the *K-way vertex cut problem* on general graphs, a *Mixed-Integer Program* formulation has been presented in [7], where bounds and validated inequalities for the proposed formulation have been studied. As well, an evolutionary framework, that uses two greedy methods embedded within two main genetic operators, has been presented in [17]. The two operators, namely reproduction and mutation, are used to repair the obtained solutions, while the greedy methods are used to guide the search in the feasible solution space.

Considering the *K-way vertex cut problem* on particular classes of graphs, it has been proved to be NP-complete on split and planar graphs [16]. Also it has been shown, by the same authors, that the problem is NP-hard to be approximated on split graphs [16], while on planar graphs it can be approximated using a polynomial-time approximation scheme (PTAS) of complexity $O(nk^2 f(\epsilon))$, where $\epsilon > 0$ and f is a function only depending on ϵ [16]. We note that a PTAS outputs an approximate solution of value at least $(1-\epsilon)$ times the optimum, and the running time is polynomial in the size of the problem. Considering the parametrized complexity on these two classes of graphs, the problem remains W[1]-hard with respect to parameter k (the number of vertices to be deleted) on split graphs [16], however on planar graphs, a fixed-parameter tractable algorithm of complexity $O(nk^{O(k)})$, with respect to k, has been proposed [16].

On trees, k-hole and series-parallel graphs, polynomial dynamic programming algorithms have been developed for solving the problem with complexity $O(n^3)$, $O(n^{3+k})$ and $O(n^3)$, respectively [14]. Also on graphs of bounded treewidth, the problem can be solved in polynomial-time using a dynamic programming algorithm with complexity $O(nk^2 w^w)$, where $w-1$ is the treewidth [16].

Table I summarizes the different results arisen from studying the *K-way vertex cut problem* on different classes of graphs.

B. Definitions and notations

Let $G = (V, E)$ be an undirected graph, where V is the set of vertices and $E \subseteq V \times V$ is the set of edges. Two distinct vertices u and v are adjacent (or neighbour) if there exists an edge $uv \in E$ connecting them. u and v are called the endpoints of the edge uv. The neighbourhood set of a vertex $v \in V$ is defined as $N(v) = \{u \in V | \{u, v\} \in E\}$. Let $deg_G(v)$ denote the degree of the vertex v, we have $deg_G(v) = |N(v)|$.

A *chain* in G is a sequence of distinct vertices $\{v_1, v_2, \dots, v_k\}$ such that $v_i v_{i+1}$ is an edge for each $0 < i < k-1$.

Given a subset of vertices $S \subseteq V$, S is called an *independent set* if there are no edges between any pair of vertices in S. We use $G[S]$ to denote the subgraph of G induced by S, and hence $G[V \setminus S]$ denotes the subgraph induced by $V \setminus S$. Also, we use $c(G, S)$ to denote the number of connected components in $G[V \setminus S]$ obtained by removing S from G. As well, $c(G, A)$ denotes the number of connected components obtained by deleting a set of edges $A \in E$.

A graph $G = (V, E)$ is a *bipartite graph* if the vertex set V can be divided into two disjoint subsets V_1 and V_2, such that every edge $e \in E$ has one endpoint in V_1 and the other endpoint in V_2. Each subset, V_1 or V_2, forms an independent set of G. G is then denoted $G = (V_1, V_2, E)$, where $n_1 = |V_1|$, $n_2 = |V_2|$ and $n_1 + n_2 = n$. G is said to be a complete bipartite graph, denoted K_{n_1,n_2}, if each vertex in V_1 is adjacent to all vertices in V_2. If one of the independent set, V_1 or V_2, is a clique G is called a split graph.

II. Bipartite graphs

In this section, we consider the *K-way vertex cut problem* on bipartite graphs. This case is relevant when the network

TABLE I
THE DIFFERENT RESULTS OBTAINED FROM STUDYING THE *K-way vertex cut problem* ON DIFFERENT CLASSES OF GRAPHS.

Graph class	Complexity	Solving approach	Time
General graphs	NP-complete [7], [16]	Genetic algorithm [17]	NC
Planar graphs		PTAS [16]	$O(nk^2 f(\epsilon))$
		FPT [16]	$O(nk^{O(k)})$
Split graphs		\	
Trees	Polynomial [14]	Dynamic programming [14]	$O(n^3)$
k-hole graphs			$O(n^{3+k})$
Series-parallel graphs			$O(n^3)$
Graphs with bounded T_w		Dynamic programming [16]	$O(nk^2 w^w)$

to be decomposed on connected groups or communities has a bipartite structure, which is the case, for example, of users vs files in a P2P system, traders vs stocks in a financial trading system, conferences vs authors in a scientific publication networks and so on.

In the following, we show that the *K-way vertex cut problem* remains NP-complete even on this class of graphs. In order to establish the complexity proof we have first to introduce the following transformation of the *k-cut problem* on general graphs [18] to the *K-way vertex cut problem* on bipartite graphs.

The K-cut problem. Given a graph $G = (V, E)$ and an integer K, find a minimal subset of edges $A \subseteq E$, whose removal partitions the graph into at least K connected components, *i.e.* such that $c(G, A) \geq K$. This problem is NP-complete on general graphs [18], and its recognition version asks whether there exists a cut-edges set A where $|A| \leq B$, for a given bound B.

We give a polynomial-time reduction from the *k-cut problem* to the *K-way vertex cut problem*. Given an instance of the *k-cut problem* on a general graph $G(V, E)$, we define an instance of the *K-way vertex cut problem* on a bipartite graph $G' = (V', E')$ as follows:

1) G' contains all vertices and all edges of G, *i.e.* $V \subseteq V'$ and $E \subseteq E'$.
2) For each vertex $v \in V$, if $deg_G(v) \geq 2$, we add to G' a chain $p_v = \{v_1, \ldots, v_k\}$ of k vertices, such that v_1 coincides with v (see *Figure 2*).
3) For each edge $uv \in E'$, we add a vertex $x \in V'$ such as we replace uv with two new edges $ux, xv \in E'$, *i.e.* we replace each edge uv by a chain $\{u, x, v\}$ such that x is an added vertex. We denote U the set of all added vertices x for which $ux, xv \in E'$ and $u, v \in V$.
4) For each chain $p_v = \{v_1, x_1, v_2, x_2, \ldots, v_k\}$ we add two edges $xx_1, x'x_1 \in E'$ where $x_1 \in P_v$ and xv, vx' are two edges sharing the vertex $v \in V$ (see *Figure 2*). Also we add edges vv_i where $2 \leq i \leq k$, and $v_i x_{i+1}$, $x_i v_{i+1}$ where $1 \leq i \leq k$ for each chain p_v.

Note that removing vertices of V from G' does not disconnect the graph G', and G' becomes disconnected only by removing vertices of U. Also, it is obvious that the transformation can be done in polynomial time, and the graph G' is bipartite. Now, we prove the following theorem.

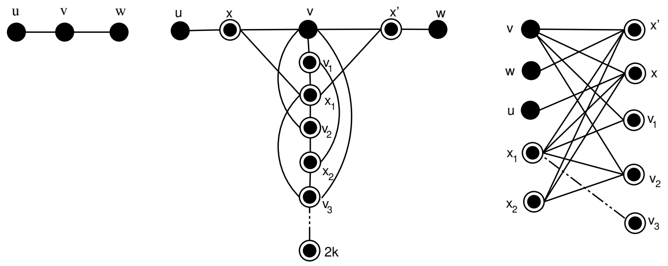

Fig. 2. The reduction *k-cut problem* $\propto$ *K-way vertex cut problem* on bipartite graphs. The added vertices are those with circles, and we have $U = \{x, x'\}$.

Theorem 2.1: The K-way vertex cut problem is NP-complete on bipartite graphs.

Proof. The *K-way vertex cut problem* is in *NP* since given a graph, we can compute in polynomial time the number of connected components in the induced graph after deleting k vertices. Now we prove that the *K-cut problem* on general graphs $\leq_p$ *K-way vertex cut problem* on bipartite graphs.

Given an instance I of the *K-cut problem* on a general graph $G = (V, E)$, we construct an instance I' of the *K-way vertex cut problem* on a bipartite graph $G'(V', E')$ as described in (1)-(4). We show that G has a cut-edge set $A \in E$ of k edges such that $c(G, A) \geq K$ if and only if G' has a cut-vertex set $S \in V'$ of k vertices such that $c(G', V' \setminus S) \geq K$.

First, let $A \subseteq E$ be a solution of I, so A contains no more than k edges whose deletion disconnects G into at least K components. In I', we select the k vertices of S as follows: for each edge $uv \in A$, we select from G' the corresponding vertex $x \in U$ such that $ux, xv \in E'$.

By deleting the vertices in S from G', no more than k vertices are deleted $|S| \leq k$ and at least K connected components are generated $c(G', V' \setminus S) \geq K$. Hence, S is a solution of the *K-way vertex cut problem* on G'.

Conversely, we prove that if there is a cut-vertex set S of size k for G', then we have a cut-edge set of size k for G. Let S be a solution of I', so S contains a set of k vertices whose deletion disconnects G' into at least K components. We can easily observe that G' becomes disconnected only by removing vertices of U. Thus, the solution satisfies the condition that only vertices from U are deleted. Indeed, if the condition is not satisfied, then S should contain the original vertices of G and/or vertices from the added paths p_i. Given such a solution an equivalent solution satisfying the condition that only vertices from U are deleted can be constructed in polynomial time. In doing so, we swap each vertex $v \in S$ and a vertex $u \in U$, *i.e.* we keep v and we delete u instead, and hence we get an induced graph with probably more components, since deleting vertices from U can disconnect G' and generates further components. Thus, the obtained solution is at least as good as S, and satisfies that only vertices from U are deleted.

Now, let $S \subseteq U$ be a solution of I'. In I, we select the k edges of A as follows: for each vertex $v \in S$, we select from G the edge $uw \in E$ such that $uv, vw \in E'$. By deleting A from G, no more than k edges are deleted, $|A| \leq k$, and at least K connected components are generated. Therefore, A is a solution of the *K-cut problem* on G.
This complete the proof.

■

Remark It is clear that for the complete bipartite graph K_{n_1,n_2} the *K-way vertex cut problem* is trivial, and the solution is obtained by deleting the partition of smaller cardinality if $n_1, n_2 \leq k$. Otherwise, the solution is to delete any k vertices that results in only one component.

III. Split graphs

Considering split graphs, we show that the *K-way vertex cut problem* is equivalent to the *Critical Node Problem* (*CNP*) [11].

Theorem 3.1: The *K-way vertex cut problem* and the CNP are equivalent on split graphs.

Proof. Given a split graph $G = (V, E)$ and a set of vertices $S \subseteq V$, we can easily notice that $G[V \setminus S]$ always contains a non-trivial connected component and isolated vertices, if any (see *Figure 3*). Note that G is a split graph if the set of vertices can be partitioned into two subsets V_1 and V_2, $V = V_1 \cup V_2$, where V_1 is an independent set and V_2 is a clique.
We recall that the recognition version of both the *CNP* and *K-way vertex cut problem* seeks for finding a set of vertices of at most k, the deletion of which, respectively, minimizes pairwise connectivity (for the *CNP*), or maximizes the number of components (for the *K-way vertex cut problem*) in the remaining graph. According to the value of k, two cases can be considered:

Case 1: $k \geq |N(V_1)|$. This is a trivial case, where the optimal solution, for both variants, is to delete the vertices of $N(V_1)$ and any $k - |N(V_1)|$ vertices from V_2. We then obtain a residual graph that has $|V_1|$ isolated vertices and a connected component of size $|V_2| - (k - |N(V_1)|)$.

Case 2: $k < |N(V_1)|$. In this case, we consider an optimal solution for the *CNP* and try to prove that it is also an optimal solution for the *K-way vertex cut problem*, and vice versa. Given an optimal solution s^* for the *CNP* on a split graph G, this solution aims to find a set of vertices $S \subseteq V$ so that the non-trivial connected component of $G[V \setminus S]$ is as

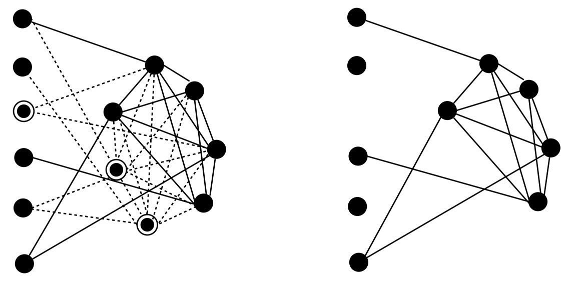

Fig. 3. Deleting any subset of vertices (eg. vertices with circles) from a split graph (see (a)) results in a non-trivial connected component and isolated vertices (see (b)).

small as possible and the surviving isolated vertices of the independent set be as large as possible. Therefore, we note that for s^* only vertices in V_2 are removed from G (*i.e.,* $S \subseteq V_2$), and given any optimal solution for the *CNP*, an equivalent solution satisfying this condition ($S \subseteq V_2$) can be constructed in polynomial time (for proof see [10]). On the other hand, to solve the *K-way vertex cut problem* we aim to obtain a maximal number of components in the residual graph. In doing so, we seek for maximizing the number of isolated vertices from V_1 once the critical vertices have been deleted. For this purpose, only vertices in V_2 are removed from G, which is exactly the solution s^*. Hence, the solution s^* is also the optimal solution of the *K-way vertex cut problem.*
Therefore, an optimal solution of one of the two problems is an optimal solution of the other, and so the *CNP* and the the *K-way vertex cut problem* are equivalent. ■

According to *Theorem 3.1* and since the *CNP* is NP-complete on split graphs [10], we have the following corollary:

Corollary 3.2: The *K-way vertex cut problem* remains NP-complete on split graphs.

This is also what has been proven by Berger *et al.* [16] through a reduction from the k-clique problem.

IV. Other results

We mentioned above that the *K-way vertex cut problem* is polynomially solvable on graphs of bounded treewidth [16]. The considered graphs are unweighted. In this section, we deduce that it remains polynomially solvable on the case of weighted graphs with bounded treewidth. Weighted graphs means that a weight $w_i \geq 0$ is associated with each node $v_i \in V$. In this case, we ask for a subset of nodes of a total weight (rather than a cardinality) no more than k, whose removal maximizes the number of connected components in the induced graph.

In [10], authors studied the *MaxNumSC* problem, for *Maximizing the Number of Small Components*, that can be formulated as follows. Let $f^c(S)$ be the function that returns the number of connected components in $G[V \setminus S]$ with a cardinality of at most c. :

Input: A graph $G = (V, E)$, and two integers c and k.
Output: $\underset{S \subseteq V}{argmax}\ f^c(S)$, where $|S| \leq k$.

Given a graph $G = (V, E)$ and two positive integers k and c, the *MaxNumSC* problem consists in maximizing the number of connected components of cardinality at most c, by deleting k vertices from G. The authors showed that the problem is polynomially solvable on weighted graphs with bounded treewidth.

It is obvious that the *K-way vertex cut problem* is a special case of the *MinMaxSC* problem where $c = |V|$, and as the *MinMaxSC* problem is polynomially solvable on weighted graphs (where $w_i \geq 0, \forall v_i \in V$) we have the following corollary.

Corollary 4.1: The *K-way vertex cut problem* is polynomially solvable on weighted graphs with bounded treewidth.

V. Conclusion and future works

In this paper, we studied the complexity of the *K-way vertex cut problem* on some particular classes of graphs, namely bipartite and split graphs. This problem asks for finding the subset of vertices in a graph, the deletion of which results in the maximum number of connected components in the induced subgraph. We proved its NP-completeness on bipartite graphs. While on split graphs, we provided its equivalence to the well-known problem, namely the *Critical Node Problem* (*CNP*). This allows any solving method for the *CNP* to be used for solving The *K-way vertex cut problem* and vice versa.

The problem still needs more investigation on both complexity study and solving methods. For future works, we can consider it on subclasses of (or related classes to) bipartite and split graphs, which can help providing bounds for the problem hardness. In fact, this is what we are already started to do by considering bipartite-permutation graphs (which is a subclass of the bipartite graph class). We found that the problem can be solved polynomially, on this class of graphs, using dynamic programming approach. Also, the problem can be investigated on other important classes of graphs, such as chordal graphs, disk graphs, etc. which will allow us to find different applications of this parameter in real-world networks.

References

[1] O. Goldschmidt and D. Hochbaum, "A polynomial algorithm for the k-cut problem for fixed k," Math. of oper. res. vol. 19(1), pp. 24–37, 1994.
[2] D. Karger and C. Stein, "A new approach to the minimum cut problem," J. ACM, vol. 43(4), pp. 601–640, 1996.
[3] Y. Kamidoi, N. Yoshida, and H. Nagamochi, "A deterministic algorithm for finding all minimum k-way cuts," SIAM J. Comp. vol. 36(5), pp. 1329–1341, 2006.
[4] H. Saran and V. Vazirani, "Finding k cuts within twice the optimal," SIAM J. Comp., vol. 24(1), pp. 101–108, 1995.
[5] R. Downey, V. Estivill-Castro, M. Fellows, E. Prieto, and F. Rosamund, "Cutting up is hard to do: The parameterised complexity of k-cut and related problems," Elec. Not. in Theor. Comp. Sci., vol. 78, pp. 209–222, 2003.
[6] T. Cormen, "Introduction to algorithms," MIT press, 2009.
[7] S. Shen, J. Smith, and R. Goli, "Exact interdiction models and algorithms for disconnecting networks via node deletions," Disc. Opt., vol. 9(3), pp. 172–188, 2012.
[8] D. Marx, "Parameterized graph separation problems," Theor. Comp. Sci., vol. 351(3), pp. 394–406, 2006.
[9] M. Lalou, M. Tahraoui, and H. Kheddouci, "Critical node detection problem in networks: A survey," Comp. Sci. rev., vol. 28, pp. 92–117, 2018.
[10] B. Addis, M. Di Summa, and A. Grosso, "Identifying critical nodes in undirected graphs: Complexity results and polynomial algorithms for the case of bounded treewidth," Disc. App. Math., vol. 161(16), pp. 2349–2360, 2013.
[11] A. Arulselvan, C. Commander, L. Elefteriadou, and P. Pardalos, "Detecting critical nodes in sparse graphs," Comp. & Oper. Res., vol. 36(7), pp. 2193–2200, 2009.
[12] A. Arulselvan, C. Commander, O. Shylo, and P. Pardalos, "Cardinality-constrained critical node detection problem," Perf. mod. and ris. manag. in comm. sys.(Springer), pp. 79–91, 2011.
[13] M. Di Summa, A. Grosso, and M. Locatelli, "Complexity of the critical node problem over trees," Comp. & Oper. Res., vol. 38(12), pp. 1766–1774, 2011.
[14] S. Shen and J. Smith, "Polynomial-time algorithms for solving a class of critical node problems on trees and series-parallel graphs," Net., vol. 60(2), pp. 103–119, 2012.
[15] M. Lalou, M. Tahraoui, and H. Kheddouci, "Component-cardinality-constrained critical node problem in graphs," Disc. App. Math., vol. 210, pp. 150–163, 2016.
[16] A. Berger, A. Grigoriev, and R. Zwaan, "Complexity and approximability of the k-way vertex cut," Net., vol. 63(2), pp. 170–178, 2014.
[17] R. Aringhieri, A. Grosso, P. Hosteins, and R. Scatamacchia, "A general evolutionary framework for different classes of critical node problems," Eng. App. of Art. Int., vol. 55, pp. 128–145, 2016.
[18] M. R Garey and D. S Johnson, "Computers and intractability: a guide to the theory of np-completeness," WH Freeman & Co., San Francisco, 1979.