Computers & Industrial Engineering 184 (2023) 109585

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Two-edge connectivity with disjunctive constraints: Polyhedral analysis and Branch-and-Cut

Intesar Almudahka [a,*], Ibrahima Diarrassouba [b], A. Ridha Mahjoub [a,c], Raouia Taktak [d]

[a] Department of Statistics and Operations Research, College of Science, Kuwait University, Kuwait
[b] University Le Havre Normandy, UNIHAVRE, LMAH, FR-CNRS-3335, Le Havre, 76600, France
[c] Université Paris-Dauphine, PSL LAMSADE CNRS UMR 7243, Paris, France
[d] ISIMS & Sm@rts/CRNS, Université de Sfax, Sfax, Tunisia

ARTICLE INFO

Keywords:
Network survivability
Connectivity constraints
Disjunctive constraints
Branch-and-cut algorithm
Facet

ABSTRACT

In this paper we consider the two-edge-connected subgraph problem with disjunctive constraints. We investigate the convex hull of the solutions to this problem. We describe several classes of valid inequalities, and discuss their facial aspect. We also devise separation routines for these inequalities. Using this, we propose a Branch-and-Cut algorithm for the problem along with an extensive computational study is presented.

1. Introduction

A graph is said to be *two-edge-connected* if between every pair of nodes there are at least two edge-disjoint paths, that is to say having no edges is common.

Given a graph $G = (V, E)$ with weights on the edges, the *Two-Edge-Connected subgraph Problem* (TECP) is to find a minimum weight two-edge-connected subgraph spanning all the nodes of G. This problem has applications in telecommunications (Grötschel, Monma, & Stoer, 1992b, 1995; Kerivin & Mahjoub, 2005; Stoer, 1992). If a traffic is routed between two nodes s and t in a telecommunication network, and a link of the routing path fails, the traffic can be rerouted on the second path. Hence, the existence of two edge-disjoint paths between every pair of nodes is a security condition for the network to continue to be functional in the event of a failure.

Two-connectivity is seen by telecommunication operators as a very convenient topology for telecommunication networks. In some special situations, two links may fail at the same time. If these edges belong to the two selected paths between s and t, the traffic may be cut (and even lost) between s and t. For this, it would be more secure in this case to have at most, one, of these two edges in the network.

In addition to the connectivity requirement, we also consider in this paper, the additional security condition, that is if two given links have important probability to fail at the same time, at most one of them will be considered in the network. This is known as the *disjunctive constraints*. In telecommunication networks, the disjunctive constraints may also be motivated by the fact that physical links are managed by different operators, and conflicts between operators induce conflicts between some pairs of links.

More precisely, we suppose given a graph $G = (V, E)$, where V represents the telecommunication nodes (centers), E the possible links between the nodes of G, a set C of disjunctive pairs of links (edges) of G, and weights on the edges. The *Disjunctive Two-Edge-Connected subgraph Problem* (DTECP) consists in finding a minimum weight two-edge-connected spanning subgraph of G containing at most one edge from each disjunctive pair of edges of C.

1.1. Contributions

In this paper we consider this problem from a polyhedral point of view. We describe several classes of valid inequalities, and give necessary conditions and sufficient conditions for these inequalities to be facet defining. We also devise separation procedures for these inequalities. Using this we propose a Branch-and-Cut algorithm for the problem along with an extensive computational study is presented.

1.2. Related works

To the best of our knowledge, the DTECP has not been considered before in the literature. However the closely related TECP has seen a great attention. As this problem is NP-hard (Erickson, Monma, &

* Corresponding author.
E-mail addresses: intesar.m@ku.edu.kw (I. Almudahka), ibrahima.diarrassouba@univ-lehavre.fr (I. Diarrassouba), ali.mahjoub@ku.edu.kw, ridha.mahjoub@lamsade.dauphine.fr (A.R. Mahjoub), raouia.taktak@isims.usf.tn (R. Taktak).

https://doi.org/10.1016/j.cie.2023.109585
Received 16 February 2023; Received in revised form 14 July 2023; Accepted 31 August 2023
Available online 11 September 2023

Veinott Jr., 1987), it follows that the DTECP so is. The TECP has been shown to be polynomially solvable in special classes of graphs, like series–parallel graphs (Winter, 1986), Halin graphs (Winter, 1987) and perfectly two-edge-connected graphs (Mahjoub, 1997).

The TECP can be seen as a linear relaxation of the traveling salesman problem, in which we seek a minimum weight Hamiltonian cycle, that is a cycle going through each node of the graph exactly once. Monma, Munson, and Pulleyblank (1990) show that if the weights satisfy the triangle inequalities ($c(e_1) \leq c(e_2) + c(e_3)$ for every triangle (e_1, e_2, e_3) where $c(e), e \in E$ are the weights on the edges), then an optimal solution of the TECP has a maximum degree 3. In Grötschel and Monma (1990), and in Grötschel, Monma, and Stoer (1992a) consider the TECP within the framework of a more general model. They describe several facet defining inequalities and propose cutting plane based algorithms. In Mahjoub (1994), Mahjoub gives a complete description of the two edge connected subgraph polytope when the underlying graph is series–parallel. He also introduces a large class of valid inequalities and gives necessary and sufficient conditions for these inequalities to be facet defining. In Barahona and Mahjoub (1995) Barahona and Mahjoub describe this polytope in the class of Halin graphs. Baïou and Mahjoub extend in Baïou and Mahjoub (1997) the characterization given by Mahjoub (1994) to the Steiner two-edge-connected subgraph polytope. And Didi Biha and Mahjoub (Biha & Mahjoub, 1996) extend this to the more general Steiner k-edge connected subgraph polytope for all k even. In the Steiner case, it is supposed that the node set is partitioned into two sets, a terminal set and a Steiner set. The terminal nodes must belong to the solution and the Steiner nodes are optional nodes, that is to say they may belong to the solution if they permit to reduce the cost. In Biha and Mahjoub (1996), Didi Biha and Mahjoub give a complete description for the k-edge connected subgraph polytope in series–parallel graphs when k is odd.

Further polyhedral results on the TECP can be found in Biha, Kerivin, and Mahjoub (2008), Grötschel and Monma (1989), Kerivin and Mahjoub (2005), Kerivin, Mahjoub, and Nocq (2004), Mahjoub and Pesneau (2008). Various variants of the TECP have been the subject of extensive research. Biha et al. (2008) consider the $(1, 2)$ survivable network design problem. Here the nodes of the network are of two types, type 1 and type 2. The nodes of type 1 should be linked to every node in the final network and those of type 2 need more survivability, they must have at least two edge-disjoint paths between every pair of them. The authors discuss the polytope of the solutions of this problem. They describe structural proprieties of this polytope and characterize it in a subclass of series–parallel graphs containing outerplanar graphs. In Kerivin et al. (2004), Kerivin et al. describe valid inequalities for the problem in both edge and node cases and devise Branch-and-cut algorithms. In Kerivin and Mahjoub (2002), Kerivin and Mahjoub show that the so-called partition inequalities can be separated in polynomial time.

Much work has also been done on the hop-constrained TECP. In this variant, the required paths between the nodes should not exceed a certain length L, where the length of a path is the number of its edges. Huygens, Mahjoub, and Pesneau (2004) consider the case when $L = 3$, and the network contains exactly one demand, that is to say only one pair of nodes to be linked by at least two bounded paths. They give a complete description of the related polytope. In Huygens, Labbé, Mahjoub, and Pesneau (2007) Huygens et al. consider the same variant with an arbitrary number of demands. They give an integer programming formulation in the space of the natural variables. They also describe valid inequalities and propose a Brand-and-Cut algorithm. More work on the hop-constrained TECP can be found in Dahl, Huygens, Mahjoub, and Pesneau (2006), Diarrassouba, Gabrel, Mahjoub, Gouveia, and Pesneau (2016), Diarrassouba, Kütücu, and Mahjoub (2016), Diarrassouba, Mahjoub, Mahjoub, and Yaman (2018), Huygens and Mahjoub (2007), Mahjoub, Poss, Simonetti, and Uchoa (2019), Mahjoub, Simonetti, and Uchoa (2013).

Fortz and Labbé (2002) consider the 2-node connected subgraph problem. Here, it is required that between each pair of nodes there are at least two node-disjoint paths and the total weight is minimum. The authors study the problem when every edge of the solution should belong to a bounded ring (cycle). They discuss the related polytope and describe several classes of facets. In Fortz, Labbé, and Maffioli (2000), Fortz et al. discuss a Branch-and-Cut algorithm for the problem. Fortz, Mahjoub, McCormick, and Pesneau (2006) consider the edge case, that is the 2-edge connected subgraph problem with bounded rings. They describe several classes of valid inequalities and devise a Branch-and-Cut algorithm. The closely related 2-node connected subgraph problem has also been widely studied. The reader is referred to Mahjoub (2017) for more details on this problem.

Disjunctive constraints are considered for several combinatorial optimization problems in the literature, some related to graph-based problems and others to packing-like problems. In Darmann, Pferschy, Schauer, and Woeginger (2011) Darwann et al. study the minimum spanning tree, the maximum matching and the shortest path problems with disjunctive constraints. They establish some complexity results for these problems. Further works, considering the disjunctive constraints, have been developed for variants of the maximum flow problem (Pferschy & Schauer, 2013; Şuvak, Altınel, & Aras, 2020, 2021). In Pferschy and Schauer (2013), Pferschy and Schauer consider the maximum flow problem when certain pairs of arcs in the graph cannot be simultaneously used for sending flow. They show that the problem with disjunctive constraints is strongly NP-hard even if the so-called conflict graph consists only either of unconnected edges or of disjoint paths of length 3. Şuvak et al. (2020) present a mixed integer programming formulation for the problem and devise Benders' decomposition and Branch-and-Bound based resolution methods. In Şuvak et al. (2021), they present further mixed integer programming formulations and exact solution techniques. Disjunctive constraints are also considered for the minimum spanning tree problem (Carrabs, Cerulli, Pentangelo, & Raiconi, 2021; Carrabs & Gaudioso, 2021; Samer & Urrutia, 2015). In Carrabs et al. (2021), Carrabs et al. present a Branch-and-Cut for the problem. A Lagrangian approach is described in Carrabs and Gaudioso (2021). In Samer and Urrutia (2015), further Branch-and-Cut algorithms are proposed. In Öncan, Şuvak, Akyüz, and Altınel (2019), Öncan et al. consider the assignment problem with conflicts. The authors characterize a special case in which the problem is polynomially solvable. They also propose a Branch-and-Bound based algorithm for the problem in general case. In Öncan, Zhang, and Punnen (2013), Öncan et al. address the minimum-cost perfect matching problem with conflicts. The problem is known to be strongly NP-hard. The authors give additional complexity results and identify new polynomially solvable cases. Heuristic algorithms are also proposed. In Cao (1992), Cao discusses a transportation problem with disjunctive constraints. A Branch-and-Bound algorithm is designed for the problem. Besides these graph-based problems with conflict constraints, packing problems with disjunctive constraints have been of great interest. Much work has been conducted on the disjunctively constrained knapsack problem. Yamada, Kataoka, and Watanabe (2002) propose a heuristic approach, and Senisuka, You, and Yamada (2005) give a Lagrangian relaxation based algorithm for the problem. Pferschy and Schauer (2009) present pseudo-polynomial algorithm for the problem for graphs with bounded tree-width and chordal graphs. Ben Salem, Hanafi, Taktak, and Ben-Abdallah (2017) propose an efficient probabilistic tabu search algorithm with multiple neighborhoods for the problem. They show that their algorithm outperforms the ones on the literature. Ben Salem, Taktak, Mahjoub, and Ben-Abdallah (2018) discuss the problem from a polyhedral point of view. The authors propose valid inequalities and devise a Branch-and-Cut algorithm for the problem. Luiz, Santos, and Uchoa (2021) describe a new class of facet defining inequalities for the problem and show that these inequalities are very effective. The knapsack problem with conflicts has also been considered within efficient Branch-and-Bound algorithms, see Bettinelli, Cacchiani, and Malaguti (2017), Coniglio, Furini, and San Segundo (2021). The disjunctive constraints have been also considered for the bin packing problem. The reader is referred to Ekici (2021), Sadykov and Vanderbeck (2013).

1.3. Organization of the paper

The paper is organized as follows. In Section 2, we give an integer programming formulation for the DTECP and introduce the associated polytope. In Section 3, we describe some classes of valid inequalities and give necessary conditions and sufficient conditions for these inequalities to be facet defining. In Section 4, we give our separation routines, describe our Branch-and-Cut algorithm and present our computational results. Section 5 will be devoted to some concluding remarks.

1.4. Notations

The rest of this section is devoted to more definitions and notations.

We consider undirected and loopless graphs. We denote a graph by $G = (V, E)$ where V is the node set and E the edge set. An edge between two nodes u and v will be denoted by uv. A path P in G is a sequence of the form $P = (v_1, e_1, v_2, e_2, \ldots, v_p, e_p, v_{p+1})$ where $e_i = v_i v_{i+1}$, for $i = 1, \ldots, p$. The nodes $v_1, \ldots, v_{p+1}$ are the *extremities* of P and p is its *length*. The path P is said to be *closed* if $v_1 = v_{p+1}$. A *cycle* is a *closed* path. If $W \subset V$, then the set of edges having one node in W and the other in $V \setminus W$ is called a *cut* and denoted by $\delta(W)$. If $W = \{v\}$ for some $v \in V$, then we write $\delta(v)$ for $\delta(\{v\})$. Given two nodes $s, t \in V$, a cut $\delta(W)$ such that $s \in W$ and $t \in V \setminus W$ is called an *st-cut*.

If U and W are two disjoint node subsets of V, then $[U, W]$ will denote the set of edges between U and W. For $u, v \in V$, we will write $[u, v]$ for $[\{u\}, \{v\}]$. Given a subset of edges $F \subseteq E$, we let $G \setminus F$ denote the graph obtained from G by deleting F.

Given a vector $x \in \mathbb{R}^E$ and an edge subset $F \subseteq E$, we let $x(F) = \sum_{e \in F} x_e$.

Given a graph $G = (V, E)$ and a set C of conflict pairs of edges, we denote by D the set of edges in the disjunctive pairs of C. We denote by $G_c = (V_c, E_c)$ the graph whose nodes correspond to the edges of D, and two nodes of V_c are linked by an edge if the corresponding edges are in conflict.

A *stable set* (*independent set*) in a graph is a subset of pairwise non-adjacent nodes. Thus the subset of edges in D taken in any solution of the DTECP corresponds to a stable set in G_c. In consequence, any valid inequality for the stable set polytope in G_c yields a valid inequality for the DTECP(G).

2. Formulation and polyhedral analysis

2.1. ILP formulation

Let $G = (V, E)$ be a graph. Let $c_e, e \in E$ be weights on the edges of E. Let C be a set of pairs of edges in *conflict*. If $(e, f) \in C$, then e and f cannot be taken together in a solution. If an edge e belongs to a pair of D, then we also write $e \in D$. One of the famous results in graph theory establishing a relation between paths and cuts is the following

Theorem 1 (*Menger* (*1927*)). *Given two nodes $s, t \in V$, and an integer $k > 0$, there exist k edge-disjoint paths between s and t if and only if every st-cut of G contains at least k edges.*

Let x_e be a binary variable which takes 1 if edge e is selected in the solution and 0 if not. From Theorem 1 it follows that the DTECP is equivalent to the following integer linear program

$$\min \sum_{e \in E} c_e x_e \tag{1}$$

$$x(\delta(W)) \geq 2 \qquad \text{for all } W \subset V, \text{ and } W \neq \emptyset, \tag{2}$$

$$x_e + x_f \leq 1 \qquad \text{for all } (e, f) \in C, \tag{3}$$

$$0 \leq x_e \leq 1 \qquad \text{for all } e \in E, \tag{4}$$

$$x_e \in \{0, 1\} \qquad \text{for all } e \in E. \tag{5}$$

Inequalities (2) express the fact that every two-edge connected set of edges contains at least two edges from each cut. Inequalities (3) are the disjunctive constraints. Constraints (4) and (5) are the trivial and the integrality constraints.

Let DTECP(G) denote the convex hull of the incidence vectors of all the solutions of the DTECP, *i.e.*,

$$\text{DTECP}(G) = conv\{x \in \{0, 1\}^E \,:\, x \,\text{satisfies}\, (2) - (5)\}.$$

The aim of this work is to study the polytope DTECP(G), and to devise a Branch-and-Cut algorithm for the DTECP.

Throughout the paper, in order to guarantee a feasible solution, we suppose that the subgraph obtained by deleting the edges in D is 2-edge connected. This is not a restrictive assumption. In practice, the number of conflict pairs is not high compared to the total number of edge-pairs in the network. As the underlying network is generally complete, the graph obtained when removing the conflicts remains 2-edge connected.

In what follows we characterize the dimension of DTECP(G).

2.2. Dimension of DTECP(G)

An edge $e \in E$ is called *essential* if e belongs to every solution of the problem. Note that if e is essential, then e cannot be in conflict with other edges of the graph. Let E^* be the set of essential edges.

Theorem 2. *The dimension of DTECP(G) is equal to $|E| - |E^*|$.*

Proof. Since each edge $e \in E^*$ yields the equation $x_e = 1$ and these equations are linearly independent, it follows that dim(DTECP(G)) $\leq |E| - |E^*|$.

Now, for each non essential edge e, there is a subset D_e of E such that $E_e = \left(E \setminus (D \cup \{e\})\right) \cup D_e$ induces a 2-edge connected subgraph without conflicts. Consider the following edge sets

$E_0 = E \setminus D$,

E_e for all $e \in E \setminus E^*$,

$E_f = E_0 \cup \{f\}$ for all $f \in D$.

These sets are solutions for the DTECP. Moreover, their incidence vectors are affinely independent. Thus dim(DTECP(G)) $\geq |E| - |E^*|$.

Hence the result follows. □

In the rest of the paper we suppose that $G \setminus D$ is 3-edge-connected. Therefore, G has no essential edges, and from Theorem 2, it follows that DTECP(G) is full dimensional. Hence, two inequalities define the same facet of DTECP(G) if and only if one is a positive multiple of the other.

2.3. Basic inequalities' facial aspect

In this section, we will discuss the facial aspect of the basic inequalities.

In what follows, we first characterize when the trivial inequalities define facets of the DTECP(G).

Theorem 3. *Inequality $x_e \geq 0$ defines a facet of DTECP(G) if and only if e is not in a 3-edge cutset (that is a cut of three edges).*

Proof. If e belongs to a 3-edge cutset $\{e, f_1, f_2\}$, then the following inequalities are valid for DTECP(G).

$$\begin{aligned} x_e + x_{f_1} + x_{f_2} &\geq 2, \\ -x_{f_1} &\geq -1, \\ -x_{f_2} &\geq -1. \end{aligned}$$

By summing these inequalities we get $x_e \geq 0$. Therefore, this inequality cannot define a facet.

Now suppose that edge e is not in a 3-edge cutset. If $e \notin D$ (resp. $e \in D$), then for every edge $f \in E \setminus \left(\{e\} \cup D\right)$ (resp. $f \in E \setminus D$),

$E \setminus (\{e,f\} \cup D)$ (resp. $E \setminus (\{f\} \cup D)$) is a solution for the DTECP. For otherwise, e would be in a 3-edge cutset. Consider the solutions

$E_e = E \setminus (\{e\} \cup D)$,
$E_f = E_e \setminus \{f\}$ for all $f \in E_e$,
$E_f = E_e \cup \{f\}$ for all $f \in D$.
(resp.
$E_e = E \setminus D$,
$E_f = E_e \setminus \{f\}$ for all $f \in E_e$,
$E_f = E_e \cup \{f\}$ for all $f \in D \setminus \{e\}$.)

These constitute a set of $|E|$ solutions of the DTECP which do not contain e. Moreover their incidence vectors are affinely independent. Hence $x_e \geq 0$ defines a facet of DTECP(G). □

Theorem 4. *Inequality $x_e \geq 1$ defines a facet of DTECP(G) if and only if for all $f \in E \setminus \{e\}, (e,f) \notin C$.*

Proof. First recall that G has no essential edges.

If $(e,f) \in C$ for some $f \in E \setminus \{e\}$, then for every solution of the face $\{x \in DTECP(G) : x_e = 1\}$ we have $x_f = 0$. However $x_e \leq 1$ is not a positive multiple of $x_f \geq 0$. Thus the former inequality cannot define a facet.

Conversely, suppose that $(e,f) \notin C$ for all $f \in E \setminus \{e\}$. Consider the solutions

$E_e = E \setminus D$,
$E_f = E_e \setminus \{f\}$ for all $f \in E \setminus D, f \neq e$,
$E_f = E_e \cup \{f\}$ for all $f \in D$.

As $G \setminus D$ is 3-edge connected, these edge sets induce solutions of the DTECP(G). Moreover, their incidence vectors satisfy $x_e \leq 1$ with equality and are affinely independent. Hence $x_e \leq 1$ is facet defining for the DTECP(G). □

Now, we discuss facet conditions for the cut inequalities.

Theorem 5. *A cut inequality $x(\delta(W)) \geq 2$ defines a facet of the DTECP(G) only if*

(1) $G(W)$ and $G(V \setminus W)$ are both 2-edge connected.
(2) There does not exist a partition W_1, W_2 of W such that the edges of $[W_1, W_2]$ are pairwise in conflict.

Proof.

(1) Suppose, for instance, that $G(W)$ is not 2-edge connected. If $G(W)$ is not connected, then there is a partition W_1, W_2 of W such that $[W_1, W_2] = \emptyset$. Hence $\delta(W) = \delta(W_1) \cup \delta(W_2)$ with $\delta(W_1) \cap \delta(W_2) = \emptyset$. Which yields $x(\delta(W)) = x(\delta(W_1)) + x(\delta(W_2)) \geq 2 + 2 = 4$. Hence, the face defined by $x(\delta(W)) \geq 2$ is an empty set, and consequently, cannot be a facet.
Now, suppose that $G(W)$ is connected. Then there is a partition W_1, W_2 of W such that $[W_1, W_2] = \{f\}$, for some $f \in E$. Thus $\delta(W_1) = [W_1, V \setminus W] \cup \{f\}$, $\delta(W_2) = [W_2, V \setminus W] \cup \{f\}$, and $\delta(W) = [W_1, V \setminus W] \cup [W_2, V \setminus W]$ (see Fig. 1).
We have

$$x(\delta(W_1)) = x_f + x[W_1, V \setminus W] \geq 2$$

$$x(\delta(W_2)) = x_f + x[W_2, V \setminus W] \geq 2$$

As $x_f \leq 1$, this yields

$$x(\delta(W)) = x[W_1, V \setminus W] + x[W_2, V \setminus W] \geq 1 + 1 = 2.$$

Thus $x(\delta(W)) \geq 2$ is redundant with respect to the inequalities $x(\delta(W_1)) \geq 2$, $x(\delta(W_2)) \geq 2$ and $x_f \leq 1$, and cannot then be facet defining.

(2) Suppose there exists a partition W_1, W_2 of W such that the edges of $[W_1, W_2]$ are pairwise in conflict. Hence any solution of the DTECP cannot take more than one edge from $[W_1, W_2]$. Moreover, any solution which does not intersect $[W_1, W_2]$ cannot belong to the face defined by $x(\delta(W)) \geq 2$. In fact, as in the previous case, the inequality would not be tight for the solution in this case. Consequently, any solution of the face defined by $x(\delta(W)) \geq 2$ uses exactly one edge from $[W_1, W_2]$. As $x(\delta(W)) \geq 2$ is not a positive multiple of $x[W_1, W_2] = 1$, the former inequality cannot define a facet. □

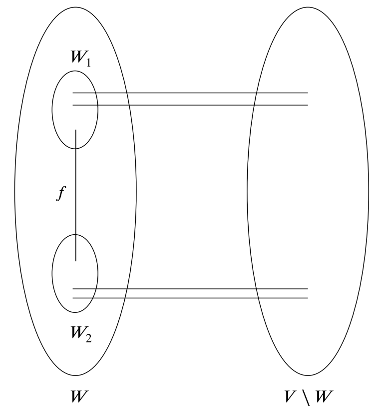

Fig. 1. Cut $\delta(W)$ and partition of W.

For a node set $U \subseteq V$, let $E_0(U) = E(U) \setminus D$, and $G_0(U) = (U, E_0(U))$, that is the subgraph obtained from $G(U)$ by removing all the edges in conflict.

Theorem 6. *A cut inequality $x(\delta(W)) \geq 2$ defines a facet of the DTECP(G) if the following are satisfied.*

(1) Conditions (1) and (2) of Theorem 5 are satisfied.
(2) $G_0(W)$ and $G_0(V \setminus W)$ are 2-edge connected.
(3) If an edge $e = uv$ belongs to a 2-edge cutset $[W_1, W_2]$ of $G_0(W)$ with $u \in W_1$ and $v \in W_2$, then there are two edges g_1, g_2 of $\delta(W)$, not in conflict, such that $g_1 \in [W_1(u), V \setminus W]$ and $g_2 \in [W_2(u), V \setminus W]$ where $W_1(u)$ (resp. $W_2(v)$) is the node subset of W_1 (resp. W_2) of the 2-edge component of $G_0(W_1)$ (resp. $G_0(W_2)$) containing u (resp. v).
(4) The number of edges in $\delta(W)$, not in conflict with any other edge, is at least 2.

Proof. Denote $x(\delta(W)) \geq 2$ by $ax \geq \alpha$ and let $bx \geq \beta$ be a facet defining inequality of the DTECP(G) such that

$$\{x \in \text{DTECP}(G) : ax = \alpha\} \subseteq \{x \in \text{DTECP}(G) : bx = \beta\}.$$

It suffices to show that there is a scalar ρ such that $b = \rho a$.

Let $\delta(W) = \{e_1, \ldots, e_q\}$. As $G \setminus D$ is 3-edge connected, for every edge $e_i \in \delta(W)$, there should exist an edge $e_j \in \delta(W) \setminus \{e_i\}$ such that e_i and e_j are not in conflict, and there are two edges $e_l, e_k \in \delta(W) \setminus \{e_i\}$ that are not in conflict. Let $e_1, \ldots, e_r, r \leq q$ be the edges of $\delta(W) \setminus D$, that is to say the edges of $\delta(W)$ which are not in conflict with any other edge of E. Let $E_0 = E \setminus D$.

First we show that

$$b(e) = b(e') \text{ for all } e, e' \in \delta(W). \tag{6}$$

By Condition (4), $r \geq 2$. Consider the edge sets

$$S_i = \{e_i, e_{i+1}\} \cup (E_0 \setminus \delta(W)), \text{ for } i = 1, \ldots, r \pmod r.$$

Since $G_0(W)$ and $G_0(V \setminus W)$ are both 2-edge connected, and do not contain edges in conflict, $S_1, \ldots, S_q$ induce solutions of the DTECP. Moreover, their incidence vectors satisfy $ax^{S_i} = \alpha$ for $i = 1, \ldots, q$. Hence $bx^{S_i} = \beta$ for $i = 1, \ldots, q$.

Thus, we have that $b(e_i) = b(e_{i+2})$ for $i = 1, \dots, r \pmod r$. This implies that

$$b(e_i) = b(e_j) \text{ for all } i, j \in \{1, \dots, r\},\ i \neq j. \tag{7}$$

Now consider the sets

$$S_i = \{e_1, e_i\} \cup (E_0 \setminus \delta(W)), \text{ for } i = r+1, \dots, q,$$

which induce solutions for the DTECP. As their incidence vectors satisfy $ax \geq \alpha$ with equality, it follows that $bx^{S_i} = \beta$ for $i = r+1, \dots, q$.

Therefore

$$b(e_i) = b(e_j) \text{ for all } i, j \in \{r+1, \dots, q\}. \tag{8}$$

Moreover, as $ax^{S_1} = ax^{S_{r+1}} = \alpha$, $0 = ax^{S_1} - ax^{S_{r+1}} = b(e_2) - b(e_{r+1})$. Therefore, $b(e_2) = b(e_{r+1})$. This, together with (7) and (8) yields (6).

Next we show that $b(e) = 0$ for all $e \in E \setminus \delta(W)$. Let e be an edge of $G_0(W)$. If $G_0(W) \setminus e$ is 2-edge connected, then the edge set

$$S_e = \{e_1, e_2\} \cup (E_0 \setminus (\delta(W) \cup \{e\}))$$

is still a solution of the DTECP. Since $ax^{S_e} = \alpha$, we have that $bx^{S_e} = \beta$. As $ax^{S_1} = \alpha$, and hence $bx^{S_1} = \beta$, it follows that

$$0 = bx^{S_1} - bx^{S_e} = b(e).$$

Thus $b(e) = 0$. Here S_1 is the solution given above.

Now suppose that $G_0(W) \setminus e$ is not 2-edge connected. As $G_0(W)$ is 2-edge connected, there should exist a partition W_1, W_2 of W such that $[W_1, W_2] \setminus \{e\} = \{f\}$ for some edge f. Moreover $G_0(W_1)$ and $G_0(W_2)$ must be both connected. For otherwise, $G_0(W)$ would not be 2-edge connected, which is a contradiction.

Hence W_1 (resp. W_2) can be partitioned into $(W_1^1, \dots, W_1^l)$ (resp. $(W_2^1, \dots, W_2^m)$) such that

(1) $G_0(W_1^j)$ (resp. $G_0(W_2^j)$) is 2-edge connected for $j = 1, \dots, l$ (resp. $j = 1, \dots, m$).
(2) The graph $G_0'(W_1)$ (resp. $G_0'(W_2)$) obtained from $G_0(W_1)$ (resp. $G_0(W_2)$) by contracting the sets $W_1^j, j = 1, \dots, l$ (resp. $W_2^j, j = 1, \dots, m$) is a tree T (resp. T'). Furthermore, as $G_0(W)$ is 2-edge connected and $|[W_1, W_2]| = 2$, the trees T and T' should be chordless paths, and their extremities are linked by edges e and f (see Fig. 2).

So the graph $G_0'(W)$ obtained from $G_0(W)$ by the contraction of $w_1^1, \dots, w_1^l, w_2^1, \dots, w_2^m$ is a cycle. By condition (3) there must exist two edges g_1 and g_2 not in conflict such that $g_1 \in [W_1(u), V \setminus W]$ and $g_2 \in [W_2(v), V \setminus W]$. As $G_0(V \setminus W)$ is 2-edge connected, it follows that the edge sets

$$S = \{g_1, g_2\} \cup (E_0 \setminus (\delta(W) \cup \{e\}))$$
$$S' = S \cup \{e\}$$

both induce solutions of the DTECP.

Moreover we have $ax^S = ax^{S'} = \alpha$, implying that $bx^S = bx^{S'} = \beta$. This yields $b(e) = 0$. As e is an arbitrary edge in $E_0(W)$, it follows that $b(e) = 0$ for all $e \in E_0(W)$. Similarly, we have that $b(e) = 0$ for all $e \in E_0(V \setminus W)$.

Now, if e is an edge of $D \setminus \delta(W)$, then the edge set $S_1' = S_1 \cup \{e\}$ is a solution of the problem where S_1 is the solution above. As $ax^{S_1} = ax^{S_1'} = \alpha$ and hence $bx^{S_1} = bx^{S_1'} = \beta$, we obtain that $b(e) = 0$. So altogether we obtain that

$$b(e) = \rho \text{ for all } e \in \delta(W), \text{ for some } \rho \in \mathbb{R},$$
$$b(e) = 0 \text{ otherwise}.$$

Thus we have that $b = \rho a$, which completes the proof. □

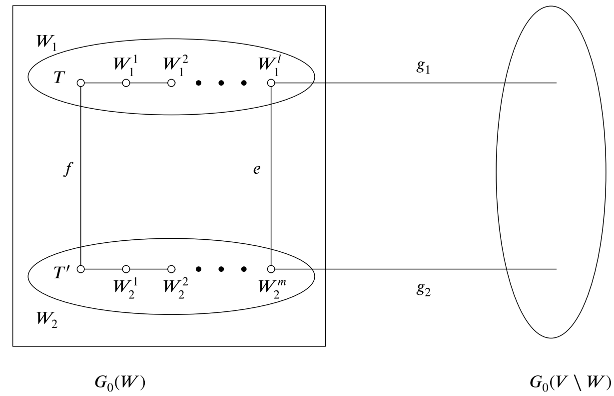

Fig. 2. Graph $G \setminus (D \cup (\delta(W) \setminus \{g_1, g_2\}))$.

3. Valid inequalities and facets

In this section we will introduce some valid inequalities for the DTECP(G), and give necessary conditions and sufficient conditions for these inequalities to be facet defining. First we describe valid inequalities related to the stable structure given by inequalities (3).

Recall that the conflict graph is denoted by $G_c = (V_c, E_c)$, where V_c refers to the edges of G that are in conflict and $E_c = C$.

3.1. Clique inequalities

Given an undirected graph, a *clique* is a subset of vertices such that every two distinct vertices are adjacent.

A clique is said to be *maximal* if it is not strictly contained in another clique.

Consider the conflict graph G_c. If K is a clique of G_c, then all the vertices of K are pairwise in conflict. This implies that one can consider at most one edge from K in a solution. Therefore the following inequality is valid for DTECP(G).

$$\sum_{e \in K} x_e \leq 1. \tag{9}$$

Inequalities of type (9) will be called *clique inequalities*.

3.2. Odd cycle inequalities

A *cycle* in a graph is a sequence of nodes and edges beginning and ending at the same node. We will denote a cycle C by its sequence of nodes and write $C = (v_1, \dots, v_k)$. A cycle of k nodes is said to be of *length* k. A cycle is said to be *even* (*odd*) if its length is even (odd). A *chord* of a cycle is an edge joining two non-consecutive nodes of the cycle. A cycle is called *simple* if its nodes $v_1, \dots, v_k$ are all different.

Odd cycles induce valid inequalities for the independent set problem and hence for the DTECP(G). Consider a cycle C of the conflict graph G_c where nodes are $e_1, \dots, e_k$ with k odd. Then the inequalities

$$x_{e_i} + x_{e_{i+1}} \leq 1, \text{ for all } i = 1, \dots, k,$$

where the indices are taken modulo k, are valid for DTECP(G). By summing these inequalities, dividing by 2 and rounding down the right hand side, we obtain the inequality

$$\sum_{e \in C} x_e \leq \frac{k-1}{2}, \tag{10}$$

which is valid for DTECP(G). Inequalities of type (10) are called *Odd Cycle Inequalities*.

Next we introduce a further class of valid inequalities combining both structures, the 2-edge connectivity and the edge-conflict.

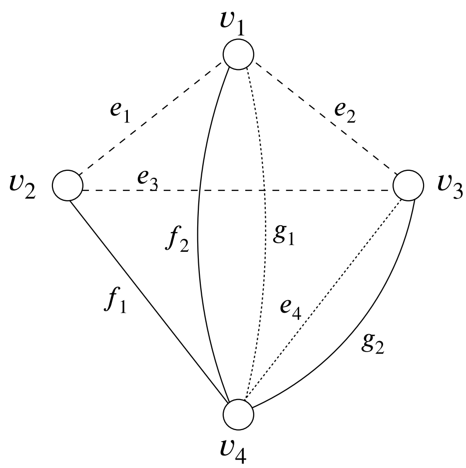

Fig. 3. Fractional point cut by a disjunctive F-partition inequality.

3.3. Disjunctive F-partition inequalities

Consider the graph H of Fig. 3 and suppose $D = \{g_1, g_2\}$. Let $\overline{x} \in \mathbb{R}^8$ be the solution given by $\overline{x}_{e_1} = \overline{x}_{e_2} = \overline{x}_{e_3} = \frac{1}{2}$, $\overline{x}_{f_1} = \overline{x}_{f_2} = \overline{x}_{g_2} = 1$, et $\overline{x}_{e_4} = \overline{x}_{g_1} = 0$.

It is easy to see that $\overline{x}$ satisfies inequalities (2)–(4). Moreover, it is a unique solution of the system

$$\begin{array}{rrl}
x(\delta(v_1)) = & x_{e_1} + x_{f_2} + x_{g_1} + x_{e_2} = & 2,\\
x(\delta(v_2)) = & x_{e_1} + x_{e_3} + x_{f_1} = & 2,\\
x(\delta(v_3)) = & x_{e_2} + x_{e_3} + x_{e_4} + x_{g_2} = & 2,\\
 & x_{f_1} = & 1,\\
 & x_{f_2} = & 1,\\
 & x_{g_2} = & 1,\\
 & x_{e_4} = & 0,\\
 & x_{g_1} = & 0,
\end{array}$$

which is non-singular. Therefore $\overline{x}$ is an extreme point of DTECP(G). On the other hand, one can see that the inequality

$$x_{e_1} + x_{e_2} + x_{e_3} + x_{e_4} \geq 2 \tag{11}$$

is valid for DTECP(G) and cut $\overline{x}$.

Moreover, inequality (11) defines a facet of DTECP(G). In what follows, we show that inequality (11) is a special case of a more general class of valid inequality for the DTECP(G).

Theorem 7. *Let $V_0, V_1, \ldots, V_p$ be a partition of V. Let $K_1, \ldots, K_q$ be q sets of edges such that $K_i \subseteq \delta(V_0)$, for $i = 1, \ldots, q$, $K_i \cap K_j = \emptyset, i \neq j$, and the edges of K_i are pairwise in conflict, (that is K_i is a complete graph in the conflict graph) for $i = 1, \ldots, q$. Let F be a set of edges such that $F \cap K_i = \emptyset$ for $i = 1, \ldots, q$, $F \subseteq \delta(V_0)$, and $|F| + q$ is odd. Then the inequality*

$$x(\delta(V_0, V_1, \ldots, V_p) \setminus (F \cup \bigcup_{i=1}^{q} K_i)) \geq p - \lfloor \frac{|F| + q}{2} \rfloor \tag{12}$$

is valid for the DTECP(G) (see Fig. 4).

Proof. The following inequalities are valid for the DTECP(G),

$$x(\delta(V_i)) \geq 2, \text{ for } i = 1, \ldots, p,$$

$$-x_e \geq -1, \text{ for all } e \in F,$$

$$-x(K_i) \geq -1, \text{ for } i = 1, \ldots, q,$$

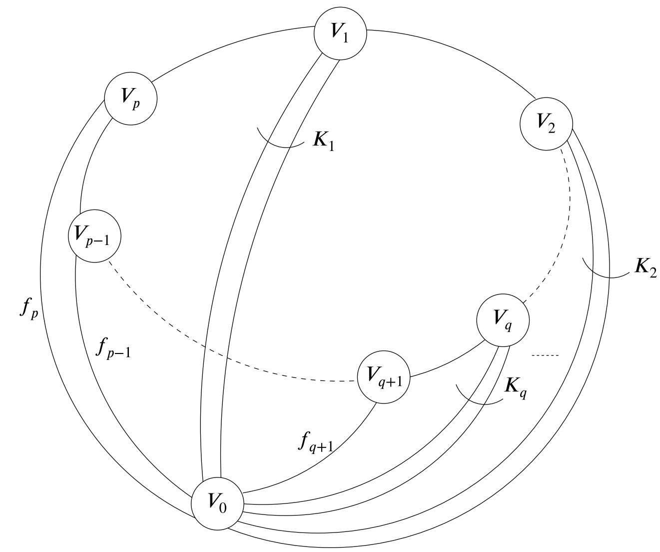

Fig. 4. A disjunctive F-partition configuration.

$$x_e \geq 0, \text{ for all } e \in \delta(V_0) \setminus (F \cup \bigcup_{i=1}^{q} K_i).$$

By summing these inequalities, we obtain

$$2x(\delta(V_0, V_1, \ldots, V_p)) \setminus (F \cup \bigcup_{i=1}^{q} K_i) \geq 2p - |F| - q.$$

By dividing by 2 and rounding up the right hand side, we obtain (12). $\square$

The inequalities of type (12) will be called *disjunctive F-partition inequalities*. These inequalities generalize the classical so-called F-partition inequalities (Mahjoub, 1994). The latter correspond to the case where $q = 0$. Observe that inequality (11) is a disjunctive F-partition inequality where $p = 3$, $q = 1$, $V_i = \{v_i\}$, $i = 1, 2, 3$, $V_0 = \{v_4\}$, $F = \{f_1, f_2\}$, and $K_1 = \{g_1, g_2\}$.

In what follows we describe some necessary conditions for inequalities (12) to be facet defining.

Theorem 8. *Inequality (12) defines a facet of DTECP(G) only if:*

(1) $|F| + q$ is odd and $\lfloor \frac{|F|+q}{2} \rfloor < p$.
(2) $G(V_i)$ is connected for $i = 1, \ldots, p$.
(3) If $\lfloor \frac{|F|+q}{2} \rfloor = k$, then V_0 is adjacent to at least $k+1$ V_i's, $i \neq 0$.
(4) There does not exist a partition V_i^1, V_i^2 of a set $V_i, i \in \{1, \ldots, p\}$, such that $[V_i^1, V_i^2]$ induces a complete graph in the conflict graph.
(5) The K_i's are maximal complete subgraphs in the conflict graph.
(6) F does not contain edges in conflict.

Proof.

(1) is easily seen to be true.
(2) Suppose $G(V_i)$ is not connected, and let V_i^1, V_i^2 be a partition of $G(V_i)$ such that $[V_i^1, V_i^2] = \emptyset$. Consider the partition $V_0, \ldots, V_{i-1}, V_i^1, V_i^2, V_{i+1}, \ldots, V_p$ of V. Inequality (12), with respect to this partition, yields

$$x(\delta(V_0, \ldots, V_{i-1}, V_i^1, V_i^2, V_{i+1}, \ldots, V_p)) \setminus (F \cup \bigcup_{i=1}^{q} K_i) \geq p + 1 - \lfloor \frac{|F| + q}{2} \rfloor.$$

As $[V_i^1, V_i^2] = \emptyset$, this inequality dominates (12), and hence the latter cannot define a facet.

(3) Suppose that V_0 is adjacent to l sets say $V_1, V_2, \dots, V_l$ with $l \leq k$. Any solution of the DTECP satisfies the inequalities

$$x(\delta(V_i)) \geq 2, \text{ for } i = l+1, \dots, p,$$

$$x(\delta(V_{l+1} \cup \cdots \cup V_p)) \geq 2.$$

By summing these inequalities, we obtain

$$2x(\delta(V_{l+1}, \dots, V_p)) + 2x(\delta(V_{l+1} \cup \cdots \cup V_p)) \geq 2(p - l + 1).$$

Since $\lfloor \frac{|F|+q}{2} \rfloor = k$, and $l \leq k$, it follows that

$$x\big(\delta(V_{l+1}, \dots, V_p)\big) + x\big(\delta(V_{l+1} \cup \cdots \cup V_p)\big) \geq p - \lfloor \frac{|F|+q}{2} \rfloor + 1.$$

As $x_e \geq 0$ for all $e \in E$, the left hand side of the above inequality is less than or equal to that of (12). This yields

$$x\big(\delta(V_0, V_1, \dots, V_p) \setminus (F \cup \bigcup_{i=1}^{q} K_i)\big) \geq p - \lfloor \frac{|F|+q}{2} \rfloor + 1,$$

which dominates (12). Implying that (12) cannot be facet defining.

(4) Suppose that there is a partition V_i^1, V_i^2 of a certain $V_i, i \in \{1, \dots, p\}$, such that $[V_i^1, V_i^2]$ induces a complete subgraph in the conflict graph. Consider the partition $V_0, \dots, V_{i-1}, V_i^1, V_i^2, V_{i+1}, \dots, V_p$. Inequality (12) with respect to this partition can be written as

$$x\big(\delta(V_0, \dots, V_{i-1}, V_i^1, V_i^2, V_{i+1}, \dots, V_p) \setminus (F \cup \bigcup_{i=1}^{q} K_i)\big) \geq p + 1 - \lfloor \frac{|F|+q}{2} \rfloor,$$

which is equivalent to

$$x\big(\delta(V_0, V_1, \dots, V_p) \setminus (F \cup \bigcup_{i=1}^{q} K_i)\big) + x[V_i^1, V_i^2] \geq p + 1 - \lfloor \frac{|F|+q}{2} \rfloor. \quad (13)$$

Since $[V_i^1, V_i^2]$ is a complete graph in the conflict graph, we have that

$$x[V_i^1, V_i^2] \leq 1 \quad (14)$$

is valid for DTECP(G).

Now inequality (12) can be obtained by summing (13) and (14). Hence (12) cannot define a facet.

(5) Suppose that for some $i_0 \in \{1, \dots, q\}$, K_{i_0} is not maximal, that is to say, there is a complete graph K'_{i_0} in the conflict graph strictly containing K_{i_0}. We have the following valid inequalities

$$x(\delta(V_i)) \geq 2, \text{ for } i = 1, \dots, p,$$

$$-x_e \geq -1, \text{ for all } e \in F,$$

$$-x((K_i)) \geq -1, \ i = 1, \dots, q; i \neq i_0,$$

$$-x(K'_{i_0}) \geq -1,$$

$$x_e \geq 0 \text{ for all } e \in \delta(V_0) \setminus (F \cup \bigcup_{i=1, i \neq i_0}^{q} K_i \cup K'_{i_0}).$$

By summing these inequalities, dividing by 2, and rounding up the right hand side, we obtain

$$x(\delta(V_0, V_1, \dots, V_p) \setminus (F \cup \bigcup_{i=1}^{q} K_i \cup (K'_{i_0} \setminus K_{i_0}))) \geq p - \lfloor \frac{|F|+q}{2} \rfloor.$$

As $K'_{i_0} \setminus K_{i_0} \neq \emptyset$ and $x_e \geq 0$ for all $e \in K'_{i_0} \setminus K_{i_0}$, this inequality dominates (12). Therefore (12) cannot define a facet.

(6) Suppose that F contains two edges, say e_1, e_2, in conflict. Consider the valid inequalities

$$x(\delta(V_i)) \geq 2 \text{ for } i = 1, \dots, p,$$

$$-x(K_i) \geq -1 \text{ for } i = 1, \dots, q,$$

$$-x(e_1) - x(e_2) \geq -1,$$

$$-x_e \geq -1 \text{ for all } e \in F \setminus \{e_1, e_2\},$$

$$x_e \geq 0 \text{ for all } e \in \delta(V_0) \setminus (F \cup \bigcup_{i=1}^{q} K_i).$$

By summing these inequalities, dividing by 2 and rounding up the right hand side, we get

$$x(\delta(V_0, V_1, \dots, V_p)) \setminus (F \cup \bigcup_{i=1}^{q} K_i) \geq p - \lfloor \frac{|F|-1+q}{2} \rfloor = p - \lfloor \frac{|F|+q}{2} \rfloor + \frac{1}{2},$$

as $|F| + q$ is odd.

This implies that inequality (12) cannot be satisfied with equality by any solution. Hence, it cannot define a facet. □

Theorem 9. *Inequality (12) defines a facet of the DTECP(G) if the following hold.*

(1) K_i is a maximal clique for $i = 1, \dots, q$, and p is odd.
(2) No edge of $E(K_i)$ is in conflict with edges of $E(K_j), i \neq j$ or edges of $\delta(V_0, V_1, \dots, V_p) \setminus (F \cup \bigcup_{i=1}^{q} K_i)$.
(3) $G(V_i) \setminus D$ is 3-edge connected, for $i = 0, 1, \dots, p$.
(4) $[V_i, V_{i+1}] \geq 1$ for $i = 1, \dots, p \,(\mathrm{mod}\, p)$.
(5) None of the edges of $[V_i, V_j]$ is in conflict with edges of $E \setminus [V_i, V_j]$.
(6) $K_i \subseteq [V_0, V_i]$ for $i = 1, \dots, q$.
(7) $F \subseteq E \setminus D$, that is the edges of F are pairwise not in conflict, $F \cap \delta(V_i) = \emptyset$ for $i = 1, \dots, q$, and $|F \cap \delta(V_i)| = 1$ for $i = q+1, \dots, p$.
(8) There are q edges, $f_1, \dots, f_q$ of $K_1, \dots, K_q$, respectively, and $p - q$ edges $f_{q+1}, \dots, f_p$ of $F \cap \delta(V_{q+1}), \dots, F \cap \delta(V_p)$, respectively, which are not in conflict with any edge of $\bigcup_{i=0}^{p} E(V_i)$.

Proof. Let us denote inequality (12) by $ax \geq \alpha$, and let $bx \geq \beta$ be a facet defining inequality such that

$$\{x \in \text{DTECP}(G) \mid ax = \alpha\} \subseteq \{x \in \text{DTECP}(G) \mid bx = \beta\}.$$

we will show that there is ρ such that $b = \rho a$.

First we will show that $b(e) = b(e')$ for all $e, e' \in [V_i, V_{i+1}]$, for $i = 1, \dots, p$ (where the indices are taken modulo p).

By Conditions (4), (5), there is an edge, say e_i, between V_i and V_{i+1}, for $i = 1, \dots, p \,(\mathrm{mod}\, p)$, which is not in conflict with any edge of $E \setminus [V_i, V_{i+1}]$. And by condition (8) there are edges $f_1, \dots, f_q$ of $K_1, \dots, K_q$ and $f_{q+1}, \dots, f_p$ of $F \cap \delta(V_{q+1}), \dots, F \cap \delta(V_p)$, respectively, which are not in conflict with any edge of $\bigcup_{i=0}^{p} E(V_i)$. Let $E_i = E(V_i) \setminus D$, that is the set of edges in $E(V_i)$ which are not in conflict with other edges. Consider the edge set

$$S_0 = \{e_1, e_3, \dots, e_p\} \cup \{f_1, f_2, \dots, f_q, f_{q+1}, \dots, f_p\} \cup \bigcup_{i=0}^{p} E_i.$$

By Condition (3), $G(E_i) = G(V_i) \setminus D$ is 3-edge connected. By Conditions (2), (5), (7) it follows that S_0 is a solution of the DTECP.

Let $S_1 = (S_0 \setminus \{e_p\}) \cup \{e_{p-1}\}$. Clearly S_1 is also a solution of the DTECP. Moreover, $ax^{S_0} = ax^{S_1} = \alpha$. Hence $bx^{S_0} = bx^{S_1} = \beta$, implying that $b(e_p) = b(e_{p-1})$. By symmetry, we obtain that

$$b(e) = b(e') \text{ for all } e, e' \in [V_i, V_{i+1}], i = 1, \dots, p. \quad (15)$$

Let $S_3 = S_0 \setminus \{f_1\}$ (Recall that $f_1 \in K_1$). It is easy to see that S_3 remains feasible for the DTECP. As $ax^{S_0} = ax^{S_3} = \alpha$, it follows that $0 = bx^{S_0} - bx^{S_1} = b(f_1)$. By symmetry we obtain that

$$b(f) = 0 \text{ for all } f \in K_i, i = 1, \dots, q. \quad (16)$$

Let $S_4 = S_1 \setminus \{f_{p-1}\}$. As S_4 is also a solution of the DTECP and $ax^{S_1} = ax^{S_4} = \alpha$, this yields $0 = bx^{S_1} - bx^{S_4} = b(f_{p-1}) = 0$. By symmetry, it follows that

$$b(f) = 0 \text{ for all } f \in F. \tag{17}$$

Now let $e \in [V_i, V_j], j \in \{1, \dots, p\} \setminus \{i-1, i+1\}$. Without loss of generality, we may suppose that $i = 1$ and $j = 2r$ for some integer $r \geq 2$. Consider the solution

$$S_5 = (S_0 \setminus \{e_1, e_3, \dots, e_{2r-1}\}) \cup \{e, e_2, e_4, \dots, e_{2r-2}\}.$$

It is easily seen that S_5 is a solution for the DTECP. Moreover, $ax^{S_5} = \alpha$. Therefore $bx^{S_5} = bx^{S_0} = \beta$. By (15) it then follows that $b(e) = b(e_1)$. By symmetry, we obtain that

$$b(e) = b(e') \text{ for all } e, e' \in \delta(V_1, \dots, V_p) \setminus \delta(V_0). \tag{18}$$

Now consider an edge $e \in [V_0, V_i] \setminus (F \cup \bigcup_{i=1}^{q} K_i)$, for $i \in \{1, \dots, p\}$. Without loss of generality, we may suppose that $i = p$. Let $S_6 = (S_0 \setminus \{e_p\}) \cup \{e\}$.

Since $G(V_i) \setminus D$ is 3-edge connected, we have that S_6 is feasible for the problem. As $ax^{S_6} = ax^{S_0} = \alpha$, this implies that $0 = bx^{S_0} - bx^{S_6} = b(e_p) - b(e)$. Hence $b(e) = b(e_p)$. By symmetry, and from (18), it follows that

$$b(e) = b(e') \text{ for all } e, e' \in \delta(V_0, V_1, \dots, V_p) \setminus (F \cup \bigcup_{i=1}^{q} K_i). \tag{19}$$

Now we show that $b(e) = 0$ for all $e \in G(V_i)$ for $i = 0, 1, \dots, p$.

Let $e \in E(V_i)$, for some $i \in \{0, 1, \dots, p\}$. If $e \in E \setminus D$, then as $G(V_i) \setminus D$ is 3-edge connected, we have that $S_0 \setminus \{e\}$ is still a solution of the problem, implying that $b(e) = 0$.

So suppose that $e \in D$. Let

$$S_7 = S_0 \cup \{e\}.$$

By condition (8), e is not in conflict with any edge of S_0. Hence, S_7 is feasible for the DTECP. As $ax^{S_7} = ax^{S_0} = \alpha$, it follows that $0 = bx^{S_7} - bx^{S_0} = b(e)$.

Therefore

$$b(e) = 0 \text{ for all } e \in E(V_i), \text{ for } i = 0, 1, \dots, p. \tag{20}$$

From (16)–(20), it follows that

$$b(e) = \rho \text{ for all } \; e \in \delta(V_0, V_1, \dots, V_p) \setminus (F \cup \bigcup_{i=1}^{q} K_i), \text{ for some } \; \rho \in \mathbb{R},$$

$$b(e) = 0 \text{ for all } \; e \in E \setminus (\delta(V_0, V_1, \dots, V_p) \setminus (F \cup \bigcup_{i=1}^{q} K_i)).$$

Therefore $b = \rho a$, and the proof is complete. □

3.4. Reduction operations

Let $G = (V, E)$ be a graph and D a set of pairs of edges in conflict in G. Let $P(G)$ be the linear relaxation of DTECP(G), that is to say the polytope given by the cut, disjunctive and trivial inequalities. Let $\overline{x} \in \mathbb{R}^E$ be an extreme point of $P(G)$. In what follows we discuss some reduction operations with respect to $\overline{x}$ that generalize those introduced by Fonlupt and Mahjoub (2006) for the 2-edge connected polytope.

- O_1: Delete an edge with $\overline{x}(e) = 0$.
- O_2: Contract an edge if one of its extremities is of degree 2.
- O_3: Contract a node set U such that $G(U)$ is 2-edge connected, $\overline{x}(e) = 1$ for all $e \in E(U)$.

Note that the edges such that $\overline{x}(e) = 1$ are not in conflict. We have the following consequence whose proof is similar to that of Theorem 1 in Kerivin et al. (2004).

Proposition 10. *Let G' and $\overline{x}'$ be the graph and the solution obtained from G and $\overline{x}$ by repeated applications of O_1, O_2, O_3. Then $\overline{x}'$ is an extreme point of $P(G')$.*

Operations O_1, O_2, O_3 can be performed in polynomial time. Also they can be used in a preprocessing phase within a cutting plane algorithm for the DTECP. Moreover we have the following result which is easily seen to be true.

Proposition 11. *Let $G' = (V', E')$ and $\overline{x}'$ be the graph obtained from G and $\overline{x}$ by repeated applications of operations O_1, O_2, O_3.*

(1) If $a'x \geq \alpha'$ is a valid inequality for the $DTECP(G')$ of type (2) or (12), then the inequality $ax \geq \alpha$ such that $a(e) = a'(e)$ for all $e \in E'$, $a(e) = 0$ for all $e \in E \setminus E'$ and $\alpha = \alpha'$ is valid for the DTECP(G). Moreover, if $a'x \geq \alpha$ is violated by $\overline{x}'$, then $ax \geq \alpha$ is violated by $\overline{x}$.

(2) If there is an inequality of type (2) (resp. (12)) that is violated by $\overline{x}$, then there is an inequality of type (2) (resp. (12)) violated by $\overline{x}'$.

Using the previous polyhedral analysis, we propose a Branch-and-Cut algorithm to solve the DTECP. Separation routines and experiments held on a set of problem instances are discussed in the next section.

4. Branch-and-Cut algorithm

4.1. Algorithm description

The ILP formulation proposed in Section 2 contains an exponential number of cut inequalities. As a consequence, we first consider a restricted version of the corresponding linear program (LP) as an initial LP, where a restricted number of cut inequalities is generated; *i.e.*, we generate only degree inequalities on the nodes of the graph. The first iteration of the Branch-and-Cut algorithm consists hence is solving the initial linear program, denoted by LP_0, whose constraints are the cut inequalities (2) written only for the nodes, the conflict inequalities (3) as well as the trivial inequalities (4).

Denote by $\overline{x} \in \mathbb{R}^E$ the solution of the linear relaxation for the DTECP. $\overline{x}$ is optimal for the restricted LP if and only if it satisfies all the cut inequalities (2). In general, this is not the case. Therefore, violated cut inequalities are added to the restricted LP, by solving a subproblem called *separation problem*. The process is repeated until no more violated inequality is found. The final solution, is hence optimal for the linear relaxation. If the solution is integral then it is optimal for the related DTECP. If not, then we create new subproblems by branching on a fractional variable. The separation routine is then considered at each new node of the tree and the process continues until no branching is needed.

In our Branch-and-Cut algorithm, along with the basic cut inequalities we also separate the valid inequalities described in Section 3. The inequalities are separated in the following order:

1. Cut inequalities,
2. Cliques inequalities,
3. Odd-Cycles inequalities,
4. Disjunctive F-partition inequalities.

In what follows, we describe the separation routines used to separate the inequalities mentioned above. Depending on the class of the valid inequality, we devise exact or heuristic procedures of separation.

4.2. Separation routines

4.2.1. Cut inequalities separation

The cut separation problem consists in finding one or more cut inequality (2) that are violated by the current solution $\overline{x}$, or to say that such inequality does not exist. This can be done using an exact algorithm based on the so-called Gomory–Hu tree (Gomory & Hu,

1961) computed using the efficient implementation of Gusfield (Gusfield, 1987, 1990). The minimum cut separating any pair of vertices s and t in graph G is nothing but the minimum cut separating s and t in the Gomory–Hu tree.

Recall that, by the maximum flow - minimum cut theorem (Ford & Fulkerson, 1956), the minimum cut problem can be solved in polynomial time. Consequently, in our Branch-and-Cut algorithm, for all the cut separation problems, we use the algorithm of Goldberg and Tarjan (1988), which is one of the most powerful implementations of the maximum flow problem.

4.2.2. Clique inequalities separation

Clique inequalities (9) represent an interesting family of valid inequalities that belongs to the independent set structure of the problem.

Since identifying the whole set of cliques in the conflict graph $G_c = (V_c, E_c)$ (where $E_c = D$) is NP-hard, we choose to generate a set of cliques $\mathcal{K}$ using a greedy heuristic algorithm. Let $\mathcal{K} = \emptyset$ be the set of cliques that we are looking to identify, and assume that initially $K = \emptyset$. For each vertex $v \in V_c$, we iterate the following. We first consider $K = \{v\}$. Then, we add to K a maximum-degree vertex $u \in V_c$ such that $(u, v) \in D$. Then, among all the other nodes of V_c, we add the one that is universal to vertices in K (*i.e.*, that is adjacent to all the vertices of K), and with a maximum degree in G_c. The process is repeated until no more universal node can be added. At the end, if $|K| > 2$, we obtain a clique K that we add to the family of cliques $\mathcal{K}$.

Consider now the separation phase. Recall that $\overline{x}$ is the optimal solution obtained for the linear relaxation of the cut formulation after separating all the cut inequalities. We choose to apply, with slight modifications, a simple greedy heuristic presented by Nemhauser and Sigismondi (1992) for the independent set problem. The heuristic can be described as follows. We first choose a vertex, say $i \in V_c$, of maximum-weight with respect to $\overline{x}$, and set $K = \{i\}$. Then the following process is repeated. Determine, if there exists in G_c, a maximum-weight vertex according to $\overline{x}$, say j, that is universal to K (*i.e.*, j is adjacent to every vertex in K). Add j to K and reiterate this step until no more universal vertex can be found. If $|K| > 2$ and $\overline{x}(K) > 1$, then the corresponding clique inequality is violated. The whole process is then repeated for another initial vertex i until either we find a violated inequality or a maximum number of iterations is reached. In our algorithm, we choose to generate at most one violated clique inequality per iteration, since in general this kind of inequalities were rarely found during the separation phase (see Section 4.3).

4.2.3. Odd cycle inequalities separation

Separating the odd cycle inequalities (10) can be computed exactly in polynomial time as it has been proposed by Grötschel, Lovász, and Schrijver (2012). The separation procedure is described in Mahjoub (2010). Consider the current solution $\overline{x}$ obtained after separating the cut and clique inequalities. Consider $(e_i, e_j) \in C$ and let $z_e = 1 - \overline{x}_{e_i} - \overline{x}_{e_j}$. It is clear that, since $\overline{x}$ satisfies (3) and (4), $z_e \geq 0$ for all $e = (e_i, e_j) \in C$. Using this, inequalities (10) can be written as

$$\sum_{e \in C} z_e \geq 1 \qquad \text{for all} \quad C \text{ odd cycle of } G_c. \tag{21}$$

As a consequence, separating inequalities (10) regarding $\overline{x}$ reduces to separating inequalities (21) regarding z, and this can be done as follows. Consider graph G_c and let z_e be the weight of conflict-edge for all $e = (e_i, e_j) \in C$. In order to separate inequalities (21), one has to look for a minimum-weight cycle in a graph with nonnegative weights. This problem can be solved in polynomial time. To this end, consider the bipartite graph $\widetilde{G_c} = (V_c^1 \cup V_c^2, \widetilde{D})$ obtained from G_c in the following way. For each vertex $v \in V_c$, consider two vertices $v_1 \in V_c^1$ and $v_2 \in V_c^1$, and for each edge $(u, v) \in D$ consider two edges $(u_1, v_2) \in \widetilde{D}$ and $(u_2, v_1) \in \widetilde{D}$ with the same weight $\tilde{z}_{u_1 v_2} = \tilde{z}_{u_2 v_1} = z_{uv}$. Obviously, looking for a minimum-weight odd cycle in G_c with respect to z going through a vertex v is nothing but determining a minimum-weight path in $\widetilde{G_c}$ with respect to $\tilde{z}$ between v_1 and v_2. Since $\tilde{z} \geq 0$, this can be done in polynomial time, using for instance Dijkstra's algorithm. At the end of this step, we have (if there is any) a minimum-weight cycle in G_c, say C. If $\sum_{e \in C} z_e < 1$, then a violated odd cycle inequality is detected.

4.2.4. Disjunctive F-partition inequalities separation

Separating the disjunctive F-partition inequalities (12) is an NP-hard problem, since it is a generalization of the classical F-partition inequalities introduced by Mahjoub (1994). In order to separate the disjunctive F-partition inequalities, we propose the following heuristic. First we apply the reduction operations until no more operation can be used. Then, we determine an odd cycle, $(v_1, v_2, \ldots, v_p)$, $p \geq 3$, in the resulting graph whose edges have fractional values and whose nodes are tight (that is to say that $\overline{x}(\delta(v_i)) = 2$, for all $i = 1, \ldots, p$). We let V_i be the set obtained by expanding v_i for $i = 1, \ldots, p$, and $V_0 = V \setminus \bigcup_{i=1}^{p} V_i$. We choose the cliques K_i, $i = 1, \ldots, q$ and the edges F among the edges of $\delta(V_0)$ having values greater than $\frac{1}{2}$ such that $|F| + q$ is odd. If such a configuration cannot be found, then we look for another odd cycle. Since the cycle is obtained by a direct labeling procedure, our heuristic runs in a linear time.

In the following section, we present the results obtained by the Branch-and-Cut algorithm on several sets of instances.

4.3. Computational results

We devise the Branch-and-Cut algorithm described in the previous section. The algorithm is implemented in C++ using SCIP (https://scipopt.org/) to manage the Branch-and-Cut tree and Cplex (http://www.ilog.com/products/cplex/) as a linear solver. We set a time limit to 2 hours. Our algorithm is tested over several sets of instances generated from TSPlib (http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/).

We consider complete graphs TSPlib-based instances generated as follows:

- A first set of instances where conflicts are randomly generated.
- A second set of instances where conflicts are taken from the conflict graphs of the graph coloring problem taken from https://mat.tepper.cmu.edu/COLOR/instances.html.

For both sets of instances, it is guaranteed that the graph $G \setminus D$ is at least 3-edge connected, which means that the instances are solvable.

Detailed results are reported in Table 1 (random conflicts), Tables 2 and 3 (graph-coloring problem conflicts).

Entries of the tables are the following:

Name	:	Name of the instance;
$\lvert V\rvert$	:	number of vertices in the graph G ranging from 51 to 574;
$\lvert D\rvert$	:	number of pairs in conflict ranging from 650 to 49629;
FP	:	number of generated F-partition inequalities;
Cliques	:	number of generated Cliques inequalities;
O-Cycles	:	number of generated Odd Cycles inequalities;
Cuts-1/Cuts-2	:	number of separated cuts for the formulation without/with valid inequalities;
Sub-1/Sub-2	:	number of nodes in the Branch-and-Cut tree (number of treated subproblems) for the formulation without/with valid inequalities;
Gap-1/Gap-2 (%)	:	relative error between the best upper bound; and the lower bound obtained at the root for the formulation without/with valid inequalities;
CPU-1/CPU-2 (s)	:	total execution time in seconds for the formulation without/with valid inequalities.

Overall, we tested over 100 instances. Our Branch-and-Cut algorithm of the formulation with valid inequalities was able to solve 95 instances to optimality within the time limit. 57% of the instances were solved

Table 1
Branch-and-Cut results (1).

Name	$\|V\|$	$\|D\|$	Cuts-1	Gap-1	Sub-1	CPU-1	Cuts-2	FP	Cliques	O-cycles	Gap-2	Sub-2	CPU-2
a150_650	150	650	554	0.69	17	00:00:08	133	72	0	0	**0**	1	00:00:03
a150_700	150	700	696	0.17	19	00:00:11	125	16	0	0	**0**	1	00:00:02
a150_800	150	800	552	0.17	10	00:00:09	106	119	0	0	**0**	1	00:00:03
a150_900	150	900	536	0.17	11	00:00:09	123	15	0	0	**0**	1	00:00:02
a200_650	200	650	2063	0.44	126	00:00:31	650	201	0	0	31	4	00:00:15
a200_700	200	700	2320	0.47	318	00:00:43	175	62	0	0	23	2	00:00:07
a200_900	200	900	1562	0.44	170	00:00:37	401	120	0	0	30	3	00:00:09
d300_1250	300	1250	3045	0.42	1983	00:05:06	1617	1253	0	0	0.18	139	00:15:28
d300_1500	300	1500	9349	0.42	3290	00:13:01	1493	1592	0	0	0.22	226	00:20:27
d300_750	300	750	5722	0.42	2516	00:05:30	1985	1392	0	0	0.15	141	00:15:39
d350_1000	350	1000	4392	0.36	1276	00:05:47	1834	2227	0	0	**0**	91	00:32:14
d350_1250	350	1250	3903	0.36	586	00:03:48	1061	1599	0	0	**0**	39	00:19:18
d350_1500	350	1500	5674	0.36	1566	00:06:25	1150	2305	0	0	0.13	133	00:36:10
d350_2000	350	2000	2884	0.36	866	00:04:02	1372	1683	0	0	0.12	57	00:19:46
d350_750	350	750	3984	0.36	753	00:03:57	1060	644	0	0	**0**	9	00:05:04
d400_1000	400	1000	3353	0.22	751	00:07:10	1040	1118	0	0	**0**	9	00:07:51
d400_1250	400	1250	1646	0.22	172	00:04:06	1437	1931	0	0	**0**	45	00:19:35
d400_1500	400	1500	1835	0.22	331	00:04:57	948	1019	0	0	**0**	12	00:08:34
d400_2000	400	2000	3106	0.22	590	00:06:53	947	1086	0	0	**0**	12	00:09:55
d400_750	400	750	2864	0.22	454	00:05:55	4234	4383	0	0	0.2	561	01:59:27
d450_1250	450	1250	17977	0.27	5713	00:41:06	4274	2064	0	0	0.26	175	02:00:00
d450_1500	450	1500	23931	0.27	7575	00:46:44	4743	4973	0	0	0.2	522	00:41:01
gr125_650	125	650	820	34	13	00:00:10	279	35	0	0	10	13	00:00:04
gr125_700	125	700	600	**40**	5	00:00:05	676	49	0	0	**59**	5	00:00:04
gr125_800	125	800	635	11	5	00:00:06	277	31	0	0	7.6	13	00:00:03
gr125_900	125	900	346	7.6	14	00:00:07	567	44	0	0	20	4	00:00:04
gr175_650	175	650	493	8.2	14	00:00:09	356	62	0	0	6.3	2	00:00:05
gr175_700	175	700	806	27	14	00:00:13	326	94	0	0	20	2	00:00:05
gr175_800	175	800	779	6.1	13	00:00:20	226	9	0	0	**0**	1	00:00:03
gr175_900	175	900	605	9.9	7	00:00:10	573	89	0	0	9.7	5	00:00:08
gr200_650	200	650	522	17	7	00:00:10	458	100	0	0	8	3	00:00:09
gr200_700	200	700	493	16	7	00:00:11	255	13	0	0	**0**	1	00:00:04
gr200_800	200	800	1105	8.9	22	00:00:24	510	59	0	0	21	2	00:00:07
gr200_900	200	900	560	10	8	00:00:15	507	41	0	0	43	3	00:00:09
usa500_1000	500	1000	22096	0.5	**10748**	00:17:10	2445	758	0	0	**0**	**93**	00:31:13

to optimality within less than 1 min, and overall only 14 instances were solved to optimality within more than 10 min. Concerning the basic formulation, 62% of the instances were solved within less than 1 minute, and only 8 were solved to optimality within more than 10 minutes. Note that in terms of execution time, the basic formulation is slightly better than the one with valid inequalities for some instances. This can be explained by the fact that the formulation with valid inequalities spends more time than the basic one while separating valid inequalities.

Having more computational time for the formulation with valid inequalities helps improve the linear relaxation compared with the basic formulation. In fact, the linear relaxation of the formulation with valid inequalities outperforms the basic one (except for 12 instances). 26 instances were solved with a gap equal to 0 for the formulation with valid inequalities while only 8 instances have a gap equal to 0 for the basic formulation (see lines with bold gaps in all the tables). However, for both formulations, we attain important values for the gap reaching 40% for the basic formulation and 59% for the formulation with valid inequalities (see instance $gr125_700$ for both formulations in Table 1, gaps are in bold). This means that more valid inequalities would be necessary to improve the linear relaxation. One should mainly detect facets to cut better the current fractional points, since the separated valid inequalities may not be facet defining within the tested conditions.

Concerning the Branch-and-Cut tree, the formulation with valid inequalities is able to solve to optimality 10 instances at the root node compared with the basic formulation that solves to optimality only 2 instances. Overall, we note that the formulation with valid inequalities is devised within smaller Branch-and-Cut trees for all the instances. For many instances, the basic formulation is solved using an extended Branch-and-Cut tree while reaching only few nodes for the formulation with valid inequalities. For example, instance usa500_1000 (Table 1 in bold) is solved with 10748 subproblems for the basic formulation and only 93 subproblems for the formulation with valid inequalities. This observation is supported by the results depicted in Fig. 5. In this figure, we plot the cumulative number of subproblems in the Branch-and-Cut trees for all the tested instances. The blue-'*' curve is related to the basic formulation, that is without generating valid inequalities. The green-'+' one corresponds to the formulation with valid inequalities. We clearly see through this figure that the total number of subproblems is significantly more important in the basic formulation compared with the one with valid inequalities. This proves the efficiency of the added valid inequalities in cutting fractional solutions and enabling to fathom many branches of the Branch-and-Cut tree.

Along with solving instances using small Branch-and-Cut trees, the valid inequalities obviously help generating less cut inequalities (cuts-2) compared to the basic formulation (cuts-1). For example, for the instance rd400_15668 (Table 3 in bold) the number of cuts that are generated is 25603 compared to only 1847 for the formulation with valid inequalities.

The results illustrated in Fig. 6 come to support this remark. This figure illustrates the cumulative number of generated cut inequalities for both formulations, the basic one (blue-'*' curve) and the one with valid inequalities (green-'+' curve). It is clear through this figure that the number of generated cut inequalities for the basic formulation is quite larger than the one for the formulation with valid inequalities. Thus, these valid inequalities help cutting fractional points without necessarily generating a huge number of cut inequalities.

In particular, we mainly note that F-partition inequalities play a crucial role in improving the linear relaxation and well cutting some fractional solutions. For several instances, we are generating an important number of F-partitions, which helps accelerating the resolution mainly in terms of subproblems. For the instance rd400_15668 (Table 3 in bold), separating 3563 F-partition inequalities helps solving the instance using a Branch-and-Cut tree of only 211 compared to the basic formulation that solves the same instance within 9468 subproblems.

Table 2
Branch-and-Cut results (2).

Name	$\|V\|$	$\|D\|$	Cuts-1	Gap-1	Sub-1	CPU-1	Cuts-2	FP	Cliques	O-cycles	Gap-2	Sub-2	CPU-2
a280_12458	280	12458	2543	0.43	545	00:00:40	2543	923	0	0	11	5	00:07:19
berlin52_12458	52	12458	207	**0**	48	00:00:06	207	129	4	0	**0**	34	00:00:06
ch130_12458	130	12458	616	1.2	67	00:00:08	616	741	0	0	**0**	21	00:00:13
eil51_12458	51	12458	587	4	403	00:00:13	587	470	23	1	3.6	223	00:00:16
eil76_12458	76	12458	388	0.59	73	00:00:05	388	114	2	0	**0**	10	00:00:03
kroA100_12458	100	12458	585	1.2	172	00:00:08	585	298	0	0	0.47	27	00:00:09
kroA150_12458	150	12458	402	0.47	53	00:00:06	402	701	0	0	0.083	6	00:00:13
kroA200_12458	200	12458	6250	0.73	2685	00:01:30	6250	912	0	0	0.44	168	00:02:26
kroB100_12458	100	12458	428	1.1	30	00:00:04	428	350	0	0	**0**	10	00:00:07
kroB150_12458	150	12458	1340	1.1	349	00:00:23	1340	733	0	0	0.71	114	00:00:41
kroB200_12458	200	12458	3723	0.78	1427	00:00:59	3723	1001	0	0	0.22	80	00:01:25
kroC100_12458	100	12458	182	**0**	11	00:00:03	182	395	0	0	1.3	11	00:00:08
lin105_12458	105	12458	117	2.4	4	00:00:02	117	3	0	0	**0**	1	00:00:01
lin318_12458	318	12458	3443	0.31	839	00:01:30	3443	4533	0	0	0.29	138	00:51:16
pr107_12458	107	12458	88	**0**	1	00:00:01	88	0	0	0	**0**	1	00:00:02
pr76_12458	76	12458	660	1.7	690	00:00:09	660	131	0	0	0.28	13	00:00:04
rat195_12458	195	12458	1320	0.7	1056	00:00:52	1320	2066	0	0	0.5	120	00:01:29
rat99_12458	99	12458	351	**0**	7	00:00:02	351	29	0	0	0.083	4	00:00:02
rd100_12458	100	12458	351	0.47	4	00:00:03	351	4	0	0	10	2	00:00:02
rd400_12458	400	12458	38310	0.54	7376	00:16:19	38310	4511	0	0	0.33	162	02:00:00
st70_12458	70	12458	168	0.56	11	00:00:04	168	137	0	0	**0**	6	00:00:04
ts225_12458	225	12458	74	1.5	881	00:00:16	74	104	0	0	40	3	00:00:07
u159_12458	159	12458	685	2.3	10	00:00:04	685	84	0	0	29	2	00:00:04
u574_12458	574	12458	75135	0.39	18463	01:37:25	75135	2617	0	0	0.8	79	02:00:00
a280_15668	280	15668	4304	0.43	1261	00:01:09	1721	896	0	0	22	6	00:01:25
berlin52_15668	52	15668	465	16	513	00:02:03	543	1161	409	2	15	457	00:00:42
ch130_15668	130	15668	834	0.65	178	00:00:12	417	462	0	0	0.19	15	00:00:12
eil51_15668	51	15668	2042	17	7788	00:07:15	728	2340	776	7	16	1064	00:01:00
eil76_15668	76	15668	289	2.6	41	00:00:21	147	404	22	0	2.4	39	00:00:09
kroA100_15668	100	15668	352	3	155	00:00:41	272	467	11	0	1.9	102	00:01:19
kroA150_15668	150	15668	722	0.69	372	00:00:56	444	585	0	0	0.37	43	00:01:48
kroA200_15668	200	15668	10672	0.79	3623	00:06:58	3047	2132	0	0	0.49	643	00:33:19
kroB100_15668	100	15668	386	0.86	37	00:00:05	232	229	10	0	**0**	12	00:00:07
kroC100_15668	100	15668	271	1.5	84	00:00:05	297	330	0	0	1.4	41	00:00:10

Table 3
Branch-and-Cut results (3).

Name	$\|V\|$	$\|D\|$	Cuts-1	Gap-1	Sub-1	CPU-1	Cuts-2	FP	Cliques	O-cycles	Gap-2	Sub-2	CPU-2
lin105_15668	105	15668	78	1.7	15	00:00:04	227	63	1	1	1.6	10	00:00:06
lin318_15668	318	15668	2104	0.31	835	00:01:38	1940	2193	0	0	0.34	20	00:06:16
pr107_15668	107	15668	434	**0**	7	00:00:04	162	17	9	1	0.63	2	00:00:04
pr76_15668	76	15668	355	7.8	346	00:00:48	197	150	33	1	6.2	30	00:00:11
rat195_15668	195	15668	2852	0.87	530	00:00:47	409	2080	0	0	0.52	84	00:01:21
rat99_15668	99	15668	517	2.2	161	00:00:08	200	140	7	0	**0**	28	00:00:06
rd100_15668	100	15668	112	**0**	1	00:00:02	112	1	0	0	**0**	1	00:00:01
rd400_15668	400	15668	**25603**	0.59	**9468**	00:18:54	**1847**	**3563**	0	0	0.28	**211**	01:35:25
st70_15668	70	15668	176	4	22	00:00:05	221	487	25	0	3.2	36	00:00:10
ts225_15668	225	15668	24	1.7	817	00:00:17	159	105	6	0	34	3	00:00:13
u159_15668	159	15668	802	7.4	21	00:00:05	486	299	1	0	5.6	3	00:00:06
u574_15668	574	15668	89430	0.39	19079	01:16:59	9266	8091	0	0	0.43	501	02:00:00
a280_49629	280	49629	1885	0.47	466	00:01:34	1411	1082	0	0	0.29	112	00:02:14
berlin52_49629	52	49629	22335	23	128460	02:00:00	15899	11799	15272	244	23	80867	02:00:00
ch130_49629	130	49629	868	1.4	287	00:00:52	422	472	1	0	1.1	47	00:00:39
eil76_49629	76	49629	588	4	521	00:01:19	722	1142	63	0	3.9	471	00:01:31
kroA100_49629	100	49629	333	3.3	149	00:00:45	400	487	10	0	2.6	89	00:01:07
kroA150_49629	150	49629	608	0.93	428	00:00:54	565	838	4	0	0.32	61	00:01:05
kroA200_49629	200	49629	7692	0.73	3456	00:04:31	2445	2373	0	0	0.48	520	00:07:03
kroB100_49629	100	49629	464	3.2	379	00:01:02	406	1045	15	0	2.9	177	00:01:21
kroB150_49629	150	49629	2238	1	496	00:00:47	713	1085	1	0	0.55	47	00:00:54
kroC100_49629	100	49629	684	2.4	210	00:00:46	353	640	5	0	2.1	111	00:01:02
lin105_49629	105	49629	197	**0**	11	00:00:09	140	51	1	0	**0**	3	00:00:09
lin318_49629	318	49629	2309	0.31	1344	00:03:46	1019	3686	0	0	0.16	85	00:07:59
pr107_49629	107	49629	201	**0**	23	00:00:12	272	24	2	0	0.58	21	00:00:13
pr76_49629	76	49629	747	5.3	1089	00:01:09	277	344	42	0	4.2	249	00:01:02
rat195_49629	195	49629	2463	0.79	1380	00:02:04	616	2936	0	0	0.53	106	00:01:44
rat99_49629	99	49629	671	2.4	179	00:00:24	265	411	8	0	2	78	00:00:33
rd100_49629	100	49629	533	2.1	161	00:00:54	465	1590	4	0	1.8	154	00:01:19
st70_49629	70	49629	2496	10	8273	00:06:53	2405	4318	636	7	10	5327	00:08:42
ts225_49629	225	49629	58	1.4	724	00:00:27	46	38	0	0	**0**	1	00:00:06
u159_49629	159	49629	720	0.67	26	00:00:11	714	107	0	0	0.67	13	00:00:13

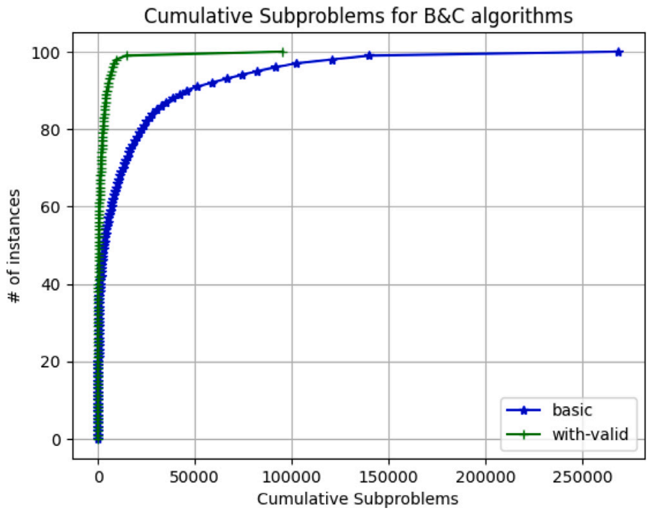

Fig. 5. Number of subproblems in the Branch-and-Cut trees for formulations without and with valid inequalities.

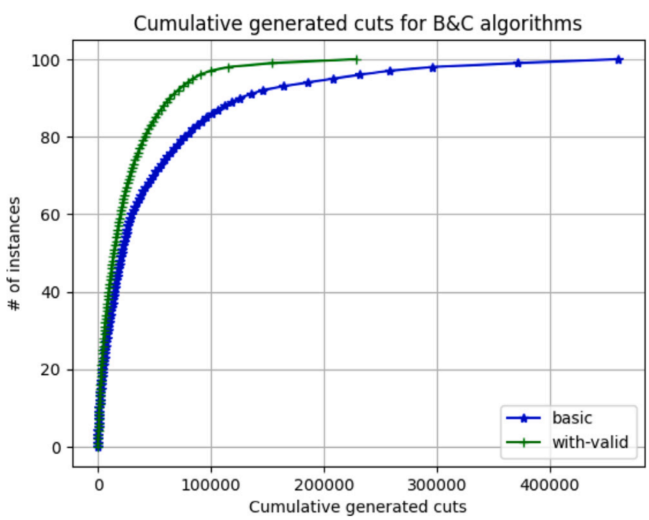

Fig. 6. Number of generated cut inequalities for formulations without and with valid inequalities.

Unfortunately, the execution time is more important when separating valid inequalities. We may here think about improving the complexities of the current separation routines in the future.

Concerning the families of valid inequalities related to the independent-set structure, we note that clique and odd cycle inequalities were more difficult to find, mainly for the random instances where we could not separate any of them (see Table 1). This can be explained by the fact that randomly generated conflicts may not necessarily induce conflict graphs with cliques and odd-cycles.

We also remark that the random instances (Table 1) were easier to solve in this case. First, the number of conflicts is small not exceeding 2000 pairs of conflicts. Moreover, the conflicts are randomly generated, which means that the conflict graph may be somehow sparse. Instances where the conflicts are generated based on the coloring-problem conflicts are clearly more difficult to solve (see Tables 2 and 3). In fact, we are dealing with instances having 12458, 15668 and 49629 conflicts, where it is a bit easier to find cliques and odd cycles. This is due to the fact that the conflict graphs for these instances are dense.

5. Concluding remarks

In this paper, we studied the Disjunctive Two Edge Connected subgraph Problem (DTECP) that is a variant of the Two Edge Connected Subgraph Problem, a well-studied problem in network design. In this variant, some pairs of edges are in conflict, which means that they cannot be considered simultaneously in the final solution. We proposed an ILP formulation for the DTECP, and identified new classes of valid inequalities. We studied the associated polytope and discussed sufficient conditions and necessary conditions for the basic and valid inequalities to be facet-defining. We also discussed the separation routines and devise Branch-and-Cut algorithm for the basic formulation, and the formulation with valid inequalities. Experiments show that the valid inequalities, though difficult to find for some configurations, have shown to be effective. They have helped strengthening the linear relaxation of the basic formulation for the majority of the tested instances, and solving them to optimality.

In the future, we aim at more improving the linear relaxation by adding new classes of valid inequalities, mainly facet-defining ones. In fact, even if the current formulation with valid inequalities is able to solve the majority of the instances, the value of the gap in the root is important for some instances and need to be reduced.

CRediT authorship contribution statement

Intesar Almudahka: Conceptualization, Methodology, Investigation, Writing – original draft. **Ibrahima Diarrassouba:** Conceptualization, Methodology, Software. **A. Ridha Mahjoub:** Supervision, Methodology, Investigation, Writing – original draft, Writing – review & editing. **Raouia Taktak:** Methodology, Software, Writing – original draft, Writing – review & editing.

Data availability

Data will be made available on request.

Appendix. Facet-defining proofs for the clique and odd-cycle inequalities

Theorem 12. *Inequality (9) defines a facet of DTECP(G) if and only if K is maximal in G_c.*

Proof. *Necessity:* If K is not maximal in G_c, then there is a node $f \in V_c \setminus K$ of G_c such that $K' = K \cup \{f\}$ is a clique. Therefore $\sum_{e\in K'} x_e = \sum_{e\in K} x_e + x_f \leq 1$ is valid for DTECP(G). Since this inequality dominates (9), the latter cannot define a facet.

Sufficiency: Let us denote inequality (9) by $ax \leq \alpha$ and let $bx \leq \beta$ be a facet defining inequality of DTECP(G) such that

$$\{x \in \text{DTECP}(G) : ax = \alpha\} \subseteq \{x \in \text{DTECP}(G) : bx = \beta\}.$$

We will show that there is $\rho \in \mathbb{R}$ such that $b = \rho a$.

First, we show that $b(e) = b(e')$ for all $e, e' \in K$. Consider the following sets

$$T_e = (E \setminus D) \cup \{e\} \quad \text{for} \quad e \in K.$$

As $G(E \setminus D)$ is 3-edge connected, clearly, these sets are solutions of DTECP(G). Hence $ax^{T_e} = ax^{T_{e'}} = \alpha$, $b_e - b_{e'} = bx^{T_e} - bx^{T_{e'}} = 0$ for all $e, e' \in K$, which implies that

$$b_e = b_{e'} = \rho \qquad \begin{array}{l}\text{for all } e, e' \in K,\\ \text{for some } \rho \in \mathbb{R}.\end{array} \tag{22}$$

Now we will show that $b_e = 0$ for all $e \in E \setminus K$.

Let $f \in D \setminus K$. As K is maximal, there should exist $e \in K$ such that e and f are not in conflict. Consider the set

$$T_f = T_e \cup \{f\},$$

where T_e is introduced above. Trivially T_f is a solution of DTECP(G) and $ax^{T_f} = \alpha$. Hence $bx^{T_f} = \beta$. As $bx^{T_e} = \alpha$, it follows that $b_f = bx^{T_f} - bx^{T_e} = 0$. Consequently,

$$b_f = 0 \qquad \text{for all } \ f \in D \setminus K. \tag{23}$$

Now we will show that $b_g = 0$ for all $g \in E \setminus D$. Let

$$T_g = T_e \cup \{g\}.$$

As $G(E \setminus D)$ is 3-edge connected, T_g is a solution of DTECP(G).

As $ax^{T_g} = \alpha$, and hence $bx^{T_g} = \beta$, we have $b_g = bx^{T_e} - bx^{T_g} = 0$ yielding

$$b_g = 0 \qquad \text{for all } \ g \in E \setminus D. \tag{24}$$

From (22)–(24), we obtain that

$$\begin{array}{ll} b_e = \rho & \text{for all } \ e \in K. \\ b_e = 0 & \text{for all } \ e \in E \setminus D. \end{array} \tag{25}$$

Therefore, $b = \rho a$, which ends the proof. □

Remark 1. Remark that the disjunctive inequalities are inequalities of type (9). So a disjunctive inequality $x_e + x_f \leq 1$ defines a facet of DTECP(G) if and only if $\{e, f\}$ is not contained in a larger complete graph of G_c.

Theorem 13. *Inequality (10) defines a facet for DTECP(G) if C is chordless, and for every $e \in D \setminus C$, there is a set $C_e \subset C$, $|C_e| = \frac{k-1}{2}$ such that the elements of $C_e \cup \{e\}$ are pairwise not in conflict.*

Proof. If C contains a chord (e_{i_0}, e_{j_0}), $(i_0 < j_0)$, then one of cycles $C_1 = (e_1, \ldots, e_{i_0}, e_{j_0}, \ldots, e_k)$ and $C_2 = (e_{i_0}, \ldots, e_{j_0})$, say C_1, is odd. It is not hard to see that (10) can be obtained as linear combination of the odd cycle inequality induced by C_1 and the disjunctive inequality $x_{e_{i_0+l}} + x_{e_{j_0+l}} \leq 1$, $l = 1, \ldots, j_0 - 2$. Hence inequality (10) cannot define a facet.

Let $ax \leq \alpha$ denote inequality (10) and let $bx \leq \beta$ be a facet defining inequality of DTECP(G) such that

$$\{x \in \text{DTECP}(G) : ax = \alpha\} \subseteq \{x \in \text{DTECP}(G) : bx = \beta\}.$$

We will show that is ρ such that $b = \rho a$.

First we show that $b_e = b_{e'}$ for all $e, e' \in C$.

Consider the edge set

$$T_i = \{e_i, e_{i+2}, \ldots, e_{i+k-3}\} \cup (E \setminus D), \qquad \text{for } i = 1, 2, \ldots, k \,(\text{mod}\, k).$$

Since C is chordless, and hence $e_i, e_{i+2}, \ldots, e_{i+k-3}\,(\text{mod}\, k)$ are pairwise non adjacent in G_c. $e_i, e_{i+2}, \ldots, e_{i+k-3}\,(\text{mod}\, k)$ are pairwise not in conflict. As $E \setminus D$ induces a 3-edge connected graph, the sets T_i, $i = 1, 2, \ldots, k\,(\text{mod}\, k)$ are solutions of DTECP(G).

Moreover as $ax^{T_i} = \alpha$, it follows that $bx^{T_i} = \beta$ for $i = 1, \ldots, k\,(\text{mod}\, k)$.

Let A be the $k \times k$ matrix given by the restrictions of the incidence vectors $x^{T_1}, \ldots, x^{T_k}$ on the edges of C, $e_1, \ldots, e_k$.

As k is odd, A is a cyclic matrix of the following form.

$$A = \begin{pmatrix} 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 1 \\ \vdots & 1 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 1 \\ 1 & \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 1 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & 0 & \ddots & & & & & & \vdots \\ \vdots & \vdots & & \ddots & & & & & \vdots \\ \vdots & \vdots & & & \ddots & & & & \vdots \\ \vdots & \vdots & & & & \ddots & & & \vdots \\ \vdots & \vdots & & & & & \ddots & & \vdots \\ \vdots & \vdots & & & & & & \ddots & 0 \\ 0 & 0 & & & & & & & 1 \end{pmatrix}$$

Moreover, $b_{e_1}, \ldots, b_{e_k}$ is a solution of the system

$$A^T \tilde{b} = \tilde{\beta} = \beta - \sum_{e \in E \setminus D} b_e$$

where $\tilde{b} = (b_{e_1}, \ldots, b_{e_k})$.

As A is nonsingular, the system above has a unique solution given by

$$b_{e_i} = \frac{2\beta'}{k-1} \qquad \text{for } i = 1, \ldots, k.$$

Hence

$$b_e = \rho \qquad \begin{array}{l} \text{for all } \ e \in C, \\ \text{for some } \ \rho \in \mathbb{R}. \end{array} \tag{26}$$

Next we will show that $b_e = 0$ for all $e \in C$.

Let $e \in D \setminus C$. By assumption, there should exist a set $C_e \subset C$ such that the elements of $C_e \setminus \{e\}$ are pairwise not in conflict. Set C_e should be one, say T_1, of the sets $T_1, \ldots, T_k$ introduced above. Let $\tilde{T}_1 = T \cup \{e\}$. As $ax^{\tilde{T}_1} = \alpha$, and hence $bx^{\tilde{T}} = \beta$, it follows that $b_e = 0$. Consequently

$$b_e = 0 \qquad \text{for all } \ e \in D \setminus C. \tag{27}$$

Now, as in the proof of the previous theorem, we can show that $b_e = 0$ for all $e \in E \setminus D$. This, together with (26) and (27) yields

$$\begin{array}{ll} b_e = \rho & \text{for all } \ e \in C. \\ b_e = 0 & \text{for all } \ e \in E \setminus C. \end{array} \tag{28}$$

Therefore, $b = \rho a$ and the proof is complete. □

References

Baïou, Mourad, & Mahjoub, A. Ridha (1997). Steiner 2-edge connected subgraph polytopes on series-parallel graphs. *SIAM Journal of Discrete Mathematics*, *10*(3), 505–514.

Barahona, Francisco, & Mahjoub, A. Ridha (1995). On two-connected subgraph polytopes. *Discrete Mathematics*, *147*(1–3), 19–34.

Ben Salem, Mariem, Hanafi, Saïd, Taktak, Raouia, & Ben-Abdallah, Hanêne (2017). Probabilistic tabu search with multiple neighborhoods for the disjunctively constrained knapsack problem. *RAIRO-Operations Research*, *51*(3), 627–637.

Ben Salem, Mariem, Taktak, Raouia, Mahjoub, A. Ridha, & Ben-Abdallah, Hanêne (2018). Optimization algorithms for the disjunctively constrained knapsack problem. *Soft Computing*, *22*(6), 2025–2043.

Bettinelli, Andrea, Cacchiani, Valentina, & Malaguti, Enrico (2017). A branch-and-bound algorithm for the knapsack problem with conflict graph. *INFORMS Journal on Computing*, *29*(3), 457–473.

Biha, M. Didi, Kerivin, Hervé, & Mahjoub, A. Ridha (2008). On the polytope of the (1, 2)-survivable network design problem. *SIAM Journal of Discrete Mathematics*, *22*(4), 1640–1666.

Biha, M. Didi, & Mahjoub, A. Rhida (1996). K-edge connected polyhedra on series-parallel graphs. *Operations Research Letters*, *19*(2), 71–78.

Cao, Buyang (1992). Transportation problem with nonlinear side constraints a branch and bound approach. *Zeitschrift für Operations Research*, *36*, 185–197.

Carrabs, Francesco, Cerulli, Raffaele, Pentangelo, Rosa, & Raiconi, Andrea (2021). Minimum spanning tree with conflicting edge pairs: a branch-and-cut approach. *Annals of Operations Research*, *298*(1), 65–78.

Carrabs, Francesco, & Gaudioso, Manlio (2021). A Lagrangian approach for the minimum spanning tree problem with conflicting edge pairs. *Networks*, *78*(1), 32–45.

Coniglio, Stefano, Furini, Fabio, & San Segundo, Pablo (2021). A new combinatorial Branch-and-Bound algorithm for the knapsack problem with conflicts. *European Journal of Operational Research*, *289*(2), 435–455.

Dahl, Geir, Huygens, David, Mahjoub, A. Ridha, & Pesneau, Pierre (2006). On the k edge-disjoint 2-hop-constrained paths polytope. *Operations Research Letters*, *34*(5), 577–582.

Darmann, Andreas, Pferschy, Ulrich, Schauer, Joachim, & Woeginger, Gerhard J. (2011). Paths, trees and matchings under disjunctive constraints. *Discrete Applied Mathematics*, *159*(16), 1726–1735.

Diarrassouba, Ibrahima, Gabrel, Virginie, Mahjoub, A. Ridha, Gouveia, Luís, & Pesneau, Pierre (2016). Integer programming formulations for the k-edge-connected 3-hop-constrained network design problem. *Networks*, *67*(2), 148–169.

Diarrassouba, Ibrahima, Kütücu, Hakan, & Mahjoub, A. Ridha (2016). Two node-disjoint hop-constrained survivable network design and polyhedra. *Networks*, *67*(4), 316–337.

Diarrassouba, Ibrahima, Mahjoub, Meriem, Mahjoub, A. Ridha, & Yaman, Hande (2018). K-node-disjoint hop-constrained survivable networks: polyhedral analysis and branch and cut. *Annals of Telecommunications*, *73*(1), 5–28.

Ekici, Ali (2021). Bin packing problem with conflicts and item fragmentation. *Computers & Operations Research*, *126*, Article 105113.

Erickson, Ranel E., Monma, Clyde L., & Veinott Jr., Arthur F. (1987). Send-and-split method for minimum-concave-cost network flows. *Mathematics of Operations Research*, *12*(4), 634–664.

Fonlupt, Jean, & Mahjoub, A. Ridha (2006). Critical extreme points of the 2-edge connected spanning subgraph polytope. *Mathematical Programming*, *105*, 289–310.

Ford, L. R., & Fulkerson, D. R. (1956). Maximal flow through a network. *Canadian Journal of Mathematics*, *8*(3), 399–404.

Fortz, Bernard, & Labbé, Martine (2002). Polyhedral results for two-connected networks with bounded rings. *Mathematical Programming*, *93*(1), 27–54.

Fortz, Bernard, Labbé, Martine, & Maffioli, Francesco (2000). Solving the two-connected network with bounded meshes problem. *Operations Research*, *48*(6), 866–877.

Fortz, Bernard, Mahjoub, A. Ridha, McCormick, S. Thomas, & Pesneau, Pierre (2006). Two-edge connected subgraphs with bounded rings: Polyhedral results and Branch-and-Cut. *Mathematical Programming*, *105*(1), 85–111.

Goldberg, A. V., & Tarjan, R. E. (1988). A new approach to the maximum-flow problem. *Journal of the Association for Computing Machinery*, *35*(4), 921–940.

Gomory, R. E., & Hu, T. C. (1961). Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, *9*(4), 551–570.

Grötschel, Martin, Lovász, László, & Schrijver, Alexander (2012). *Geometric algorithms and combinatorial optimization*. Springer Science & Business Media.

Grötschel, Martin, & Monma, Clyde L. (1989). Integer polyhedra associated with certain network design problems with connectivity constraints. *SIAM Journal on Discrete Mathematics*.

Grötschel, Martin, & Monma, Clyde L. (1990). Integer polyhedra arising from certain network design problems with connectivity constraints. *SIAM Journal of Discrete Mathematics*, *3*(4), 502–523.

Grötschel, Martin, Monma, Clyde L., & Stoer, Mechthild (1992a). Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints. *Operations Research*, *40*(2), 309–330.

Grötschel, Martin, Monma, Clyde L., & Stoer, Mechthild (1992b). Facets for polyhedra arising in the design of communication networks with low-connectivity constraints. *SIAM Journal on Optimization*, *2*(3), 474–504.

Grötschel, Martin, Monma, Clyde L., & Stoer, Mechthild (1995). Design of survivable networks. *Vol. 7*, In *Handbooks in operations research and management science* (pp. 617–672). Elsevier.

Gusfield, D. (1987). Very simple algorithms and programs for all pairs network flow analysis. *Technical report*, Computer Science Division, University of California, Davis.

Gusfield, D. (1990). Very simple methods for all pairs network flow analysis. *SIAM Journal of Computing*, *19*(1), 143–155.

Huygens, David, Labbé, Martine, Mahjoub, A. Ridha, & Pesneau, Pierre (2007). The two-edge connected hop-constrained network design problem: valid inequalities and branch-and-cut. *Networks: An International Journal*, *49*(1), 116–133.

Huygens, David, & Mahjoub, A. Ridha (2007). Integer programming formulations for the two 4-hop-constrained paths problem. *Networks: An International Journal*, *49*(2), 135–144.

Huygens, David, Mahjoub, A. Ridha, & Pesneau, Pierre (2004). Two edge-disjoint hop-constrained paths and polyhedra. *SIAM Journal of Discrete Mathematics*, *18*(2), 287–312.

Kerivin, Hervé, & Mahjoub, A. Ridha (2002). Separation of partition inequalities for the (1, 2)-survivable network design problem. *Operations Research Letters*, *30*(4), 265–268.

Kerivin, Hervé, & Mahjoub, A. Ridha (2005). Design of survivable networks: A survey. *Networks: An International Journal*, *46*(1), 1–21.

Kerivin, Hervé, Mahjoub, A. Ridha, & Nocq, Charles (2004). *(1, 2)-survivable networks: facets and Branch-and-Cut* (pp. 121–152). SIAM.

Luiz, Thiago Alcântara, Santos, Haroldo Gambini, & Uchoa, Eduardo (2021). Cover by disjoint cliques cuts for the knapsack problem with conflicting items. *Operations Research Letters*, *49*(6), 844–850.

Mahjoub, A. Ridha (1994). Two-edge connected spanning subgraphs and polyhedra. *Mathematical Programming*, *64*(1), 199–208.

Mahjoub, A. Ridha (1997). On perfectly two-edge connected graphs. *Discrete Mathematics*, *170*(1–3), 153–172.

Mahjoub, A. Ridha (2010). Polyhedral approaches. In Vangelis Paschos (Ed.), *Concepts of combinatorial optimization* (pp. 261–324). ISTE-WIELY.

Mahjoub, Meriem (2017). The survivable network design problems with high node-connectivity constraints: Polyhedra and algorithms (Ph.D. thesis), Université Paris-Dauphine PSL and Université de Tunis El Manar.

Mahjoub, A. Ridha, & Pesneau, Pierre (2008). On the steiner 2-edge connected subgraph polytope. *RAIRO-Operations Research*, *42*(3), 259–283.

Mahjoub, A. Ridha, Poss, Michael, Simonetti, Luidi, & Uchoa, Eduardo (2019). Distance transformation for network design problems. *SIAM Journal on Optimization*, *29*(2), 1687–1713.

Mahjoub, A. Ridha, Simonetti, Luidi, & Uchoa, Eduardo (2013). Hop-level flow formulation for the survivable network design with hop constraints problem. *Networks*, *61*(2), 171–179.

Menger, Karl (1927). Zur allgemeinen kurventheorie. *Fundamenta Mathematicae*, *10*, 96–1159.

Monma, Clyde L., Munson, Beth Spellman, & Pulleyblank, William R. (1990). Minimum-weight two-connected spanning networks. *Mathematical Programming*, *46*(1), 153–171.

Nemhauser, George L., & Sigismondi, Gabriele (1992). A strong cutting plane/branch-and-bound algorithm for node packing. *Journal of the Operational Research Society*, *43*(5), 443–457.

Öncan, Temel, Şuvak, Zeynep, Akyüz, M. Hakan, & Altınel, İ Kuban (2019). Assignment problem with conflicts. *Computers & Operations Research*, *111*, 214–229.

Öncan, Temel, Zhang, Ruonan, & Punnen, Abraham P. (2013). The minimum cost perfect matching problem with conflict pair constraints. *Computers & Operations Research*, *40*(4), 920–930.

Pferschy, Ulrich, & Schauer, Joachim (2009). The knapsack problem with conflict graphs. *Journal of Graph Algorithms and Applications*, *13*(2), 233–249.

Pferschy, Ulrich, & Schauer, Joachim (2013). The maximum flow problem with disjunctive constraints. *Journal of Combinatorial Optimization*, *26*(1), 109–119.

Sadykov, Ruslan, & Vanderbeck, François (2013). Bin packing with conflicts: a generic Branch-and-Price algorithm. *INFORMS Journal on Computing*, *25*(2), 244–255.

Samer, Phillippe, & Urrutia, Sebastián (2015). A branch and cut algorithm for minimum spanning trees under conflict constraints. *Optimization Letters*, *9*(1), 41–55.

Senisuka, Aminto, You, Byungjun, & Yamada, Takeo (2005). Reduction and exact algorithms for the disjunctively constrained knapsack problem. In *International symposium, operational research bremen*. Citeseer.

Stoer, Mechthild (1992). *Design of survivable networks, lecture notes in mathematics*. Springer-Verlag.

Şuvak, Zeynep, Altınel, İ Kuban, & Aras, Necati (2020). Exact solution algorithms for the maximum flow problem with additional conflict constraints. *European Journal of Operational Research*, *287*(2), 410–437.

Şuvak, Zeynep, Altınel, İ Kuban, & Aras, Necati (2021). Minimum cost flow problem with conflicts. *Networks*, *78*(4), 421–442.

Winter, Pawel (1986). Generalized steiner problem in series-parallel networks. *Journal of Algorithms*, *7*(4), 549–566.

Winter, Pawel (1987). Steiner problem in Halin networks. *Discrete Applied Mathematics*, *17*(3), 281–294.

Yamada, Takeo, Kataoka, Seija, & Watanabe, Kohtaro (2002). Heuristic and exact algorithms for the disjunctively constrained knapsack problem. *Information Processing Society of Japan Journal*, *43*(9).