Max-multiflow/min-multicut for $G + H$ series-parallel

Denis Cornaz

LAMSADE, Université Paris-Dauphine, Place du Maréchal de Lattre de Tassigny, 75775 Paris CEDEX 16, France

Abstract

We give a new characterization of series-parallel graphs which implies that the maximum integer multiflow is equal to the minimum capacity multicut if $G + H$ is series-parallel, where $G + H$ denotes the union of the support graph $G$ and the demand graph $H$.

We investigate the difference between a result of the type "the cut-condition is sufficient for the existence of a multiflow in some class" and a result of the type "max-multiflow = min-multicut for some class".

1. Introduction

Given an undirected graph $G = (V, E)$ with positive integer edge capacity $u \in \mathbb{Z}_+^E$ and a list $s_1t_1, \ldots, s_kt_k$ of pairs of distinct vertices, the maximum integer multiflow problem consists in maximizing the total amount of integer flow between any pair of vertices of the list subject to capacity constraints. Let $P = \{ p_1, \ldots, p_{|P|} \}$ be the set of $s_jt_j$-paths of $G$ and denote the union $\bigcup_{j=1}^k P_j$ of them by $P := \{ p_1, \ldots, p_{|P|} \}$, then it can be formulated as follows:

$$\max \sum_{i=1}^{|P|} f_i$$

$$\sum_{i \text{ s.t. path } p_i \ni e} f_i \leq u_e \text{ for all } e \in E,$$  

$$f_i \geq 0 \text{ for } i = 1, \ldots, |P|,$$  

$$f_i \in \mathbb{Z} \text{ for } i = 1, \ldots, |P|.$$  

A multicut separating each pair $s_jt_j$ is a subset of edges the removing of which destroy any path linking the vertices $s_j$ and $t_j$ for $j = 1, \ldots, k$. The capacity of a multicut $D$ is the sum $u(D) := \sum_{e \in D} u_e$ of the capacities of its edges. The minimum multicut problem is to find a minimum capacity multicut separating each pair $s_jt_j$. It can be formulated as:

$$\min \sum_{e \in E} u_e c_e$$

$$\sum_{e \text{ s.t. } e \text{ path } p_i} c_e \geq 1 \text{ for } i = 1, \ldots, |P|.$$
By the duality theorem, the linear program (1)–(3) has the same optimum as the linear program (5)–(7) and by the max-flow/min-cut theorem, the integer programs (1)–(8) have the same optimum when there is only one pair in the list (that is $k = 1$) [12]. But in general both problems are NP-hard, even if $G$ is a (undirected) tree [14]. The graph $H = (V, R)$ where $R$ is the set of the pairs $s_t$ is called the demand graph and then the graph $G$ is called the support graph. If $G$ is inner Eulerian and if $H$ is bi-stable, then max-multiflow = min-multicut, that is (1)–(8) are equal, see [13]. (In particular, the complete graph and $2K_2$ are bi-stable.) Bentz et al. [5] proved that if $G$ is a rectilinear grid with uniform capacities, then, except for some very special cases, max-multiflow = min-multicut. Let $G + H = (V, E \cup R)$ denote the union of $G$ and $H$. (Notice that $G + H$ may have parallel edges so $E \cap R = \emptyset$.) The maximum integer multiflow problem remains NP-hard if $G + H$ is planar [18]. The minimum multicut problem is tractable if $G + H$ has bounded tree width [15], in particular, if $G + H$ is series–parallel, that is, it can be constructed by starting with a forest, adding loops, and repeatedly replacing edges by parallel edges or by edges in series (series–parallel graphs are graphs with tree-width 2). Lomonosov [17] proved that if $|R| = 2$ and if $G + H$ is planar and not Z-special, then max-multiflow = min-multicut. For a survey, see [21,7,4].

The integer multiflow feasibility problem is, given an additional list $d_1, \ldots, d_k$ of positive integers, to decide if an integer multiflow exists, that is, if the system (2)–(4) together with (9) is empty or not.

\[
\sum_{i \text{ s.t. path } p_i \in \mathcal{P}} f_i \geq d_j \quad \text{ for } j = 1, \ldots, k.
\]

(9)

(Note that it can be formulated as a maximum integer multiflow problem by adding some edges with capacity $u_i = d_i$). The integer multiflow feasibility problem is NP-complete even if $G$ is series–parallel [20]. It is tractable if $G + H$ has bounded tree-width [28]. Sebő [22] showed that if $G + H$ is planar and if $|R|$ is bounded, then the problem is solvable in polynomial time. A cut of a graph $G$ is the subset $\delta(V_1, \ldots, V_k)$ of all edges between distinct vertex-subsets of some partition $V_1, \ldots, V_k$ of its vertex-set, and if the partition has at most two vertex-subsets, the multicut $\delta(V_1, \ldots, V_k)$ is also called a cut denoted by $\delta(V_i)$. A necessary condition for the existence of a multiflow is the so-called cut-condition that, for any cut $D$ of $G + H$, the sum of the capacities $u_i$ over all $e \in D \setminus R$ is greater than or equal to the sum of the $d_j$ over all $s_t \in D \cap R$. Checking the cut-condition is NP-complete [1]. There are several necessary conditions for the existence of a multiflow and several characterizations for some of these conditions to be sufficient. The cut-condition is sufficient for the existence of an integer multiflow if $G + H$ has no odd--$K_4$ minor where $R$ is the set of the signed edges [23], or for the larger class of $G + H$ without odd-$K_5$ minor, which contains that of $G + H$ planar, if moreover the Euler condition holds [25], see also [21, p. 1342]. For a survey, see [21, p. 1234] and [19].

Given integer weight $w \in \mathbb{Z}^E$ on the edges of a graph $G$, where a weight may be negative, the weight of a cut or multicut $D$ is the sum $w(D) := \sum_{e \in D} w_e$ of the weights of its edges. The maximum cut problem in $(G, w)$ is to find the maximum weight of a cut of $G$. The problem is NP-hard even if $G$ has no minor $K_5$ and polynomial if it has no minor $K_5$ [2], furthermore a linear description of the cut polytope is known for these graphs [3]. The maximum multicut problem in $(G, w)$ is to find the maximum weight of a multicut of $G$. The problem is NP-hard in general and a linear description of the multicut polytope is known if $G$ is series–parallel (or equivalently $G$ has no minor $K_4$) [6]. So there is a (non combinatorial) polynomial algorithm for these graphs. Seymour established a min–max formula for the chromatic index of series-parallel graphs [26]. Facets of the multicut polytope are given in [16,27]. See also [21,8].

This paper is motivated by finding a class of (planar) $G + H$ for which max-multiflow = min-multicut, and by the following question: Is it possible to transform a result of the type “the cut-condition is sufficient for the existence of a multiflow in some class” into a result of the type “max-multiflow = min-multicut for some class?”

In this paper we introduce the minimum multiflow loss problem which is, given a weighted graph $(G, w)$ with $w \in \mathbb{Z}^E$, to find the minimum amount of demand whose removal ensures the existence of an integer multiflow. It generalizes the maximum integer multiflow problem since $w$ can be chosen so as to represent infinite amounts of demand with capacity constraints; minimizing the demand loss is then equivalent to maximizing the multiflow. (A formal definition is given later and actually both problems are equivalent). We prove that it has the same optimum than the maximum multicut problem, for any weight $w \in \mathbb{Z}^E$, if and only if $G$ is series–parallel. It implies that max-multiflow = min-multicut holds for $G + H$ series–parallel.

The paper is organized as follows. In Section 2, we give definitions and basic results needed in the paper. In Section 3, we prove the characterization of series–parallel graphs involving the minimum multiflow loss and the maximum multicut problems. Actually, we also show that it implies a similar characterization involving the minimum cut-condition problem (defined later) which is an optimization version of checking the cut-condition. In Section 4, we derive consequences concerning a new characterization of series–parallel graphs and TDIness (defined in Section 4). In Section 5, we derive consequences concerning a new characterization of series–parallel graphs and packing and covering. We also show some difficulties for translating a result of the type “the cut-condition is sufficient for the existence of a multiflow in some class” into a relation of the type “max-multiflow = min-multicut for this class.”
2. Preliminaries

2.1. Basics

First we recall that a nonempty inclusionwise minimal cut is called a bond and a circuit is a nonempty subset \{e_1, e_2, \ldots, e_k\} of edges such that there exist \(k\) distinct vertices \(v_1, v_2, \ldots, v_k\) and \(e_i = v_i v_{i+1}\) (with \(v_{k+1} = v_1\)).

Both sets of circuits and of bonds of \(G\) satisfy the (matroid) exchange property, that is, if two circuits (bonds, respectively) intersects properly, one can find a third circuit (bond, respectively) excluding an edge of the intersection and including an edge outside the intersection. Formally:

\[ \forall C_1, C_2 \in \mathcal{C}, \forall e \in C_1 \cap C_2, \forall f \in C_1 \setminus C_2, \text{ then } \exists C \in \mathcal{C} : f \in C \subseteq (C_1 \cup C_2) \setminus \{e\}, \]

where \(\mathcal{C}\) either denotes the set of all circuits or the set of all bonds of \(G\).

Let \(\mathcal{C}\) denote the set of all circuits, and \(\mathcal{D}\) denotes the set of all bonds of \(G\). An edge \(e\) is called a loop (bridge, resp.) if \(e\) is a circuit (bond, resp.). Two edges \(e, f\) are said to be parallel (in series, resp.) if \(\{e, f\}\) is a circuit (bond, resp.). So two edges are parallel if they share the same distinct endpoints. A particular case of edges in series arises with two parallel edges belong to the same bond. Letting an edge \(e\) means replacing \(E\) with \(E \setminus \{e\}\) and contracting \(e\) means to delete \(e\) and to identify its vertices. Deleting \(e\) replaces \(C\) with \(\{ C \in \mathcal{C} : e \not\in C \}\) and contracting \(e\) replaces \(C\) with \(\{ C \in \mathcal{C} : e \not\in C \} \cup \{ \emptyset \neq P \subseteq E : P \cup \{e\} \in \mathcal{C} \}\). On the contrary, letting \(e\) replaces \(D\) with \(\{ D \in \mathcal{D} : e \not\in D \} \cup \{ \emptyset \neq P \subseteq E : P \cup \{e\} \in \mathcal{D} \}\) and contracting \(e\) replaces \(D\) with \(\{ D \in \mathcal{D} : e \not\in D \}\). A minor of a graph \(G\) is a graph obtained from \(G\) by a series of deletions and contractions.

Let \(K_n\) be the complete graph on \(n\) vertices. Recall that \([9]\)

\[ G \text{ is series–parallel if and only if it has no minor } K_4. \]

It follows that series–parallelism is closed under taking minor. Recall also that \([9]\)

\[ \text{if } G \text{ is series–parallel, then it has either a circuit or a bond of size } \leq 2. \]

2.2. Notation and preliminary results

Let \(G = (V, E)\) be an undirected graph and \(w \in \mathbb{Z}^E\) an integer weight vector. The optimum of the maximum multicut problem is denoted by max-multicut\((G, w)\). We define a multiflow problem in such a way that the input is also the weighted graph \((G, w)\):

- Let \(R \subseteq E\) be the set of all edges of \(G\) with positive weight, then an edge \(e \in R\) is called a demand of \((G, w)\) and \(w_e\) is the amount of demand of \(e \in R\);
- An edge \(f\) with negative weight is called a link of \((G, w)\) and \(|w_f|\) is the capacity of \(f \in E \setminus R\);
- A flow in \((G, w)\) is a circuit \(C \subseteq E\) with \(|C \cap R| = 1\);
- Denoting by \(\mathcal{F}\) the set of all flows of \((G, w)\), a multiflow in \((G, w)\) is a nonnegative integer vector \(y \in \mathbb{Z}^\mathcal{F}\) satisfying \((13)-(14)):

\[
\sum_{C \in \mathcal{F} : C \ni e} y_C \geq w_e \quad \text{for } e \in R, \tag{13}
\]

\[
\sum_{C \in \mathcal{F} : C \ni f} y_C \leq -w_f \quad \text{for } f \in E \setminus R. \tag{14}
\]

By removing one unit of demand we mean reset \(w_e := w_e - 1\) for some \(e \in R\) and eventually delete \(e\) if then \(w_e = 0\). The minimum multflow-loss problem in \((G, w)\) is to find the minimum number of units of demand the removal of which makes a multiflow exist; denote it by min-mflowloss\((G, w)\). It can be formulated as minimizing \(\sum_{e \in R} l_e\) over all nonnegative integer vectors \(l \in \mathbb{Z}^R\) satisfying \((14)-(15)) for some nonnegative integer vector \(y \in \mathbb{Z}^\mathcal{F}:

\[
\sum_{C \in \mathcal{F} : C \ni e} y_C \geq w_e - l_e \quad \text{for } e \in R. \tag{15}
\]

Since demands have positive weights and links have negative weights, the cut-condition in \((G, w)\) is that no cut of \(G\) has a positive weight. One associates a minimization problem with the cut-condition, namely the minimum cut-condition problem in \((G, w)\) which is to find the minimum number of units of demands the removal of which satisfies the cut-condition. We write min-cutcond\((G, w)\) for the optimum of this problem.

Recall that:

\[ |C \cap D| \text{ is even for any circuit } C \text{ and any bond } D, \text{ and} \]

\[ D \text{ is a multicut if and only if } |C \cap D| \neq 1 \text{ for any circuit } C. \]
Necessity in (17) follows directly from (16). To see sufficiency in (17), remove the edges in $D$ and assume for contradiction that $D$ is not the set of all edges of $G$ between the different subsets of the partition of $V$ induced by the connected components of the graph $G \setminus D$ that we have obtained. So, there is an edge $e \in D$ and a path $P$ of $G \setminus D$ linking the vertices of $e$. Yet $C = P \cup \{e\}$, is a circuit of $G$ such that $|C \cap D| = 1$.

Notice that:

$$\text{If we find an edge with weight 0, we delete it;}$$

$$\text{If we find a loop, we delete it;}$$

$$\text{If we find an edge with weight 0, we delete it;}$$

$$\text{If we find a bridge, we delete it; we do } \beta \leftarrow \beta + \max\{0, w_e\};$$

$$\text{We contract it; (by } -\infty \text{ we mean less than the negative sum of the positive weights of the current graph).}$$

$$\text{We return the reduced graph } \tilde{G} \text{ with its new weights and the new (increased) value of } \beta.\]
For any input \((G, w)\), we define \((\tilde{G}, \tilde{w}, \beta)\) as follows:

- \((C_1)\) If we find two parallel edges \(e, f\), we delete \(f\) and we do \(w_e \leftarrow w_e + w_f\); 
- \((C_2)\) If we find two edges \(e, f\) in series, we do:
  - \((C_{2a})\) If \(w_e, w_f > 0\), we delete \(e\) and \(f\), and we do \(\beta \leftarrow \beta + w_e + w_f\);
  - \((C_{2b})\) If \(w_e, w_f < 0\), we contract \(f\) and we do \(w_e \leftarrow \min\{w_e, w_f\}\);
  - \((C_{2c})\) If \(w_e > 0\) and \(w_f < 0\), we contract \(f\) and we do \(\beta \leftarrow \beta + \max\{0, w_e + w_f\}\), and we do \(w_e \leftarrow \min\{w_e, |w_f|\}\).

Assuming that the reduction algorithm is valid (which is ensured by Propositions 3.2 and 3.3), the proof is easy:

**Proof of Theorem 3.1.** (i) \(\Rightarrow\) (ii): It follows directly from (21).

(ii) \(\Rightarrow\) (iii): If \(G\) is not series-parallel, by (11) we can assume that the reduced graph \(\tilde{G}\) is a \(K_4\). Indeed, we only have to give the weight 0 for each edge that needs to be deleted, the weight \(-\infty\) for each edge that needs to be contracted, and any different weight for the six remaining edges. Thus, we can assume that we are in the case of Example 1.

(iii) \(\Rightarrow\) (i): If \(G\) is series-parallel, it follows by (12) that the reduced graph \(\tilde{G}\) has no edge. Consequently, max-multicut \((\tilde{G}, \tilde{w}) = 0 = \min\text{-mflowloss}(\tilde{G}, \tilde{w})\); and then max-multicut \((G, w) = \beta = \min\text{-mflowloss}(G, w)\). \(\square\)

We only need now to check the validity of the algorithm. For simplicity in the proofs below, we assume that the reduction operations are done in the order we presented them.

**Proposition 3.2.** For any input \((G, w)\) of the reduction algorithm and its output \((\tilde{G}, \tilde{w}, \beta)\), then (22) holds.

**Proof.** Initially (22) is true, since \(\tilde{G} = G\) and \(\beta = 0\). We show that (22) is closed under any reduction, that is, that (22) is true when \((\tilde{G}, \beta)\) is obtained by one reduction operation of some edge \(e\), where the operation and the edge \(e\) are chosen arbitrarily.

If \(w_e = 0\) for the edge \(e\), then \(e\) is a link without capacity, and then we can remove from \(F\) the circuits containing \(e\), that is, we can delete \(e\). If \(w_e = -\infty\) the capacity is infinite on \(e\), and then we can replace each circuit \(C\) containing \(e\) by \(C\setminus\{e\}\), we can contract \(e\). Hence (22) is closed under \((A_1)\) and \((A_2)\). We assume in the following that no edge has weight 0. A loop \(e\) is either a useless link, or a demand that can always be routed on the circuit \(\{e\}\); hence \(e\) creates no multiflow loss and we can delete it, that is doing \((B_1)\). Since no circuit contains a bridge, a bridge is either a useless link, or a lost demand; we can do \((B_2)\). Let \(e\) and \(f\) be two parallel edges, thus if \(C\) is a circuit containing \(f\) and not \(e\), then by (10) \(C\setminus\{f\}\) is a circuit. If \(e\) and \(f\) have the same sign, \((C_{11})\) preserves (22). Indeed, \(\{e, f\} \notin F\), then removing the circuits containing \(f\) preserves (22) if the demand (or the capacity) of \(e\) becomes the sum of the demands (or of the capacities) of \(e\) and \(f\). If \(e\) and \(f\) have different signs, say \(w_f < 0\), then \(\{e, f\} \notin F\). Since the flows play a symmetric role in the objective function, by (10) we can always assume that the maximum possible amount of the demand \(e\) is routed on the circuit \(\{e, f\}\). Thus if \(f\) is saturated by flow, we can delete \(e\) and decrease the demand of \(e\) by \(|w_f|\). Otherwise, the demand of \(e\) is satisfied, and then we can delete \(e\) and decrease the capacity of \(f\) by \(|w_e|\). Hence (22) is closed under \((C_1)\).

Now we let \(e\) and \(f\) be two edges in series. We have that \((C_{2a})\) and \((C_{2b})\) preserve (22). Indeed, if \(w_e, w_f > 0\), then both demands \(e\) and \(f\) are lost, and if \(w_e, w_f < 0\), we can replace every circuit \(C\) containing \(e\) and \(f\) by \(C\setminus\{f\}\) if the capacity of \(e\) becomes \(\min\{|w_e|, |w_f|\}\). Assume now that \(w_e > 0\) and \(w_f < 0\). If \(w_e < |w_f|\), then we can do \((C_{2c})\). Indeed we have \(w_e + w_f \leq 0\) and furthermore we can contract \(f\) since \(f\) will never be saturated. Finally (22) is closed under \((C_{2c})\) since if \(w_e \geq |w_f|\), the maximum amount of the demand that can be achieved is \(|w_f|\). Hence we must accept losing the amount \(w_e - |w_f| = w_e + w_f\) of demand, and then we can contract \(f\). \(\square\)

**Proposition 3.3.** For any input \((G, w)\) of the reduction algorithm and its output \((\tilde{G}, \tilde{w}, \beta)\), then (23) holds.

**Proof.** It is easily seen that \((A_1)\), \((A_2)\) and \((B_1)\) preserve (23). We can suppose now that no edge of \(G\) has weight 0, and we show that (23) still holds after any other reduction operation of an edge \(e\). Let \(e\) be a bridge. If \(w_e > 0\), then \(e\) belongs to all maximum multcuts of \(G\), and if \(w_e \leq 0\), there are maximum multcuts without \(e\); it follows that \((B_2)\) preserves (23). Obviously (23) is closed under \((C_1)\).

Now we can suppose that \(e\) and \(f\) are two edges in series, that is \(\{e, f\}\) is a bond. By (18) if \(w_e\) and \(w_f\) are positive, then \(e\) and \(f\) belong to every maximum multcut. Hence we can delete \(e\) and \(f\) if we increase \(\beta\) by \(w_e + w_f\); then (23) is closed.
under \((C_{2a})\). We can suppose now that \(w_f < 0\). Note that since \([e,f]\) is a bond, by (10), if \(D\) is a bond containing \(f\) and not \(e\) then \(D \setminus \{f\} \cup \{e\}\) is a bond. Consequently, if \(w_e \geq w_f\) and if the weight of the bond \([e,f]\) is not positive, then we can remove all bonds \(D\) containing \(f\). Hence doing \((C_{2a})\) preserves (23). Moreover, providing that \(w_e + w_f \leq 0\), we can do \((C_{2c})\) (indeed since we have \(w_e \leq |w_f|\)). Finally we can assume that \(w_e > 0\), \(w_f < 0\), and \(w_e + w_f > 0\) (and so \(|w_f| < w_e\)). By (18), \(e\) belongs to every maximum multicut. The weight contribution of both edges \(e\) and \(f\) to a maximum multicut is either \(w_e\) or (at least) \(w_e - |w_f|\); furthermore it is \(w_e\) if and only if there is a maximum multicut containing \(e\) and not \(f\). Hence by (18) we can remove all bonds \(D\) containing \(f\) if we increase \(\beta\) by \(w_e - |w_f|\) and if we the weight of \(e\) becomes \(|w_f|\). Finally, the property is closed under \((C_{2c})\). \(\Box\)

The rest of the paper is devoted to the corollaries of the theorem. But first we give the one concerning the max-multiflow/min-multicut equality:

**Corollary 3.4.** If the union \(G + H\) of the support graph \(G = (V, E)\) and of the demand graph \(H = (V, R)\) is series–parallel, then the maximum integer multiflow problem and the minimum multicut problem have the same optimum, for any capacity \(u \in \mathbb{Z}_+^E\).

**Proof.** Given a support graph \(G = (V, E)\) with positive integer edge capacity \(u \in \mathbb{Z}_+^E\) and a demand graph \(H = (V, R)\), we define \(M := |R| \times \sum_{e \in E} u_e\), so obviously \(M\) is an upper bound for the maximum multiflow problem. Let \(w \in \mathbb{Z}^{E \times R}\) be the vector defined as follows:

\[
w_e := \begin{cases} 
\sum_{e \in R} u_e & \text{if } e \in R, \\
- u_e & \text{if } e \in E.
\end{cases}
\]

Thus \(\text{min-}\text{mflowloss}(G + H, w)\) is equal to \(M\) minus the optimum of the maximum integer multiflow problem. Obviously, any cut in \((G + H, w)\) containing at least one edge in \(R\) has a nonnegative weight. Since \(R\) has no loop each edge of \(R\) belongs to a cut, then by (18), a maximum multicut of \((G + H, w)\) contains \(R\). Then \(\text{max-multicut}(G + H, w)\) is equal to \(M\) minus the optimum of the maximum multicut problem. The result follows from the equality \(\text{min-}\text{mflowloss}(G + H, w) = \text{max-multicut}(G + H, w)\). \(\Box\)

Given a min–max theorem, an interesting question is to associate it with a system of linear inequalities having both primal and dual integer optimal solutions. This is the goal of the next section with the min–max relations of **Theorem 3.1**.

4. TDI

The dual of \(z_{LP} := \max c^T x\) over \(\{Ax \leq b, x \geq 0\}\) is \(\psi_{LP} := \min y^T b\) over \(\{y^T A \geq c^T, y \geq 0\}\) and the linear system \(\{Ax \leq b, x \geq 0\}\) is said to be totally dual integral – TDI for short – if for any integer objective function \(c\) such that \(\psi_{LP}\) admits a feasible solution then \(\psi_{LP}\) has an integer optimal solution (where \(A \in \mathbb{R}^{m \times n}, x \in \mathbb{R}^n\) and \(y, b \in \mathbb{R}^m\)).

Often, there is a min–max relation associated with a TDI system and vice-versa. Schrijver [21, page 505] pointed out that \(G\) is series–parallel if and only if the linear system

\[
\begin{align*}
& x_e \geq 0 \quad \text{for each } e \in E, \quad (24) \\
& x_e \leq \sum_{f \in D \setminus \{e\}} x_f \quad \text{for each cut } D \text{ of } G \text{ and each edge } e \in D, \quad (25)
\end{align*}
\]

is TDI. But no min–max relation is associated with (24)–(25) since the objective function of the dual is the zero function. In this section, we give a TDI system associated with the min–max relation (1) of the theorem.

TDI implies integrality [10] but of course the converse is generally false, especially for the system (24)–(25) which is integral for any graph \(G\) (and describes its circuit cone) [24]. Chopra [6] showed that \(G\) is series–parallel if and only if

\[
\begin{align*}
& 0 \leq x_e \leq 1 \quad \text{for each } e \in E, \quad (26) \\
& x_e \leq \sum_{f \in C \setminus \{e\}} x_f \quad \text{for each circuit } C \text{ of } G \text{ and } e \in C \quad (27)
\end{align*}
\]

is integral, yielding the description of the multicut polytope of these graphs. Removing the upper-bound constraints, the system obtained, that is (24) and (27), is integral for the larger class of graphs \(G\) with no minor \(K_5\) (and describes the cut cone of \(G\)) [25]. In the same way, adding the upper bounds constraints to (24)–(25), the system obtained, that is (25)–(26), is not integral anymore, as expected from the NP-hardness of the maximum bridgeless subgraph problem [11].

**Theorem 3.1** has the following corollary:

**Corollary 4.1.** The system (26)–(27) is TDI if and only if \(G\) is series–parallel.
Proof. Necessity follows from [6], hence we only need to show sufficiency. Let \( G \) be a series-parallel graph with edge weight \( w \in \mathbb{Z}_+^E \). Recall that \( R \) is the set of all edges with positive weight and that \( C \) is the set of circuits of \( G \). The dual of \( z_{IP} := \max \sum_{e \in E} w_e x_e \) over (26)–(27) is \( \psi_{IP} := \min \sum_{e \in E} l_e \) over all nonnegative vectors \( l \in \mathbb{R}^E \) such that

\[
\sum_{C \in C \cap \{ e \}} (y_C^e - \sum_{f \in C \setminus \{ e \}} y_C^f) \geq u_e - l_e \quad \text{for each } e \in E.
\]

for some nonnegative vector \( y \in \mathbb{R}^{E \times E} \). Since \( x_e = 0 \) and \( y_C^e = 0 \), \( y_C^f = 0 \) yield feasible solutions for both problems, then by linear programming duality, \( z_{IP} = \psi_{IP} \). Since (14)–(15) admits a trivial feasible solution \( l_e = w_e \) and \( y_C = 0 \), there exists \( l \in \mathbb{Z}_+^E \) a nonnegative integer vector satisfying (14)–(15) with \( \sum_{e \in E} l_e \) minimum. Let \( \hat{y} \) be the corresponding vector in \( \mathbb{Z}_+^E \).

One obtains a feasible integer solution \( (\hat{l}, \hat{y}) \) for \( \psi_{IP} \) as follows:

\[
\hat{l}_e := \begin{cases} l_e & \text{for } e \in R, \\ 0 & \text{otherwise}, \end{cases} \quad \hat{y}_C^e := \begin{cases} \bar{y}_C^e & \text{for } C \in \mathcal{F} \text{ and } e \in C \cap R, \\ 0 & \text{otherwise}. \end{cases}
\]

Since the solution \( (\hat{l}, \hat{y}) \) has the same value as \( (\hat{l}, \hat{y}) \), the value of \( (\hat{l}, \hat{y}) \) equals min-mflowloss\((G, w)\). Moreover, since the incidence vectors of the multicut of \( G \) are feasible solutions for \( z_{IP} \), then max-multicut\((G, w)\) \( \leq z_{IP} = \psi_{IP} \). Hence Theorem 3.1 implies that \( (\hat{l}, \hat{y}) \) is optimal. \( \square \)

The proof of Corollary 4.1 is that the equality max-multicut\((G, w)\) = min-mflowloss\((G, w)\) for any \( w \in \mathbb{Z}_+^E \) implies the TDiiness of (26)–(27). Besides the converse holds since both happen exactly when \( G \) is series-parallel. In the following we point out that, in a similar way, the equality min-cutcond\((G, w)\) = min-mflowloss\((G, w)\) for any \( w \in \mathbb{Z}_+^E \) implies the TDiiness of the system defined by (24) and (27). Again in both properties are equivalent, since both are equivalent to series-parallelism, yet we can prove the implication for the larger class of graphs without minor \( K_5 \). Clearly, the equality min-cutcond\((G, w)\) = min-mflowloss\((G, w)\) implies that the cut-condition is sufficient for the existence of a multiflow. Since \( G \) has no minor \( K_5 \), as mentioned above, the cut-cone of \( G \), that is, the cone pointed in \( 0 \) generated by the incidence vectors of the cuts of \( G \), is described by (24) and (27). If \( w(D) > 0 \) for some cut \( D \), then the maximum of \( w^T x \) over (24) and (27) is infinite. So, max \( w^T x \) over the cut-cone has an optimal solution (with value zero) if and only if \( w(D) \leq 0 \) for any cut \( D \), that is, if and only if the cut-condition holds. If the system (24) and (27) admits a feasible solution, since the cut-condition is sufficient, then \( (G, w) \) admits a multiflow \( y \). As in the proof of Corollary 4.1, this multiflow \( y \) can be used in order to obtain an integer feasible solution for the dual. Optimal and feasible are the same for the dual, since it consists in minimizing 0 over nonnegative \( y \in \mathbb{R}^{E \times E} \) such that

\[
\sum_{C \in C \cap \{ e \}} (y_C^e - \sum_{f \in C \setminus \{ e \}} y_C^f) \geq w_e \quad \text{for each } e \in E.
\]

Hence the system (24) and (27) is TDi.

Remark also that the cut inequalities (25) and the circuit inequalities (27) are equivalent for the class of planar graphs, which contains series-parallel graphs. Series-parallelism being closed under taking the dual graph (since the dual graph of \( K_4 \) is still \( K_4 \)) we can restate Corollary 4.1 as follows:

Corollary 4.2. The system (25)–(26) is TDi if and only if \( G \) is series-parallel. \( \square \)

To end the section we note that it seems unlikely that a TDi system associated with the min–max relation (ii) in Theorem 3.1 exists although it is not difficult to find, for the maximum multicut problem, formulations using variables associated with cuts or multicuts whose integer dual formulates the minimum cut-condition problem. However, since the weights then are associated with multicuts and not with edges, the system may not even be integral for e.g. triangles. So one should find a formulation using only edge variables \( x \) different from the circuit formulation. It seems impossible to express that \( x \) is the union of cuts without additional variables.

One could not have associated a TDi result with the relation min-mflowloss = max–multicut if we had restrictions on the weight function. However, as a by product of Theorem 3.1, we have the equality min-cutcond\((G, w)\) = min-mflowloss\((G, w)\) for series-parallel \( G \) and any \( w \), which is a particular case of the result of [23] cited in the Introduction. Indeed, as this result implies, the equality min-cutcond\((G, w)\) = min-mflowloss\((G, w)\) if the signed graph associated with \( (G, w) \) has no odd-minor \( K_4 \). The next section investigates a generalization of the equality min-mflowloss = max–multicut using signed graphs.

5. Covering and packing flows of signed graphs

A signed graph is a pair \((G, R)\) where \( G = (V, E) \) is a loopless graph and \( R \subseteq E \). Any edge in \( R \) is said to be signed in \((G, R)\). An odd circuit of \((G, R)\) is a circuit \( C \) of \( G \) that contains an odd number of edges in \( R \), and if moreover \(|C \cap R| = 1\), call \( C \) a flow of \((G, R)\). The cut-condition in \((G, R)\) is that \(|D \cap R| \leq |D \setminus R| \) for any cut \( D \) of \( G \). (In terms of weighted graphs, this is equivalent to \( w(D) \leq 0 \) where the edges in \( R \) have weight 1 and the other edges have weight \(-1\).

An odd circuit cover of \((G, R)\) is a subset \( T \subseteq E \) intersecting each odd circuit of \((G, R)\). Every odd circuit cover is also a flow cover (that is, an edge subset intersecting each flow). Given a signed graph \((G, R)\), we use the following notation:

1963
For every cut $C$,

**Corollary 5.3** \((\text{Fig. 3})\)

Theorem 5.1 generalizes the max-flow/min-cut theorem since, when the cut-condition holds, by Proposition 5.7, one can always resign so $\nu_{\text{odd}} = \tau_{\text{odd}}$ implies the equality $\nu_{\text{flow}} = \tau_{\text{flow}}$. Yet in general, as noticed before, the statements are unrelated. Proposition 5.7 shows that the case where the cut-condition holds is very special since in this case, for every signed graph \((G, R)\), the equalities $\nu_{\text{odd}} = \tau_{\text{odd}}$ and $\nu_{\text{flow}} = \tau_{\text{flow}}$ are actually equivalent. If the cut-condition does not hold, dealing with flows instead of odd circuits seems to change everything, especially because unlike odd circuits, the symmetric difference of an odd number of flows may contain no flow even if $G$ is series-parallel as illustrated by Fig. 3. Actually, Fig. 3 shows that, even if $G$ is series-parallel, the flow hypergraph of \((G, R)\), that is the pair \((E, \mathcal{F})\), is neither binary, nor balanced.

**Theorem 5.1** \((\text{[23])}\). If \((G, R)\) has no minor odd-$K_4$, then the maximum number of edge-disjoint odd circuits is equal to the minimum size of an odd circuit cover.

A consequence of it is **Corollary 5.3** (to make our contribution clear later in **Proposition 5.7** we give the proofs):

**Lemma 5.2** \((\text{[21], p. 1334})\). If the cut-condition holds in \((G, R)\), then $R$ is a minimum odd circuit cover of \((G, R)\).

**Proof.** For every cut $D$, if the cut-condition holds, then $|R \triangle D| = |D \setminus R| + |R \setminus D| = |D \cap R| + |R \setminus D| = |R|$. So, by (28), every odd circuit cover has at least $|R|$ edges. Hence $R$ is a minimum odd circuit cover.

**Corollary 5.3** \((\text{[21], p. 1334})\). If \((G, R)\) has no minor odd-$K_4$ and if the cut-condition holds in \((G, R)\), then the maximum number of edge-disjoint flows is equal to the minimum size of a flow cover.

**Proof.** Since the cut-condition holds, by **Lemma 5.2**, $R$ is a minimum odd circuit cover. Then, by **Theorem 5.1**, there exists a collection $C_1, \ldots, C_{|R|}$ of edge-disjoint odd circuits. Yet each $C_i$ contains one edge of $R$ and then it is a flow.

Notice that **Corollary 5.3** generalizes the max-flow/min-cut theorem since, when $H$ has only one multiple edge, then $G + H$ has no minor odd-$K_4$. Besides, the proof of **Corollary 5.3** shows that, for every signed graph, if the cut-condition holds, then the equality $\nu_{\text{odd}} = \tau_{\text{odd}}$ implies the equality $\nu_{\text{flow}} = \tau_{\text{flow}}$. Yet in general, as noticed before, the statements are unrelated. **Proposition 5.7** shows that the case where the cut-condition holds is very special since in this case, for every signed graph \((G, R)\), the equalities $\nu_{\text{odd}} = \tau_{\text{odd}}$ and $\nu_{\text{flow}} = \tau_{\text{flow}}$ are actually equivalent. If the cut-condition does not hold, dealing with flows instead of odd circuits seems to change everything, especially because unlike odd circuits, the symmetric difference of an odd number of flows may contain no flow even if $G$ is series-parallel as illustrated by Fig. 3. Actually, Fig. 3 shows that, even if $G$ is series-parallel, the flow hypergraph of \((G, R)\), that is the pair \((E, \mathcal{F})\), is neither binary, nor balanced.

![Fig. 2. Two signed graphs \((G, R)\), where $R$ is the set of the edges in bold. For the signed graph on the left: $\nu_{\text{flow}} = 1 < 2 = \tau_{\text{flow}} = \tau_{\text{odd}} = \nu_{\text{odd}}$. For the signed graph on the right: $\nu_{\text{flow}} = 2 < 3 = \nu_{\text{odd}}$.](image)
Lemma 5.4. If \( D \) is a multicut of \( G \), then \( T = R \Delta D \) is a flow cover of \( (G, R) \).

Proof. Let \( D \) be a subset of edges and let \( T = R \Delta D \). Assume that \( T \) is not a flow cover, that is, there exists a flow \( C \) of \( (G, R) \) with \( C \cap T = \emptyset \). Let \( e \) be the demand of \( C \). Since \( e \) is in \( R \) but not in \( T \), then \( e \) belongs to \( R \cap D \). Moreover for each \( f \in C \setminus \{e\} \), since \( f \) is not in \( R \cup T \), then \( f \) is not in \( D \). It follows that \( e \) is the only edge in \( C \cap D \); hence by (17) \( D \) is not a multicut.

Lemma 5.5. If \( T \) is an inclusionwise minimal flow cover of \( (G, R) \), then \( D = R \Delta T \) is a multicut of \( G \).

Proof. Let \( T \) be a minimal flow cover of \( (G, R) \). Note that, since \( T \) is minimal, for each edge \( f \in T \) there exists a flow such that \( f \) is the only edge in the flow and in \( T \). For each edge \( f \in T \), we let \( C_f \) be a flow with \( C_f \cap T = \{f\} \). Now assume for contradiction that \( D = R \Delta T \) is not a multicut of \( G \). By (17), there exists a circuit \( C \) such that \( |C \cap D| = 1 \). Let \( e \) be the (unique) edge in \( C \cap D \). Observe that every edge in \( T \cap C \setminus \{e\} \) is necessarily in \( R \), and that every edge in \( R \cap C \setminus \{e\} \) is necessarily in \( T \). We denote by \( F \) the set of the edges in \( R \cap T \cap C \setminus \{e\} \). By (10), the union of \( C \setminus F \) and of the paths \( C_f \setminus \{f\} \) for \( f \in F \) contains a circuit \( \tilde{C} \) without an edge in \( R \cup T \cup \{e\} \). Since \( T \) is a flow cover, it follows that \( e \) is not in \( R \cup T \). Indeed, otherwise \( \tilde{C} \) is a flow but \( \tilde{C} \cap T = \emptyset \). Thus \( e \) is in \( T \setminus R \). Let \( e' \) be the demand of the flow \( C_e \) (note that \( e \) and \( e' \) are distinct). By (10), the union of \( \tilde{C} \setminus \{e\} \) and of the path \( C_{e'} \setminus \{e\} \) contains a flow, the demand of which is \( e' \). We have a contradiction since this flow is not covered by \( T \). \( \square \)

In order to state the last lemma, let us associate a weighted graph \((G, w)\) to a signed graph \((G, R)\) by defining the weight of each edge \( e \) as:

\[
w_e := \begin{cases} +1 & \text{if } e \in R, \\ -1 & \text{if } e \in E \setminus R. \end{cases}
\]

Clearly, there is a one-to-one correspondence between signed graphs and weighted graphs with weight in \([-1, +1]\). In multiflow and multicut problems we can assume that the weights are in \([-1, +1]\) since these problems remain unchanged after the following transformation:

Replace each edge \( e \) of \((G, w)\) by \(|w_e|\) parallel edges with weight \( \frac{w_e}{|w_e|} \).

In the following we assume that \((G, R)\) and \((G, w)\) are in correspondence. Combining the two first lemmas we obtain that finding a minimum flow cover in a signed graph and finding a maximum multicut in a weighted graph are equivalent, as stated by the last lemma below:

Lemma 5.6. Let \( T \) be a minimum size flow cover of \((G, R)\) and let \( D \) be a maximum weight multicut of \((G, w)\), then \( |T| + w(D) = |R| \).

Proof. Let \( T \) be a minimum flow cover and let \( D \) be a multicut maximizing \( |D \cap R| - |D \setminus R| \) over all multicuts \( D \) of \( G \). Notice that \( |R \Delta D| + |D \cap R| - |D \setminus R| = |R| \). Since by Lemma 5.4, \( R \Delta D \) is a flow cover, it follows that \( |T| + w(D) \leq |R| \). Now remark that \( |T| + |(R \Delta T) \cap R| - |(R \Delta T) \setminus R| = |T| + |R \setminus T| - |T \setminus R| = |R| \). Since by Lemma 5.5, \( R \Delta T \) is a multicut, it follows that \( |T| + w(D) \geq |R| \). \( \square \)

Now we can prove the propositions.

Proposition 5.7. If the cut-condition holds in \((G, R)\), then the following propositions are equivalent:

(i) The maximum number of edge-disjoint odd circuits is equal to the minimum size of an odd circuits cover;
(ii) The maximum number of edge-disjoint flows is equal to the minimum size of a flow cover.

If, moreover, (i)–(ii) hold, then \( \tau_{\text{flow}} = \tau_{\text{odd}} = \nu_{\text{odd}} = \nu_{\text{flow}} = |R| \).

(see [21, p. 1439] for a definition). Proposition 5.8 shows that the equality \( \nu_{\text{flow}} = \tau_{\text{flow}} \) and the equality max-multicut = min-multiflow are equivalent. Let us start the proof of these propositions with three small lemmas.
Proof. (i) ⇒ (ii): It follows from the proof of Corollary 5.3.

(ii) ⇒ (i): If the cut-condition holds then the maximum of \( w(D) \) over all multicuts is 0, and hence, by Lemma 5.6, \( R \) is a minimum flow cover. Then there are \( |R| \) edge-disjoint odd circuits, yet \( R \) is an obvious (minimum) odd circuit cover.

Proposition 5.8. The maximum number of edge-disjoint flows is equal to the minimum size of a flow cover in \((G, R)\) if and only if \( \min\text{-mflowloss}(G, w) = \max\text{-multicut}(G, w) \).

Proof. Clearly, the maximum number of disjoint flows is equal to \( (21) \).

Lemma 5.6

Corollary 5.3

The maximum number of edge-disjoint flows is equal to the minimum size of a flow cover in \( G \) is series–parallel.


References


L.R. Ford, D.R. Fulkerson, Maximal flow through a network RAND corporation, Santa Monica, California, 1954.


