Woodall's conjecture on Packing Dijoins: a survey

Paulo Feofiloff
http://www.ime.usp.br/~pf/dijoins/

21/10/2005, updated 30/11/2005

Abstract

Woodall's conjecture asserts the following about every directed graph: if every directed cut of the graph has k or more edges then the graph has k or more mutually disjoint dijoins. Here, a *dijoin* is a set J of arcs such that any vertex is connected to any other by a path all of whose forward-directed arcs are in J. This talk is a little survey of the counterexamples to a generalized version of the conjecture.

1 Introduction

Work on Woodall's conjecture (see Schrijver's book [Sch03, cap.56]) has been slow for two decades. Then, some 5 years ago, Cornuéjols[1] and his student Guenin brought the subject up again by studying an abstract generalization of the conjecture [CGM00]. More recently, Williams[2] wrote a masters's thesis [Wil04] on the subject under the supervision of Guenin.

2 Graphs, cuts and joins

A **graph** is a pair (V, E) where V is a finite set and E is a set of ordered pairs of elements of V.[3] The elements of V are called **vertices** and those of E are **edges**.[4] Actually, E is a *multi*set: it may contain several copies of each edge. But, to make

[1] Cornuéjols came to our Ubatuba Pronex Workshop in 2003, but did not speak about the conjecture.
[2] Williams came to GRACO 2005 and spoke about his work.
[3] Usually such object is called "directed graph" or "digraph".
[4] Usually the elements of E are called "arcs".

language simpler, we shall treat E as an ordinary set. As usual, the sets of vertices and edges of a graph G will be denoted by V_G and E_G respectively.

For any subset X of V, let $\nabla^+(X)$ be the set of edges that leave X and $\nabla^-(X)$ the set of edges that enter X. A **source-set** is any proper nonempty subset X of V_G such that $\nabla^-(X) = \emptyset$. A **sink-set** is defined by transposition.[5]

A **cut**[6] is any set of the form $\nabla^+(X)$ such that X is a source-set. Of course any set of the form $\nabla^-(Y)$ where Y is a sink-set is also a cut. (See figure 1.) A graph is (weakly) connected if and only if all its cuts are nonempty.

A **join**[7] is a set J of edges having the following property: for each pair u,v of vertices, there is a path from u to v whose forward-directed edges are in J. (See figure 1.) In other words, a set J of edges is a join if its contraction makes the graph strongly connected.

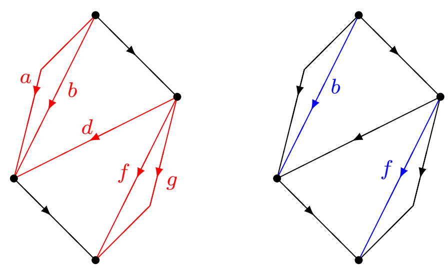


Figure 1: On the left, the set of edges $\{a, b, d, f, g\}$ is a cut. On the right, the set $\{b, f\}$ is a join.

A cut C is **peripheral** if $C = \nabla^+(X)$ and X is a (inclusionwise) minimal source-set[8] or if $C = \nabla^-(Y)$ and Y is a (inclusionwise) minimal sink-set.

DAGs. We may restrict ourselves, with no loss of generality, to DAGs (directed acyclic graphs), since cuts are disjoint from directed circuits. In a DAG, the minimal source-sets and the minimal sink-sets consist of single vertices. In other words, each peripheral cut is of the form $\nabla^+(\{u\})$ or of the form $\nabla^-(\{v\})$. In the first case, we say that u is a **source**; in the second, we say that v is a **sink**. We also say that u and v are **peripheral** vertices.

In a DAG, a set J of edges is a join if and only if for each source u e each sink v there exists a path from u to v whose forward-directed edges are in J.

[5] The **transpose** (or **directional dual**) of a graph G is the graph obtained by inverting the orientation of every edge of G.)

[6] It would have been more correct to say "nontrivial directed cut" or "nontrivial dicut".

[7] It would have been more correct to say "dijoin".

[8] A source-set X is minimal if and only if $G[X]$ is strongly connected.

3 Minimum cuts and disjoint collections of joins

A collection $\mathcal{D}$ of joins is **disjoint** if its elements are pairwise disjoint. For any graph G,

$$\nu(G) \equiv \max\{|\mathcal{D}| : \mathcal{D} \text{ is a disjoint collection of joins of } G\}$$

(see figure 2).[9] No polynomial algorithm is known to compute $\nu(G)$ (but there is no evidence that the problem is NP-hard). For any graph G,

$$\tau(G) \equiv \min\{|C| : C \text{ is a cut of } G\}$$

(see figure 2).[10] There exists a polinomial algorithm (essencially the max-flow-min-cut algorithm) to calculate $\tau(G)$.

Every cut intercepts every join, i.e., $C \cap J \neq \emptyset$ for every cut C and every join J. Hence, we have the following relation:

Lemma 1 *For any graph G, one has $\nu(G) \leq \tau(G)$.*

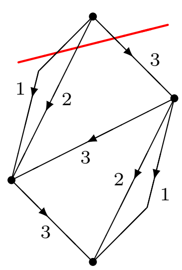


Figure 2: The figure shows (colored lines) a cut of size 3 and three pairwise disjoint joins (labels 1, 2 and 3). Hence, $\tau = 3$ e $\nu = 3$.

4 Linear programming formulation

The relation between cuts and joins is more intimate than suggested at the end of the previous section: every transversal of cuts (i.e., every set of edges that intersects all the cuts) includes a join and, conversely, every transversal of joins includes a cut. Therefore, ν and τ may be defined by the following pair of integer linear programs:

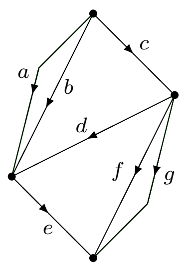
$$\text{maximize} \quad y1 \quad \text{subject to the constraints } y \in \{0,1\}^{\mathcal{J}} \text{ and } yM \leq 1\,, \tag{1}$$

$$\text{minimize} \quad 1x \quad \text{subject to the constraints } x \in \{0,1\}^{E} \text{ and } Mx \geq 1\,, \tag{2}$$

[9] If G is strongly connected then $\nu(G) = \infty$, since $\emptyset$ is a join and any multicollection of such joins is disjoint.

[10] If G is strongly connected then $\tau(G) = \infty$, since G has no cuts.

where $\mathcal{J}$ is the set of all (inclusionwise minimal) joins of G and M is the matrix whose rows are the characteristic vectors os the elements of $\mathcal{J}$ (see figure 3).[11] Of course (1) and (2) are dual to each other. The optimal value of program (1) is $\nu(G)$ and the optimal value of program (2) is $\tau(G)$.



a	b	c	d	e	f	g
1	–	–	–	–	1	–
1	–	–	–	–	–	1
–	1	–	–	–	1	–
–	1	–	–	–	–	1
–	–	1	1	1	–	–
1	–	–	–	1	–	–
–	1	–	–	1	–	–
–	–	1	–	–	1	–
–	–	1	–	–	–	1

Figure 3: The rows of the matrix are the characteristic vectors of all the minimal joins of the graph.

5 Woodall's conjecture

The following min-max conjecture [Woo78a, Woo78b] is still open:

Conjecture 1 (Woodall) *For any graph G, one has $\nu(G) = \tau(G)$.*

We say that a graph G **satisfies Woodall's min-max** (or that **the joins of G pack**) if $\nu(G) = \tau(G)$ (see figure 4).

Woodall's conjecture is dual to the Lucchesi–Younger theorem [LY78]. According to that theorem, any maximum disjoint collections of cuts has the same cardinality as a minimum join. One of the consequences of this theorem is that all the vertices of the polyhedron

$$\{x : x \in \mathbb{R}^E_+ \text{ and } Mx \geq 1\}$$

are integer.[12] Hence, the constraint "$x \in \{0,1\}^E$" in (2) can be replaced by "$x \in [0,1]^E$".

(Here is a curiosity: A simple construction [Sch03, p.968] shows that $\tau(G) \geq 2$ implies $\nu(G) \geq 2$.[13] However, ee do not know whether there exists a constant k_3 such that

[11] The "1" in "$y1$" is the vector indexed by $\mathcal{J}$ whose components are all equal to 1.

[12] In more technical terms, one says that the clutter of joins is *ideal*.

[13] Let H be a strongly connected re-orientation of G. (Such re-orientation does exist since the undirected graph underlying G is 2-edge-connected.) Let J be the set $E_G \cap E_H$ and let $\tilde{J}$ be the set $\{uv \in E_G : vu \in E_H\}$. Of course $J \cap \tilde{J} = \emptyset$. Since each edge of H belongs to a directed circuit in H,

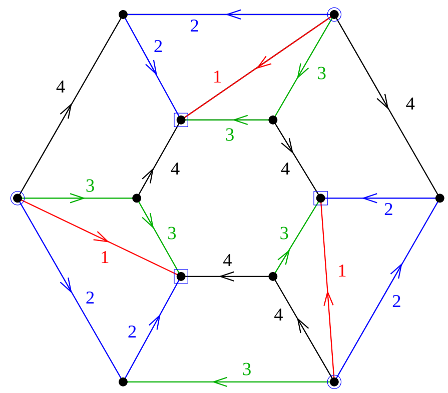


Figure 4: For this graph, $\nu = 4$ and $\tau = 4$. Hence, the graph satisfies Woodall's min-max. The colors (and the labels) indicate mutually disjoint joins. The graph is a DAG. The sources and sinks are indicated by circles and aquares respectively.

$\tau(G) \geq k_3$ implies $\nu(G) \geq 3$.)

6 Analogy with maximum flow

Woodall's conjecture is analogous (to some extent) to the Max-Flow-Min-Cut theorem. The Max-Flow-Min-Cut theorem states that every maximum flow from a vertex s to a vertex t has the same size as a minimum semi-cut. Here, a **flow** is a set of pairwise edge-disjoint *directed*[14] paths from s to t. A **semi-cut** is any set of the form $\nabla^+(X)$ such that $s \in X \subseteq V_G \setminus \{t\}$. (See figure 5.)

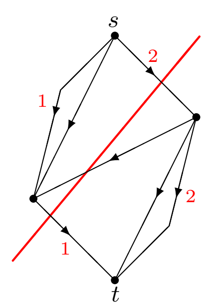


Figure 5: A maximum flow (labels 1 and 2) and a minimum semi-cut (colored line).

we conclude that $J \cap C \neq \emptyset$ and $\tilde{J} \cap C \neq \emptyset$ for each cut C. Hence, J and $\tilde{J}$ are joins.

[14] All edges are forward-directed; there are no reverse-directed edges.

The analogy is not perfect, of course. In Max-Flow-Min-Cut, the paths have no reverse-directed edges and there are two fixed vertices (s and t), while in Woodall's conjecture only the forward-directed edges of the paths are taken into account and there are no fixed vertices.[15]

The Max-Flow-Min-Cut theorem has a "capacitated" version. In this version, each edge e has a capacity $c(e)$ in $\mathbb{N}$, where $\mathbb{N}$ is the set of natural numbers.[16] An edge e cannot be used more than $c(e)$ times by the flow and contributes $c(e)$ to the size of each semi-cut that contains it.

It is difficult to immagine how the non-capacitated version of the Max-Flow-Min-Cut theorem could hold without the capacitated version holding too, since the proof of the theorem is an induction in c.

7 Capacitated generalization of Woodall's conjecture

The analogy with the Max-Flow-Min-Cut theorem suggests that we should study the capacitated version of Woodall's conjecture. A **capacity function** for a graph G is any function c from E_G to $\mathbb{N}$. A collection $\mathcal{D}$ de joins is c-**disjoint** if each edge e belongs to at most $c(e)$ elements of $\mathcal{D}$. The definitions of ν and τ must be adjusted accordingly:

$$\begin{aligned}\nu(G,c) &\equiv \max\{|\mathcal{D}| : \mathcal{D} \text{ is a } c\text{-disjoint multicollection of joins of } G\} \quad \text{e}\\ \tau(G,c) &\equiv \min\{c(C) : C \text{ is cut of } G\}\,.\end{aligned}$$

These definitions may be formulated in the language of integer linear programming: $\nu(G,c)$ is the optimum value of the program

$$\text{maximize } \; y1 \; \text{ subject to } \; y \in \mathbb{N}^{\mathcal{J}} \text{ and } yM \leq c \tag{3}$$

and $\tau(G,c)$ is the optimum value of the dual program

$$\text{minimize } \; cx \; \text{ subject to } \; x \in \mathbb{N}^{E} \text{ and } Mx \geq 1\,, \tag{4}$$

where M is the matrix whose rows are the characteristic vectors of the joins. It is easy to verify that $\nu(G,c) \leq \tau(G,c)$ for all G and all c.

Here is the capacitated version of Woodall's conjecture, also known as the Edmonds–Giles [EG77] conjecture:

[15] For the maximum flow between two vertices of an *undirected* graph there is second diference, subtle but important: unlike with joins, every undirected path between s and t has an *odd* number of edges in common with every undirected cut.

[16] We consider 0 a natural number. Hence, $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$.

Conjecture 2 (capacitated Woodall) *For every graph G and any c in $\mathbb{N}^E$, one has $\nu(G,c) = \tau(G,c)$.*

The identity $\nu(G,c) = \tau(G,c)$ holds for certain classes of graphs. It holds, for example, when G is a source-sink-connected[17] DAG [FY87].[18] In particular, the identity holds when G is a DAG with a single source[19] (or a single sink).[20] In particular, the identity holds when G is um DAG with a single source *and* a single sink.This special case follows from the Max-Flow-Min-Cut theorem mentioned in section 6 and is the basis of a polynomial algoritmo that calculates $\tau(G,c)$.

In geral, the capacitated conjecture *is false*. Counter-examples will be shown shortly. We do not even know whether there exists a constant k_2 such that $\nu(G,c) \geq 2$ for every capacitated graph (G,c) with $\tau(G,c) \geq k_2$.

8 Capacities versus active edges

The capacitated version of the Max-Flow-Min-Cut theorem does not add generality to the "pure" version, since each edge with capacity k may be "simulate" by k edges in parallel. When $k = 0$, in particular, we simply delete the edge from the graph.

At first sight, the same seems to hold for the capacitated version of Woodall's conjecture. In fact, an edge e with capacity $c(e) \geq 1$ may be simulated by $c(e)$ copies of e in parallel. But the case $c(e) = 0$ of this construction has bad side-effects: the deletion of an edge e may create new cuts, i.e., cuts that are not of the form $C \setminus \{e\}$ where C is one of the original cuts.[21] This problem can be circumvented by replacing the usual deletion of an edge by the

"deactivation" of the edge.

This will require us to remember which edges of the graph are "active" and which are "inactive". It will be necessary, therefore, to replace the concept of a graph by that of a graph-with-active-and-inactive-edges, or *graaph*.

A **graaph** is a pair (G, A) where G is a graph and A is a part of E_G. The edges in A are **active** and those in $E_G \setminus A$ are **inactive**. The joins containing an inactive edge are

[17] Every source connected to every sink by a directed path.

[18] The identity also holds when G is a tree with arbitrarily oriented edges plus any transitive edges (i.e., edges uv such that there exista a directed path from u to v in the tree). The identity also holds when G is series-parallel.

[19] Same proof technique as that of the Edmonds' Disjoint Branchings theorem [Edm73].

[20] The validity of the conjecture in these cases has a simple but curious consequence [SV05]: for every graph G and any capacity function c from E_G to $\{0, 2, 4, 6, \ldots\}$, onde has $\nu(G,c) \geq \frac{1}{2}\,\tau(G,c)$.

[21] For example, give an appropriate orientation to the edges of an undirected circuit.

ignored.[22] The definition of ν and τ for graaphs is obvious:

$$\begin{aligned} \nu(G, A) &\equiv \max\{|\mathcal{D}| : \mathcal{D} \text{ is a disjoint collection of joins and } \bigcup \mathcal{D} \subseteq A\} \text{ and} \\ \tau(G, A) &\equiv \min\{|A \cap C| : C \text{ is a cut}\}\,. \end{aligned}$$

Conjecture 2 is, therefore, equivalent to the following:

Conjecture 3 (Woodall for graaphs) *For any graaph* (G, A)*, one has* $\nu(G, A) = \tau(G, A)$*.*

9 Active edges plus capacities

For later use (see section 14), it will be convenient to assign capacities to the active edges. Thus, a **capacitated graaph** is a triple (G, A, c) where (G, A) is a graaph and c is a capacity function for (G, A), i.e., a function c from E_G to $\mathbb{N}$ such that $c(e) = 0$ for all e in $E_G \setminus A$. For any capacitated graaph (G, A, c), define

$$\begin{aligned} \nu(G, A, c) &\equiv \max\{|\mathcal{D}| : \mathcal{D} \text{ is a } c\text{-disjoint multicollection of joins}\} \text{ and} \\ \tau(G, A, c) &\equiv \min\{c(C) : C \text{ is a cut}\}\,. \end{aligned}$$

Of course $\nu(G, A, c) \leq \tau(G, A, c)$. The conjectures 2 and 3 will now take the following form:

Conjecture 4 (Woodall for capacitated graaphs) *For any capacitated graaph* (G, A, c)*, one has* $\nu(G, A, c) = \tau(G, A, c)$*.*

We shall say that (G, A, c) **satisfies Woodall's min-max** (or that **the joins of** (G, A, c) **pack**) if $\nu(G, A, c) = \tau(G, A, c)$.

10 Schrijver's counterexample

Schrijver [Sch80] found the first counterexample to conjecture 3 (and therefore also to conjectures 2 and 4.) Schrijver's graaph, denoted by (G_1, A_1), is defined in figure 6.[23]

[22] The idea of inactive edges is very natural if we consider the following heuristic for finding a disjoint collection of joins: (1) find a join J and (2) look for a disjoint colection of joins that do not use any edge of J. (In general, of course, the size of the collection found by this heuristic is much smaller than ν.)

[23] Since not all edges of the graaph are active, (G_1, A_1) is not a counterexample to the original conjecture.

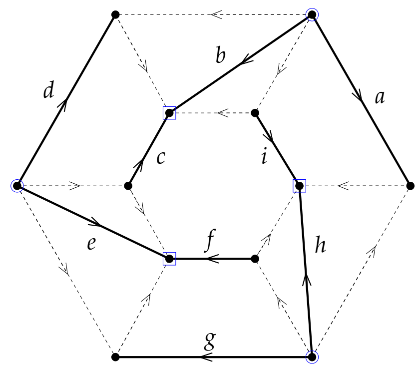


Figure 6: Schrijver's (G_1, A_1) graaph. The edges in A_1 are indicated by solid lines; the other edges are dashed. Graph G_1 is a DAG. The sources are marked by circles and the sinks by squares. The graaph does not satisfy Woodall's min-max: $\nu = 1$ while $\tau = 2$.

Fact 2 $\nu(G_1, A_1) = 1$ *and* $\tau(G_1, A_1) = 2$.

PROOF: Observe the following properties of (G_1, A_1):

G_1 is a DAG,
$\tau(G_1, A_1) = 2$,
each minimum cut is peripheral,
each component of the "active subgraph" $G_1[A_1]$ is a path,
each internal vertex of each "active path" is a minimum cut.

It is easy to check that $\nu(G_1, A_1) \geq 1$ and $\tau(G_1, A_1) \geq 2$. Now suppose for a moment that $\nu(G_1, A_1) = 2$. Let J and $\tilde{J}$ be two mutually disjoint joins, both contained in A_1. The edges of each active path are alternately in J and $\tilde{J}$. Since there are three active paths, there are $\frac{1}{2}2^3 = 4$ possible patterns for the pair $J, \tilde{J}$ (see figure **??**). For each of these patterns, either J or $\tilde{J}$ is not a join. Specifically, J or $\tilde{J}$ does not cover one of the four cuts that intersect each active path only once (see figure 8). This contradiction shows that $\nu(G_1, A_1) < 2$. □

Schrijver's counterexample is a "ring" of length $2i$ with $i = 3$. The analogous graaphs with $i = 5, 7, 9, \ldots$ are also conter-examples. Yet the analogous graaphs with $i = 2, 4, 6, \ldots$ all satisfy Woodall's min-max (see figure 9).

a	b	c	d	e	f	g	h	i
J	$\tilde{J}$	J	J	$\tilde{J}$	J	J	$\tilde{J}$	J
J	$\tilde{J}$	J	J	$\tilde{J}$	J	$\tilde{J}$	J	$\tilde{J}$
J	$\tilde{J}$	J	$\tilde{J}$	J	$\tilde{J}$	J	$\tilde{J}$	J
J	$\tilde{J}$	J	$\tilde{J}$	J	$\tilde{J}$	$\tilde{J}$	J	$\tilde{J}$

Figure 7: Trying to find two disjoint joins in Schrijver's graaph. The rows of the table correspopnd to the three active paths of the Schrijver's graaph (G_1, A_1) (figure 6). Each row shows a configuration of two potential mutually disjoint joins J e $\tilde{J}$. There are $\frac{1}{2}2^3 = 4$ different configurations.

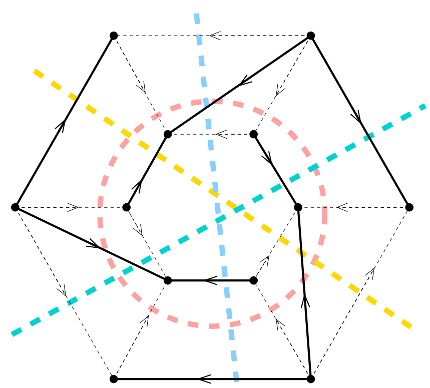


Figure 8: The four "special" cuts in (G_1, A_1): each cut intersects each active path only once.

Figure 9: In spite of its superficial similarity with (G_1, A_1), the graaph in this figure has $\nu = 2 = \tau$. The labels 1 and 2 indicate two mutually disjoint joins.

11 The counterexamples by Cornuéjols–Guenin

For two decades, Schrijver's counterexample was the only one known. Then, in 2002, Cornuéjols and Guenin published [CG02] two new counterexamples, which we shall denote by (G_2, A_2) and (G_3, A_3). There graaphs are defined by the figures 10 and 11.

Fact 3 *$\nu(G_2, A_2) = 1$ and $\tau(G_2, A_2) = 2$.*

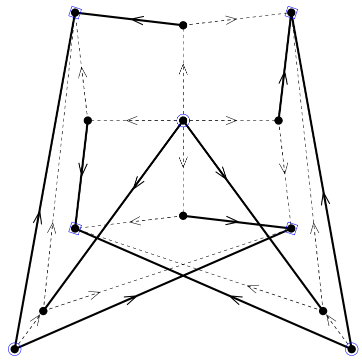


Figure 10: The Cornuéjols–Guenin graaph (G_2, A_2). Edges in A_2 are indicated by solid lines; the other edges are dashed. The graph G_2 is a DAG; the sources are marked by circles and the sinks by squares. The graaph does not satisfy Woodall's min-max, since $\nu = 1 < 2 = \tau$.

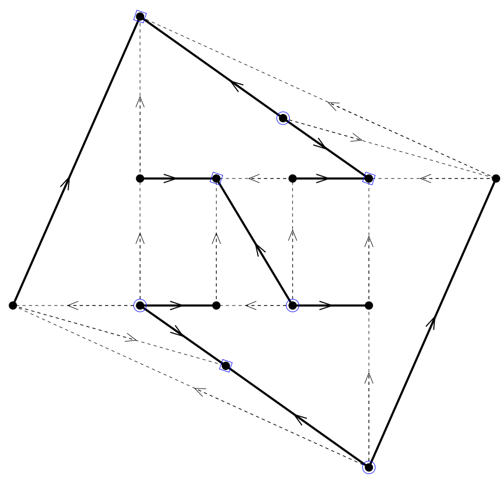


Figure 11: The Cornuéjols–Guenin graaph (G_3, A_3). Edges in A_3 are indicated by solid lines; the other edges are dashed. The graph G_3 is a DAG; the sources are marked by circles and the sinks by squares. The graaph does not satisfy Woodall's min-max, since $\nu = 1 < 2 = \tau$.

Fact 4 *$\nu(G_3, A_3) = 1$ and $\tau(G_3, A_3) = 2$.*

The proofs of these two facts are similar to the proof os fact 2.

The graaphs (G_2, A_2) e (G_3, A_3) were obtained from the enumeration [CGM00] of certain objects more general than graaphs, known as ideal clutters. Seymour's Q_6 clutter [Sey77] played an important rôle in this enumeration.

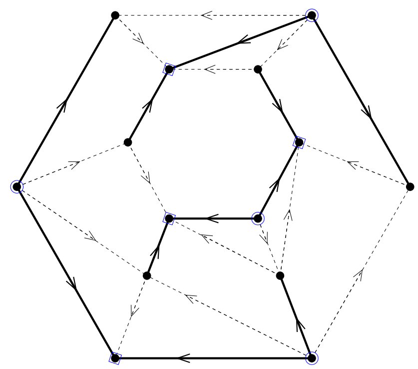


Figure 12: A different drawing of the Cornuéjols–Guenin graaph (G_3, A_3). This is the drawing favored by Williams [Wil04, WG05].

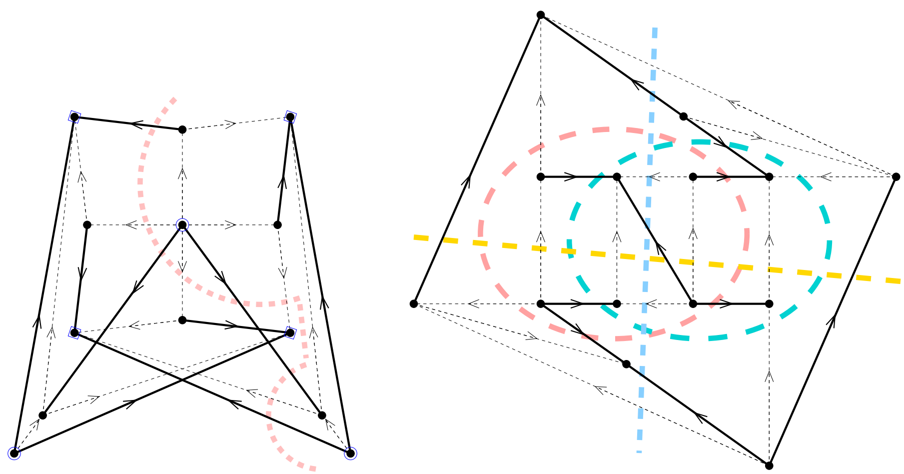


Figure 13: The first figure indicates one of the "special" cuts in (G_2, A_2). The other three special cuts are defined by simmetry. The second figure indicates the four special cuts in (G_3, A_3). Each of the special cuts intersects each active path only once.

12 Other counterexamples?

Of course any graaph isomorphic[24] to Schrijver's is also a counterexample to Woodall's min-max. But this should not really count a "different" counteresxample.

[24] An **isomorphism** between graaphs (G, A) and (G', A') is a bijection φ from V_G to $V_{G'}$ such that, for each pair (u, v) of vertices of G, one has $uv \in A$ if and only if $\varphi(u)\varphi(v) \in A'$ and $uv \in E_G \setminus A$ if and only if $\varphi(u)\varphi(v) \in E_{G'} \setminus A'$.

Another trivial observation of this nature involves transposition.[25] We shall say that a graph G is **dirmorphic**[26] to a graph G' if G is isomorphic to the transpose of G'. Similarly, a graaph (G, A) is dirmorphic to a graaph (G', A') is (G, A) is isomorphic to the transpose of (G', A'). Of course if (G, A) does not satisfy Woodall's min-max then no graaph dirmorphic to (G, A) will satisfy the min-max. the transpose of any counterexample (G, A). But such counterexamples should not really count as "different" from (G, A).

By the way, it is curious that Schrijver's graaph (G_1, A_1) is dirmorphic to itself. The same is true of the Cornuéjols–Guenin graaphs (G_2, A_2) e (G_3, A_3). But not all counterexamples have such property, as we shall see next.

There are conterexamples that differ very little from those of Cornuéjols and Guenin but are neither isomorphic nor dirmorphic to them. We shall exhibit a family of such examples in what follows.

Let u and x be the vertices of the Cornuéjols–Guenin graaph (G_3, A_3) indicated in figure 14. Consider the graaph $(G_3 - ux, A_3)$. One can verify (see the proof of fact 2) that $\nu(G_3 - ux, A_3) = 1 < 2 = \tau(G_3 - ux, A_3)$, and therefore $(G_3 - ux, A_3)$ does not satisfy Woodall's min-max. Note that $(G_3 - ux, A_3)$ is neither isomorphic nor dirmorphic to any of the three counterexamples discussed so far. As a curiosity, we remark that $(G_3 - ux, A_3)$ is also not dirmorphic to itself.[27] (But $(G_3 - ux, A_3)$ is dirmorphic to $(G_3 - u'x', A_3)$.).

The graaph $(G_3 - uy, A)$ (see figure 14) has similar properties: it does not satisfy Woodall's min-max but is neither isomorphic nor dirmorphic to any of the previous counterexamples. (Graaph $(G_3 - uy, A)$ is not dirmorphic to itself, but it is dirmorphic to $(G_3 - u'y', A)$.)

The graaph $(G_3 - \{ux, u'x'\}, A)$ is another counterexample. It is neither isomorphic nor dirmorphic to the previous ones (but is dirmorphic to itself).

The graaph $(G_3 - \{ux, u'y'\}, A)$ is another counterexample. It is neither isomorphic

[25] The **transpose** (or **directional dual**) of a graph G is the graph $(V_G, \vec{E_G})$ where $\vec{E}$ is the set $\{vu : uv \in E\}$. The **transpose** of a graaph (G, A) is the graaph $(\vec{G}, \vec{A})$, where $\vec{G}$ is the transpose of G.

[26] I've just invented this definition; it is not standard.

[27] PROOF: To simplify notation, we shall use the abbreviation $(G, A) \equiv (G_3 - ux, A_3)$ in this footnote. Observe that $G[A]$ has three components: two paths of length 4 and one path of length 3. Let P be the path of length 4 that contains u, let Q be the other path of length 4, and let R be the path of length 3. We shall say that these are the active paths of (G, A). Let $(\vec{G}, \vec{A})$ be the transpose of (G, A) and let P', Q', and R' be the paths in $\vec{G}[\vec{A}]$ corresponding to P, Q, and R respectively. We shall say that these are the active paths of $(\vec{G}, \vec{A})$. Suppose that tere exists an isomorphism φ from (G, A) to $(\vec{G}, \vec{A})$. Since u is a source in $G[A]$ and belongs to P, thus $\varphi(u)$ can only be 14 or 7. Since 14 is incident to three inactive edges in $\vec{G}$, we have $\varphi(u) \neq 14$, and therefore $\varphi(u) = 7$. But all the inactive edges incident to u in G have their other end in Q, while in $\vec{G}$ the inactive edges incident to 7 have their other end in R'. Contradiction.

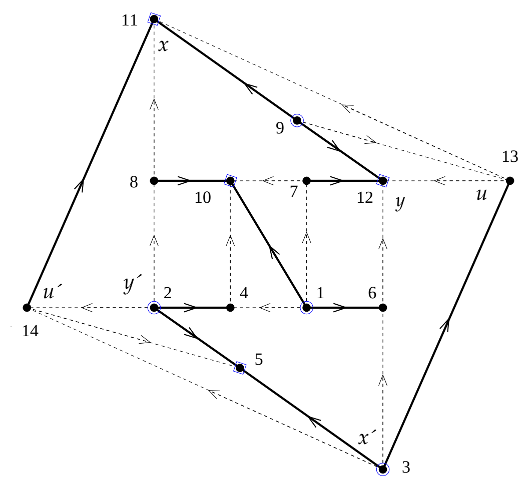


Figure 14: The graaph (G_3, A_3) is a counterexample to Woodall's conjecture. The graaph $(G_3 - ux, A_3)$ is another counterexample. So are $(G_3 - uy, A_3)$ and $(G_3 - \{uy, u'x'\}, A_3)$ and $(G_3 - \{uy, u'y'\}, A_3)$. Other variations, like $(G_3 - u'x', A_3)$, are also counterexamples, but they are isomorphic or dirmorphic to the one of already mentioned.

nor dirmorphic to the previous ones.(but is dirmorphic to itself).

We would like to build a catalogue of all the "essencial" or "minimal" counterexamples to Woodall's generalized conjecture. If the minimality is defined properly, every minimal counterexample (G, A) will have (at least) the following properties:

- G um DAG,
- every minimum cut is peripheral,
- every edge in A belongs do a minimum cut,
- the "active subgraph" $G[A]$ is a forest (edge orientations ignored),
- every vertex of G in incident to some element of A.

Looking for an appropriate definition of "minimal counterexample" brings up the idea of *minor*, which we review briefly in the next section.

13 Graph minors

This section makes a brief review of the characterization of graph properties by means of *minor*s. We are referring, of course, o properties which are invariant under isomorphism. We shall examine two classical characterizations: the bipartite graphs and the planar graphs. Unlike in the rest of this text, all graphs in this section are undirected.

Bipartite graphs and deletion of edges. As is well known, a graph is bipartite if and only if it does not contain an odd circuit. This characterization has two ingredients: a (infinite) set of "forbiden", clearly non-bipartirte, graphs — the odd circuits — and a "reduction" operation — the deletion of edges (and the deletion of isolated vertices) — that transforms a graph int a smaller one. The operation of deletion of edges *preserves the bipartite character of the graph*. Hence, if there exists a sequence of reductions that transforms a given graph G into an odd circuit then G is not bipartite.

The set of forbidden graphs is "complete": any non-bipartite graph can be reduced to a one of the forbidden graphs, Hence, the set of forbidden graphs contains all non-bipartite graphs that are minimal under the reduction: the deletion of any edge of an odd circuit produces a bipartite graph.

Though we began this discussion by speaking of two ingredients — the set of forbidden graphs and the reduction operation — we see now that only one ingredient is needed: *a reduction operation that preserves the bipartite character of the graph*. The forbidden graphs are then, simply, all the minimally non-bipartite graphs, i.e, all non-bipartite graphs that become bipartite as a result of any reduction.

Planarity and deletion/contraction of edges. Consider the weel-known characterization of planar graphs: a graph is planar if and only if it "contains" neither K_5 nor $K_{3,3}$.

In this case, we have two reduction operations: deletion of edges and contraction of edges.[28] Both *preserve planarity*. Hence, if successive reductions of a graph G produce a non-planar graph (K_5 or $K_{3,3}$ for example) then G is not planar.

Only two non-planar graphs are minimal under the deletion and contraction operations: K_5 and $K_{3,3}$. These are, then, the "forbidden" graphs for planarity.

14 Minimal counterexamples for Woodall's conjecture

Let's apply the ideas of the previous section to the class of graaphs that satisfy Woodall's min-max, i.e., to the class of graaphs (G, A) such that $\nu(G, A) = \tau(G, A)$. Note that the min-max is invariant under isomorphism and under dirmorphism[29].

We must begin by choosing a set of reduction operations that preserve the validity of the $\nu = \tau$ min-max. For lack of some better idea, let's try the operations of contraction and deletion of edges. Actually, as discussed in section 8, the deletion must be replaced by the deactivation.[30]

[28] One of these operation alone will not produce the desired effect.

[29] That is, (G, A) satisfies Woodall's min-max if and only if $(\vec{G}, \vec{A})$ also satisfies the min-max.

[30] These operation are "well-matched" with isomorphism and transposition: If (G, A) is isomorphic

For any graaph (G, A), the **deactivation** of an element a of A results in the graaph $(G, A \setminus \{a\})$. The **contraction** of a results in the graaph $(G/a, A \setminus \{a\})$. More generally, a **deactivation/contraction** of a graaph (G, A) is any graaph of the form $(G/B', A \setminus (B' \cup B''))$ such that $B' \cup B'' \subseteq A$ and $B' \cap B'' = \emptyset$. We shall assume, in general, that the deactivation/contraction is non-trivial, i.e., that $B' \neq \emptyset$ or $B'' \neq \emptyset$.

The deactivation of an active edge will often leave the value of τ unchanged but will eliminate some joins and therefore decrease the value of ν. On the other hand, the contraction of an active edge will usually not change the value of ν but may eliminate all minimum cuts and thus increase the value of τ. Thus, these operations have no chance of preserving Woodall's min-max in general.

To circumvent this difficulty, we shall resort to a dirty trick: restrict attention to a class of graaphs that is closed under contraction and deactivation *by definition*.

Good graaphs. A capacity function c for (G, A)[31] is **bad** if the capacitated graaph (G, A, c) fails to satisfy Woodall's min-max. The graaph (G, A) is **good** if it does not have a bad capacity,[32] i.e., if every capacitated graaph of the form (G, A, c) satisfies Woodall's min-max.[33]

For example, it is known [Sch03, cap.56] that if G is a DAG with only one source then (G, E_G) is good (in particular, (G, A) is good for any A).

A graaph (G, A) is **bad** if it is not good, i.e., if there exists a bad capacity for (G, A). For example, Schrijver's graaph (G_1, A_1) (figure 6) is bad. A bad capacity for this graaph is defined by $c(e) = 1$ if $a \in A_1$ and $c(e) = 0$ otherwise. The Cornuéjols and Guenin graaphs (G_2, A_2) and (G_3, A_3) (figures 10 e 11) are also bad.

The class of good graaphs is trivially closed under deactivation and contraction of edges. (To show that the deactivation of an edge a in a good graaph produces another good graaph, we can "simulate" the deactivation of a by setting $c(a)$ to 0. To show that the contraction of a produces another good graaph, we "simulate" the contraction by setting $c(a)$ to $1 + \tau(G, A, c)$.) It follows that any graaph that can be reduced by contractions and deactivations to a bad graaph is bad.

At this point, it is convenient to introduce one more reduction operation: the contraction of a directed circuit. Let O be a directed circuit in (G, A) (some of the edges of O

to (G', A') and e and e' are corresponding edges of G and G' then the result of the deactivation of e in G is isomorphic to the result of the deactivation of e' in G'. Analogous observations hold for contractions and for the transposition.

[31] Remember that we assume $c(e) = 0$ for all e not in A.

[32] By adopting this definition of a good graaph I am loosing touch with computational reality, since I do not have a polynomial algorithm for deciding whether a given capacitated graaph satisfies Woodall's min-max.

[33] Cornuéjols–Guenin–Margot [CGM00, Cor01] conjecture that the good nature of a graaph depends only on capacity functions having values in $\{0, 1, \infty\}$. Specifically, they conjecture the following: if a graaph (G, A) and all its deactivations/contractions satisfy Woodall's min-max then (G, A) is good.

may be in A and some others out of A). The **contraction of** O produces the graaph $(G/E_O, A \setminus E_O)$, where E_O is the set of edges of O. It is easy to verify that the class of good graaphs is closed under the contraction of directed circuits.

A bad graaph (G, A) is **minimally bad** if every reduction operation — i.e., every deactivation of an active edge, contraction of an active edge, or contraction of a directed circuit — produces a good graaph. For example, Schrijver's graaph and the two Cornuéjols–Guenin graaphs are minimally bad.

If the set of all minimally bad graaphs turns out to be easy to describe (for example, if it turns out to be small), we shall have a useful (if not algorithmic) characterization of the class of good graaphs.

15 Properties of minimally bad graaphs

We describe here some properties of minimally bad graaphs. The first two are very simple. The second, in particular, is a mere play of words. Propriedade 10 is the most interesting.

Property 5 *Every minimally bad graaph is a* DAG.

PROOF: Let O be a directed circuit in a graaph (G, A). Let (G', A') be the result of contracting O. Of course (G', A') is good if and only if (G, A) is good. Hence, (G, A) is not minimally bad. □

Property 6 *For any bad capacity c of a minimally bad graaph (G, A) one has $c(a) \geq 1$ for all a in A.*

PROOF: Suppose for a moment that $c(a) = 0$ for some a in A. By virtue of minimality, the graaph $(G, A \setminus \{a\})$ is good. Hence, $\nu(G, A \setminus \{a\}, c) = \tau(G, A \setminus \{a\}, c)$. But $\nu(G, A, c) = \nu(G, A \setminus \{a\}, c)$ and $\tau(G, A, c) = \tau(G, A \setminus \{a\}, c)$. Therefore, $\nu(G, A, c) = \tau(G, A, c)$, and that is a contradiction. □

Property 7 *For any bad capacity c of a minimally bad graaph (G, A) one has $c(a) \leq \tau(G, A, c)$ for all a in A.*

PROOF: Suppose for a moment that $c(a) > \tau(G, A, c)$ for some a in A. Let (G', A') be the result of contracting a. By virtue of minimality, (G', A') is good, whence $\nu(G', A', c') = \tau(G', A', c')$, where c' is the restriction of c to $E_G \setminus \{a\}$. Let $\nu' \equiv \nu(G', A', c')$, $\tau' \equiv \tau(G', A', c')$, $\nu \equiv \nu(G, A, c)$ and $\tau \equiv \tau(G, A, c)$. It is clear that

$\nu' \geq \nu$ and $\tau' \geq \tau$. Since no minimum cut in (G, A) contains a, we have

$$\tau' = \tau \,.$$

We will show next that $\nu' = \nu$. Let $\mathcal{D}'$ be a c'-disjoint collection of joins in (G', A') such that $|\mathcal{D}'| = \nu'$. Let $\mathcal{D}'_a$ be the collection $\{J \cup \{a\} : J \in \mathcal{D}'\}$. Of course $\mathcal{D}'_a$ is a collection of joins in (G, A). Since $|\mathcal{D}'_a| = \nu' \leq \tau' = \tau < c(a)$, the collection $\mathcal{D}'_a$ does not use edge a more than $c(a)$ times. Hence, $\mathcal{D}'_a$ is c-disjoin in (G, A), and so $\nu \geq |\mathcal{D}'_a| = \nu'$. Therefore,

$$\nu' = \nu \,.$$

We conclude that $\nu = \tau$. That is a contradiction, since c is bad. This contradiction shows that, in fact, $c(a) \leq \tau$ for every edge a. □

Property 8 *For any minimally bad graaph (G, A) and any bad capacity c for (G, A), all the minimum cuts are peripheral.*

PROOF: Suppose some minimum cut C is not peripheral. Let X be a source-set such that $\nabla^+(X) = C$. Of course $1 < |X| < |V_G|$. Now contract all edges having both ends in X. The resulting graaph is good. Repeat the procedure for $Y \equiv V_G \setminus X$. A standard construction will produce a c-disjoint collection of joins in (G, A) whose size is $|C|$. □

Property 9 *In any minimally bad graaph (G, A) and for any bad capacity c, each edge in A belongs to some minimum cut.*

PROOF: Suppose for a moment that an edge $a \in A$ belongs to no minimum cut. Then $\tau(G, A) = \tau(G, A')$, where $A' \equiv A \setminus \{a\}$. Let $\mathcal{D}'$ be a maximum disjoint collection of joins in (G, A'). Since (G, A) is minimal, $|\mathcal{D}'| = \tau(G, A')$. Since each element of $\mathcal{D}'$ is a join in (G, A), we have $\nu(G, A) \geq |\mathcal{D}'| = \tau(G, A') = \tau(G, A)$. Hence, (G, A, c) satisfies Woodall's min-max, and that is a contradiction, since c is bad. □

Property 10 *For any minimally bad graaph (G, A), the active subgraph $G[A]$ is a forest (if we ignore the orientation of the edges).*

PROOF: For any two different capacity functions c and d, there exists a natural number j such that $|c^{-1}(i)| = |d^{-1}(i)|$ for $i = 0, \ldots, j-1$ but $|c^{-1}(j)| \neq |d^{-1}(j)|$. We say that c is **lexicographically smaller** than d if

$$|c^{-1}(j)| > |d^{-1}(j)| \,.^{34}$$

[34] This could be restated as folows. Let $e_1, \ldots, e_m$ be the edges of the graph and let $\mathrm{ord}(c(e_1), \ldots, c(e_m))$ be a permutation of $(c(e_1), \ldots, c(e_m))$ in increasing order. Then the function c

It is not difficult to verify that there is no infinite sequence $(c^1, c^2, c^3, \ldots)$ of functions such that c^{i+1} is lexicographically smaller than c^i for arch i. Among all the bad capacity functions for (G, A), choose a lexicographically smallest; let c be the chosen function.

Suppose now, for a moment, that $G[A]$ has a (not necessarily directed) circuit O. Let k be the minimum value of c in O. Adjust the notation so that a reverse-directed edge of O has capacity k. Let c' be the capacity defined as follows:

$$c'(a) = \begin{array}{ll} c(a) - k & \text{if } a \text{ is a reverse-directed edge of } O \\ c(a) + k & \text{if } a \text{ is a forward-directed edge of } O \\ c(a) & \text{otherwise .} \end{array}$$

Observe that $c'(C) = c(C)$ for every cut C, whence

$$\tau(G, A, c') = \tau(G, A, c) \ .$$

Let $\nu(d) \equiv \nu(G, A, d)$ and $\tau(d) \equiv \tau(G, A, d)$ for all d. We can then write $\tau(c') = \tau(c)$.

Since $c'(a) = 0$ for some edge a of O, c' is not bad for (G, A), according to property 6. Hence,

$$\nu(c') = \tau(c') \ .$$

Let $\mathcal{D}'$ be a c'-disjoint multicollection of joins such that $|\mathcal{D}'| = \tau(c')$. Let J' be an element of $\mathcal{D}'$ and define capacity c'' as follows:

$$c''(a) = \begin{array}{ll} c(a) - 1 & \text{if } a \in J' \text{ and} \\ c(a) & \text{otherwise.} \end{array}$$

Of course $c''(C) = |\mathcal{D}' \smallsetminus \{J'\}| = \tau(c') - 1 = \tau(c) - 1$ for each cut C, whence

$$\tau(c'') = \tau(c) - 1 \ .$$

Since c'' is lexicographically smaller than c, the capacitated graaph (G, A, c'') satisfies Woodall's min-max

$$\nu(c'') = \tau(c'') \ .$$

Let $\mathcal{D}''$ be a c''-disjoint multicollection of joins such that $|\mathcal{D}''| = \tau(c'')$. Since $\{J'\} \cup \mathcal{D}''$ is a c-disjoint collection of joins,

$$\begin{aligned} \nu(c) &\geq \left|\mathcal{D}'' \cup \{J'\}\right| \\ &= \left|\mathcal{D}''\right| + 1 \\ &= \tau(c'') + 1 \\ &= \tau(c) - 1 + 1 \\ &= \tau(c) \ . \end{aligned}$$

is lexicographically smaller than the function d if and only if the sequence $\text{ord}(c(e_1), \ldots, c(e_m))$ is lexicographically smaller, in the usual sense, than the sequence $\text{ord}(d(e_1), \ldots, d(e_m))$.

Therefore, $\nu(c) = \tau(c)$. But this is a contradiction. Hence, $G[A]$ has no circuits. □

Suppose (G, A) is a minimally bad graaph. According to properties 5, 8, and 10, A has a (unique) partition $\mathcal{A}$ such that, for each A' in $\mathcal{A}$,

- $G[A']$ is a tree,[35]
- the internal vertices of $G[A']$ are minimum sources or sinks.[36]

Moreover, by virtue of property 9, $|A'| \geq 2$ for all A'. In the three known counterexamples, the structure of the active forest is particularly simple: for each A' in $\mathcal{A}$, the graph $G[A']$ is a component of $G[A]$ (edge orientations ignored).

16 More reductions

Williams observed [Wil04, WG05] that the set of "forbidden graaphs" — i.e., minimally bad graaphs defined by the operations of deactivation and contration — seems to be large and complex. In particular, it contains several simple variations of Schrijver's graaph and the Cornuéjols–Guenin graaphs. See, for example, figure 15.

Figure 15: Thsi graaph is bad ($\nu = 1 < 2 = \tau$). Contract edges e, f, g, h, then deactivate edge x. Now the graaph is almost equal to Schrijver's (G_1, A_1). To make it equal, all we need to do it delete the parallel edges y and z. This suggests adding a nwew operation to the repertoir of reductions.

In an attempt to simplify the set os minimally bad graaphs, Williams invented [Wil04, WG05] new reduction operations, that transform a good graaph (G, A) into a smaller good graaph:

1. *deletion of a transitive edge:* if $uv \notin A$ and there exists a directed paths form u to v in $G - uv$ then change (G, A) into $(G - uv, A \setminus \{uv\})$;

2. *biclique substitution:* suppose no edge incident to a vertex r is in A; let B be the set of all the pairs (u, v) such that $ur \in E_G$ and $rv \in E_G$; change (G, A) into $((G - r) + B, A)$;

3. *folding:* this is a rather complex operation; we shall not attempt to describe it.

[35] But $G[A']$ is not necessarily a component of $G[A]$ (edge orientations ignored).

[36] In other words, if x in an internal vertex the either x is a source and $\nabla^+(\{x\}) = \tau(G, A)$ or x is a sink and $\nabla^-(\{x\}) = \tau(G, A)$.

These new reductions preserve the good character of a graaph. Hence, they can be added to the set of reductions used so far. When this is done, we have a new definition of **minimally bad graaph**. Properties 5, 6, 7, 8, 9, and 10 remain valid.

With the new reductions, many of the bad graaphs are no longer minimally bad. However, certain simple variations of counterexample (G_2, A_2) resist the reductions. These variations are obtained by eliminationg some inactive edges (see figure 16). Thsi produces a family $(G_2^1, A_2^1), \ldots, (G_2^8, A_2^8)$ of minialy bad graaphs. To make notation uniform, we shall denote the graaph (G_2, A_2). by (G_2^0, A_2^0). Someting similar happens with (G_3, A_3) (see figure 16).

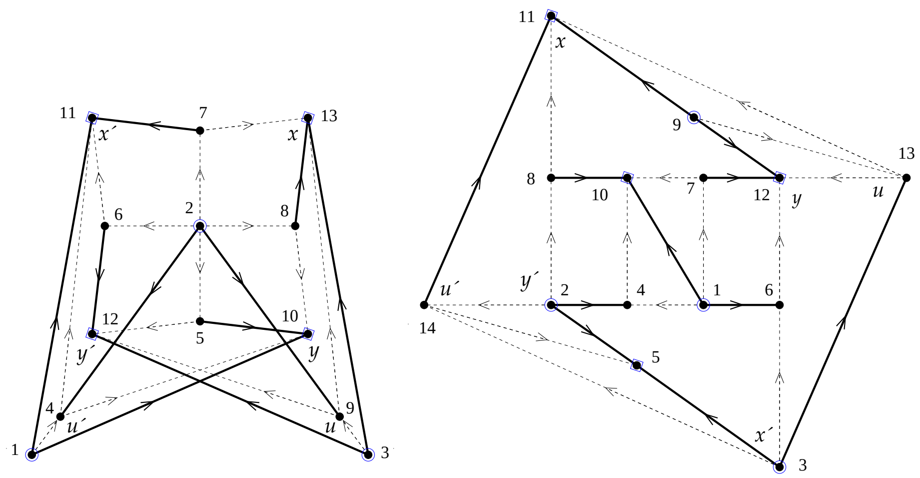


Figure 16: If we delete no more than one of the edges ux and uy and (independently) no more than one of $u'x'$ and $u'y'$, we obtain a new minimally bad graaph. These new graaphs will be denoted by $(G_2^1, A_2^1), \ldots, (G_2^8, A_2^8)$ and $(G_3^1, A_3^1), \ldots, (G_3^8, A_3^8)$.

17 Conterexamples with $\tau = 2$

With the new set of reductions (and the definition of minimally bad graaphs that follows from it), Williams studied [Wil04] graaphs with $\tau = 2$. Minimally bad graaphs of this type satisfy properties 5, 6, 7, 8, 9, and 10. In particular, the active subgraph $G[A]$ is a forest and A admits a (unique) partition $\mathcal{A}$ such that, for each A' in $\mathcal{A}$,

$$G[A'] \text{ is a path}^{37}$$

with two or more edges whose internal vertices are minimum sources or sinks. We say that each $G[A']$ is an **active path**. Williams has shown that

[37] But $G[A']$ is not necessarily a componente of $G[A]$ (orientations of the edges ignored).

Teorema 11 (Williams) *The only minimally bad graaphs with* $\tau = 2$ *and three active paths are* (G_1, A_1), (G_2^i, A_2^i) *and* (G_3^i, A_3^i), *with* $i = 0, \ldots, 8$.

Teorema 12 (Williams) *There are no minimally bad graaphs with* $\tau = 2$ *and four active paths.*

18 The significance of the case $\tau = 2$

The class of graaphs for which $\tau = 2$ seem to be important. Cornuéjols–Guenin–Margot [CGM00, Cor01] conjecture[38] that

$$\tau(G, A) = 2$$

(e $\nu(G, A) = 1$) for any minimaly bad graaph (G, A).

References

[CG02] G. Cornuéjols and B. Guenin. Note on dijoins. *Discrete Mathematics*, 243:213–216, 2002. 10

[CGM00] G. Cornuéjols, B. Guenin, and F. Margot. The packing property. *Mathematical Programming (Ser. A)*, 89:113–126, 2000. 1, 11, 16, 22

[Cor01] G. Cornuéjols. *Combinatorial Optimization: Packing and Covering*, volume 74 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM (Society for Industrial and Applied Mathematics), 2001. 16, 22

[Edm73] J. Edmonds. Edge-disjoint branchings. In R. Rustin, editor, *Combinatorial Algorithms*, volume 9 of *Courant Computer Science Symposium*, pages 91–96. Algorithmics Press, 1973. 7

[EG77] J. Edmonds and R. Giles. A min-max relation for submodular functions on graphs. In P.L. Hammer et al., editors, *Studies in Integer Programming*, volume 1 of *Annals of Discrete Mathematics*, pages 185–204. North-Holland, 1977. 6

[FY87] P. Feofiloff and D.H. Younger. Directed cut transversal packing for source-sink connected graphs. *Combinatorica*, 7(3):255–263, 1987. 7

[LY78] C.L. Lucchesi and D.H. Younger. A minimax theorem for directed graphs. *J. of the London Math. Soc. (2)*, 17:369–374, 1978. 4

[38] The conjecture is rather general: it applies to all ideal minimally bad clutters.

[Sch80] A. Schrijver. A counterexample to a conjecture of Edmonds and Giles. *Discrete Math.*, 32:213–214, 1980. 8

[Sch03] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, 2003. [Three volumes]. 1, 4, 16

[Sey77] P.D. Seymour. The matroids with max-flow min-cut property. *J. of Combinatorial Theory (Ser. B)*, 23:189–222, 1977. 11

[SV05] F.B. Shepherd and A. Vetta. Visualizing, finding and packing dijoins. In D. Avis, A. Hertz, and O. Marcotte, editors, *Graph Theory and Combinatorial Optimization*, chapter 8, pages 219–254. Springer, 2005. 7

[WG05] A.M. Williams and B. Guenin. Advances in packing directed joins. *Electronic Notes in Discrete Mathematics*, 19:249–255, 2005. [http://www.sciencedirect.com/science/journal/15710653] [Proceedings of the 2nd Brazilian Symposium on Graphs, Algorithms, and Combinatorics (GRACO 2005)]. 12, 20

[Wil04] A.M. Williams. Packing directed joins. Master's thesis, University of Waterloo, 2004. Supervisor: Bertrand Guenin. 1, 12, 20, 21

[Woo78a] D.R. Woodall. Menger and König systems. In Y. Alavi and D.R. Lick, editors, *Theory and Applications of Graphs*, volume 642 of *Lecture Notes in Mathematics*, pages 620–635. Springer, 1978. 4

[Woo78b] D.R. Woodall. Minimax theorems in graph theory. In L.W. Beineke and R.J. Wilson, editors, *Selected Topics in Graph Theory*, pages 237–269. Academic Press, 1978. 4