| | |
|---|---|
| **18.997 Topics in Combinatorial Optimization** | May 13th, 2004 |

Lecture 23

*Lecturer: Michel X. Goemans* *Scribe: Dan Stratila*

Consider a planar graph $G = (V, E)$ and a set of terminal pairs $R = \{(s_i, t_i) : i = \overline{1, k}\}$. Assume $G$ is planar, $(V, E \cup R)$ is Eulerian, and all terminals lie on the outer face of $G$. In this lecture, we will cover the following results.

- The Okamura-Seymour theorem on the equivalence between the existence of $s_i$-$t_i$ edge-disjoint paths and the cut condition $|\delta_E(S)| \leq |\delta_R(S)|, \forall S \subseteq V$ [OS81].
- The Wagner-Weihe linear-time algorithm for finding the edge-disjoint paths [WW93, RLWW95, WW95].

Chapter 74 of [Sch03] contains a proof of the Okamura-Seymour theorem, as well as a survey of related results.

# 1 The Okamura-Seymour Theorem

We begin with the main theorem.

**Theorem 1 (Okamura-Seymour)** *Consider an undirected planar graph $G = (V, E)$ and a set of terminal pairs $R = \{(s_i, t_i) : s_i \in V, t_i \in V, i = \overline{1, k}\}$ s.t. the following conditions are satisfied:*

1. *The terminals are on the boundary of the outside face of $G$.*[1]
2. *The Euler condition: $(V, E \cup R)$ is Eulerian.*
3. *The cut condition: $|\delta_E(S)| \leq |\delta_R(S)|, \forall S \subseteq V$.*

*Then there exist edge disjoint paths between $s_i$ and $t_i, i = \overline{1, k}$.*

Note that since given 1 and 2, the necessity of 3 is obvious, the Theorem can also be stated as an equivalence condition, as is done, for example, in [Sch03].

To prove Theorem 1, we will use two lemmas. First, we show it suffices to consider cuts that result in connected subgraphs, and then we reduce the problem to the 2-connected case. As before, let $d_E(S) := |\delta_E(S)|$.

**Lemma 2** *For any edge-disjoint path problem, $d_E(S) \geq d_R(S), \forall \emptyset \neq S \subsetneq V$ if and only if $d_E(S) \geq d_R(S), \forall \emptyset \neq S \subsetneq V$ s.t. $G[S]$ and $G[V \setminus S]$ are connected.*

**Proof:** W.l.o.g. assume that $G[S]$ is disconnected and consists of connected components $G[S_1]$ and $G[S_2]$. If $d_E(S) < d_R(S)$, then

$$d_E(S_1) + d_E(S_2) \overset{(*)}{=} d_E(S) < d_R(S) \overset{(**)}{\leq} d_R(S_1) + d_R(S_2), \tag{1}$$

where $(*)$ holds since $G[S]$ is disconnected, and $(**)$ holds since $\delta(S) \subseteq \delta(S_1) \cup \delta(S_2)$. Since $d(\cdot) \geq 0$, this implies $d_E(S_1) < d_R(S_1)$ or $d_E(S_2) < d_R(S_2)$. $\square$

[1] We will assume there is a fixed planar embedding associated with $G$.

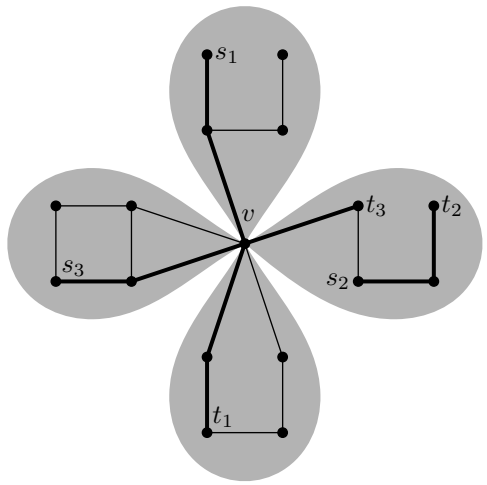


Figure 1: A 1-connected graph separated by cut vertex $v$ into 4 smaller components. Path edges are in bold; the original requirement set is $R = \{(s_1, t_1), (s_2, t_2), (s_3, t_3)\}$, and the new requirement set is $R' = \{(s_1, v), (t_1, v), (s_2, t_2), (s_3, v), (t_3, v)\}$.

**Lemma 3** *We can assume w.l.o.g. that $G$ is 2-connected.*

**Proof:** If $G$ is not connected, we can inductively reduce the problem to a set of smaller subproblems, one for each connected component. If $G$ is 1-connected, then there exists a vertex $v$ s.t. $G - v$ is disconnected.

For any $i$, if $s_i$ and $t_i$ are in the same component of $G - v$, then any solution can be transformed into a solution where the $s_i$-$t_i$ path is entirely contained in that component. If $s_i$ and $t_i$ are in different components, then $v$ will belong to any $s_i$-$t_i$ path, and replacing $R$ with

$$R \setminus \{(s_i, t_i)\} \cup \{(s_i, v), (t_i, v)\} \tag{2}$$

yields an equivalent instance (see Figure 1).

Repeating this for all $i$ we reduce the problem to a set of smaller problems with terminals on the boundary and that satisfy the Euler condition. Moreover, each problem satisfies the (induced) cut condition if and only the original problem does. By induction, we assume that each subproblem has a solution to the edge-disjoint paths problem if the cut condition is satisfied. Since we can reconstruct the solution to the original problem from the subproblem solutions, it follows that the original problem has a solution to the edge-disjoint paths problem if the cut condition is satisfied. □

We can also begin with the counterexample construction, and show that a $(2|E| - |R|)$-minimal counterexample needs to be 2-connected, as is done in [Sch03].

**Outline of proof of Theorem 1:**

1. We take a minimal (according to some criterion) counterexample.
2. We show that properties (3, 4, 5) hold for the counterexample.
3. We take a cardinality minimal tight set $X \subseteq V$, and using (3, 4, 5) derive the final contradiction.

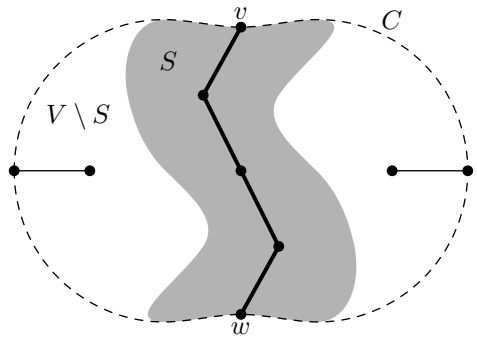


Figure 2: Example of set $S$ that contains non-consecutive vertices of $C$, and thus disconnects $G[V \setminus S]$. The path $P$ is in bold.

**Proof of Theorem 1:** Suppose the theorem is not true, and consider the set of edge-minimal counterexamples; from this set, select a counterexample with as many terminals as possible. Let $C$ be the simple cycle along the border of the outside face.

Suppose there is pair s.t. $s_i$ and $t_i$ are adjacent in $E$, then we can remove both the edge from $E$ and the pair from $R$. The resulting instance satisfies the Euler condition, since we deleted two parallel edges from $(V, E \cup R)$; it also satisfies the cut condition, since any cut either crosses both the edge and $(s_i, t_i)$ or neither. However, since a feasible solution to the modified instance yields a feasible solution to the original instance, the modified instance has no feasible solution., and is thus a counterexample. This contradicts the minimality of the original counterexample, hence

$$\text{there is no pair with } (s_i, t_i) \in E. \tag{3}$$

We will call a set $S$ tight if $\delta_E(S) = \delta_R(S)$. Suppose that there is no tight set, and remove some edge $e = (u, v) \in E(C)$ and add $(u, v)$ to $R$. Let the corresponding sets of the new instance be $E'$ and $R'$. Fix a set $S \subset V$, and note that $\delta_{E'}(S) - \delta_{R'}(S)$, and $\delta_E(S) - \delta_R(S) \geq 0$ are both even. Therefore the cut condition can only be violated for $S$ in the new instance if $S$ was tight in the original instance. It follows that the new instance satisfies the cut condition, which contradicts the minimality of the counterexample (again because we can translate a feasible solution to the new instance into one for the original instance), thus

$$\text{the original instance contains at least one tight set.} \tag{4}$$

Next, we show that since $G[S]$ and $G[V \setminus S]$ are connected, both $S$ and $V \setminus S$ cannot contain non-consecutive elements of $V(C)$. To see this, suppose w.l.o.g. that $S$ contains vertices $v, w \in V(C)$ s.t. for either side of $C$ between $v$ and $w$, not all vertices between $v$ and $w$ are in $S$. Since $G[S]$ is connected, there is a path $P$ from $v$ to $w$ in $G[S]$. Since $G$ is planar and $v, w$ belong to the outside face, this path implies $G[V \setminus S]$ is disconnected (see Figure 2). This is a contradiction, thus

$$\text{both } S \text{ and } V \setminus S \text{ contain consecutive elements of } V(C), \tag{5}$$

or in other words $|\delta_E(S) \cap E(C)| = 2$.

Take a $|X|$-minimal tight set $X$, let $(w, u) \in E(C)$ with $w \in X, u \notin X$, and select a terminal pair $(s_r, t_r)$ s.t. $s_r \in X, t_r \notin X$, and $t_r$ is closest (w.r.t the number of edges in a

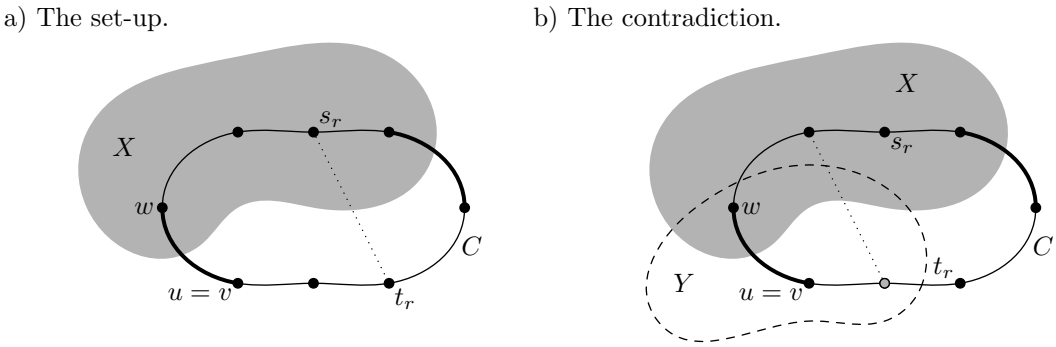


Figure 3: The counterexample construction for Theorem 1. Edges crossing the cut $X$ are in bold. Demand pairs are denoted by dotted lines, and the node yielding the contradiction is drawn hollow.

path) to $u$. By (3), we can take $v \in \{w, u\} \setminus \{s_r, t_r\}$ (see Figure 3.a)). Consider a new set of demand pairs

$$R' = R \setminus \{(s_r, t_r)\} \cup \{(s_r, v), (t_r, v)\}, \tag{6}$$

and note that the terminals are still on the boundary of the outside face, and the Euler condition is still satisfied, because in $(V, E \cup R)$ the degree of $v$ increased by 2, while all other degrees remained unchanged. Therefore, the cut condition must be violated for $R'$, otherwise it would contradict the minimality of the counterexample.

This implies we can pick a violating set $Y, v \in Y$, which immediately implies by construction of $R'$, $s_r \notin Y, t_r \notin Y$, hence $v \notin \{s_r, t_r\}$ at all. Moreover, $d_E(Y) < d_{R'}(Y)$ implies $d_E(Y) = d_R(Y)$, i.e. $Y$ was tight. Now, note that

$$\begin{aligned} d_R(X \cap Y) + d_R(X \cup Y) &\overset{\text{(i)}}{=} d_R(X) + d_R(Y) \overset{\text{(ii)}}{=} d_E(X) + d_E(Y) \\ &\overset{\text{(iii)}}{\geq} d_E(X \cap Y) + d_E(X \cup Y) \overset{\text{(iv)}}{\geq} d_R(X \cap Y) + d_R(X \cup Y), \end{aligned} \tag{7}$$

since:

(i) $d_R(X \cap Y) + d_R(X \cup Y) \leq d_R(X) + d_R(Y)$ due to submodularity, and a strict inequality can only occur if there is a demand pair from $X \setminus Y$ to $Y \setminus X$. This cannot happen, since $t_r \notin Y$ and $u \in Y$ imply the endpoint of this pair in $Y \setminus X$ would be closer to $u$, due to (5) (see Figure 3.b)).

(ii) This is simply because both $X$ and $Y$ are tight.

(iii) This is again due to submodularity of cut cardinality.

(iv) The cut condition implies $d_E(X \cap Y) \geq d_R(X \cap Y)$ and $d_E(X \cup Y) \geq d_R(X \cup Y)$.

Therefore, equality holds throughout, and since $d_E(X \cap Y) \geq d_R(X \cap Y)$ and $d_E(X \cup Y) \geq d_R(X \cup Y)$ and all quantities are nonnegative, we get $d_E(X \cap Y) = d_R(X \cap Y)$. However $|X \cap Y| < |X|$, since $s_r \in X, s_r \not\in Y$.

Unless $X \cap Y = \emptyset$, this violates the minimality of $X$ (if $X \cap Y$ is disconnected, then one of the components will be violating the cut condition, and the minimality of $X$). If $X \cap Y = \emptyset$, then $w \notin Y$, hence $v = u \in Y$. Since $w \in X$ and $u \notin X$, this implies $(u, w)$ connects $X \setminus Y$ and $Y \setminus X$, which contradicts the "sandwich" equality (iii) of (7). $\square$

## 2 The Wagner-Weihe algorithm

In this section we present the Wagner-Weihe algorithm. The algorithm either finds the $s_i$-$t_i$ edge-disjoint paths, or a proof of infeasibility in the form of a violated cut. The running time is $O(n)$, and, for our description, we assume a planar embedding and a sorted list of edges to be given. In the algorithm we will also assume, w.l.o.g., that each terminal source or destination is a separate node connected to the original node by an edge.

**Step 1: Re-pair.**

a) Choose a terminal $x \in \cup_{i=1}^{k}\{s_i, t_i\}$, and enumerate the vertices of $V(C)$ counterclockwise. When encountering a vertex in a demand pair $\{s_i, t_i\}$ for the first time, associate to it an opening parenthesis, and when encountering one for the second time, associate to it a closing parenthesis.

b) Obviously, the parenthesis in the result string are correctly nested. Each pair of matching parenthesis will define a new demand pair $\{s'_i, t'_i\}$, and we will number them in order of closing parenthesis.

   For example, for the graph in Figure 4, we renumber starting from the pair 5 terminal on the left side of the graph, and obtain (the new numbering is denoted, through a slight abuse of notation, by $1', 2', \ldots$):

| Original demand pair | 5 | 3 | 4 | 2 | 4 | 5 | 1 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| Parenthesis | ( | ( | ( | ( | ) | ) | ( | ) | ) | ) |
| New demand pair | $5'$ | $4'$ | $2'$ | $1'$ | $1'$ | $2'$ | $3'$ | $3'$ | $4'$ | $5'$ |

c) For $i = 1'$ to $k'$, proceed from $s_i$ by "right-first" search, i.e. when at a node, always take the right-most edge (clockwise) that is not already in use. If stuck in the graph, or if reached the wrong terminal, *stop*, one can find a violated cut.

**Step 2: return to original pairing.**

a) Remove edges unused in the paths from step 1, and orient the remaining edges in the direction used.

b) Enumerate the pairs of $R$ in the order of their closing parenthesis. For each pair, do "right-first search" on the resulting graph. If stuck in the graph, or if reached the wrong terminal, *stop*, one can find a violated cut.

Denote the instance obtained through re-pairing starting with terminal $x$ by $I_x$, and its requirement set by $R_x$. We will only prove the following lemma.

**Lemma 4** *The cut condition holds for the original instance $I$ if and only if it holds for any re-paired instance $I_x$.*

**Proof:** For necessity, let $S$ be a violated cut for some instance $I_x$. Recall that the number of edges is $\delta_E(S)$ is unchanged, so we evaluate $\delta_R(S)$ and $\delta_{R_x}(S)$.

W.l.o.g. we can assume that $x \notin S$, and $S$ contains consecutive nodes of $V(C)$. Then, the terminal pairs in $S$ represent a consecutive set of paranthesis in the string obtained at step 1.b). Since the parenthesis are correctly nested, the set of parenthesis corresponding

to demand pairs crossing $S$ consists of $k' \geq 0$ unmatched closing parenthesis, followed by a $k'' \geq 0$ unmatched opening parenthesis; $k' + k'' = \delta_{R_x}(S)$.

Let $S', S''$ be the terminals in $S$ before and including the last unmatched ")", and in $S$ after the last unmatched ")", respectively. Then, $S'$ contains $k'$ more ")" than "(", thus there are $k'$ pairs in $R$ crossing it, and since in $R$ every ")" is preceded by its ")", these pairs also cross $S$. Similarly, $S'$ contains $k''$ more "(" than ")", thus there are $k''$ more pairs in $R$ crossing $S$. Hence $\delta_R(S) \geq \delta_{R_x}(S) > \delta_E(S)$.

Conversely, to see sufficiency, let $S$ be a cut violated w.r.t $R$. Choose $x$ to be the first vertex in $S$. Then, every pair in $R$ that crosses $S$ will have an opening parenthesis, but not a closing one associated to it. As a result, $S$ will contain $\delta_R(S)$ more "(" than ")" in it, and thus $\delta_{R_x}(S) \geq \delta_R(S) > \delta_E(S)$. □

A successful run of the algorithm is illustrated in figures 4–7; an unsuccessful run, together with the violating cut is illustrated in figures 8–11. For the latter, the re-pairing is:

| Original demand pair | 3 | 2 | 1 | 4 | 3 | 2 | 1 | 4 |
|---|---|---|---|---|---|---|---|---|
| Parenthesis | ( | ( | ( | ( | ) | ) | ) | ) |
| New demand pair | $4'$ | $3'$ | $2'$ | $1'$ | $1'$ | $2'$ | $3'$ | $4'$ |

A violated cut can be obtained by using the path that got stuck or arrived at the wrong destination, and is illustrated in Figure 12.

## References


[OS81] Haruko Okamura and P. D. Seymour. Multicommodity flows in planar graphs. *J. Combin. Theory Ser. B*, 31(1):75–81, 1981.

[RLWW95] Heike Ripphausen-Lipa, Dorothea Wagner, and Karsten Weihe. Efficient algorithms for disjoint paths in planar graphs. In *Combinatorial optimization (New Brunswick, NJ, 1992–1993)*, volume 20 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 295–354. Amer. Math. Soc., Providence, RI, 1995.

[Sch03] Alexander Schrijver. *Combinatorial optimization. Polyhedra and efficiency. Vol. C*, volume 24 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 2003. Disjoint paths, hypergraphs, Chapters 70–83.

[WW93] Dorothea Wagner and Karsten Weihe. A linear-time algorithm for edge-disjoint paths in planar graphs. In *Algorithms—ESA '93 (Bad Honnef, 1993)*, volume 726 of *Lecture Notes in Comput. Sci.*, pages 384–395. Springer, Berlin, 1993.

[WW95] Dorothea Wagner and Karsten Weihe. A linear-time algorithm for edge-disjoint paths in planar graphs. *Combinatorica*, 15(1):135–150, 1995.

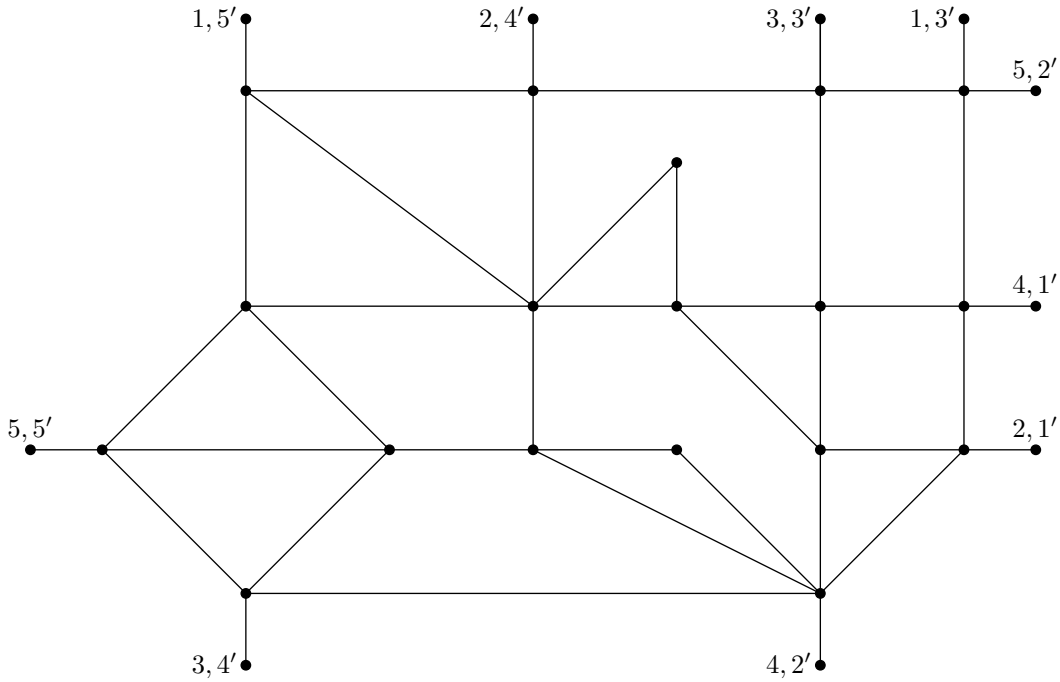


Figure 4: Initial graph for the Wagner-Weihe algorithm, together with the terminal pairs resulting from re-pairing. We will illustrate a successful run of the algorithm on this graph.

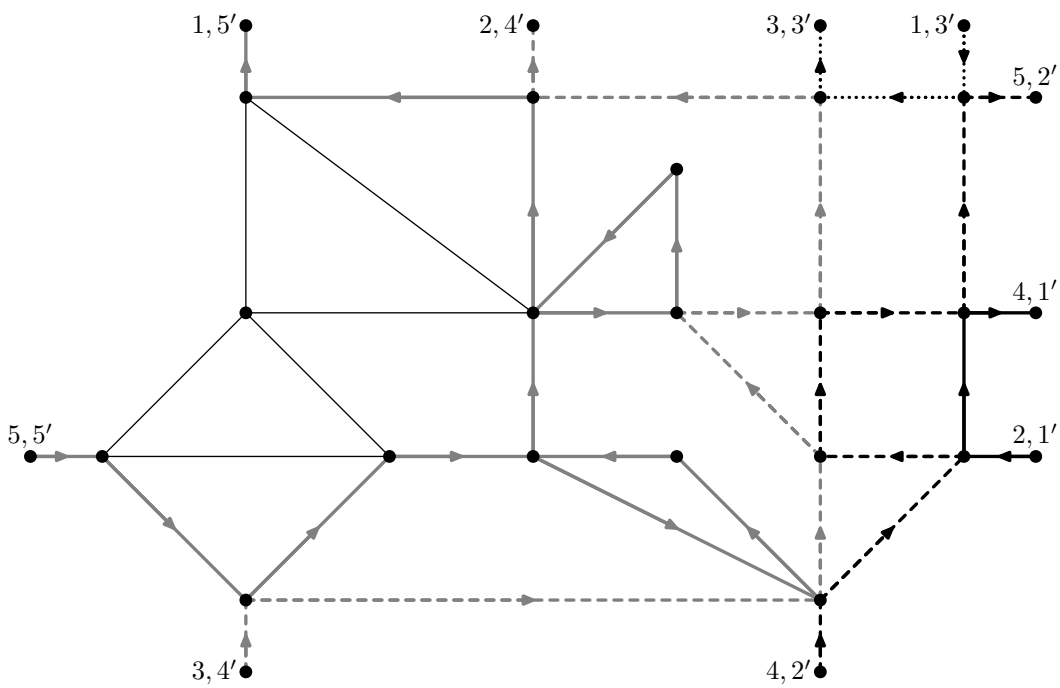


Figure 5: The results of the "right-first search" w.r.t. the re-paired terminals. Path between different terminal pairs are styled differently. Plain undirected edges are unused.

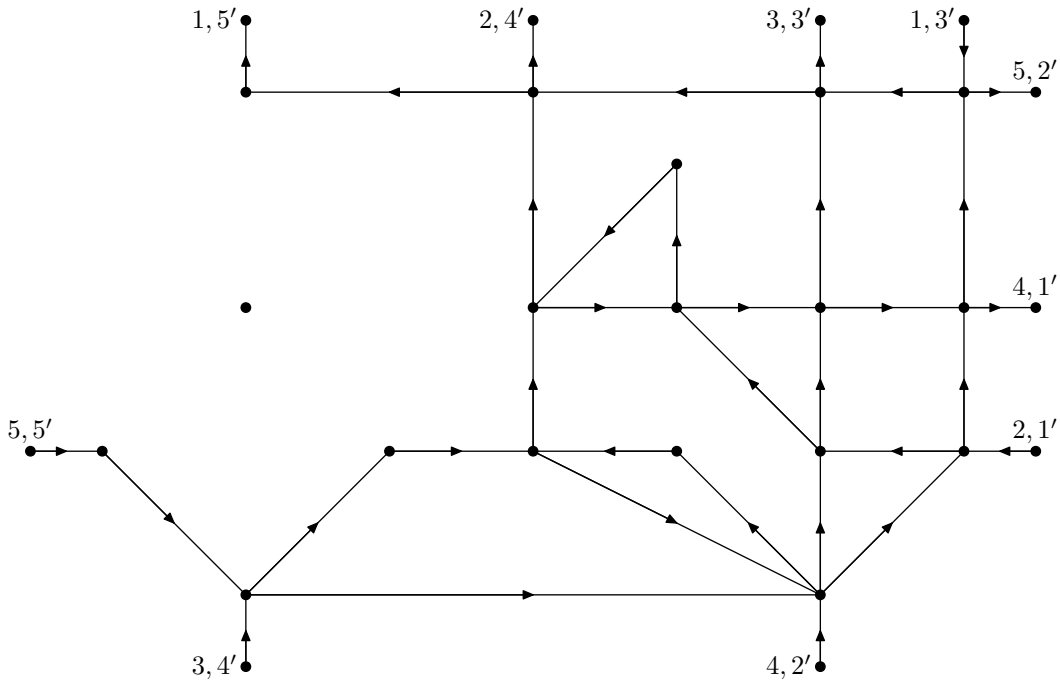


Figure 6: The resulting directed graph with the unused edges deleted.

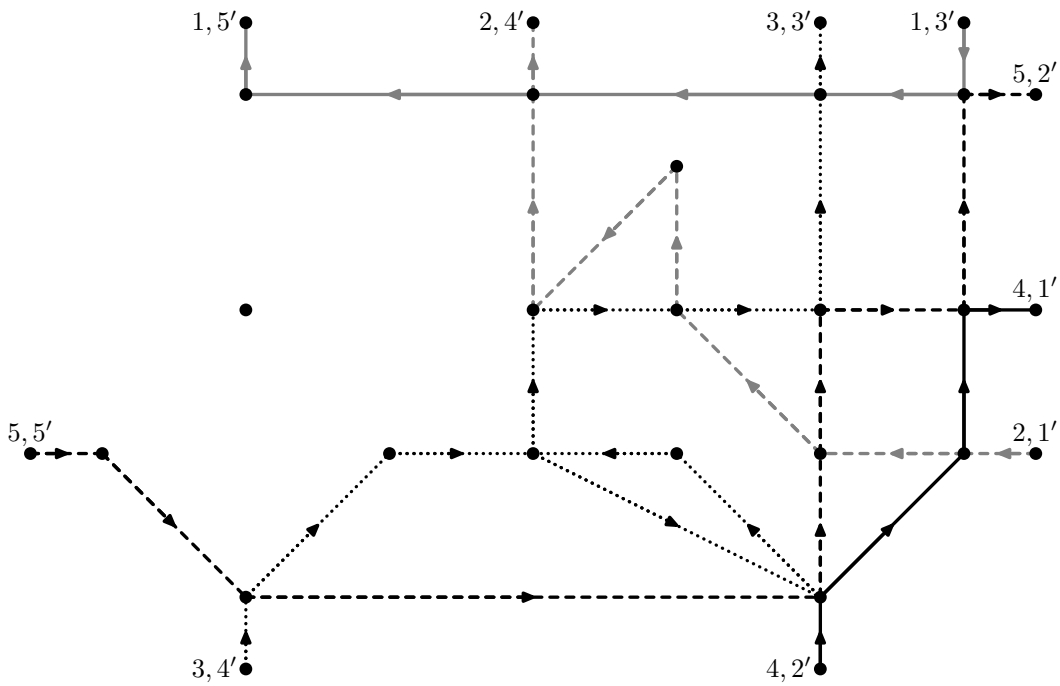


Figure 7: The results of "right-first search" w.r.t. the original terminal pairs. This is also a solution to the original disjoint paths problem. Edges not used in the final solution are omitted.

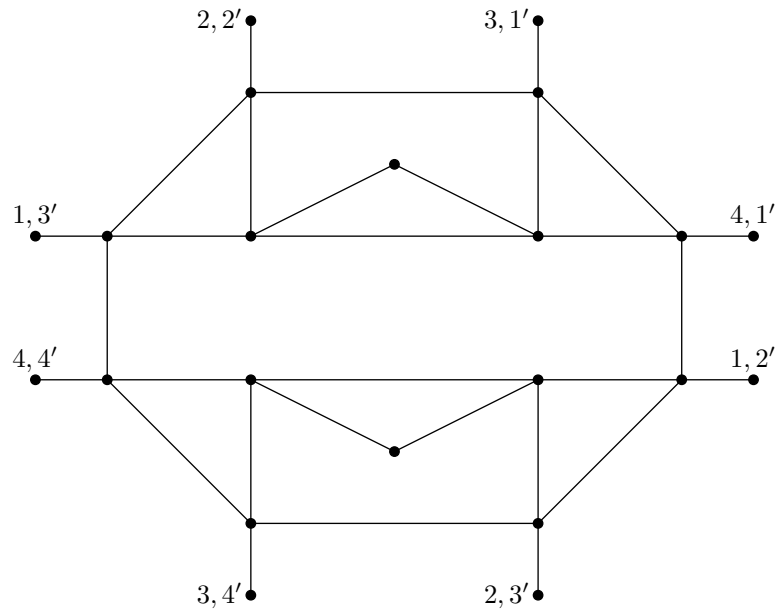


Figure 8: Initial graph for the Wagner-Weihe algorithm, together with the terminal pairs resulting from re-pairing. We will illustrate an unsuccessful run of the algorithm on this graph.

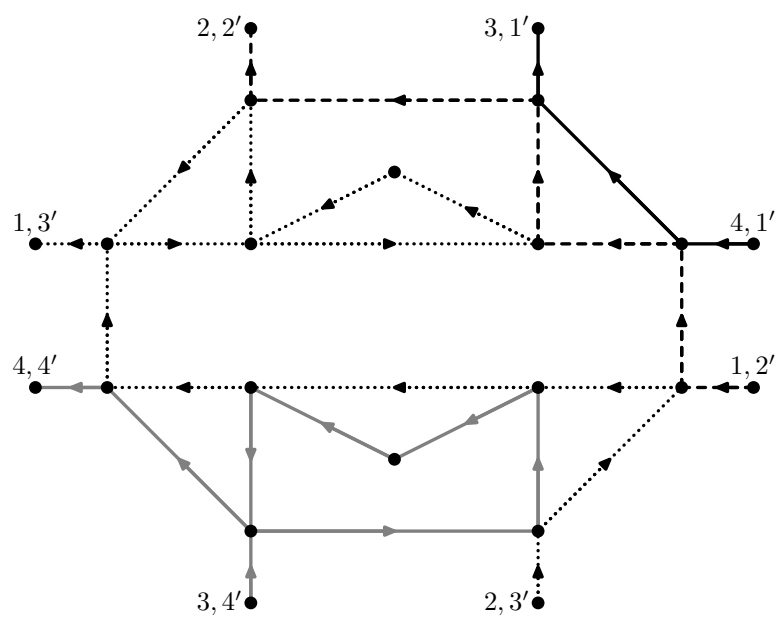


Figure 9: The results of the "right-first search" w.r.t. the re-paired terminals. Path between different terminal pairs are styled differently. Plain undirected edges are unused.

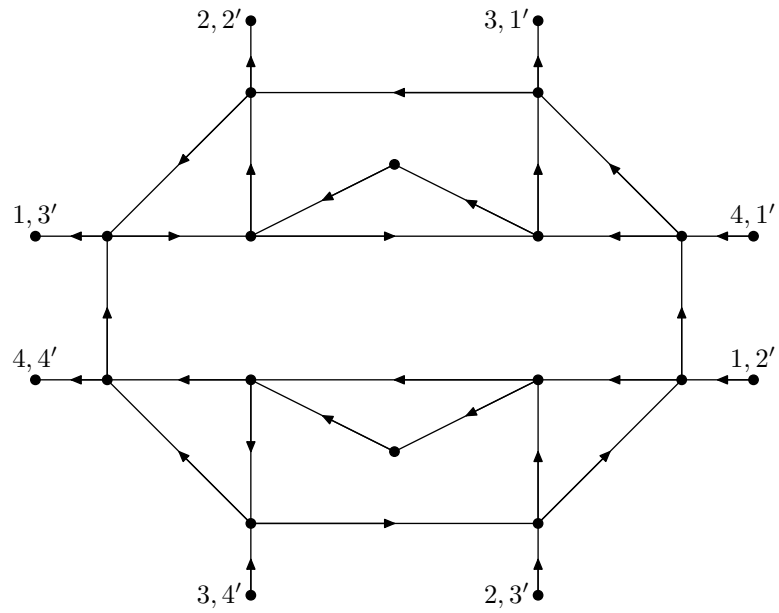


Figure 10: The resulting directed graph with the unused edges deleted.

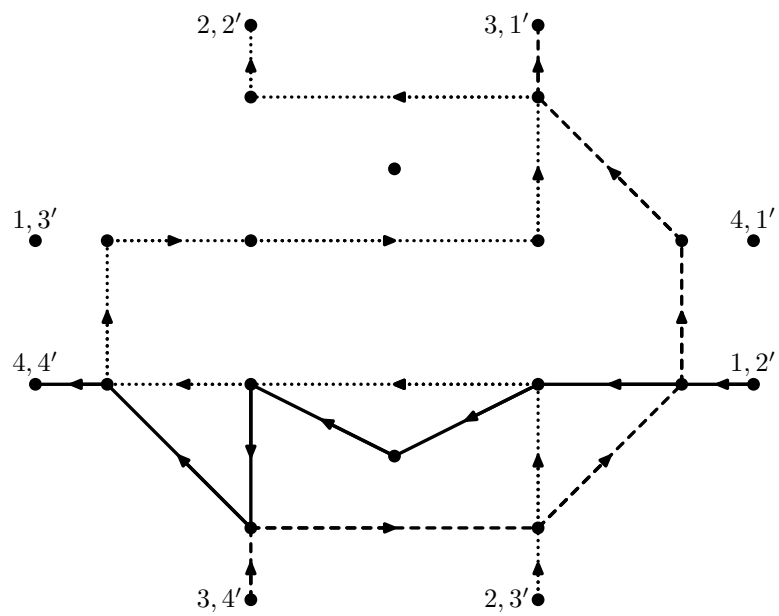


Figure 11: The results of "right-first search" w.r.t. the original terminal pairs. Note that path from source 1 arrives at 4 instead of destination 1, disconnects the graph into two parts, and thus defines a cut. Edges not used in the drawn paths are omitted.

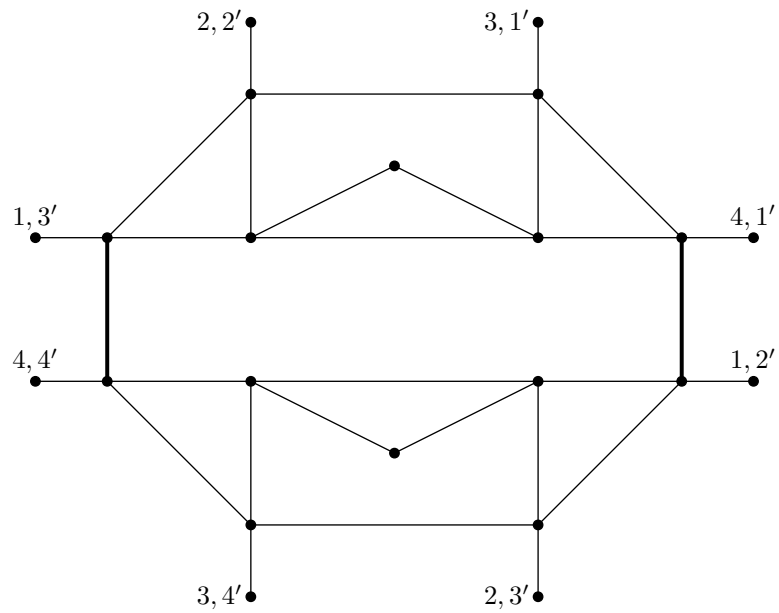


Figure 12: The path from the previous figure results in the following violated cut, with $\delta_R(S) = 4 > \delta_E(S) = 2$.