

The Two-Edge Connectivity Survivable Network Problem in Planar Graphs

Glencora Borradaile^{1,*} and Philip Klein^{2,**}

¹ Department of Combinatorics and Optimization, University of Waterloo
glencora@uwaterloo.ca

² Computer Science Department, Brown University
klein@cs.brown.edu

Abstract. Consider the following problem: given a graph with edge-weights and a subset Q of vertices, find a minimum-weight subgraph in which there are two edge-disjoint paths connecting every pair of vertices in Q . The problem is a failure-resilient analog of the Steiner tree problem, and arises in telecommunications applications. A more general formulation, also employed in telecommunications optimization, assigns a number (or *requirement*) $r_v \in \{0, 1, 2\}$ to each vertex v in the graph; for each pair u, v of vertices, the solution network is required to contain $\min\{r_u, r_v\}$ edge-disjoint u -to- v paths.

We address the problem in planar graphs, considering a popular relaxation in which the solution is allowed to use multiple copies of the input-graph edges (paying separately for each copy). The problem is SNP-hard in general graphs and NP-hard in planar graphs. We give the first polynomial-time approximation scheme in planar graphs. The running time is $O(n \log n)$.

Under the additional restriction that the requirements are in $\{0, 2\}$ for vertices on the boundary of a single face of a planar graph, we give a linear-time algorithm to find the optimal solution.

1 Introduction

In the field of telecommunications network design, an important requirement of networks is resilience to link failures [19]. The goal of the survivable network problem is to find a graph that provides multiple routes between pairs of terminals. In this work we focus on edge-disjoint paths, though vertex-disjoint paths have also been the subject of research. More formally for Z a set of non-negative integers, the input to the Z -edge connectivity problem is a weighted, undirected graph G and an assignment of *connectivity requirements* $r_v \in Z$ to vertices v . The goal is to find a minimum-weight subgraph such that, for each pair u, v of vertices, the subgraph contains at least $\min\{r_u, r_v\}$ edge-disjoint u -to- v paths. Because it is considered unlikely that two links would fail simultaneously, some research has focused on requiring at most two paths between vertices that need

* Work done while at Brown University.

** Supported by NSF grant CCF-0635089. Work done while visiting MIT.

to be connected. There is a wealth of literature on such *low-connectivity network design problems*. Resende and Pardalos [19] survey the literature, which includes heuristics, structural results, polyhedral results, computational results using cutting planes, and approximation algorithms.

This work focuses on $\{0, 1, 2\}$ -edge connectivity. We consider the well-studied relaxation wherein the solution subgraph is allowed to contain multiple copies of each edge of the input graph. We call such a subgraph a *sub-multigraph* and the weight of the edges appearing twice in the solution is counted according to multiplicity. For two-connectivity, at most two copies of an edge are needed. This version of the problem, like the other variants, is SNP-hard in general graphs [6]. In [3], Berger and Grigni gave a polynomial-time approximation scheme (PTAS) for $\{1, 2\}$ -edge connectivity (ie. the spanning case) in planar multigraphs. A year later, a PTAS was given for $\{0, 1\}$ -edge connectivity (that is, the Steiner tree problem) in planar graphs. Here we give a PTAS for the $\{0, 1, 2\}$ -edge connectivity (ie. the subset case) for planar multigraphs. The running time is significantly lower than that of [3]. In the following, OPT denotes the weight of the optimal solution to the problem at hand.

Theorem 1. *Let G be a planar graph with nonnegative edge-weights and integer requirements $r_v \in \{0, 1, 2\}$ for each vertex v . For any $0 < \epsilon < 1$, there is an $O(n \log n)$ algorithm that finds a sub-multigraph H of G such that for every pair u, v of vertices, there are at least $\min\{r_u, r_v\}$ edge-disjoint u -to- v paths in H . Further, the total weight of the edges in H is at most $(1 + \epsilon)\text{OPT}$.*

An important special case involves finding a sub-multigraph that achieves two-edge connectivity between a given set Q of vertices. Our approximation scheme addresses this problem (i.e. $r_v = 2$ for all $v \in Q$ and $r_v = 0$ for all $v \notin Q$). In addition, for the special case where the vertices of Q are on the boundary of a common face, we give a linear-time algorithm to find the optimal solution:

Theorem 2. *There is a linear-time algorithm that, given a planar embedded graph with edge-weights and a subset Q of the vertices on the boundary of a single face, finds a minimum-weight two-edge-connected sub-multigraph of G spanning Q .*

For ease of exposition, we will take the the face on which the vertices Q lie to be the outermost or infinite face of the planar embedded graph. That is, the vertices of Q lie on the boundary of the planar graph.

Both results rely on a common observation (Theorem 3, Section 2) concerning the structure of two-edge connectivity between boundary vertices of planar graphs.

1.1 Related Work

Two-edge-connected spanning subgraph. A special case that has received much attention is the problem of finding a minimum-weight subgraph of G in which every pair of vertices is two-connected. This problem is called *two-edge-connected*

spanning subgraph, and is NP-hard [8] and max-SNP complete [6] in general graphs. Frederickson and JáJá [9] gave a 3-approximation algorithm for this problem. The approximation ratio was improved to 2 by Khuller and Vishkin [14]. For the unweighted case, they gave a 1.5-approximation algorithm. Jothi, Raghavachari, and Varadarajan [13] improved the approximation ratio to $5/4$.

In planar graphs the problem is NP-hard. Berger et al. [2] and Berger and Grigni [3] gave PTASes for the unweighted and weighted cases, respectively, in planar graphs. In both cases, the degrees of the polynomial depend on the desired precision ϵ . All the above algorithms work for the case where the output is not allowed to duplicate edges. For the case where duplication is allowed, the techniques of Klein [15] can be applied to obtain a linear-time approximation scheme.

Beyond spanning. For the more general case where a subset Q of the vertices need only be spanned, Ravi [18] showed that Frederickson and JáJá's approach could be generalized to give a 3-approximation algorithm (in general graphs). Klein and Ravi [17] gave a 2-approximation for a more general problem in which the input specifies which pairs of vertices must be connected up. This result was greatly generalized by Williamson, Goemans, Mihail, and Vazirani [20], Goemans, Goldberg, Plotkin, Shmoys, Tardos, and Williamson [10], and Jain [12]. These algorithms did not require duplication of edges.

In their recent paper on the spanning case, Berger and Grigni raise the question of whether there is a PTAS for finding a minimum-weight two-edge-connected subgraph of a planar graph. In this paper, we answer that question in the affirmative, at least when edge duplications are allowed.

1.2 Notation

For a path P , $P[x, y]$ denotes the x -to- y subpath of P for vertices x and y of P . For paths A and B , $A \circ B$ denotes the concatenation of A and B . For a subgraph H of a graph G , we use $V(H)$ to denote the set of vertices in H . We similarly use the notation $V(P)$, etc.

We employ the usual definitions of planar embedded graphs. For a face f , the cycle of edges making up the boundary of f is denoted ∂f . We assume the planar graph G is connected and is embedded in the plane, so there is a single infinite face, and we denote its boundary by ∂G .

For a cycle C in a planar embedded graph, $C[x, y]$ denotes an x -to- y path in C for vertices x and y of C . There are two such paths and the choice between the two possibilities will be disambiguated by specifying an orientation of the cycle (clockwise or counterclockwise). A cycle C is said to *enclose* the faces that are embedded inside it. C encloses an edge/vertex if the edge/vertex is embedded inside it or on it. In the former case, C *strictly encloses* the edge/vertex.

See Figure 1 for an illustration of the notion of paths crossing. A cycle is non-self-crossing if every pair of subpaths of the cycle do not cross. Two trees are noncrossing if no path in one crosses a path of the other.

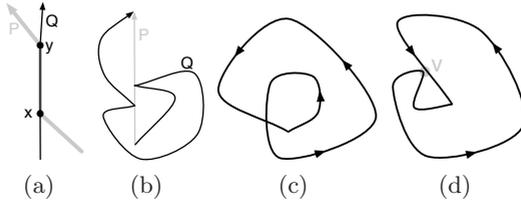


Fig. 1. (a) P crosses Q . (b) P and Q are noncrossing. (c) A self-crossing cycle. (d) A non-self-crossing cycle (non-self-crossing allows for repeated vertices, i.e. v).

1.3 Outline

In Section 2, we establish some key properties of two-edge-connectivity between boundary vertices of a planar graph. In Section 3, we prove Theorem 2 by giving a linear-time algorithm for the special case of finding the minimum two-edge-connected subgraph containing a subset of the boundary vertices of a planar embedded graph. In Section 4, we build on the results in Section 2 to give a decomposition of solutions to the two-edge connectivity survivable network problem in planar graphs where all terminals are on the boundary.

The remainder of the paper is devoted to proving the PTAS of Theorem 1. The approximation scheme employs an approach used by Borradaile, Klein, and Mathieu [5] to obtain an approximation scheme for Steiner tree. In Section 5, we outline the approach. In particular, what is needed is a structural theorem that states that the interaction between different parts of an optimal solution can be restricted to be “simple” while paying only a small penalty (in relative terms) in weight. We restate this theorem (Theorem 4) as given in [5] for the Steiner tree problem. The corresponding theorem for two-edge connectivity (Theorem 5) appears in Section 6. The proof draws on the results of Sections 2 and 4 and the corresponding structure theorem for Steiner trees. Finally, in Section 7, we briefly outline the dynamic program that is at the heart of the computation.

2 Basic Structural Properties of Boundary Connectivity

The results of this section hold for both subgraphs and sub-multigraphs. In this section, we investigate the structure of sub-(multi)graphs of G that achieve up to $\{0, 1, 2\}$ -edge-connectivity between vertices of ∂G .

Since we are only interested in connectivity up to and including two-edge connectivity, we define the following: for a graph H and vertices x, y , let

$$c_H(x, y) = \min\{2, \text{maximum number of edge-disjoint } x\text{-to-}y \text{ paths in } H\}.$$

For two sub-multigraphs H and H' of a common graph G and for a subset S of the vertices of G , we say H' achieves the two-connectivity of H for S if $c_{H'}(x, y) \geq c_H(x, y)$ for every $x, y \in S$. We say H' achieves the boundary two-connectivity of H if it achieves the two-connectivity of H for $S = V(\partial G)$.

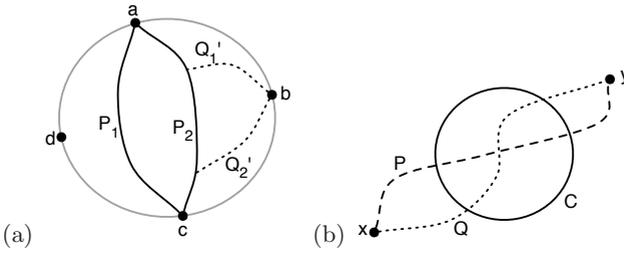


Fig. 2. (a) An illustration of the paths in Lemma 2. (b) An illustration of the proof of Lemma 3: there are edge-disjoint x -to- y paths that do not use edges enclosed by C .

Lemma 1 (Transitivity). For any graph H , for vertices $u, v, w \in V(H)$, $c_H(u, w) \geq \min\{c_H(u, v), c_H(v, w)\}$

Lemma 2 (Crossing). Let G be a planar embedded graph, and let its boundary be $v_1v_2 \dots v_n$. For integers $1 \leq i < j < k < \ell \leq n$, for any subgraph H of G , $c_H(v_i, v_j) \geq \min\{c_H(v_i, v_k), c_H(v_j, v_\ell)\}$.

Proof (Sketch). For the case of connectivity two, see Figure 2(a). Given 2 edge-disjoint a -to- c paths (P_1 and P_2) and 2 edge-disjoint b -to- d paths (whose prefixes are Q'_1 and Q'_2), it is easy to construct 2 a -to- b edge-disjoint paths. The proof for connectivity one is similar but simpler. \square

Lemma 3. Let H be a sub-(multi)graph of G and let C be a non-self-crossing cycle of H . Let H' be the subgraph of H obtained by removing the edges of H that are strictly enclosed by C . H' achieves the boundary 2-connectivity of H .

Proof. See Figure 2(b). Without loss of generality, let C be a simple cycle that is clockwise according to the planar embedding. Consider two vertices x and y of ∂G . We show that there are $c_H(x, y)$ edge-disjoint x -to- y paths in H that do not use edges strictly enclosed by C . There are two non-trivial cases: $c_H(x, y) = 1$ and $c_H(x, y) = 2$. We omit the former case, as the latter is illustrative.

Let P and Q be edge-disjoint x -to- y paths in H . If Q does not intersect C , then P' and Q are edge-disjoint paths, neither of which has a dart strictly enclosed by C (where P' is as defined above). Suppose that both P and Q intersect C . Let x_Q and y_Q be vertices of Q defined as for P . Suppose these vertices are ordered x_P, x_Q, y_Q, y_P around C . Then $P[x, x_P] \circ C[x_P, y_Q] \circ Q[y_Q, y]$ and $Q[x, x_Q] \circ rev(C[y_P, x_Q]) \circ P[y_P, y]$ are edge disjoint x -to- y paths that do not use any edges enclosed by C . This case is illustrated in Figure 2(b); other cases follow similarly.

We have shown that we can achieve the boundary two-connectivity of H without using any edges enclosed by a cycle of H . The lemma follows. \square

Corollary 1. Let H be a subgraph of G and let H' be a minimal subgraph of H that achieves the boundary two-connectivity of H . Then in H' every cycle C strictly encloses no edges.

Lemma 4. *Let H be a subgraph of G . Let S be a subset of $V(\partial G)$ such that, for every $x, y \in S$, $c_H(x, y) = 2$. Then there is a non-self-crossing cycle C in H such that $S \subseteq V(C)$ and the order that C visits the vertices in S is the same as their order along ∂G .*

Proof (sketch). Assume that the vertices of S are in the order s_1, s_2, \dots, s_k along ∂G . Let $\partial G[s_{i+1}, s_i]$ denote the subpath of the boundary of G between s_{i+1} and s_i that does not go through s_j for $j \neq i, i + 1$. Let P_i be the s_i -to- s_{i+1} path in H (taking the indices mod k) such that the cycle $P_i \circ \partial G[s_{i+1}, s_i]$ encloses only one s_i -to- s_{i+1} path (namely, P_i). One can show that P_i does not cross P_j for any pair i, j . The cycle $C = P_1 \circ P_2 \circ \dots \circ P_{k-1}$ has the properties required by the lemma. □

3 Linear-Time Exact Algorithm for a Boundary Two-Edge-Connectivity Problem

Here we give a linear-time algorithm for the following problem: given a weighted, planar graph G and a subset Q of the vertices of ∂G , find a minimum-weight two-edge-connected sub-multigraph of G that spans Q . This will prove Theorem 2, as stated in the Introduction. The algorithm whose correctness will follow from Lemma 4, is:

BOUNDARY2EC(G, Q)

1. Let q_1, q_2, \dots be the cyclic ordering of the terminals in Q along ∂G .
2. For $i = 1, \dots$, let P_i be the shortest q_i -to- q_{i+1} path in G (taking the indices mod $|Q|$).
3. Return the disjoint union $\cup_i P_i$.

Using the following lemma, we show that BOUNDARY2EC can be implemented in linear time using the linear-time shortest path algorithm for planar graphs [11].

Lemma 5. *Let a, b and c be vertices ordered along the clockwise boundary ∂G of a planar graph G . Let T_a be the shortest-path tree rooted at a . Then there is a shortest b -to- c path in G that is enclosed by the cycle $\partial G[b, c] \circ T[c, b]$.*

Proof (sketch). Suppose that the shortest b -to- c path P in G is not enclosed by the cycle $\partial G[b, c] \circ T[c, b]$. Then there is a subpath of P that contradicts the shortness of T . □

A linear-time implementation of BOUNDARY2EC is: compute a shortest-path tree T rooted at terminal q_1 in linear time; for each i , consider the graph G_i enclosed by $C_i = \partial G[q_i, q_{i+1}] \circ T[q_{i+1}, q_i]$; compute the shortest q_i -to- q_{i+1} path P_i in G_i . By Lemma 5, P_i is a shortest q_i -to- q_{i+1} path in G . Since each edge of G appears in at most two subgraphs G_i and G_j , the paths P_i can be computed in linear time.

We now argue that BOUNDARY2EC finds the minimum-weight two-edge-connected multi-subgraph of G that spans Q . Certainly BOUNDARY2EC returns a 2-edge-connected multi-subgraph that spans Q . We show that the graph

BOUNDARY2EC finds is of minimum weight. Let H be the optimal solution. By Lemma 4, there is a cycle C in H that visits the vertices q_1, q_2, \dots in order. This cycle can be written as $L_1 \circ L_2 \circ \dots$ where L_i is a q_i -to- q_{i+1} path. Let P_i be the shortest q_i -to- q_{i+1} path. Then $w(P_1 \circ P_2 \circ \dots) \leq w(L_1 \circ L_2 \circ \dots) \leq w(H)$.

4 Decomposition Result for Boundary Connectivity

For the theorem given in this section, we have to generalize the notion of connectivity requirements. Connectivity requirements so far assign an integer to each vertex; the corresponding subgraph must ensure connectivity at least $\min\{r_u, r_v\}$ between u and v . One can instead specify a connectivity requirements for each pair of vertices, using a function from the set of two-element subsets of $V(\partial G)$ (written $\binom{V(\partial G)}{2}$) to $\{0, 1, 2\}$.

Theorem 3. *Let G be a connected planar embedded graph. Let $r : \binom{V(\partial G)}{2} \rightarrow \{0, 1, 2\}$ be a function specifying connectivity requirements among the boundary vertices. There is a collection $\mathcal{X} = \{X_1, \dots, X_k\}$ of subsets of $V(\partial G)$ that are noncrossing with respect to ∂G such that a minimal subgraph H of G satisfies connectivity requirements $r(\cdot)$ iff H contains edge-disjoint non-crossing trees T_1, T_2, \dots, T_k where, for each i , T_i spans X_i .*

In the following we will assume for notational convenience that the boundary of the graph G is a simple cycle; that is, a vertex appears at most once along ∂G . Let us see why it suffices to prove the theorem with this assumption. Suppose the boundary of G is not simple: there is a vertex v that appears at least twice along ∂G . Partition G into two graphs G_1 and G_2 such that v appears exactly once along ∂G_1 and $E(\partial G) = E(\partial G_1) \cup E(\partial G_2)$. Let x be a vertex of ∂G_1 and let y be a vertex of ∂G_2 . Then $c_G(x, y) = \min\{c_{G_1}(x, v), c_{G_2}(v, y)\}$.

Let $a_1 a_2 a_3 a_4 \dots a_m$ be the alternating sequence of vertices and edges of ∂G in the order in which they are encountered during a clockwise traversal. We say $\{a_i, a_k\}$ and $\{a_j, a_\ell\}$ cross if $i < j < k < \ell$.

We start with some definitions that will lead to the definition of the sets making up \mathcal{X} :

- \sim_2 is a relation on the vertices of ∂G : $u \sim_2 v$ if $r(\{u, v\}) = 2$.
- \sim_2^* is the transitive and crossing closure of \sim_2 . That is, \sim_2^* is the minimal superset of \sim_2 such that if $x \sim_2^* y$ and $u \sim_2^* v$ and either $\{x, y\}$ crosses $\{u, v\}$ or $y = u$, then $x \sim_2^* v$.
- \sim_1 is a relation on the vertices of ∂G : $x \sim_1 y$ if $r(\{x, y\}) \geq 1$. Let \sim_1^* be the transitive and crossing closure of \sim_1 .
- $r_1^* : \binom{V(\partial G)}{2} \rightarrow \{0, 1\}$ is a requirement function such that $r_1^*(\{u, v\}) = 1$ iff $u \sim_1^* v$.
- \sim_0 is a relation on the edges of ∂G : $a \sim_0 b$ if there is no set $\{u, v\} \subset V(\partial G)$ that crosses $\{a, b\}$ such that $u \sim_2^* v$. It is easy to see that \sim_0 is an equivalence relation.

Let E_1, \dots, E_ℓ be the equivalence classes of \sim_0 . For $i = 1, \dots, \ell$, let $Z_i = \bigcup\{\text{endpoints of } e : e \in E_i\} \cap V(H)$, and let $\mathcal{X}_i = \{W \cap Z_i : W \text{ an equivalence class of } r_1^*\}$. Let $\mathcal{X} = \bigcup_i \mathcal{X}_i$, and write $\mathcal{X}_i = \{X_1, \dots, X_k\}$. There are two parts to the proof of the theorem.

Part 1: For $i = 1, \dots, k$, let T_i be a tree that connects X_i where the T_i 's are edge-disjoint. Let $H = \bigcup_i T_i$. We will show that H satisfies the original connectivity requirements $r(\cdot)$, thus proving the forward direction of the theorem. We must show (A) if $r(\{x, y\}) = 2$ then $c_H(x, y) = 2$, and (B) if $r(\{x, y\}) = 1$ then $c_H(x, y) \geq 1$.

Let Y_1, \dots, Y_ℓ be the equivalence classes of \sim_2^* . For each Y_i , we will show that H contains a cycle C_i through the vertices of Y_i , which will prove (A). Let the vertices of Y_i be x_0, x_1, \dots, x_{p-1} , numbered according to their occurrence in a clockwise traversal of ∂G .

Claim 1: For $j = 0, \dots, p - 1$, there is some $X \in \mathcal{X}$ that contains x_j and $x_{j+1 \bmod p}$.

Proof. Let e be the edge immediately after x_j in clockwise traversal of ∂G , and let e' be the edge immediately before $x_{j+1 \bmod p}$. Suppose there were a subset $\{u, v\} \subset V(\partial G)$ that crosses $\{e, e'\}$ such that $u \sim_2^* v$. Assume without loss of generality that u occurs after e and before e' in clockwise traversal of ∂G . Then v occurs after e' and before e . It follows that one of the following must hold: $v = x_j$ or $v = x_{j+1 \bmod p}$ or $\{u, v\}$ crosses $\{x_j, x_{j+1 \bmod p}\}$. In each case, since \sim_2^* is closed under crossing and transitivity, $u \sim_2^* x_j$, contradicting the fact that x_j and $x_{j+1 \bmod p}$ are consecutive elements of Y_i . This shows that $e \sim_0 e'$, which shows in turn shows that x_j and $x_{j+1 \bmod p}$ are in a common set $Z \in \mathcal{Z}$. Since $r(\{x_j, x_{j+1}\}) = 2$, we infer $r_1^*(\{x_j, x_{j+1}\}) = 1$, so there is a set $X \in \mathcal{X}$ (with $X \subset Z$) that contains x_j and x_{j+1} . □

Let P_j be the x_j -to- x_{j+1} path in T (the tree that spans X). By combining these paths for $j = 0, 1, \dots, p - 1$, we obtain a cycle C_i , proving (A).

Now we prove (B). Let U_1, \dots, U_s be all the equivalence classes of \sim_2^* such that there is pair in U_i crosses $\{x, y\}$ for every i . Assume that these sets are ordered according to their distance from x along ∂G (in, say, the clockwise direction). Let u_i and v_i be two distinct vertices of U_i chosen such that u_1 is the vertex of U_1 closest to x along ∂G and u_s is the vertex of U_s closest to y along ∂G . (See Figure 3.)

If $s = 0$ then there are edges e_x and e_y adjacent to x and y respectively such that $e_x \sim e_y$. So x and y are in a common set $Z \in \mathcal{Z}$. Since $r(\{x, y\}) = 1$ and $x \sim_1^* y$ then $r^*(\{x, y\}) = 1$ and x and y are in a common set $X_i \in \mathcal{X}$. Therefore T_i (and hence H) contains an x -to- y path.

Suppose that $s > 0$. The argument is illustrated in Figure 3. Since $x \sim_1^* y$ and \sim_1^* is closed under crossing, $x \sim_1^* u_i$ for $i = 1, \dots, s$. Since \sim_1^* is closed under transitivity, $u_i \sim_1^* u_{i+1}$ for $j = 1, \dots, s - 1$. By choice of u_1 , $x \sim_0 u_1$, so x and u_1 are in a common set $X_i \in \mathcal{X}$. Therefore T_i (and hence H) contains an x -to- u_1 path. Similarly H contains a u_s -to- y path.

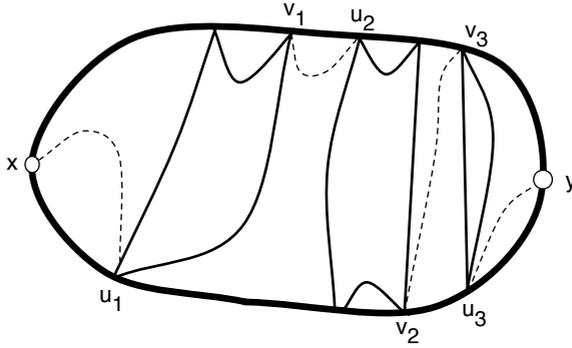


Fig. 3. The argument for one-connectivity is illustrated. Because $\{x, y\}$ crosses $\{u_1, u_2\}$, a connectivity requirement arises between x and u_2 . Hence there is an x -to- u_2 path. Moreover, for each equivalence class, there is a cycle (indicated in medium-bold) connecting the members u_i . Combining the paths with the cycles yields an x -to- y path.

By (A), H contains a u_i -to- v_i path for $i = 1, \dots, s$. For $i = 1, \dots, s - 1$ we argue that H contains a u_i -to- u_{i+1} path. Since $u_i \sim_0 u_{i+1}$ and by the transitivity of \sim_1^* , u_i and u_{i+1} are in a common set X_i , so T_i contains such a path. Combining these paths, we obtain an x -to- y path in H , proving (B) and the forward direction of Theorem 3.

Part 2: Let H be a subgraph of G that satisfies the connectivity requirements $r(\cdot)$. Assume without loss of generality that H is edge-minimal subject to this condition. We will show how to decompose H into noncrossing, edge-disjoint subgraphs T_1, \dots, T_k , so that T_i spans X_i .

By Lemmas 1 and 2, for any vertices $x, y \in V(\partial G)$, if $x \sim_2^* y$ then $c_H(x, y) = 2$. As in the proof of Part 1, let Y_1, \dots, Y_ℓ be the equivalence classes of \sim_2^* . For each Y_j , let C_j be the corresponding non-self-crossing cycle in H whose existence is guaranteed by Lemma 4. By Corollary 1, C_j does not strictly enclose any edges.

Let R be the subgraph of G consisting of $\partial G \cup \bigcup_{j=1}^\ell C_j$. Let \mathcal{F} be the set of faces of R other than the infinite face and the faces in the interiors of cycles C_j . For each face $f \in \mathcal{F}$, let H_f be the subgraph of H consisting of edges enclosed by ∂f .

Claim 2: For distinct faces $f_1, f_2 \in \mathcal{F}$, H_{f_1} and H_{f_2} are edge-disjoint.

Proof. The set of edges strictly enclosed by f_1 and the set of edges enclosed by f_2 are clearly disjoint. We need to address the case of edges not strictly enclosed by f_1 , i.e. edges of ∂f_1 . Every edge e of R belongs either to ∂G or to some cycle C_i , so e is on the boundary of some face not in \mathcal{F} . Hence e is on the boundary of at most one face in \mathcal{F} . □

Claim 3: For any face $f \in \mathcal{F}$ and any vertices $x, y \in V(\partial f \cap \partial G)$, if $x, y \in X \in \mathcal{X}$ then H_f contains an x -to- y path.

Proof. By Lemmas 1 and 2, if $x \sim_1^* y$ then $c_H(x, y) \geq 1$. Hence H contains such a path P . Suppose P is not a path of H_f , and consider a maximal subpath P' of P that is not enclosed by ∂f . By maximality, the endpoints of P' must lie on ∂f . Since P' is enclosed by ∂G , the endpoints of P' must lie on a subpath Q of $\partial f \cap (\bigcup_{j=1}^{\ell} C_j)$. Thus Q belongs to H , and therefore to H_f . The subpath P' of P can therefore be replaced by Q . Similarly replacing each such subpath yields an x -to- y path in H_f . \square

Claim 4: Let f be a face in \mathcal{F} , and let a, b be edges of $\partial G \cap \partial f$. Then there is some equivalence class E_i of \sim_0 that contains a and b .

Proof. Assume for a contradiction that there is a subset $\{u, v\} \subset V(\partial G)$ that crosses $\{a, b\}$ such that $u \sim_2^* v$. There is some subset Y_j containing u and v , and therefore some cycle C_j that passes through u and v . Since the edges of C_j belong to R , this contradicts the fact that a and b lie on the boundary of a common face of R . \square

Now we can complete the proof of Part 2. For $i = 1, \dots, k$, let W_i be the set of faces f in \mathcal{F} such that $V(\partial f)$ intersects Z_i . By Claim 4, the W_i 's are disjoint. Let $H_i = \bigcup_{f \in W_i} H_f$. By Claim 2, the H_i 's are edge-disjoint. By Claim 3, H_i spans X_i . By the disjointness of the W_i 's, no path in H_{i_1} crosses a path in H_{i_2} if $i_1 \neq i_2$. Since the connectivity requirements are $\{0, 1\}$, each H_i contains a forest F_i that satisfies the requirements $r_1^*(\cdot)$ among vertices of Z_i . The components of F_i span the sets in \mathcal{X}_i . The union of all these trees is a subgraph that, by Part 1, satisfies connectivity requirements $r(\cdot)$. This completes the proof of Part 2 and the reverse direction of Theorem 3.

5 A PTAS Framework for Connected Problems in Planar Graphs

In this section, we review the approach used in [5] to give a PTAS for the Steiner tree problem in planar graphs as we will use the same approach for this survivable network problem.

The framework relies on an algorithm for finding a subgraph MG of G , called the *mortar graph* [5]. The mortar graph spans Q and has total weight no more than $f(\epsilon)$ times the minimum weight of a Steiner tree in G spanning Q (and so has weight no more than $f(\epsilon) \cdot \text{OPT}$ where OPT denotes the optimal value of the Steiner tree or the survivable network problem). The first step in constructing MG is to find an approximate Steiner tree and recursively augmenting this with short paths.

The mortar graph is a grid-like subgraph (the bold edges in Figure 4(a)). For each cell or face of the mortar graph, the subgraph of G enclosed by that face is called a *brick* (Figure 4(b)). The properties of bricks needed for this work are summarized by the following lemma.

Lemma 6 (from Lemma 4 [5]). *The boundary of a brick B , in counterclockwise order, is the concatenation of four paths $W_B \cup S_B \cup E_B \cup N_B$ such that:*

- (B1) The set of edges $B \setminus \partial B$ is nonempty.
- (B2) Every terminal of Q that is in B is on N_B or on S_B .
- (B3) N_B and S_B are ϵ -short.

A path P in a graph G is ϵ -short if for every pair of vertices x and y on P , the distance from x to y along P is at most $(1 + \epsilon)$ times the distance from x to y in G : $dist_P(x, y) \leq (1 + \epsilon)dist_G(x, y)$.

The mortar graph and the bricks are building blocks of the structural properties required for designing an approximation scheme. In [5], it was demonstrated that there is a near-optimal Steiner tree whose interaction with the mortar graph is “simple”. To formalize this notion (Theorem 4), we say that there is near-optimal Steiner tree that *joins* the boundary of each brick a small number of times. A *joining vertex* of graph H with a path P is a vertex of P that is the endpoint of an edge of $H \setminus P$. The intersection of a tree with a brick might not be connected, and so the theorem applies to forests inside bricks.

Theorem 4 (Structural property of bricks for $\{0, 1\}$ -edge connectivity, Theorem 4 [5]). *Let B be a plane graph with boundary $N \cup E \cup S \cup W$ satisfying the brick properties of Lemma 6. Let F be a subgraph of B . There is a forest \tilde{F} of B with the following properties:*

- (F1) If two vertices of $N \cup S$ are connected in F , then they are connected in \tilde{F} .
- (F2) The number of joining vertices of \tilde{F} with both N and S is at most $\alpha(\epsilon)$.
- (F3) $\ell(\tilde{F}) \leq (1 + c\epsilon)\ell(F)$.

In the above, $\alpha(\epsilon) = o(\epsilon^{-5.5})$ and c is a fixed constant.

This theorem is a key ingredient to the proof of correctness of the PTAS for Steiner tree and will be used in proving a similar theorem (Theorem 5) for the $\{0, 1, 2\}$ -edge connectivity problem we solve here.

5.1 Approximation Scheme

The approximation scheme consists of the following steps. Only Step 5 depends on the specifics of the optimization problem, though Step 4 depends on a constant that comes out of the Structure Theorem for the problem.

Step 1: Find the mortar graph MG .

Step 2: Decompose MG into “parcels”, subgraphs with the following properties:

- (a) Each parcel consists of the boundaries of a disjoint set of faces of MG . Since each edge of MG belongs to the boundaries of exactly two faces, it follows that each edge belongs to at most two parcels.
- (b) The weight of all boundary edges (those edges belonging to two parcels) is at most $(1/\eta)weight(MG)$. We choose η so that this bound is $(\epsilon/2)weight(OPT)$.
- (c) The planar dual of each parcel has a spanning tree of depth at most $\eta + 1$.

Each parcel P corresponds to a subgraph of G , namely the subgraph consisting of the bricks corresponding to the faces making up P . Let us refer to this subgraph as the *filled-in* version of P .

Step 3: Select a set of “artificial” terminals on the boundaries of parcels so that for each filled-in parcel, there is a feasible (with respect to original and artificial terminals) solution whose weight is at most the parcel’s boundary plus the weight of the intersection of OPT with the filled-in parcel, and the union over all parcels of such feasible solutions is a feasible solution for the original graph.

Step 4: For each brick, designate as *portals* a constant number of vertices on the boundary of each brick. The constant is chosen, depending on the Structure Theorem, so that there exists a near-optimal feasible solution that is *portal-respecting*, i.e. passes through a portal whenever it passes from one face of MG to another.

Step 5: For each filled-in parcel, find an optimal portal-respecting solution. Output the union of these solutions.

Step 1 can be carried out in $O(n \log n)$ time. Details are in [5,4,16]. Step 2 can be done in linear time. It consists of doing breadth-first search in the planar dual of MG , and then applying a “shifting” technique in the tradition of Baker [1]. Details are in [5]. Step 3 uses the fact that each parcel’s boundary consists of edge-disjoint, noncrossing cycles. If such a cycle strictly encloses an original terminal and does not enclose all terminals, a vertex on the cycle is designated an artificial terminal. Under this condition, any feasible solution for the original graph must cross the cycle; by adding the edges of the cycle, we get a feasible solution that also spans the artificial terminal. Step 3 can be implemented in linear time. Step 5 is achieved in linear time using dynamic programming.

Step 4 uses a simple greedy algorithm to designate portals along the boundary ∂B of a brick B so that there are at most $\theta + 1$ portals chosen, and that each vertex on the boundary is within distance at most $weight(\partial B)/\theta$ of some portal. We discuss the choice of θ presently.

5.2 Portal-Connected Graph

In order to make more precise the notion of a portal-respecting feasible solution, we introduce an auxiliary graph, the *portal-connected graph* (PCG) of a parcel. See Figure 4. Starting with a parcel (which consists of edges of the mortar graph), within each face, insert a duplicate of the brick corresponding to that face, and use artificial zero-weight edges to connect the occurrences of the portals in the duplicate brick to the occurrences of the same vertices in the parcel.¹

A path P in the filled-in parcel from a vertex x interior to a brick B to a vertex y on the boundary of the brick B corresponds to a path \tilde{P} in the PCG from x to the occurrence of y in the parcel; \tilde{P} must take a detour through an artificial edge, and must therefore go through a portal. The increase in weight is at most $2weight(\partial B)/\theta$.

¹ Our usage of the term *PCG* differs slightly from that in [5], where the PCG was defined for the entire graph, not just a parcel.

The Structure Theorem states that the subgraph of OPT embedded strictly inside a brick can be modified so that it touches the boundary of the brick at no more than $\alpha(\epsilon)$ vertices. Rerouting each of these connections so it occurs at a portal incurs a weight of $2weight(\partial B)/\theta$, for a total of $2\alpha(\epsilon)weight(\partial B)/\theta$. The sum of boundary lengths of all bricks is twice the length of the mortar graph, which in turn is at most $f(\epsilon)$ times the value of OPT. The value of θ is chosen to ensure that the total rerouting weight is at most ϵ times the value of OPT. In order to find a nearly optimal solution in the filled-in parcel, therefore, it suffices to find an optimal solution in the PCG.

Recall that the planar dual of the parcel has a spanning tree of depth $\eta + 1$. Since each brick has at most $\theta + 1$ portals, it follows that the planar dual of the PCG has a spanning tree of depth at most $(\eta + 1)(\theta + 1)$. It follows that there is a rooted spanning tree of the PCG (the primal) such that, for each vertex v , there at most $2(\eta + 1)(\theta + 1) + 1$ edges from descendants of v to non-descendants. This spanning tree is used to guide the dynamic program (Section 7).

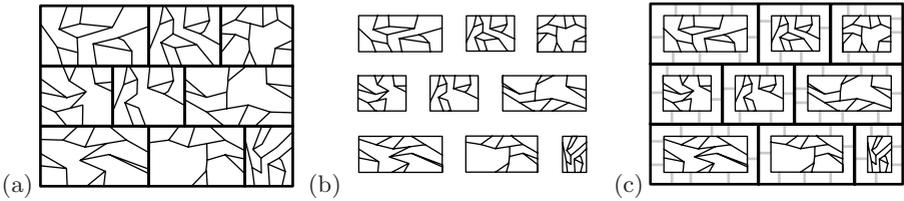


Fig. 4. The mortar graph in bold (a), the set of bricks (b), and the portal connected graph (c)

6 Applying the PTAS Framework

Theorem 4 applies directly to the Steiner-tree problem: the intersection of a tree with a brick is a forest and since the terminals are vertices of MG , it is enough to maintain connectivity between vertices on the boundary of a brick. However, for the 2-EC problem, the intersection of a solution with a brick has a more complicated structure.

In this section we prove the following counterpart to Theorem 4 that maintains up to 2 connectivity between vertices on the north and south boundary of a brick.

Theorem 5 (Structural property of bricks for $\{0, 1, 2\}$ -edge connectivity). *Let B be a plane graph with boundary $N \cup E \cup S \cup W$ and satisfying the brick properties of Lemma 6. Let H be a subgraph of B . There is another subgraph \widehat{H} that is the disjoint union of three forests $\widehat{F}_1, \widehat{F}_2, \widehat{F}_3$ of B with the following properties:*

- (H1) \widehat{H} achieves the 2-connectivity of H for vertices of $N \cup S$.
- (H2) The number of joining vertices of \widehat{H} with both N and S is at most $2\alpha(\epsilon)$.
- (H3) $\ell(\widehat{H}) \leq (1 + c\epsilon)\ell(H)$.

In the above, $\alpha(\epsilon) = o(\epsilon^{-5.5})$ and c is a fixed constant.

Proof. The theorem is proved as follows. We first show that there is a subgraph H' of H that is the disjoint union of a set of trees $\mathcal{T} = \{T_1, T_2, \dots, T_k\}$ (where k can be very large) such that H' achieves the 2-connectivity of H (Theorem 3). We then show that we can partition this set of trees into three sets such that the disjoint union of each set is a forest. We then apply Theorem 4 to each of these forests, proving Theorem 5.

Let H' be a minimal subgraph of H such that H' achieves the 2-connectivity of H for vertices on $N \cup S$.

By the *only-if* direction of Theorem 3, H' is the union of a set of noncrossing edge-disjoint trees $\mathcal{T} = \{T_1, T_2, \dots\}$, where each tree T_i achieves connectivity between a set X_i of vertices of $N \cup S$. Partition \mathcal{T} into two sets:

$$\begin{aligned} \mathcal{T}_1 &= \{T_i \in \mathcal{T} \text{ such that } X_i \subseteq V(N) \text{ or } X_i \subseteq V(S)\}. \\ \mathcal{T}_2 &= \{T_i \in \mathcal{T} \text{ such that } X_i \text{ has vertices in both } V(N) \text{ and } V(S)\}. \end{aligned}$$

We further partition \mathcal{T}_2 into two sets. Let T_i and T_j be two trees in \mathcal{T}_2 . Since T_i and T_j do not cross each other, if the vertices $X_i \cap V(S)$ appear before $X_j \cap V(S)$ along S then the vertices in $X_i \cap V(N)$ appear before $X_j \cap V(N)$ along N . It follows that there is an ordering of the trees in \mathcal{T}_2 from left to right in the brick, ordered according to the vertices in S to which they connect. Let \mathcal{T}_A be the set of trees of \mathcal{T}_2 that are even-numbered in this ordering and let \mathcal{T}_B be the set that are odd-numbered. That is, the trees in \mathcal{T}_2 alternate between \mathcal{T}_A and \mathcal{T}_B .

Any two trees in \mathcal{T}_A are separated by a tree in \mathcal{T}_B . Assume for a contradiction that a cycle was formed by some trees in \mathcal{T}_A . Then the cycle would have to strictly enclose an edge of a tree in \mathcal{T}_B . This contradicts Corollary 1. This shows that the trees in \mathcal{T}_A form a forest. Similarly, the trees in \mathcal{T}_B form a forest.

Consider a tree $T_i \in \mathcal{T}_1$. We describe how to select a corresponding tree \widehat{T}_i . Suppose that $X_i \subseteq V(N)$. Let \widehat{T} be the minimal subpath of N that spans X_i . The case where $X_i \subseteq V(S)$ is analogous.

Let \widehat{F}_1 be the *disjoint* union of $\{\widehat{T} : T \in \mathcal{T}_1\}$. (That is, the multiplicity of an edge in \widehat{F}_1 is the sum of its multiplicities in $\{\widehat{T} : T \in \mathcal{T}_1\}$.) Let \widehat{F}_A be the forest guaranteed by Theorem 4 for the forest obtained by taking the union of the trees in \mathcal{T}_A . Similarly define \widehat{F}_B . Let \widehat{H} be the union of $\widehat{F}_1, \widehat{F}_A, \widehat{F}_B$.

We show that \widehat{H} achieves the required properties.

It is clear from the construction that \widehat{F}_1 does not have any joining vertices with N or S . By Theorem 4, each of \widehat{F}_A and \widehat{F}_B has at most $\alpha(\epsilon)$ joining vertices with $N \cup S$. Therefore \widehat{H} has at most $2\alpha(\epsilon)$ joining vertices with $N \cup S$, proving Property H2.

Since N and S are ϵ -short paths, $\ell(\widehat{F}_1)$ is at most $1 + \epsilon$ times the total length of all trees in \mathcal{T}_1 . By Theorem 4, $\ell(\widehat{F}_A) \leq (1 + c\epsilon)\ell(\widehat{F}_A)$ and $\ell(\widehat{F}_B) \leq (1 + c\epsilon)\ell(\widehat{F}_B)$. It follows that $\ell(\widehat{H}) \leq (1 + c\epsilon)\ell(H)$, proving Property H3.

We now show that if two vertices of $N \cup S$ are 2-edge connected in H' , then they are 2-edge connected in \widehat{H} . Showing this for 1-edge connectivity is simpler; the argument is omitted here. This will complete the proof. We were particular

in partitioning the trees into forests $\mathcal{T}_1, \mathcal{T}_A, \mathcal{T}_B$ because applying Theorem 4 to a tree with two edges incident to a vertex $v \in \partial B$ could result in a tree with only one edge incident v . This could remove edge connectivity.

Let a and b be vertices of $N \cup S$ that are 2-edge connected in H' . Let C be the minimal cycle 2-connecting x and y as guaranteed by Lemma 4 and let $Y = V(C \cap (N \cup S))$. Let y_1, y_2, \dots, y_k be the order of the vertices of Y along the boundary of the brick. Let X_i be the set such that $y_i, y_{i+1} \in X_i$ and $X_i \subseteq V(\partial B[y_i, y_{i+1}])$ (as guaranteed by the construction given in Theorem 3). There are two cases:

$Y \subseteq V(N)$ or $Y \subseteq V(S)$: Without loss of generality, assume that y_1 is the first vertex and y_k is the last vertex of Y along N . Then X_1, \dots, X_{k-1} are subsets of N . X_k may contain vertices of S . Let \widehat{T}_i be a tree in \widehat{F}_1 that spans X_i (for $i = 1, \dots, k - 1$). Since \widehat{F}_1 is the disjoint union of these trees, there is a path P in \widehat{F}_1 that visits each vertex y_1, \dots, y_k in order. If X_k spans a vertex of S then $X_k \in F_A$ (without loss of generality). The vertices X_k are spanned by \widehat{F}_A and so there is a y_k -to- y_1 path Q in \widehat{H} that is edge disjoint from P . $P \circ Q$ is a cycle such that $Y \subseteq V(P \cup Q)$. The vertices in Y are 2-edge connected in \widehat{H} .

$Y \cap V(N) \neq \emptyset$ and $Y \cap V(S) \neq \emptyset$: Without loss of generality, assume that y_1 and y_l are the first and last vertices of Y along N . Then y_k and y_{l+1} are the first and last vertices of Y along S . By the argument used in the above case, there is a path P in \widehat{H} that visits the vertices y_1, \dots, y_l in order. Likewise, there is a path Q in \widehat{H} that visits the vertices y_{l+1}, \dots, y_k in order. We now argue that there are edge-disjoint y_l -to- y_{l+1} and y_k -to- y_1 paths in \widehat{H} by showing that T_l (the tree corresponding to X_l) is in F_A and T_k (the tree corresponding to X_k) is in F_B : by Lemma 1, there are no trees enclosed by C in H' , so T_l and T_k are ordered sequentially in \mathcal{T}_B . □

7 Dynamic Program

Here we give an outline of the dynamic program used to find an optimal solution in each *filled-in parcel*. As discussed at the end of Section 5.2, we use a rooted tree such that, for each vertex v , there at most $2(\eta + 1)(\theta + 1) + 1$ edges from descendants of v to non-descendants. Each vertex gives rise to a subproblem in the dynamic program. Both θ and η depend polynomially on $1/\epsilon$. The interaction between two subproblems is limited to this set of edges. Each brick in the brick decomposition corresponds to a base case of the dynamic program. All other base cases are trivial, corresponding to single vertices in our input graph.

For each subproblem, we consider all possible $\{0, 1, 2\}$ -connectivity patterns (or *configurations*) on the vertex set U . (A configuration is given by a forest with no degree-2 vertices whose vertices correspond to 2-edge connected components and whose edges correspond to adjacency between these components. Such a forest corresponds to a block-cut tree of the solution it encodes.) The leaves of the forests are identified with edges in the cut corresponding to the vertex set U .

Since there are $O(\theta\eta)$ edges in the cut, there are at most $O((\theta\eta)(\theta\eta)^{\theta\eta})$ forests representing configurations (by way of Cayley's formula).

It remains to show that we can solve a base case corresponding to a brick. The number of edges between a brick and the rest of the parcel is the number of portal edges, η , that connects the brick in the filled-in parcel. A configuration for the brick is a set of 2-connectivity requirements between the portal edges. Given such a set of requirements, we can use the algorithm implied by Theorem 3 to find a set of subsets \mathcal{X} of the portal edges such that independently connecting each set in \mathcal{X} will satisfy the given 2-connectivity requirements (Theorem 3). For each set in \mathcal{X} , we find the minimum-length Steiner tree using the algorithm of Erickson et al. [7]. For a constant number of terminals, using the algorithm of [11], this algorithm can be implemented to run in linear time. The resulting running time of the dynamic program, including the dependence on ϵ is $2^{o(\epsilon^{-9.5})}n$.

Comments

The PTAS framework used is potentially applicable to problems where (i) the input consists of a planar graph G with edge-weights and a subset Q of the vertices of G (we call Q the set of *terminals*), and where (ii) the output spans the terminals. Steiner tree and two-edge connectivity have been solved using this framework. The PTAS for the subset tour problem [16] (which was the inspiration for this framework) can be reframed using this technique. Recently, with David Pritchard, we have extended this work to give a PTAS for the $\{0, 1, \dots, k\}$ -edge connectivity problem in planar multigraphs. Details will follow in a longer version.

References

1. Baker, B.: Approximation algorithms for NP-complete problems on planar graphs. *J. ACM* 41(1), 153–180 (1994)
2. Berger, A., Czumaj, A., Grigni, M., Zhao, H.: Approximation schemes for minimum 2-connected spanning subgraphs in weighted planar graphs. In: Brodal, G.S., Leonardi, S. (eds.) *ESA 2005*. LNCS, vol. 3669, pp. 472–483. Springer, Heidelberg (2005)
3. Berger, A., Grigni, M.: Minimum weight 2-edge-connected spanning subgraphs in planar graphs. In: Arge, L., Cachin, C., Jurdziński, T., Tarlecki, A. (eds.) *ICALP 2007*. LNCS, vol. 4596, pp. 90–101. Springer, Heidelberg (2007)
4. Borradaile, G., Kenyon-Mathieu, C., Klein, P.: A polynomial-time approximation scheme for Steiner tree in planar graphs. In: 18th SODA, pp. 1285–1294 (2007)
5. Borradaile, G., Klein, P., Mathieu, C.: Steiner tree in planar graphs: An $O(n \log n)$ approximation scheme with singly exponential dependence on epsilon. In: Dehne, F., Sack, J.-R., Zeh, N. (eds.) *WADS 2007*. LNCS, vol. 4619, pp. 275–286. Springer, Heidelberg (2007)
6. Czumaj, A., Lingas, A.: On approximability of the minimum cost k -connected spanning subgraph problem. In: 10th SODA, pp. 281–290 (1999)
7. Erickson, R., Monma, C., Veinott, A.: Send-and-split method for minimum-concave-cost network flows. *Math. Op. Res.* 12, 634–664 (1987)

8. Eswaran, K., Tarjan, R.: Augmentation problems. *SIAM J. Comput.* 5(4), 653–665 (1976)
9. Frederickson, G., Jájá, J.: Approximation algorithms for several graph augmentation problems. *SIAM J. Comput.* 10(2), 270–283 (1981)
10. Goemans, M., Goldberg, A., Plotkin, S., Shmoys, D., Tardos, É., Williamson, D.: Improved approximation algorithms for network design problems. In: 5th SODA, pp. 223–232 (1994)
11. Henzinger, M., Klein, P., Rao, S., Subramanian, S.: Faster shortest-path algorithms for planar graphs. *J. Comput. System Sci.* 55(1), 3–23 (1997)
12. Jain, K.: A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica* 21(1), 39–60 (2001)
13. Jothi, R., Raghavachari, B., Varadarajan, S.: A $5/4$ -approximation algorithm for minimum 2-edge-connectivity. In: 14th SODA, pp. 725–734 (2003)
14. Khuller, S., Vishkin, U.: Biconnectivity approximations and graph carvings. *J. ACM* 41(2), 214–235 (1994)
15. Klein, P.: A linear-time approximation scheme for planar weighted TSP. In: 46th FOCS, pp. 647–647 (2005)
16. Klein, P.: A subset spanner for planar graphs, with application to subset TSP. In: 38th STOC, pp. 749–756 (2006)
17. Klein, P., Ravi, R.: When cycles collapse: A general approximation technique for constrained two-connectivity problems. In: 3rd IPCO, pp. 39–55 (1993)
18. Ravi, R.: Approximation algorithms for Steiner augmentations for two-connectivity. Technical Report TR-CS-92-21, Brown University (1992)
19. Resende, M., Pardalos, P. (eds.): *Handbook of Optimization in Telecommunications*. Springer, Heidelberg (2006)
20. Williamson, D., Goemans, M., Mihail, M., Vazirani, V.: A primal-dual approximation algorithm for generalized Steiner network problems. In: 35th STOC, pp. 708–717 (1993)