Algorithmic Operations Research Vol.3 (2008) 30–42

$K_{3,3}$ Minors and the Maximum-Flow Problem

Donald K. Wagner

Mathematical, Computer, and Information Sciences Division, Office of Naval Research, Arlington, VA 22203, U.S.A.

Abstract

Let G be a graph, and let e be an edge of G. The main result of this paper is that any instance of the maximum-flow problem on G having e as the "return" edge can be solved in linear time provided G does not have a $K_{3,3}$ minor containing e. The primary tool in proving this result is a new graph decomposition. In particular, it is shown that if G is 3-connected and does not have a $K_{3,3}$ minor containing e, then it can be decomposed into planar graphs, "almost" planar graphs, and copies of K_5.

Key words: Maximum flow, Planar graphs, $K_{3,3}$ minor

1. Introduction

This paper is concerned with the maximum-flow problem. A basic knowledge of this problem is assumed – see, for example, Ahuja, Magnanti, and Orlin [1]. For the present purpose, an instance of the problem is specified as (G, s, t, u), where G is an undirected graph, s and t are distinguished *source* and *sink* nodes of G, and u is a vector of real-valued, nonnegative, edge *capacities*. A *flow* is a pair (D, x), where D is an orientation of G and x is a nonnegative, real-valued vector indexed on the arc set of D. For a given flow (D, x) and node i of D, define *net flow* at i to be $x(\delta^+(i)) - x(\delta^-(i))$, where $\delta^+(i)$ (respectively, $\delta^-(i)$) denotes the set of arcs that have node i as their head (respectively, tail). (Here, for a subset of A of arcs or edges, $x(A) := \sum_{f \in A} x_f$.) A flow (D, x) is *feasible* if, for each arc f of D, $x_f \leq u_f$ and for each node i of D other than s and t, the net flow at i equals zero. The *value* of a flow (D, x) is equal to the net flow at t. A *maximum* flow is one of maximum value. A *minimum cut* is an st-cut (in either the directed or undirected sense, as appropriate) of minimum capacity.

Without loss of generality, it can be assumed that G has a unique edge joining s and t of capacity zero. Such an edge is called the *return* edge. Throughout this paper, the existence of the return edge is taken as a given.

Given the existence of a return edge, an instance of the maximum-flow problem can equivalently be specified by (G, e, u), where G and u are as before, and e is an edge of capacity zero. Note, however, if an instance is specified as (G, e, u), and $e = st$, there is nothing to distinguish the source node s from the sink node t. This does not cause any confusion, however, since a flow (D, x) maximizes the net flow at t if and only if (D', x) maximizes the net flow at s, where D' is obtained from D by reversing the orientation on each arc.

Now, consider an instance (G, e, u) of the maximum-flow problem where G is planar. (Note, given the assumption about the existence of the return edge $e = st$, this is equivalent to saying that $G \backslash e$ is "st-planar"; that is, it has a planar embedding in which two designated nodes, s and t, lie on the same face.) Ford and Fulkerson [5] provided a specialized augmenting-path procedure for computing a maximum flow on (G, e, u) assuming a planar embedding of G was given; this algorithm pre-dates their well-known algorithm [6] for general directed graphs by a year or so. (See Schrijver [15] for a comprehensive treatment of the history of the maximum-flow problem.) Berge and Ghouila-Houri [3] extended the Ford and Fulkerson algorithm to directed planar graphs. An $O(n \log n)$ version of the Berge and Ghouila-Houri algorithm was developed in Itai and Shiloach [11]. (Throughout, n and m denote the number of nodes and edges, respectively, of G.) A different approach for planar graphs was developed by Hassin [7], who provided an $O(n)$-time reduction of the maximum-flow problem on G to a shortest-path problem on its planar dual. Combined with the $O(n)$-time algorithm for solving the shortest-path problem on planar graphs due to Henzinger, Klein, Rao, and Subramanian [8], this yields an $O(n)$-time algorithm for solving the maximum-flow problem on (G, e, u).

Email: Donald K. Wagner [don.wagner@navy.mil].

The main purpose of this paper is to show that the linear-time solvability of the maximum-flow problem can be extended beyond planar graphs, and in particular, to graphs that do not have a $K_{3,3}$ minor containing the return edge. This is done by employing a new graph decomposition that essentially reduces the problem to the planar case.

Others have looked at solving graph optimization problems when the $K_{3,3}$ minors are excluded entirely; see, for example, Manor and Penn [13] and the references contained therein. Additionally, there has been work on solving optimization problems when other minors are excluded, such as K_5; see, for example, Barahona [2]. An important distinguishing feature of the present work is that the minor $K_{3,3}$ is excluded only "locally" (i.e., through the return edge), not "globally". This feature can also be seen in the work of Seymour [16], Truemper [20], and Tseng and Truemper [21] in the context of matroids.

The remainder of this paper is structured as follows. The next section contains the decomposition for graphs that do not have a $K_{3,3}$ minor containing a fixed edge e. This decomposition is completely independent of the maximum-flow problem. The decomposition forms the basis of two algorithms to follow. The first, contained in Section 3, is a linear-time algorithm for recognizing graphs that do not have a $K_{3,3}$ minor containing a fixed edge. The second, contained in Section 5, is a linear-time algorithm for solving the maximum-flow problem on a graph that does not have a $K_{3,3}$ minor containing the return edge. Section 4 is a brief section containing an algorithm for the target-flow problem, which is needed for the maximum-flow algorithm that follows. The final section briefly indicates how to extend the algorithm of Section 5 to solve the directed version of the maximum-flow problem.

2. Graph decomposition

A basic knowledge of graph theory is assumed. Undefined terminology is standard; see, for example, Diestel [4] or West [24].

Graph connectivity plays a central role in the graph decomposition described here. The notion of connectivity used here is that of Tutte [22]. In particular, a connected graph G is *n-connected*, for $n \geq 2$, if it does not have a k-separation for any $k < n$, where a *k-separation*, for a positive integer k, of G is a partition $\{E_1, E_2\}$ of the edge set such that $|E_1| \geq k \leq |E_2|$ and the edge-induced subgraphs $G[E_1]$ and $G[E_2]$ have at most k nodes in common. If $\{E_1, E_2\}$ is a k-separation, then each of E_1 and E_2 are *k-separators*. A k-separation $\{E_1, E_2\}$ of a connected graph G is an *internal k-separation* if $|E_1| \geq k+1 \leq |E_2|$. A k-connected graph is *internally $(k+1)$-connected* if it does not have an internal k-separation.

In this paper, 2- and 3-separations play a crucial role, as do the related notions of 2- and 3-sum decompositions, which are defined as follows. Let $\{E_1, E_2\}$ be a 2-separation of a 2-connected graph G. Let f be a new edge, and let G' be the graph obtained by adding f to G joining the two nodes in $V(G[E_1]) \cap V(G[E_2])$. For $i \in \{1, 2\}$, define G_i to be $G'[E_i \cup \{f\}]$. Then, $\{G_1, G_2\}$ is a *2-sum decomposition* of G, and f is the *connecting* edge. Now, let $\{E_1, E_2\}$ be an internal 3-separation of a 3-connected graph G. For each pair of non-adjacent nodes in $V(G[E_1]) \cap V(G[E_2])$, add a new edge joining the pair; denote the resulting graph by G'. Let T denote the set of edges of G' that have both ends in $V(G[E_1]) \cap V(G[E_2])$. For $i \in \{1, 2\}$, define G_i to be $G'[E_i \cup T]$. Then, $\{G_1, G_2\}$ is a *3-sum decomposition* of G, and T is the *connecting* triangle.

The following properties of 2- and 3-sum decompositions are well known and straightforward to prove.

LEMMA 1. *Let $\{G_1, G_2\}$ be a k-sum decomposition of a k-connected graph G, for $k \in \{2, 3\}$. Then, G_1 and G_2 are k-connected and are isomorphic to proper minors of G.* $\square$

Two special kinds of internal 3-separations are needed. Both are defined relative to a fixed edge. Specifically, let $\{E_1, E_2\}$ be an internal 3-separation of a 3-connected graph G, and let e be a fixed edge of G. If both ends of e are in $V(G[E_1]) \cap V(G[E_2])$, then the 3-separation $\{E_1, E_2\}$ is said to be *straddled* by e. Observe that is in this case, e is in the connecting triangle of the corresponding 3-sum decomposition. The second special internal 3-separation is as follows. Suppose E_1 has exactly seven edges, say $e, f_1, \ldots, f_6$. Suppose further that $\{e, f_1, f_2\}$, $\{e, f_3, f_4\}$, and $\{e, f_5, f_6\}$ are triangles of G such that no two of $\{f_1, \ldots, f_6\}$ are parallel. Then, $G[E_1]$ is a *crown* and $\{E_1, E_2\}$ is a *crown* 3-separation of G with respect to e. Observe that the crown $G[E_1]$ has three nodes of degree two, which by the 3-connectivity of G, constitute the set $V(G[E_1]) \cap V(G[E_2])$. It also has two nodes of degree four, which are the ends of e.

Let G be a 3-connected graph, and let e be an edge of G. Let $\{E_1, E_2\}$ be a crown 3-separation of G with respect to e, where $e \in E_1$. Let $\{G_1, G_2\}$ be the corresponding 3-sum decomposition. If G_2 is planar, then G is said to be *crown-planar* with respect to e.

The primary decomposition tool used in this paper is the following. Let G be a 3-connected graph, and let $e = st$ be an edge of G. If $G\backslash\{s,t\}$ is 2-connected, then define $\{G\}$ to be the e*-decomposition* of G. Otherwise, assume that $G\backslash\{s,t\}$ has a 1-separation, and let $v_1,\ldots,v_p$ denote its cut nodes. Let $V_1,\ldots,V_k$ denote the node sets of the blocks of $G\backslash\{s,t\}$. Let $(G,e)^+$ denote the graph obtained from G by adding a *virtual* edge joining every non-adjacent pair of nodes in $\{s,t\}\times\{v_1,\ldots,v_p\}$. Where $H := (G,e)^+$, the e*-decomposition* of G is defined to be the set $\{H[V_1\cup\{s,t\}],\ldots,H[V_k\cup\{s,t\}]\}$.

The main goal of this section is to prove the following theorem.

THEOREM 1. *Let G be a 3-connected graph, and let e be an edge of G. Then, G does not have a $K_{3,3}$ minor containing e if and only if every member of the e-decomposition of G is either planar, crown-planar with respect to e, or isomorphic to K_5.*

The proof of Theorem 1 is broken down into a sequence of results, which will be collected into a concise proof later in the section.

The next lemma relates 3-sum decompositions and e-decompositions. Let D be the e-decomposition of a 3-connected graph G, and let H be a member of D. Using the above notation, if $D = \{G\}$, define G to be the (unique) *end* member of D; otherwise, define the *end* members of D to be those members that correspond to the *end* blocks of $G\backslash\{s,t\}$; that is, those blocks that contain exactly one of $\{v_1,\ldots,v_p\}$.

LEMMA 2. *Let G be a 3-connected graph, let $e = st$ be an edge of G, and suppose that $G\backslash\{s,t\}$ is not 2-connected. Let G_1 be an end member of the e-decomposition D of G. Then, there exists an internal 3-separation of G straddled by e such that $\{G_1,G_2\}$, for some G_2, is the corresponding 3-sum decomposition. Moreover, $D-\{G_1\}$ is the e-decomposition of G_2.*

Proof. Observe that each end block of $G\backslash\{s,t\}$ contains at least two nodes. Therefore, each end member of D contains at least four nodes.

Since G_1 is an end member of D, there exist a corresponding end block of $G\backslash\{s,t\}$. Let B denote this end block, and let z denote unique cut node of $G\backslash\{s,t\}$ in B.

Let E_1 be those edges of G_1 that are edges of G, and define $E_2 := E(G)-E_1$. From the definition of the e-decomposition, it follows that $V(G[E_1])\cap V(G[E_2])$ consists of exactly three nodes, namely s, t, and z. From the above, G_1 has at least four nodes, and therefore at least one node not in $V(G[E_1])\cap V(G[E_2])$. By the 3-connectivity of G, this node is incident to at least three edges. These three incident edges plus the edge e are in E_1, and therefore $|E_1|\geq 4$. Similarly, G_2 has a node not in $V(G[E_1])\cap V(G[E_2])$, and therefore $|E_2|\geq 3$. Observe that G has at least five nodes, and since it is 3-connected, it has at least eight edges. Thus, either $|E_2|\geq 4$ or $|E_1|\geq 5$. In the latter case, re-define E_1 and E_2 by transferring e from E_1 to E_2. It follows that $\{E_1,E_2\}$ is an internal 3-separation of G straddled by e, the 3-sum decomposition of which is equal to $\{G_1,G_2\}$ for an appropriately defined graph G_2.

Now, observe that the graph $G_2\backslash\{s,t\}$ is equal to the graph $G\backslash\{s,t\}$ with the end block B removed. (That is, deleting all of the nodes of B, except for z, from $G\backslash\{s,t\}$.) It follows that $D-\{G_1\}$ is the e-decomposition of G_2. □

In many situations in this section, it is easier to deal with $K_{3,3}$ subdivisions rather than minors. The following result, which is straightforward and used without further reference, makes this possible.

LEMMA 3. *Let G be a 3-connected graph, and let e be an edge of G. Then, G has a $K_{3,3}$ minor containing e if and only if G has a $K_{3,3}$ subdivision containing e.* □

The next lemma shows that the property of having a $K_{3,3}$ minor containing e is inherited under e-decompositions.

LEMMA 4. *Let G be a 3-connected graph, and let e be an edge of G. Then, some member of the e-decomposition of G has a $K_{3,3}$ minor containing e if and only if G does.*

Proof. Let $e = st$, and let $H := (G,e)^+$.

First, let J be a member of the e-decomposition of G that has a $K_{3,3}$ minor containing e. Then, it has a $K_{3,3}$ subdivision, say K, containing e. If K does not contain any virtual edges, then it is also a subgraph of G, as required. If K does contain a virtual edge, it is shown each such edge of K can be replaced by a path of G in such a way that the result is a $K_{3,3}$ subdivision of G.

Let B be the unique block of $G\backslash\{s,t\}$ such that $J = H[V(B)\cup\{s,t\}]$. Let f be a virtual edge of K. Then, f joins s (say) to a node z_1 of B, where z_1 is also a cut node of $G\backslash\{s,t\}$ Observe that no other virtual edge of K can be incident to z_1, for if such an edge were to exist, it would join z_1 to t, implying that K contains a triangle on nodes $\{s,t,z_1\}$, which is impossible in a $K_{3,3}$ subdivision. Since z_1 is a cut node of $G\backslash\{s,t\}$

and since G is 3-connected, there exists a path, say Z_1, from z_1 to s in G, no edges of which are from B. Similarly, one can construct paths $Z_2, \ldots, Z_k$, one for each remaining virtual edge of K. Observe that the paths $Z_1, \ldots, Z_k$ are pairwise internally node disjoint. Thus, replacing the virtual edges of K with the paths $Z_1, \ldots, Z_k$ produces a $K_{3,3}$ subdivision of G.

Now, suppose G has a $K_{3,3}$ subdivision K containing e. Observe that $K \backslash \{s,t\}$ contains a cycle; let B be the block of $G \backslash \{s,t\}$ that contains this cycle, and let J be the unique member of the e-decomposition of G such that $J = H[V(B) \cup \{s,t\}]$. If K is a subgraph of J, then the lemma is proved. Thus, assume this is not the case. Now, observe that the edges of K not in J can be partitioned into internally node-disjoint paths, each of which starts at a node of B and ends at either s or t. Let $Z_1, \ldots, Z_k$ denote these paths, and let $z_1, \ldots, z_k$ denote their respective starting nodes. Observe that the $z_1, \ldots, z_k$ are distinct cut nodes of $G \backslash \{s,t\}$. Therefore, by the definition of the e-decomposition, each one of $z_1, \ldots, z_k$ is adjacent is to both s and t by an edge of J not in K. It follows that the paths $Z_1, \ldots, Z_k$ of K can be replaced by edges of J to produce a $K_{3,3}$ subdivision of J. $\square$

The next theorem is a well-known result of K. Wagner [23].

THEOREM 2. *Let G be a 3-connected graph. Then, G does not have a $K_{3,3}$ minor if and only if G is planar or isomorphic to K_5.* $\square$

The next result, on $K_{3,3}$ subdivisions, is due to Širáň [17]. If a graph H is a subdivision of a graph K, and s and t are non-adjacent nodes of K, then s and t are *independent* in H.

LEMMA 5. *Let G be a 3-connected graph, and let $e = st$ be an edge of G. If e is not contained in any $K_{3,3}$ subdivision of G, then, for every $K_{3,3}$ subdivision H of G, s and t are independent degree-three nodes of H.* $\square$

Let G be a graph, and let H be a subgraph of G. Let P be a path of G, the end nodes of which are nodes of H, and the internal nodes of which are not nodes of H. Then, the subgraph $H \cup P$ of G is said to be obtained from H by *adjoining* P, and P is an *adjoinable* path of G with respect to H.

Let G be a graph, and let e be an edge of G. Let H be a $K_{3,3}$ subdivision of G, and suppose that e joins two independent degree-three nodes of H. Since $K_{3,3}$ has nine edges, the graph H consists of nine paths, each of which is a subdivision of an edge of $K_{3,3}$. The six such paths that share an end with e are called the *principal* paths of H with respect to e; the remaining three paths are the *support* paths of H. The $K_{3,3}$ subdivision H of G is *good* (respectively, *bad*) with respect to e if all six (respectively, at most five) of the principal paths with respect to e consist of a single edge.

LEMMA 6. *Let G be a 3-connected graph, and let e be an edge of G. Assume that G does not have a $K_{3,3}$ minor that contains e or an internal 3-separation that is straddled by e. Then, every $K_{3,3}$ subdivision of G is good with respect to e.*

Proof. If G is planar or isomorphic to K_5, then, vacuously, every $K_{3,3}$ subdivision of G is good with respect to e. Thus, Theorem 2 implies that G has a $K_{3,3}$ minor, and thus, a $K_{3,3}$ subdivision. Let $e = st$. By Lemma 5, s and t are independent degree-three nodes in every $K_{3,3}$ subdivision of G. If the theorem is not true, then there exists a $K_{3,3}$ subdivision of G, say H, in which some principal path with respect to e, say Q_1, has at least two edges. Let y denote the end node of Q_1 not in $\{s,t\}$. Let Q_2 denote the other principal path that has y as an end node, and let S_1 denote the support path that has y as an end node. Denote the other end node of S_1 by z. Consistent with the above, assume H and Q_1 are chosen so that the number of edges in S_1 is as small as possible.

Claim: If an adjoinable path of G with respect to H has one end that is an internal node of either Q_1 or Q_2, then the other end of the path is a node of $V(Q_1 \cup Q_2 \cup S_1)$.

Proof of Claim: If the other end of the path is not in $V(Q_1 \cup Q_2 \cup S_1)$, then it is easy to check that adjoining the path to H results in a graph that has a $K_{3,3}$ minor that contains e, a contradiction. *End of Claim.*

Observe that $\{Q_1 \cup Q_2 \cup \{e\}, E(H) - (Q_1 \cup Q_2 \cup \{e\})\}$ is an internal 3-separation of H straddled by e. Since G does not have an internal 3-separation straddled by e, there exists an adjoinable path R_1 of G, one end of which, say p_1, is a internal node of Q_1 (say), and the other end of which, say r_1, is not in $V(Q_1 \cup Q_2)$. By the Claim, r_1 is a node of S_1; if it is an internal node of S_1, then a contradiction to the choice of H is obtained by adjoining R_1 to H and deleting the internal nodes of the yp_1-subpath of subpath of Q_1. Thus, $r_1 = z$.

Observe that $\{Q_1 \cup Q_2 \cup S_1, E(H) - (Q_1 \cup Q_2 \cup S_1)\}$ is an internal 3-separation of H straddled by e. Since G does not have an internal 3-separation straddled by e, there exists an adjoinable path R_2 of G with respect to H, one end of which, say p_2, is in $V(Q_1 \cup Q_2 \cup S_1)$, the other end of which, say r_2, is not in $V(Q_1 \cup Q_2 \cup S_1)$, and neither end of which is in $\{s, t, z\}$. By the Claim, p_2 is a node of S_1 and r_2 is a node of S_2 (say). Moreover, p_2 must equal y, for otherwise a contradiction to the choice of H is obtained by adjoining R_2 to H and deleting the internal nodes of the zr_2-subpath of S_2. By the Claim, R_1 and R_2 are node disjoint. Now, $H \cup R_1 \cup R_2$ has a $K_{3,3}$ minor that contains e, a contradiction. □

The following result can be seen an a generalization of Theorem 2. It was inspired by a similar type result for matroids due to Tseng and Truemper [21]. A version of this result was independently discovered by Mohar [14].

LEMMA 7. *Let G be a 3-connected graph, and let e be an edge of G. Assume that G does not have a $K_{3,3}$ minor that contains e or an internal 3-separation that is straddled by e. Then, G is planar, crown-planar with respect to e, or isomorphic to K_5.*

Proof. By Theorem 2, either G is planar, isomorphic to K_5, or has a $K_{3,3}$ minor. Thus, by Lemma 6, G has a $K_{3,3}$ subdivision, say H, that is good with respect to e. Let $e = st$, and let z denote the common end node of the three support paths of H. Let v, w, and y denote the remaining degree-three nodes of H. Let S_1, S_2, and S_3 denote the three support paths of H with respect to e, and without loss of generality, assume that the ends of S_1 are y and z.

Observe that $\{\{S_1, sy, ty, e\}, E(H) - \{S_1, sy, ty, e\}\}$ is an internal 3-separation of H straddled by e. Since G does not have an internal 3-separation straddled by e, there exists an adjoinable path R_1 of G with respect to H, one end of which is in $V(S_1)$, the other end of which is in $V(S_2)$ (say), and neither end of which is equal to z. Similarly, there exists an adjoinable path R_2 of G with respect to H, one end of which is $V(S_3)$, the other end of which is in $V(S_1)$ (say), and neither end of which is equal to z.

Observe that $\{\{e, sv, tv, sw, tw, sy, ty\}, S_1 \cup S_2 \cup S_3 \cup R_1 \cup R_2\}$ is a crown 3-separation with respect to e of $H \cup R_1 \cup R_2$. Thus, either G has a crown 3-separation with respect to e or there exists an adjoinable path R_3 of G with respect to $H \cup R_1 \cup R_2$, one end of which is in $\{s, t\}$ and the other end of which, call it r_3, is in $V(S_1 \cup S_2 \cup S_3 \cup R_1 \cup R_2)$. If $r_3 \neq z$, then observe that $H \cup R_1 \cup R_2 \cup R_3$ contains a bad $K_{3,3}$ subdivision with respect to e, which, by Lemma 6, is a contradiction. (Note, if $r_3 \in \{v, w, y\}$, then, by the 3-connectivity of G, R_3 has at least two edges.) Thus, $r_3 = z$. It can now be checked that $H \cup R_1 \cup R_2 \cup R_3$ contains a $K_{3,3}$ minor containing e, a contradiction. Thus, it can be assumed that $\{\{e, sv, tv, sw, tw, sy, ty\}, E(G) - \{e, sv, tv, sw, tw, sy, ty\}\}$ is a crown 3-separation of G. Let $\{G_1, G_2\}$ be the corresponding 3-sum decomposition.

To show that G is crown-planar with respect to e, it suffices to show that G_2 is planar. To this end, suppose this is not the case. By Lemma 1, G_2 is 3-connected. By Theorem 2, G_2 is either isomorphic to K_5 or has a $K_{3,3}$ minor. In the former case, it is easy to see that G has a $K_{3,3}$ minor containing e, a contradiction. In the latter case, G has a $K_{3,3}$ subdivision for which e does not join independent nodes, contradicting Lemma 5. □

Lemma 7 has as a corollary the following result, which was proved by Thomas [19]. The corollary, in turn, generalizes a result of Širáň [17].

COROLLARY. *Let G be an internally 4-connected graph, and let e be an edge of G. Assume that G does not have a $K_{3,3}$ minor that contains e. Then, either G is planar or isomorphic to K_5.* □

The next result examines further the structure of crown-planar graphs occuring in an e-decomposition, showing that they are "almost" planar.

LEMMA 8. *Let G be 3-connected a graph, and let e be an edge of G. Assume that G does not have a $K_{3,3}$ minor containing e or an internal 3-separation straddled by e. Let $\{E_1, E_2\}$ be a crown 3-separation of G with respect to e with $e \in E_1$. Then, G is crown-planar with respect to e if and only if $G \backslash f$ is 3-connected and planar for any edge f adjacent to e.*

Proof. First, suppose that G is crown-planar with respect to e. Let $e = st$, and let $f = sy$ be an edge of G adjacent to e. It is first shown that y has degree at least four in G. If not, then the three edges incident to y together with the edge e comprise a 3-separator of G, the corresponding 3-separation of which is an internal 3-separation straddled by e, a contradiction. Thus, y has degree at least four in G. Also, by definition, s has degree four in G.

To show that $G \backslash f$ is 3-connected, suppose that it has a 2-separation $\{F_1, F_2\}$ with $e \in F_1$. Let p and q denote the two nodes in $V(G[F_1]) \cap V(G[F_2])$. Observe that $\{F_1, F_2 \cup \{f\}\}$ is a 3-separation of G. Moreover, since both ends of f have degree at least four, it is an internal 3-separation. Also, observe that since e and f are in a triangle of G, it must be the case that $t \in \{p, q\}$. This shows that $\{F_1, F_2 \cup \{f\}\}$ is straddled by e, a contradiction.

If $G \backslash f$ is not planar, then Theorem 2 implies that it is either isomorphic to K_5 or has a $K_{3,3}$ minor and, therefore, a $K_{3,3}$ subdivision. Evidently, s has degree three in $G\backslash f$. If $G\backslash f$ is isomorphic to K_5, s has degree four in $G\backslash f$, a contradiction. Similarly, if $G\backslash f$ has a $K_{3,3}$ subdivision, then, by Lemma 5, s has degree at least four in $G\backslash f$, a contradiction.

Now, suppose that $G\backslash f$ is 3-connected and planar for any edge f adjacent to e. Let $\{G_1, G_2\}$ be the 3-sum decomposition corresponding to the internal 3-separation $\{E_1, E_2\}$. Evidently, G_2 is isomorphic to a minor of $G\backslash f$, and thus is planar. □

The proof of Theorem 1 can now be presented.

Proof of Theorem 1. First, assume every member of the e-decomposition of G is planar, crown-planar with respect to e, or isomorphic to K_5, and suppose that G has a $K_{3,3}$ minor containing e. By Lemma 4, some member J of the e-decomposition has a $K_{3,3}$ minor containing e. Clearly, J cannot be planar or isomorphic to K_5. Thus, J is crown-planar with respect to e.

Let K be a $K_{3,3}$ subdivision of J containing e. Since the maximum degree in K is three, there exists an edge f of J adjacent to e that is not in K. Thus, K is a subgraph of $J\backslash f$, contradicting Lemma 8.

Now, suppose that G has no $K_{3,3}$ minor containing e. Let J be a member of the e-decomposition of G. By Lemma 4, J has no $K_{3,3}$ minor containing e. By definition of the e-decomposition, J does not have an internal 3-separation straddled by e. Thus, by Lemma 7, J is either planar, crown-planar with respect to e, or isomorphic to K_5. □

The final two results of the section are useful in the derivation of the time complexity of the algorithms to follow. In particular, it is shown that if a simple 2-connected graph G has an edge e that is not contained in a $K_{3,3}$ minor, then the number of edges of G is bounded by $5n - 12$. The analogous well-known bound for simple planar graphs is $3n - 6$; see, for example, Diestel [4].

LEMMA 9. *Let G be a simple 2-connected graph having at least three nodes. If, for some edge e, G does not have a $K_{3,3}$ minor containing e, then G has at most $5n - 12$ edges.*

Proof. First, assume that G is 3-connected, and so Lemma 7 applies. If G is planar, then G has at most $3n-6$ edges, which since $n \geq 3$, is at most $5n-12$. If $G\backslash f$ is planar for some edge f of G and $n \geq 5$ (which is true if G is crown-planar with respect to e (by Lemma 8) or isomorphic to K_5), then G has at most $3n - 5$ edges. Since $n \geq 5$, $3n-5 \leq 5n-12$. Thus, by Lemma 7, it can be supposed that G has an internal 3-separation straddled by e. Let $\{G_1, G_2\}$ be the corresponding 3-sum decomposition, and let k denote the number of edges of G in $E(G_1) \cap E(G_2)$. By Lemma 1, G_1 and G_2 are both 3-connected and neither has a $K_{3,3}$ minor containing e. Now, using the facts that $m = |E(G_1)| + |E(G_2)| - 6 + k$, $|V(G_1)| + |V(G_2)| = n+3$ and $k \leq 3$, the result follows by induction.

Second, suppose that G is 2-connected, but not 3-connected. Let $\{E_1, E_2\}$ be a 2-separation of G with $e \in E_1$. Let $\{G_1, G_2\}$ be the corresponding 2-sum decomposition, and let f be the connecting edge. Observe G_1 (say) might not be simple because of a possible edge parallel to f; if such an edge exists, denote it by g. By Lemma 1, G_1, G_2, and $G_1\backslash g$ (if g exists) are 2-connected. Also, it is straightforward to check neither G_1 nor $G_1\backslash g$ (if g exists) has a $K_{3,3}$ minor containing e, and that G_2 does not have a $K_{3,3}$ minor containing f. Now, using the facts that $|V(G_1)| + |V(G_2)| = n+2$ and $m = |E(G_1)| + |E(G_2)| - 2$, the result follows by induction. □

LEMMA 10. *Let G be a 3-connected graph, and let e be an edge of G. Then, the total number of edges occuring in the members of the e-decomposition of G is at most $m + 5n$.*

Proof. Let k_1 denote the total number of edges that belong to exactly one member of the e-decomposition, let k_2 denote the total number of times that the edge e appears in some member of the e-decomposition, and let k_3 denote the total number of remaining edges; that is, those edges, other than e, that appear in more than one member of the e-decomposition. Thus, the total number of edges occuring in the members of the e-decomposition of G is $k_1 + k_2 + k_3$.

Consider k_1. Observe that the only edges that appear in exactly one member of the e-decomposition must be edges of G. Thus, $k_1 \leq m$.

Consider k_2. Let $e = st$. The edge e occurs exactly once in every member of the e-decomposition. The number of members of the e-decomposition is equal to the number of blocks of $G\backslash\{s, t\}$, which in turn is bounded by n. Thus, $k_2 \leq n$.

Consider k_3. Observe that the edges, other than e, that appear in more than one member of the e-decomposition are precisely those edges of $H := (G, e)^+$ that join either s or t to a cut node of $G\backslash\{s, t\}$. Let $\{v_1, \ldots, v_p\}$ denote the cut nodes of $G\backslash\{s, t\}$. Consider an edge f joining v_1 to s. Then, the number of times f appears in some member of the e-decomposition is equal to the number of blocks of $G\backslash\{s,t\}$ that contain v_1; denote this number by d_1. Thus, $k_3 = 2\sum_{i=1}^{p} d_i$. Now, it is easy to see (by induction, for example) that $\sum_{i=1}^{p} d_i$ is bounded by twice the number of blocks of $G\backslash\{s,t\}$, which in turn is bounded by $2n$. Therefore, $k_3 \leq 4n$. □

Combining Lemmas 9 and 10 evidently yields an $O(n)$ bound on the total number of edges occuring in the members of the e-decomposition of a graph G having no $K_{3,3}$ minor containing e.

3. A recognition algorithm

This section provides an algorithm for recognizing whether a graph G contains a $K_{3,3}$ minor containing a fixed edge e of G. The algorithm relies on having the e-decomposition of G on hand. Thus, the first step is to compute the e-decomposition.

Algorithm Decomp below computes the e-decomposition by essentially implementing its definition. For this algorithm, let G be a 3-connected graph, and let $e = st$ be an edge of G. It assumed that G is represented by adjacency lists. For a node y of G, denote the adjacency list of y by $L_G(y)$.

algorithm decomp;
begin
  delete nodes s and t from G and let H be the resulting graph;
  compute the blocks $B_1, \ldots, B_k$ and cut nodes of H;
  for $i = 1, \ldots, k$ **do**
    begin
      $V(J_i) := V(B_i) \cup \{s,t\}$;
      $E(J_i) := \{vy \in E(G) | \{v,y\} \subseteq V(J_i)\} \cup$
      $\{sv | v$ is a cut node of $H; sv \notin E(G)\} \cup$
      $\{tv | v$ is a cut node of $H; tv \notin E(G)\}$;
    end;
end;

PROPOSITION 1. *Algorithm Decomp correctly computes the e-decomposition of G in $O(m+n)$ time.*

Proof. The correctness of the algorithm follows directly from the definition of the e-decomposition.

The first step is to construct the graph $H := G\backslash\{s,t\}$. To do this, one first copies all the adjacency lists, except for those corresponding to s and t. Then, one scans the adjacency lists corresponding to the neighbors of s or t, removing each occurrence of s and t. Clearly, this can be done in $O(m+n)$ time.

The second step is to compute the blocks of H. This can be done in $O(m+n)$ time using the algorithm of Tarjan [18].

The last step is to add the nodes s and t to each block of H. First, for each node y of H, construct a list B_y of the set of blocks of H that contain y; evidently, nodes that appear in more than one block are the cut nodes of H. This requires scanning the list of nodes in each block once, and so requires $O(\sum_{i=1}^{k} |V(B_i)|)$ time, which is easily seen (by induction, for example) to be $O(n)$. Now, consider the node s. Create an (initially empty) adjacency list for s for each block of H. Then, scan $L_G(s)$, and for each node y on $L_G(s)$ and for each $z \in B_y$, add y to $L_{J_z}(s)$ and s to $L_{J_z}(y)$. If y is a cut node of H, *mark* it. Finally, for each unmarked cut node v and for each $z \in B_v$, add v to $L_{J_z}(s)$ and s to $L_{J_z}(v)$. In this way, s (and analogously, t) can be added in $O(n)$ time. □

The recognition algorithm is now described.

algorithm recog;
begin
  if $m > 5n - 12$ **then** G has a $K_{3,3}$ minor containing e;
  compute an e-decomposition D of G;
  flag ← true;
    while $D \neq \emptyset$ **do**
      begin
        choose $H \in D$;
        if H is not planar, crown-planar with respect to e, or isomorphic to K_5
        then flag ← false;
        $D \leftarrow D - \{H\}$;
      end;
    if flag = false **then** G has a $K_{3,3}$ minor containing e;**else** G does not have a $K_{3,3}$ minor containing e;
end;

PROPOSITION 2. *Algorithm Recog correctly determines whether a 3-connected graph G has a $K_{3,3}$ minor containing e in $O(n)$ time.*

Proof. The correctness of the algorithm follows directly from Theorem 1 and Lemma 9.

Computing the e-decomposition of G requires $O(n)$ time by Proposition 1 and the first **if** statement.

Determining whether a given H is planar requires $O(|V(H)|)$ time using the algorithm of Hopcroft and Tarjan [10].

Determining whether a given H is isomorphic to K_5

can evidently be done in constant time.

Determining whether a given H is crown-planar with respect to e can be done in $O(|V(H)|)$ time as follows. First, determine if H has a crown 3-separation with respect to e; this requires constant time since a crown has just seven edges. Next, form the corresponding 3-sum decomposition, and determine if the member of the 3-sum decomposition not containing e is planar; this requires $O(|V(H)|)$ time.

By Lemmas 9 and 10, $\sum_{H \in D} |V(H)|$ is $O(n)$. □

4. The Target-Flow Problem

As part of the maximum-flow algorithm of the next section, the following *Target-Flow Problem* needs to be solved: given an instance (G, e, u) of the maximum-flow problem and a *target* flow value v, find a feasible flow in (G, e, u) of value v, or determine no such flow exists. In principle, the target-flow problem can be easily reduced to the maximum-flow problem – just subdivide the edge e into two edges, one of capacity v and one of capacity zero. The nature of the maximum-flow algorithm presented in the next section, however, precludes this approach. The following algorithm will suffice.

algorithm target;
begin
  compute a maximum flow of value z and a minimum cut C in (G, e, u);
  if $v > z$ **then** (G, e, u) does not have a target flow of value v;
  while $u(C) > v$ **do**
    begin
      choose an edge f of C having positive capacity;
      $\delta \leftarrow \min\{u_f, u(C) - v\}$;
      $u_f \leftarrow u_f - \delta$;
    end;
  compute a maximum flow in (G, e, u);
end;

PROPOSITION 3. *Algorithm Target correctly computes a target flow of value v in (G, e, u). Moreover, the running time of the algorithm is equal to that of solving the maximum-flow problem (G, e, u).*

Proof. If $v > z$, then clearly (G, e, u) does not have a target flow of value v. Thus, assume $v \leq z$. Let C be the minimum cut computed by the algorithm. Consider the first execution of the **while** loop. In particular, let f and δ be as defined, and let u' be the capacity vector resulting from the first execution of the **while** loop.

Now, consider any cut D in G. If $f \notin D$, then $u'(D) = u(D)$, and if $f \in D$, then $u'(D) = u(D) - \delta$. It follows that, after one execution of the **while** loop, C is a minimum cut in (G, e, u').

Repeating the above argument implies that after the final execution of the **while** loop, C is a minimum cut, and its capacity is equal to the target value v. Therefore, by the Max-Flow-Min-Cut Theorem [5], the final maximum-flow computation of the algorithm produces the desired flow.

With respect to the running time of the algorithm, clearly the dominant steps are the two maximum-flow computations. □

5. A Maximum-Flow Algorithm

Theorem 3 below is the main result of the paper. The basic idea of the proof is to use 2- and 3-sum decompositions to reduce the original maximum-flow problem on G to a sequence of maximum-flow problems on "easy" graphs.

The relationship between 2- and 3-sum decompositions and maximum flows is not a new; for example, it can be seen in the matroid work of Seymour [16] and Truemper [20]. In particular, the work of Truemper [20] shows how to compute a maximum flow using 2- and 3-sum decompositions. Applied here, the Truemper approach would lead to a polynomial-, but not linear-time, algorithm. The proof of Theorem 3 below yields a linear-time algorithm.

THEOREM 3. *The maximum-flow problem (G, e, u), where G is simple and has no $K_{3,3}$ minor containing e, can be solved in $O(n)$ time.*

Proof. The proof is via a sequence of reductions. Throughout the proof, it is assumed, by Lemma 9, that m is $O(n)$. Also, as usual, let $e = st$.

(I) *Reduction to the* 2*-Connected Case*

The first step is to show that it can be assumed that G is 2-connected. It is well known that, for any edge f not contained in the block of G that contains e, there exists a maximum flow in which the flow on f is zero. Thus, computing a maximum flow can be confined to the block of G that contains e, which can be computed in $O(n)$ time; see, for example, Tarjan [18]. Thus, it is assumed that G is 2-connected.

(II) *Reduction to the* 3*-Connected Case*

The next step is to show that the maximum-flow problem (G, e, u) can be reduced in linear time to solving a sequence of maximum-flow problems, where each problem in the sequence is defined on a graph that is 3-connected and does not contain a $K_{3,3}$ minor using its return edge, and such that total size of this sequence of graphs in linear in the size of G.

(IIa) *Maximum Flows and* 2*-Sum Decompositions*

The first step in defining this sequence is to examine the relationship between an instance of the maximum-flow problem and a 2-sum decomposition in the underlying graph. So, suppose that G is not 3-connected, and let $\{E_1, E_2\}$ be a 2-separation of G with $e \in E_1$. Let $\{G_1, G_2\}$ be the corresponding 2-sum decomposition, and let f be the connecting edge. By Lemma 1, G_1 and G_2 are 2-connected. Moreover, it is straightforward to verify that G_1 (respectively, G_2) does not have a $K_{3,3}$ minor containing e (respectively, f). First, consider solving the maximum-flow problem (G_2, f, u^2), where each edge of G_2, except for f, inherits is capacity from G. Let v^2 denote the maximum-flow value. Now, consider (G_1, e, u^1), where $u^1_f = v^2$ and all other edges of G_1 inherit their capacity from G. Then, it is well known and not hard to see that the maximum-flow value for (G_1, e, u^1) is equal to that of (G, e, u). Moreover, a maximum flow for (G, e, u) can be found by first computing a maximum flow (D_1, x^1) in (G_1, e, u^1), then computing a feasible flow (D_2, x^2) of value x^1_f in (G_2, f, u^2) (using Algorithm Target), and finally by combining these two flows into a flow (D, x) for (G, e, u).

Combining (D_1, x^1) and (D_2, x^2) into a maximum flow (D, x) of (G, e, u) is done as follows. First, it is assumed that the source node for the flow (D_2, x^2) coincides with the tail of f in D_1; if not, then the orientation of the arcs in D_2 need to be reversed. Second, by definition, each edge of G appears in exactly one of G_1 or G_2. Thus, one can construct a feasible flow (D, x) for (G, e, u) by simply, for each edge of G, taking its orientation and flow from either (D_1, x^1) or (D_2, x^2), as appropriate. Now, it is straightforward to see that (D, x) is a maximum flow for (G, e, u).

(IIb) *The Reduction Procedure*

To turn the above relationship between maximum flows and 2-sum decomposition into a computationally efficient algorithm requires two straightforward ideas. First, one chooses the 2-sum decomposition judiciously, and second one applies this judicious choice recursively. Tutte [22] and Hopcroft and Tarjan [9] showed that one can always find a 2-separation $\{E_1, E_2\}$ of G such that $e \in E_1$ and in the resulting 2-sum decomposition $\{G_1, G_2\}$, G_2 is either 3-connected, a cycle, or a bond (i.e., the planar dual of a cycle). Applying this choice of 2-sum decomposition recursively reduces the maximum-flow problem on G to solving a sequence of maximum-flow and target-flow problems on a collection of graphs, say $\{H_1, \dots, H_p\}$, every member of which is either 3-connected, a cycle, or a bond. Moreover, Hopcroft and Tarjan [9] showed that the sequence of 2-separations necessary to generate $\{H_1, \dots, H_p\}$ can be found in $O(n)$ time and that the size of the collection, i.e., $\sum_{i=1}^{p} |V(H_i)|$, is $O(n)$. Observe that solving a maximum-flow or target-flow problem on a cycle or bond can trivially be done in linear time (in the size of the cycle or bond). Thus, in $O(n)$ time, the maximum-flow problem on G can be reduced to solving a sequence of maximum-flow and target-flow problems, each of which is on a graph that is 3-connected and does not have a $K_{3,3}$ minor using its return edge. Moreover, the total size of the graphs in the sequence is $O(n)$. So, in particular, if each of these 3-connected maximum-flow or target-flow problems can be solved in linear time, so can the original problem (G, e, u). By Proposition 3, each of the target-flow problems is computationally equivalent to a maximum-flow problem. Thus, it suffices to consider the maximum-flow problem when G is 3-connected.

(III) *Reduction to the Planar Case*

Assume that G is 3-connected. By Lemma 7, G is either planar, crown-planar with respect to e, isomorphic to K_5, or has an internal 3-separation straddled by e. These cases are considered one at a time. As a first step, it is shown that one can recognize which case is applicable in $O(n)$ time. Clearly, recognizing if G is isomorphic to K_5 can be done in constant time. Also, it is well known that planarity can be recognized in $O(n)$ time; see, for example, Hopcroft and Tarjan [10]. Determining whether G has an internal 3-separation straddled by e can be done in $O(n)$ time using Algorithm Decomp, since G has an internal 3-separation straddled by e if and only if its e-decomposition has at least two members. Finally, by Lemma 7, the only other possibility for G is that it is crown-planar with respect to e. Moreover, in this last case, it can further be assumed that G does not have an internal 3-separation straddled by e.

$(IIIa)$ *The Base Cases*

This subcase considers that cases when G is either planar, crown-planar with respect to e, or isomorphic to K_5. For each of these three cases, it is shown how to solve the maximum-flow problem (G, e, u) in linear time. For the planar case, this is done using the Hassin [7] reduction to the shortest-path problem. For the latter two cases, this is done by reducing them to the planar case.

The Hassin reduction can be applied to either directed or undirected graphs. For what follows, it is better to use directed graphs. That is, the undirected maximum-flow problem is first converted to an equivalent directed maximum-flow problem, to which the Hassin reduction is applied. The reason for doing this is to ensure that the resulting solution (D, x) to the maximum-flow problem (G, e, u) satisfies the following property P_1. Direct application of the Hassin reduction to (G, e, u) does not ensure this.

Property P_1: No arc of D is directed into s or out of t.

Assuming G is planar, consider the following directed graph G^*. Each of G edge incident to s is directed away from s, each edge incident to t is directed towards t, and each remaining edge is replaced by a pair of oppositely directed arcs. The arc capacities for G^* are inherited from G; that is, for each arc of G^*, define its capacity to be equal to the corresponding edge of G; in particular, the two arcs in an oppositely directed pair have the same capacity. Let u^* denote the resulting capacity vector. Then, it is straightforward to see that solving the directed maximum-flow problem (G^*, s, t, u^*) solves (G, e, u).

ByHassin [7],the maximum-flow problem(G^*, s,t,u^*) can reduced to a shortest-path problem on the planar dual of G^*. Finding the planar dual can be done in $O(n)$ time; see, for example, Hopcroft and Tarjan [10] and Mehlhorn and Mutzel [12]. Solving the shortest-path problem on the planar dual requires $O(n)$ time using the algorithm of Henzinger, Klein, Rao, and Subramanian [8]. Converting the shortest-path solution into one for maximum-flow problem is straightforward, and can be done in $O(n)$ time. Thus, solving the maximum-flow problem (G, e, u) when G is planar requires $O(n)$ time.

Now, consider the case where G is isomorphic to K_5 or crown-planar with respect to e. In the latter case, it is assumed that G does not have an internal 3-separation straddled by e. Observe that e is in a triangle of G. Let f and g be the other two edges of the triangle, and without loss of generality, assume that $u_f \leq u_g$. Consider reducing the capacity of both f and g by u_f, and let u' denote the resulting capacity vector. Observe that every st-cut of G contains exactly one of f and g. From the Max-Flow-Min-Cut Theorem [5], it follows that the maximum-flow value for (G, e, u') is equal to that of (G, e, u) minus u_f. Thus, a maximum flow for (G, e, u) can be found by first solving the maximum-flow problem (G, e, u'), and then augmenting the flow on f and g by u_f units. Since $u'_f = 0$, in solving (G, e, u'), the edge f can be a priori deleted. By Lemma 8 (or by inspection, in the case of K_5), $G \backslash f$ is planar. Therefore, both the crown-planar case and the K_5 case reduce to the planar case in $O(n)$ time.

(IIb) *Maximum Flows and 3-Sum Decompositions*

The next step is to analyze the relationship between an instance of the maximum-flow problem and a 3-sum decomposition, defined relative to an internal 3-separation straddled by e, in the underlying graph. The step is conceptually similar to (IIa), although the details are more complicated. In particular, it is shown that such an instance of the maximum-flow problem can be reduced to solving maximum-flow problems on the members of the associated 3-sum decomposition.

Let $\{E_1, E_2\}$ be an internal 3-separation straddled by e. Let $\{G_1, G_2\}$ be the corresponding 3-sum decomposition, and let T be the connecting triangle. Let $\{s, t, z\}$ denote the node set of T. Two maximum-flow problems are defined on G_2. For both problems, initially define the capacity of all virtual edges to be zero; all other edges inherit their capacity from G. Let $f = sz$ and $g = tz$ denote the two edges of $T - \{e\}$. For the first maximum-flow problem on G_2, denoted (G_2, e, u^{21}), re-define the capacity of g to be ∞, and let v^{21} denote resulting maximum-flow value. For the second maximum-flow problem on G_2, denoted (G_2, e, u^{22}), re-define the capacity of f to be ∞, and let v^{22} denote the resulting maximum-flow value. Now, consider (G_1, e, u^1), where u^1 is defined as follows. All edges of G_1, except for those in T, inherit their capacity from G; edges e, f, and g have respective capacities of zero, v^{21}, and v^{22}.

Let (D_1, x^1) be a maximum flow for (G_1, e, u^1). The goal is to show that this flow can be extended to a maximum flow (D, x) for (G, e, u). It is assumed that (D_1, x_1) satisfies Property P_1. In addition, consider the following property.

Property P_2: The flow on the arc from $\{f, g\}$ having the smaller capacity with respect to u^1 is equal to its capacity, and the flow on the arc having the greater capacity is at least that of the edge having the smaller capacity.

It is claimed that, without loss of generality, it can be assumed that (D_1, x_1) satisfies Property P_2. To see this, let δ denote the smaller of the two capacities of f and g, and consider modifying u^1 by reducing the capacities of f and g by δ. Let u' denote the resulting capacity vector. Observe that every st-cut of G_1 contains exactly one of f and g. From the Max-Flow-Min-Cut Theorem [5], it follows that the maximum-flow value for (G_1, e, u') is equal to that of (G_1, e, u^1) minus δ. Now, consider a maximum flow (D', x') for (G_1, e, u'). Augmenting the flow on f and g be δ units produces a maximum flow for (D_1, e, u^1) satisfying Property P_2 as claimed.

In extending the maximum flow (D_1, x^1) for (G_1, e, u^1) to a maximum flow (D, x) for (G, e, u), two cases are considered. For the first case, which assumes $x^1_f \geq x^1_g$, the full details are below. The second case, which assumes $x^1_g \geq x^1_f$, is symmetric with the first and thus is left to the reader.

Assuming $x^1_f \geq x^1_g$, define $\gamma := x^1_f - x^1_g$. Consider the instance of the maximum-flow problem obtained from (G_2, e, u^{21}) by re-defining the capacity on the edge g to be γ if g is a virtual edge, and to be $u_g + \gamma$ otherwise; denote this problem by (G_2, e, u^*), and let (D_2, x^2) be a maximum flow for (G_2, e, u^*). It is assumed that (D_2, x^2) satisfies Property P_1.

Given (D_1, x^1) and (D_2, x^2), a flow (D, x) for (G, e, u) is constructed as follows. Each edge of G that is not in T takes its orientation and flow from either (D_1, x^1) or (D_2, x^2), as appropriate. The edges in T are handled as follows. Edge e evidently is assigned a flow of zero, and is oriented from s to t. If f is an edge of G (i.e., it is not a virtual edge of T), then it is oriented from s to z, and assigned a flow equal to the x^2_f. Finally, if g is an edge of G, then it is oriented from z to t and assigned a flow equal to $x^2_g - \gamma$. Note, Property P_1 is satisfied.

To show that (D, x) is a maximum flow for (G, e, u), it is first shown that it is feasible. It is easily seen that the flow satisfies the capacity constraints on the edges. Also, for any node other than s, t, or z, it easily seen that the net flow at the node is equal to zero. Consider node z. The net flow at z is equal to zero in both (D_1, x^1) and (D_2, x^2). In constructing (D, x), the flows on f and g from (D_1, x^1) are ignored, which, by the definition of γ and Property P_1, contributes $-\gamma$ to the net flow at z in (D, x). Also, in constructing (D, x), the flow on g in (D_2, x^2) is decreased by γ, which, by Property P_1, contributes $+\gamma$ to the net flow at z in (D, x). Thus, the net flow at z in (D, x) is zero. Finally, it needs to be shown that the nonnegativity constraints are satisfied. In particular, it needs to be shown that the flow on g is nonnegative. This follows immediately from the next claim.

Claim: *In* (D_2, x^2), $x^2_g \geq \gamma$.

Proof of Claim: The claim is proved by showing that either γ equals zero, or that the edge g is in a minimum cut of (D_2, x^2) with respect to the capacity vector u^*, which implies (by a standard network-flow result – see Ahuja, Magnanti, and Orlin [1]) that the flow on g is equal to its capacity, and therefore at least γ. First, suppose that $v^{21} < v^{22}$. Then, Property P_2 implies that $\gamma = 0$. On the other hand, suppose that $v^{22} \leq v^{21}$. By definition, $\gamma = x^1_f - x^1_g$, which implies that $\gamma \leq v^{21} - x^1_g$. By Property P_2, the right-hand side is equal to $v^{21} - v^{22}$. That is, $v^{22} + \gamma \leq v^{21}$. Observe $v^{22} + \gamma$ (respectively, v^{21}) is the capacity of a minimum-capacity st-cut of (G_2, e, u^*) that contains g (respectively, f). In other words, there exists a minimum cut of (G_2, e, u^*) that contains g, which implies that g is in a minimum cut of (D_2, x^2) as required. *End of Claim*

It has just been shown that (D, x) is a feasible flow for (G, e, u). The next step is to show that it is a maximum flow. To this end, it is first shown that the value of the flow (D, x) is equal to the value of the flow (D_1, x^1). From the definition of (D, x), this can be done by showing that the value of the flow (D_2, x^2) is equal to x^1_f. This, in turn, is done by showing that the capacity of a minimum cut in (G_2, e, u^*) equals x^1_f. If a minimum cut of (G_2, e, u^*) contains f, then the minimum-cut capacity equals v^{21}. If a minimum cut of (G_2, e, u^*) contains g, then the minimum-cut capacity equals $v^{22} + \gamma$. Thus, the minimum-cut capacity in (G_2, e, u^*) equals $\min\{v^{21}, v^{22} + \gamma\}$. Now, if $v^{21} \leq v^{22}$, then Property P_2 implies that implies the minimum-cut capacity in (G_2, e, u^*) equals x^1_f, as required. If, on the other hand, $v^{22} \leq v^{21}$, then Property P_2 implies $v^{22} = x^1_g$, which, in turn, implies that $v^{22} + \gamma = x^1_f$. Therefore, the minimum-cut capacity in (G_2, e, u^*) equals $\min\{v^{21}, x^1_f\}$, which equals x^1_f, as required.

The second, and final step, in showing that (D, x) is a maximum flow for (G, e, u) is to show that the capacity of a minimum cut in (G, e, u) is no more than that in (G_1, e, u^1). The result then follows from the Max-Flow-Min-Cut Theorem [5]. Let X_1 be a minimum cut of (G_1, e, u^1). Suppose that X_1 contains f; an analogous argument can be made if X_1 contains g. By definition, there exists an st-cut, say Y_1, of (G_2, e, u^1) that contains f and has capacity v^{21}. It is now straightforward to see that the set consisting of $X_1 \cup Y_1$ minus any virtual edges is an st-cut of G, the capacity of which is equal to that of X_1.

$(IIIc)$ *The Reduction Procedure*

The final step in the proof of the theorem is now at hand. In particular, it needs to be shown that the above relationship between maximum flows and 3-sum decompositions can be turned into a computationally efficient procedure for reducing the 3-connected case to the planar case.

Similar to (IIb), this is done by first making a judicious choice for the 3-sum decomposition, and then applying this choice recursively. In particular, by $(IIIb)$, Theorem 1, and recursive application of Lemma 2, the maximum-flow problem on G reduces to solving a sequence of maximum-flow problems on the members of the e-decomposition of G, each of which is either planar, crown-planar with respect to e, or isomorphic to K_5. By Proposition 1, finding the e-decomposition can be done in $O(n)$ time. By $(IIIa)$, all of these individual maximum-flow problems can be solved in linear time. By $(IIIa)$ and Lemma 10, the total time spent solving the maximum-flow problems is $O(n)$. Thus, it follows that the maximum-flow problem (G, e, u) can be solved in $O(n)$ time. $\square$

6. Directed Graphs

The main result of this paper, Theorem 3, can be extended to directed graphs in a straightforward manner. That is, the maximum-flow problem on a directed graph D can be solved in $O(n)$ time provided that D is simple (in the directed sense) and that the underlying graph of D does not have a $K_{3,3}$ minor containing the return edge. The proof follows the same basic steps – one decomposes the underlying graph using 2-sum and 3-sum decompositions to reduce the original problem to solving maximum-flow problems on directed planar and crown-planar graphs, and directed K_5's. The details are left to the reader.

7. Acknowledgments

I thank Robin Thomas and Klaus Truemper for helpful discussions.

References

[1] Ahuja, R. K., T. L. Magnanti, and J. B. Orlin. 1993. *Network Flows – Theory, Algorithms, and Applications,* Prentice Hall, Englewood Cliffs, New Jersey.

[2] Barahona, F. 1983. The max-cut problem on graphs not contractible to K_5. *Operations Research Letters* 2, 107 – 111.

[3] Berge, C. and A. Ghouila-Houri. 1965. *Programming, Games and Transportation Networks*. Wiley, New York.

[4] Diestel, R. 2000. *Graph Theory*. Springer-Verlag, New York.

[5] Ford. L. R. and D. R. Fulkerson. 1956. Maximal flow through a network. *Canadian Journal of Mathematics* 8, 399 – 404.

[6] Ford, L. R. and D. R. Fulkerson. 1957. A simple algorithm for finding maximal network flow and an application to the Hitchcock problem. *Canadian Journal of Mathematics* 9, 210 – 218.

[7] Hassin, R. 1981. Maximum flow in (s,t)-planar networks. *Information Processing Letters* 13, 107.

[8] Henzinger, M. R., P. Klein, S. Rao, and S. Subramanian. 1997. Faster shortest-path algorithms for planar graphs. *Journal of Computer and Systems Sciences* 55, 3 – 23.

[9] Hopcroft, J. E. and R. E. Tarjan. 1973. Dividing a graph into triconnected components. *SIAM Journal on Computing* 2, 135 – 158.

[10] Hopcroft, J. E. and R. E. Tarjan. 1974. Efficient planarity testing. *Journal of the ACM* 21, 549 – 568.

[11] Itai, A. and Y. Shiloach. 1979. Maximum flow in planar networks. *SIAM Journal on Computing* 8, 135 – 150.

[12] Mehlhorn, K. and P. Mutzel. 1996. On the embedding phase of the Hopcroft and Tarjan planarity testing algorithm. *Algorithmica* 16, 233 – 242.

[13] Manor, R. and M. Penn. 1998. An extended planar algorithm for maximum integral two-flow. *Networks* 32, 67 – 76.

[14] Mohar, B. 2004. Personnel communication.

[15] Schrijver, A. 2003. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer-Verlag, Berlin.

[16] Seymour, P. D., 1977. The matroids with the max-flow min-cut property. *Journal of Combinatorial Theory, Series B*, 23, 189 – 222.

[17] Širáň, J. 1983. Edges and Kuratowski subgraphs of non-planar graphs. *Mathematische Nachrichten* 113, 187-190.

[18] Tarjan, R. E., 1972. Depth first search and linear graph algorithms. *SIAM Journal on Computing* 1, 146 – 160.

[19] Thomas, R. 2003. Personal communication.

[20] Truemper, K. 1987. Max-flow min-cut matroids: Polynomial testing and polynomial algorithms for maximum flow and shortest routes. *Mathematics of Operations Research* 12, 72 – 96.

[21] Tseng, F. T. and K. Truemper. 1986. A decomposition of the matroids with the max-flow min-cut property. *Discrete Applied Mathematics* 15, 329 – 364.

[22] Tutte, W. T. 1984. *Graph Theory*. Addison-Wesley, Reading, Massachusetts.

[23] Wagner, K. 1937. Über eine Erweiterung eines Satzes von Kuratowski. *Deutsche Mathematik* 2, 280 – 285.

[24] West, D. B. 1996. *Introduction to Graph Theory*. Prentice Hall, Upper Saddle River, New Jersey.

Received 20 May, 2007; revised 14 January 2008; accepted 29 January 2008