

Non-Interfering Network Flows

C. McDiarmid
*Corpus Christi College,
Oxford,
England.*

B. Reed
*Institute for
Discrete Mathematics,
Bonn, Germany.*

A. Schrijver
*Centre for Mathematics
and Computer Science,
Amsterdam,
The Netherlands.*

B. Shepherd
*Institute for
Discrete Mathematics,
Bonn, Germany.*

Abstract. We consider a generalization of the maximum flow problem where instead of bounding the amount of flow which passes through an arc, we bound the amount of flow passing “near” an arc. Nearness is specified by an extra distance parameter d . When $d = 0$ we get the usual network flow and $d = 1$ corresponds to bounding the flow through the nodes. A polynomial time algorithm is given to solve the max-flow and min-cost non-interfering flow problems for $d = 2$ and it is shown that the problems become NP-hard for $d \geq 3$. A polynomial time algorithm is outlined for arbitrary d when the underlying network is planar and how an integral flow can be obtained from a fractional one. Finally, we describe relationships with induced circuits and perfect graphs, VLSI chip design and the Hilbert basis problem.

1 Introduction

Historically, network (s, t) flows were studied as a model for the problem of shipping commodities through a network from a supply source to a destination where the arteries of the network had capacities on the amount of flow which could be shipped along them. Thus a solution to the *maximum (s, t) flow problem* would be interpreted in the graph $G = (V, E)$ representing the network as a maximum collection of $s - t$ paths such that for each edge, the number of paths containing that edge is no more than its capacity. (We presently restrict ourselves to undirected networks but we consider later some directed versions of the problem.) Ford and Fulkerson showed that it was enough to solve the problem

$$(1) \quad \begin{aligned} \max & \quad 1 \cdot y \\ y \cdot M & \leq u \\ y & \geq 0 \end{aligned}$$

where $u \geq 0$ is an integral vector of capacities on the edges and M is a matrix whose rows are indexed by the set of (s, t) paths and whose columns are indexed by the edges E . For a path P , its associated row in M is the incidence vector of its edge set. A feasible vector y for (1) is called a u -capacitated $s - t$ flow and its *value* is $1 \cdot y$.

The theory of network flows continues to be of interest in particular due to its relationship to wiring problems arising in the design of microchips. This imposes new

constraints on the problem. In this paper we consider an extension which models the phenomenon of interference surrounding a wire, or path. Thus our aim is to solve (1) where we bound the amount of flow which passes within a specified distance of an edge. To be more precise, for an integer d and $S \subset E \cup V$, the d -environment of S , denoted by $E_d(S)$, consists of those elements $\beta \in E \cup V \setminus \{s, t\}$ for which there is a path P with at most d elements of $E \cup V$ such that (i) P is not incident with s or t and (ii) P connects β to an element of S (N.B. we count both edges and nodes in the path P). Note that the 1-environment of the edge set of an $s-t$ path P consists of the edges and internal nodes of that path; the 2-environment is the union of the 1-environment and the edges "hanging" from internal nodes of P .

Consider now the LP (1) where the rows of M are the incidence vectors of d -environments of the edge sets of $s-t$ paths. Let u' be an integral vector of capacities on E . An *extension* u^* of u' is a vector of capacities on V where (i) the capacity of a node is at least as large as the minimum u' capacity of an edge incident to it and (ii) for each edge xy , one of u_x^* or u_y^* is bounded above by u'_{xy} . An integral capacity vector of the form $u = (u', u^*)$ where u^* is an extension of u' , is called *tamed*. In the case that we have a tamed capacity vector and M is as described above, call the the resulting LP (1) the *maximum (u, d) -capacitated (s, t) flow* (or simply maximum (u, d) flow) *problem*. Denote this problem by $MF(G, s, t, u, d)$, or by $MF(u, d)$ when no confusion arises, and let $\mu(G, s, t, u, d)$ denote the value of a maximum (u, d) flow. For $d = 0$, this is just the usual maximum network flow problem. For $d = 1$, $MF(G, s, t, u, d)$ is equivalent to network flows with node capacities. This is because if u^* is an arbitrary vector of integral capacities on V , then there is an edge capacity vector of which u^* is an extension.

Note that when d is even, if a path weighting y is infeasible for (1), then there is a violated upper bound constraint which corresponds to an edge. This is because if some edge is in the $2k$ -environment for a node $v \neq s, t$, then it is also in the environment of any edge α incident to v . Thus if v 's constraint is violated, then so too is α 's constraint. Similarly, if d is odd, then infeasibility of a path weighting y implies that some node constraint is violated. (N.B. We use here the fact that our capacity functions are tame.) Thus for d even (respectively odd) let M_d be a matrix whose rows are the incidence vectors of the edges (respectively nodes) in the d -environments of $s-t$ paths. A vector is called a d -vector if it assigns weights to edges or nodes when d is even or odd respectively. If we let u be an integral d -vector, then the above remarks together imply that $MF(u, d)$ is equivalent to:

$$(2) \quad \begin{aligned} \max & \quad 1 \cdot y \\ & y \cdot M_d \leq u \\ & y \geq 0. \end{aligned}$$

We take this as our standard formulation of $MF(u, d)$. It is interesting to observe that this says that the "min-cuts" for (u, d) flows (i.e., dual solutions) will correspond to weighted sets of edges or nodes depending on whether d is even or odd. This provides a different perspective of the Menger edge and node cut theorems, (i.e., that they fit into this larger scheme which is based on some notion of parity).

In Section 2 we show how to solve $MF(u, 2)$ in polynomial time and show that $MF(u, d)$ is NP-hard for $d \geq 3$. We also consider the problem of computing a

minimum cost $(u, 2)$ flow.

In Section 3 we consider the maximum (u, d) -capacitated *integral* (s, t) flow problem where we require an optimal integral solution to (2). We denote this problem by $MF'(G, s, t, u, d)$ and its optimal value by $\mu'(G, s, t, u, d)$. Note that $\mu'(G, s, t, \bar{1}, 2)$ is at least two if and only if s, t lie on a common circuit without chords, i.e., an *induced circuit*. The problem of determining whether two nodes lie on an induced circuit has been shown to be NP-complete by Fellows [2]. Thus computing integral $(u, 2)$ flows (and in fact integral (u, d) flows for $d \geq 2$) is NP-hard. On the other hand, we describe a polynomial time algorithm given in [9] which solves the integral (u, d) flow problems for planar networks. This work has several extensions and applications which we discuss in Section 4.

2 Computing (u, d) Flows

2.1 Maximum $(u, 2)$ Flows

Unlike the situations for $d = 0, 1$, we are no longer guaranteed that $\mu(G, s, t, u, 2)$ is an integer. Indeed we will see that the graph of Figure 1 is a graph for which $\mu(\bar{1}, 2) = \frac{8}{3}$. We describe two approaches to solving $MF(u, 2)$ in polynomial time.

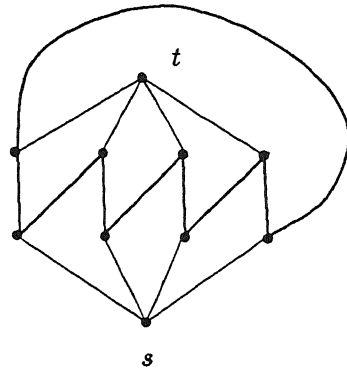


Figure 1:

Consider a directed graph D created from G by directing all edges incident with s and t in such a way that they become respectively source and sink. We then take two copies of each other edge and orient these in opposite directions. Finally we add the extra arc (t, s) . We claim that $MF(u, 2)$ is computed by solving:

$$(3) \quad \begin{array}{ll} \max & x_{(t,s)} \\ & A \cdot x = 0 \\ & x_a \geq 0 \end{array} \quad \text{for each arc } a,$$

$$\begin{aligned}
x(\delta^-(v)) &\leq u_{sv} && \text{for } sv \in E, \\
x(\delta^+(v)) &\leq u_{vt} && \text{for } vt \in E, \\
x(\delta^-(x)) + x(\delta^-(y)) - x_{(x,y)} - x_{(y,x)} &\leq u_{xy} && \text{for the remaining edges } xy \in E.
\end{aligned}$$

Here A is the node-arc incidence matrix of the digraph D and for $S \subset V$, $\delta^-(S)$ (respectively $\delta^+(S)$) denotes the set of arcs directed into (respectively out from) S . Note that except for the last three sets of *bounding* constraints this looks like the traditional LP formulation of the maximum (s, t) flow problem. Note also that if $u = \bar{1}$, then these constraints forbid assigning weight greater than $\frac{1}{2}$ to an $s-t$ paths with a chord. This is not an issue since such paths can always be short-cutted. We must use alternative methods, however, when we consider the directed problem in Section 4.1. Since (3) has a linear number of constraints, it can be solved by any polynomial algorithm for linear programming in time polynomially bounded by the input G, u, d . Moreover, the resulting solution (i.e., weights on the arcs of D) can be polynomially transformed to an optimal collection of paths.

The above approach is conceptually simple and varies little from the traditional solution to the max-flow problem. Unfortunately it does not extend to computing (u, d) flows in general. We need a different approach to solve the minimum cost and directed versions of the $(u, 2)$ flow problem. We outline this approach first for computing $\mu(u, 2)$ as it is also a means by which we can show the NP-hardness of $MF(u, d)$ for $d \geq 3$.

Note that the dual of (2) is:

$$\begin{aligned}
(4) \quad &\min u \cdot x \\
&M_d \cdot x \geq 1 \\
&x \geq 0,
\end{aligned}$$

where x is a d -vector of variables.

Results of Grötschel, Lovasz and Schrijver [6], [8] imply that in order to solve (4) in polynomial time it is enough to have a polynomial time algorithm which determines whether a vector x satisfies the constraints of (4) (i.e., a separation algorithm). When $d = 2$, this separation problem can be reduced (in polynomial time) to a shortest path problem as follows. Given $x \in \mathbb{R}^E$ it is easy to check that $x \geq 0$. Thus determining whether x is feasible for (4) is equivalent to determining whether $x \cdot \chi^{C(P)} \geq 1$ for each induced (s, t) -path P (where $\chi^{C(P)}$ denotes the incident vector of edges in $E_2(E(P))$). We define weights w^x on E as follows: if uv is not incident to s or t , then

$$w_{uv}^x = \frac{1}{2} \left(\sum_{yu \in E: y \neq v} x_{yu} + \sum_{yv \in E: y \neq u} x_{yv} \right),$$

and

$$w_{us}^x = \frac{1}{2} \sum_{yu \in E} x_{yu}, \quad w_{ut}^x = \frac{1}{2} \sum_{yu \in E} x_{yu}.$$

We have defined w^x so that the weight of an induced path P , $\sum_{\alpha \in E(P)} w_{\alpha}^x$, is simply $x \cdot \chi^{C(P)}$. Thus determining whether $x \cdot \chi^{C(P)} \geq 1$ holds for each path P is equivalent

to determining whether a smallest weight path has weight at least 1. Since $w^x \geq 0$ this latter problem is polynomially solvable. Thus the separation problem and hence the LP (4) is solvable in polynomial time. In fact, the Ellipsoid algorithm can be adapted (see [8]) to output an optimal dual solution in polynomial time. In our case this is a solution to the LP (2), i.e., a vector, y^* , of positive weights, each of which is associated with an induced (s, t) -path (in particular y^* can be represented in a polynomial number of bits).

One may now easily verify that Figure 1 depicts a graph where $\mu(G, s, t, \bar{1}, 2) = \frac{8}{3}$. This is done by exhibiting feasible solutions to (2) and (4). Namely, consider directing all edges in the graph “upwards” from s to t . Then assign weight $1/3$ to each of the 8 directed (s, t) -paths. Conversely, a feasible solution to (4) is obtained by setting each of the bold edges to a value $\frac{1}{3}$.

We close the section with a problem.

Problem 1: Give a combinatorial algorithm (not based on an algorithm which solves general linear programming) to compute $\mu(u, 2)$ in general graphs.

2.2 Minimum Cost $(u, 2)$ Flows

For the remainder of this section, c is a vector of non-negative costs on E . We define a vector c^* on the induced (s, t) -paths P such that $c_P^* = \sum_{\alpha \in E(P)} c_\alpha$. If the maximum in (2) is at least k , then the *minimum cost (u, d) -capacitated induced (s, t) flow of value k problem* $MC(G, s, t, u, d, c, k)$, or simply $MC(u, d, c, k)$, is:

$$(5) \quad \begin{aligned} \min \quad & c^* \cdot y \\ & yM \leq u \\ & 1 \cdot y = k \\ & y \geq 0. \end{aligned}$$

The dual of (5) is

$$(6) \quad \begin{aligned} \max \quad & (-u) \cdot x + k\beta \\ & Mx + c^* \geq \beta \cdot 1 \\ & x \geq 0. \end{aligned}$$

It is straightforward to check that (x, β) is a feasible solution to (6) if and only if the shortest (s, t) -path, with weights $(w^x + c)$ on the edges, is at least β . Hence, we may again use the Ellipsoid Algorithm to solve (5). Thus we have that

Theorem 2.1 $MC(u, d, k)$ can be computed in polynomially bounded time.

We raise the following question.

Problem 2: Find a combinatorial algorithm to solve $MC(G, s, t, u, d, c, k)$ for general or even for planar graphs.

2.3 Computing (u, d) Flows is Hard in General

We consider next the problems $MF(u, d)$ for $d > 2$ and we argue in a reverse manner to the previous section. Namely, if we can solve (2) for $d = 3$ for an

arbitrary integral vector u , then we can solve the separation problem for its dual. The latter problem we call the *caterpillar problem* which is defined as: Given a graph G and nonnegative integral weights w on the nodes, find an $s - t$ path P for which $w(V(P) \cup N(V(P) - \{s, t\}))$ is minimized. We show that the caterpillar problem is NP-hard for general graphs and hence so too is $MF(u, 3)$.

We give a reduction of the node cover problem in general graphs to the caterpillar problem. Recall that the node cover problem is for a graph $G = (V, E)$ and integer k , to determine whether there is a subset S of size k such that $V - S$ is a stable set. We construct a graph G^* as follows. First take $|E|$ cycles of length four each of which is identified with an edge $e = uv \in E$ by labelling two of its non-adjacent nodes as e_u and e_v . The cycles are now concatenated on the unlabelled nodes to form a long chain. We then label the two extreme (unlabelled) nodes of degree two as s and t . We also add a new node n_v for each $v \in V$ and join it to e_v for every edge $e \in E$ incident to v . Each n_v is also joined to $|E|$ new degree one nodes to ensure that any minimum cardinality $s - t$ caterpillar will not use any of the nodes n_v . So suppose that P is an $s - t$ path not using any node n_v . Let the degree two nodes traversed by P be $E_P = \{s, e_{v_1}, e_{v_2}, \dots, e_{v_{|E|}}, t\}$. Clearly $N_P = \{v \in V : e_v \in E_P \text{ for some } e \in E\}$ is a node cover of G . Moreover, the cardinality of P 's caterpillar is simply $|E| + |N_P|$. Thus finding a minimum cardinality $s - t$ caterpillar in the new graph corresponds to finding a minimum size node cover for G .

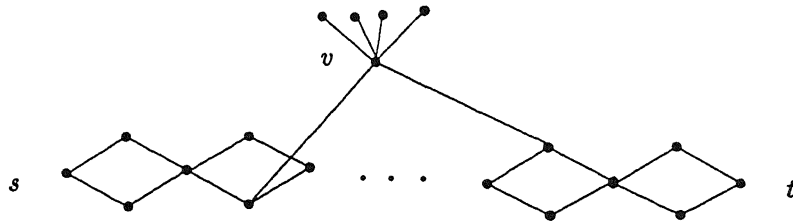


Figure 2:

The above ideas can be extended for $d > 3$ and so we have the following.

Theorem 2.2 $MF(u, d)$ is NP-hard for $d \geq 3$.

3 Planar Networks and Integral (u, d) Flows

3.1 The LP Approach to Integral $(u, 2)$ Flows

We state the following result from [11].

Theorem 3.1 *For graphs G with $G - t$ planar, a $(u, 2)$ -capacitated (s, t) flow y of value k can be transformed in polynomial time to an integral flow of value k . In particular, $\mu'(u, 2) = \lfloor \mu(u, 2) \rfloor$. Furthermore, the integral solution may be chosen to only use paths assigned positive weight in some optimal fractional solution.*

The above result is proved by uncrossing the paths assigned positive weights by y and then taking an appropriate subset of the resulting collection. Actually the latter statement of the theorem would be enough to solve $MF'(u, 2)$ in polynomial time since we could repeatedly delete nodes of G until we find a minimal subgraph $H \subseteq G$ such that $\mu(H, s, t, u_H, 2) \geq \lfloor \mu(G, s, t, u, 2) \rfloor$. At this point the graph H is precisely a collection of induced-disjoint $s - t$ paths. Thus the maximum $(u, 2)$ flow problem can be solved in polynomial time for planar graphs. In particular we have the following.

Corollary 3.2 *For graphs G such that $G - t$ is planar, seeing if s, t lie on an induced circuit can be checked in polynomial time.*

This is related to a well-known problem arising in the study of so-called perfect graphs. An *odd hole* is an odd induced circuit of size at least five. The problem is to determine the complexity of finding whether a graph contains an odd hole. Hsu [7] gives a polynomial algorithm for planar graphs. We propose the following generalization of Corollary 3.2 and Hsu's result.

Problem 3: Give an algorithm to determine whether two nodes in a planar graph lie on an odd induced circuit.

We state another result of [11] showing that minimum cost flows in planar graphs can also be transformed into integral flows.

Theorem 3.3 *If $G - t$ is planar and y is an optimal solution to $MC(G, s, t, u, d, c, k)$, then a $(u, 2)$ -capacitated integral flow of value k with the same cost can be found in polynomial time.*

Specifically, this and Theorem 2.1 imply:

Corollary 3.4 *For planar graphs G and nonnegative integral costs on the edges, a minimum cost induced circuit through two specified nodes can be found in polynomial time.*

3.2 A Combinatorial Algorithm

We now consider a combinatorial approach to finding a maximum $(\bar{1}, d)$ integral flow in a planar network. The method yields a min-max theorem which resembles Menger's edge cut theorem. The method also extends to the problem $MF(u, 2)$, i.e., with general capacities. We must first introduce some concepts.

For a graph G and $d \geq 1$, a d -path is a simple path of length at most $d - 1$. A collection of $s - t$ paths is pairwise d -separate if there is no d -path in $G - \{s, t\}$ connecting internal nodes of distinct paths in the collection. Note that if $d \geq 1$, then an integral $(\bar{1}, d)$ flow in a graph is precisely a collection of d -separate paths.

We assume that G is embedded in the 2-sphere S_2 . Let C be a closed curve in S_2 , not traversing s or t . The *winding number* $w(C)$ of C is, roughly speaking, the

number of times that C separates s and t . More precisely, consider any curve P from s to t , crossing C only a finite number of times. Let λ be the number of times C crosses P from left to right, and let ρ be the number of times C crosses P from right to left (fixing some orientation of C , and orienting P from s to t). Then $w(C) = |\lambda - \rho|$. This number can be seen to be independent of the choice of P .

We call a closed curve C (with clockwise orientation relative to s) d -alternate if C does not traverse s or t , and there exists a sequence

$$(7) \quad (C_0, p_1, C_1, p_2, C_2, \dots, p_l, C_l)$$

such that

- (i) p_i is a d -path of $G \setminus \{s, t\}$ with endpoints s_i, t_i ($i = 1, \dots, l$);
- (ii) C_i is a (noncrossing) curve of positive length from t_{i-1} to s_i traversing a face of G ($i = 1, \dots, l$ and $C_0 = C_l$);
- (iii) C traverses the paths and curves given in (7) in the described order.

Here, by traversing p_i we mean the image of C "follows" p_i from one endpoint to the other.

Intuitively it makes sense that if G has a collection of k d -separate s - t paths, then any alternate curve C must satisfy $l \geq kw(C)$. Since any member of the collection must cross C $w(C)$ times and can only cross it on the paths p_i . Furthermore, no two paths may intersect the same p_i since the paths are d -separate, hence the number of p_i 's is at least $kw(C)$. This argument is fine when $d \leq 2$, the problem for higher values of d is that there can exist a path which crosses C $w(C)$ times but does not intersect $w(C)$ of the p_i 's. Fortunately, this can only happen when a maximum collection of d -separated paths is of size one! We now state the min-max theorem. A proof may be found in [9].

Theorem 3.5 For a planar graph G :

- (i) There exists an integral $(\bar{1}, d)$ flow of value k if and only if $l(C) \geq k \cdot w(C)$ for each d -alternate closed curve C .
- (ii) A maximum value $(\bar{1}, d)$ flow from s to t can be found in polynomial time.
- (iii) The curves C in (i) can be restricted to those with $w(C) < |V|$.

Note that for the cases $d = 0, 1$ the max-flow min-cut theorems imply that we may restrict ourselves to alternate curves of winding number one. This is not the case for $d > 1$ otherwise $MF(u, d)$ would always have an integral optimum as we have seen is not the case in Figure 1. It would be attractive if one could give a better bound on the required number of windings of an alternate curve. The proof of Theorem 3.5 is by means of a path shifting algorithm which also produces a collection of paths which is optimal for (2). This yields the following min-max formula for $MF(\bar{1}, 2)$ in planar graphs.

Theorem 3.6 For a planar graph G ,
 $\mu(G, s, t, \bar{1}, d) = \min\{l(C)/w(C) : C \text{ is a } d\text{-alternate curve}\}.$

These methods can be applied to the case $d = 2$ with general capacities. In this case the *capacity* of a 2-alternate curve C , denoted by $\text{cap}(C)$, is defined as the sum of the capacities of the p_i 's and if p_i is a single node, then it is the minimum capacity of an edge incident to it.

Theorem 3.7 For a planar graph G and vector u of integers on the edges:

- (i) There exists an integral $(u, 2)$ flow of value k if and only if $\text{cap}(C) \geq k \cdot w(C)$ for each 2-alternate closed curve C .
- (ii) $\mu(G, s, t, u, 2) = \min\{\text{cap}(C)/w(C) : C \text{ is a 2-alternate curve}\}.$

Actually we do not know a counterexample to the obvious extension of this to $MF'(G, s, t, u, d)$ for $d > 2$. Of course one must change slightly the definition of capacity. The capacity of a d -path p_i becomes the minimum capacity of an element in its center. (The center of a path P are those element of $E(P) \cup V(P)$ whose d -environment contains both endpoints of P .) Vaguely speaking, the problem for higher values of d is that it is no longer obvious whether one can choose a noncrossing optimal collection of paths.

The algorithm can be extended using the techniques of [14] to find a maximum collection of 2-separated paths for graphs on a fixed surface. It gives a theorem resembling Menger's theorems. In particular, it states that for a graph embedded on a surface \mathcal{S} of genus g either there are k 2-separated $s-t$ paths or there exist $f(g)$ alternate curves whose existence shows that there is no such collection. Here the function f corresponds to an upper bound on the number of homotopic possibilities there are for collections of k disjoint $s-t$ curves in \mathcal{S} (where we identify s and t with points on \mathcal{S}). Note that in some sense the function f gives focus to why the induced circuit problem is computationally difficult in general graphs.

4 Extensions and Applications

4.1 Related Computational Problems

As far as solving for integral solutions, the combinatorial approach is generally preferable to the LP approach. The major advantage of the latter, of course is that it can be adapted to solve the minimum cost induced cycle problem. This in conjunction with Theorem 5 heightens the interest in resolving Problem 2. The shifting algorithms of [9],[10] generalize to solve the maximum (u, d) flow problem in planar graphs and can be extended (as we will note) to solve the maximum flow problems in directed planar networks. It also, as mentioned, yields a min-max theorem which is a closer analogy to the minimum cut conditions of Menger and Ford and Fulkerson. We now discuss a few more related computational problems.

1. Coloured Paths

Input: A planar graph $G = (V, E)$ (possibly with multiple edges)

whose edges are coloured red and green. Source and sink s, t .

Output: A maximum collection of green-edge paths such that no red path joins internal nodes of distinct paths.

A similar characterization as to Theorem 3.5 holds for this problem although we have to change slightly the definition of an alternate curve: the faces it traverses are now faces of the green subgraph and paths p_i are simply red edge paths.

In this problem the red edges can be thought to represent forbidden or high penalty regions in the graph. This phenomenon occurs often in the design of VLSI chips where terminals are iteratively connected by wires. As the number of wires increases, certain regions of the chips become more congested and efforts should be made to avoid routing through these areas.

2. d -Separated Directed Paths

Input: A Planar digraph D with source s and sink t and integer d .

Output: A maximum collection of d -separated $s - t$ dipaths.

The method of Section 3.2 can also be extended to solve a directed version of the problem but we must modify slightly the definitions. (The proof is more involved than that of [10] and the details may be found in [9].) A collection of $s - t$ dipaths is pairwise d -separate if there is no directed path of length less than d connecting internal nodes of distinct dipaths in the collection.

Here again, there is a min-max theorem where a d -alternate curve is defined as in (7) except that now the curves C_i are not required to only traverse a face:

- (i) p_i is a d -dipath of $D \setminus \{s, t\}$ with endpoints s_i, t_i ($i = 1, \dots, l$);
- (ii) C_i is a (noncrossing) curve of positive length from t_{i-1} to s_i and these are the only vertices of D that C_i intersects ($i = 1, \dots, l$ and $C_0 = C_l$);
- (iii) C traverses the paths and curves given in (7) in the described order;
- (iv) each C_i may cross arcs only from right to left (relative to the orientation derived from C) and may not cross any arc in any d -path p_i .

Here, the p_i 's may be directed from s_i to t_i or conversely. and C may traverse p_i in either direction. Note that roughly speaking, condition (iv) requires that the arcs of D crossed by the C_i 's point back towards s . It is interesting to note that in the directed case we do not require the paths in the collection to be induced, i.e., they may have backwards arcs. In fact, Fellows and Kratochvil [4] have recently shown that the problem of determining whether there is a single induced $s - t$ dipath in a planar digraph is NP-complete!

3. k -Star

Input: Planar graph G , node s and nodes v_1, \dots, v_k .

Output: An induced subgraph consisting exactly of node-disjoint paths from s to each of the v_i 's or proof that no such subgraph exists.

4. Internal Node

Input: Planar graph G and nodes u, v, w

Output: An induced $u - v$ path containing w or proof that none exists.

The fact that in Theorem 3.1 we need only that $G - t$ is planar implies that the above problems are also polynomially solvable.

5. Minimum Weight Environment

Input: A planar graph G , integer d , prescribed nodes s, t and a d -vector w .

Output: An $s - t$ path whose d -environment has a minimum w -weight.

Finally, it would be interesting to extend the techniques of Section 3.5 to solve the capacitated version of integral d -separated paths. This would imply (see Section 2.3) the existence of polynomial time algorithms to solve the separation problems for the duals (4), that is the d -environment problem. In particular, this would imply the polynomial solvability of the caterpillar problem in planar graphs.

4.2 Hilbert Bases and a Polyhedral Description

Let G be a graph embedded on a cylinder. A path in G is *linear* if it joins the two ends of the cylinder. Let M_2 be a matrix whose rows are the incidence vectors of 2-environments of linear paths in G . Note that solving the resulting LP (2) is then equivalent to solving $MF(u, 2)$ where s and t are nodes corresponding to the two ends of the cylinder. We have from Theorem 3.1 that an integral optimum to (2) is obtained by rounding down the fractional optimum. Systems $M_2 \cdot x \geq 1, x \geq 0$ for which this holds for each integral u are said to have the *integer round-down property* (cf [13], [1]). Giles and Orlin have shown that this implies that the rows of the matrix

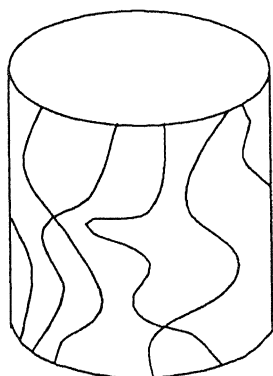


Figure 3:

$$\begin{array}{cc} M_2 & 1 \\ I & 0 \\ 0 & -1 \end{array}$$

form a Hilbert basis. Recall that a *Hilbert basis* is a set of vectors S such that each integral vector in the cone $\{x : x \cdot s \leq 0, \forall s \in S\}$ can be written as a

nonnegative, integral combination of vectors in S . Thus we have the following special class of Hilbert bases, which we denote by $H_2(G, s, t)$. The *Hilbert basis problem* is to show that any integral vector in a $\text{cone}(S)$ can be expressed as the sum of at most $\dim(\text{cone}(S))$ integral vectors in S .

Theorem 4.1 *For a planar graph G and nodes s, t , the associated Hilbert basis $H_2(G, s, t)$ satisfies the Hilbert basis conjecture.*

Proof: Suppose that (u, γ) is an integral vector in the cone generated by H_2 . Then there exists $y_1, y_0 \geq 0$ and $\psi \geq 0$ such that $y_0 \cdot M_2 + y_1 \cdot I = u$ and $\gamma = 1 \cdot y_0 - \psi$. Thus Theorem 3.1 implies that $\mu(G, s, t, u, 2) \geq \lfloor 1 \cdot y_0 \rfloor$ and so there exists an integral vector $z_0 \geq 0$ such that $z_0 M_2 \leq u$ and $z_0 \cdot 1 \geq \lfloor 1 \cdot y_0 \rfloor \geq \gamma$. Thus $(u, \gamma) = z_0 \cdot M_2 + z_1 \cdot I - (0, 0, \dots, 0, -\psi')$ for $\psi' = \lfloor 1 \cdot y_0 \rfloor - \gamma$ and an appropriate integral vector z_1 . But for planar graphs Theorem 3.1 shows that z_0 may be chosen so that its number of positive components is at most the number of components where $z_0 \cdot M_2 = u$. Thus (u, γ) has a representation with at most $|E| + 1$ vectors from H_2 , as desired. \square

We also obtain a polyhedral description for $P(G, s, t) = \mathbf{Q}^+ + \text{conv}(\{\chi^{C(P)} : P \text{ is an } s - t \text{ path}\})$ for s, t in a planar graph G . Results of Fulkerson [3] imply that it is enough to know the structure of the vertices of $\{x : M_2 \cdot x \geq 1, x \geq 0\}$. In fact Theorem 3.7 may be strengthened (see [11]) to show that each such vertex is $0 - \frac{1}{p}$ valued for some integer p .

Theorem 4.2 *For a planar graph G with nodes s, t , the polyhedron $P(G, s, t)$ is given by the nonnegativity constraints and the rank inequalities.*

Acknowledgements: The authors are very grateful to Coelho de Pina Jr. for his careful reading and insightful comments on this work. We are also grateful to Andras Sebő for stimulating conversations. This paper stemmed from discussions at a workshop at Belairs Research Institute. We thank Wayne Hunt and his colleagues for their hospitality and providing an ideal working environment.

References

- [1] S. Baum, L.E. Trotter Jr., Integer rounding for polymatroid and branching optimization problems, *SIAM Journal on Algebraic and Discrete Methods* **2**, (1981), 416-425.
- [2] M. R. Fellows, The Robertson-Seymour theorems: a survey of applications, *Contemporary Mathematics*, **89**, 1989.
- [3] D.R. Fulkerson, Anti-blocking polyhedra, *Journal of Combinatorial Theory, B* **12**, (1972), 50-71.
- [4] M. Fellows, J. Kratochvil, Personal communication, (1991).
- [5] R. Giles, J.B. Orlin, Verifying total dual integrality, *manuscript* (1981).

- [6] M. Grötschel, L. Lovász, and A. Schrijver (1981), *The ellipsoid method and its consequences in Combinatorial Optimization*, *Combinatorica* 1, 169-197.
- [7] W.-L. Hsu, Recognizing planar perfect graphs, *J. of the A.C.M.* 34, (1987), 255-288.
- [8] M. Grötschel, L. Lovász, and A. Schrijver (1988), *Geometric algorithms and combinatorial optimization*, Springer.
- [9] C. McDiarmid, B. Reed, A. Schrijver and B. Shepherd, Non-interfering dipaths in planar digraphs, *Centrum voor Wiskunde en Informatica Technical Report* (1991).
- [10] C. McDiarmid, B. Reed, A. Schrijver and B. Shepherd, Induced circuits planar graphs, submitted to *Journal of Combinatorial Theory*.
- [11] C. McDiarmid, B. Reed, A. Schrijver and B. Shepherd, Packing induced paths in planar graphs, *University of Waterloo Technical Report* (1990).
- [12] W.S. Massey, *Algebraic Topology: An Introduction*, Graduate texts in mathematics, 56, *Springer-Verlag*, (1967).
- [13] A. Schrijver, *Theory of Integer and Linear Programming*, *Wiley*, (1986).
- [14] A. Schrijver, Disjoint circuits of prescribed homotopies in a graph on a compact surface, *J. Combinatorial Theory, B* 51 (1991), 127-159.