Discrete Applied Mathematics 16 (1987) 91–99
North-Holland

EXACT ARBORESCENCES, MATCHINGS AND CYCLES

Francisco BARAHONA and William R. PULLEYBLANK
Department of Combinatorics & Optimization, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

Received 12 March 1985
Revised 1 April 1986

Suppose we are given a graph in which edge has an integral weight. An 'exact' problem is to determine whether a desired structure exists for which the sum of the edge weights is exactly k for some prescribed k.

We consider the special case of the problem in which all costs are zero or one for arborescences and show that a 'continuity' property is possessed similar to that possessed by matroids. This enables us to determine in polynomial time the complete set of values of k for which a solution exists. We also give a minmax theorem for the maximum possible value of k, in terms of a packing of certain directed cuts in the graph.

We also show how enumerative techniques can be used to solve the general exact problem for arborescences (implying spanning trees), perfect matchings in planar graphs and sets of disjoint cycles in a class of planar directed graphs which includes those of degree three. For these problems, we thereby obtain polynomial algorithms provided that the weights are bounded by a constant or encoded in unary.

1. Introduction

Given a graph (digraph) $G=(V,E)$ and a vector $w=(w_e : e\in E)$ of nonnegative integral weights, we define the weight of $F\subseteq E$ as

$$w(F)=\sum_{e\in F} w(e).$$

Let $\mathbf{F}$ be a set of 'feasible' subsets of E. An *exact problem* is to determine, for some prescribed value k, whether there exists $S\in\mathbf{F}$ such that $w(S)=k$. A special case is the following: Suppose we are given $G=(V,E)$ and a set $R\subseteq E$ of 'red' edges. For a prescribed value k, we want to know whether there exists a member of $\mathbf{F}$ containing exactly k edges of R. In Section 2 we discuss this problem for matroids and matroid intersections and give a direct solution for the case of aborescences.

Papadimitriou and Yannakakis [13] observe that the following three problems are NP-complete when the weights are encoded in binary and ask what is the complexity when the weights are encoded in unary.

Exact Spanning Tree. Given a graph, a weight function and $k>0$, is there a spanning tree of weight exactly k?

EXACT PERFECT MATCHING. Given a graph, a weight function and $k>0$, is there a perfect matching of weight exactly k?

EXACT CYCLE SUM. Given a digraph, a weight function and $k>0$, is there a set of disjoint cycles of total weight exactly k?

Even when the corresponding optimization problems are polynomially solvable it seems that for the above problems completely different algorithms are needed. In Sections 3 and 4 we use enumeration methods to prove that the first problem is polynomially solvable, and the second and third ones are polynomially solvable when restricted to planar graphs and to a class of planar digraphs (which includes cubic) respectively.

2. Zero-one weights, matroids and arborescences

In this section we discuss direct solution methods for 'exact' problems in which all weights are 0 or 1. In several such cases there is a 'continuity' property, namely if there exist feasible solutions with weights w_1 and w_2, then for any k between w_1 and w_2 there exists a feasible solution of weight k. A standard optimization algorithm can then be used to find minimum and maximum weight solutions and then the weights for which a feasible solution exists are precisely those between the maximum and minimum. Indeed, Papadimitriou and Yannakakis [13] show that the exact spanning tree problem with all edge weights 0 or 1 can be solved in precisely this fashion. Their result generalizes easily to matroids.

Proposition 2.1. (cf. [9, pp. 84–86]). *Let $M=(E,\mathbf{F})$ be a matroid, let $R\subseteq E$ and suppose there exist bases B_1, B_2 of M such that $|R\cap B_1|=k_1$, $|R\cap B_2|=k_2$ and $k_1\le k_2$. Then, for any integer k satisfying $k_1\le k\le k_2$, there exists a basis $B\subseteq B_1\cup B_2$ such that $|B\cap R|=k$.*

Proof. It is sufficient to show that either $k_1=k_2$ or elese there exists a basis B' of M such that $B'\subseteq B_1\cup B_2$ and $|B'\cap R|=k_1+1$. For any $e\in B_2\setminus B_1$, let C_e be the unique circuit obtained by adding e to B_1. If there exists $e\in B_2\setminus B_1$ such that $e\in R$ but there exists $e'\in C_e\setminus R$, then $B'=B_1\cup\{e\}\setminus\{e'\}$ is the desired basis. If no such e exists, then $B_1\cap R$ is a basis of $(B_1\cup B_2)\cap R$ and so $|B_1\cap R|\ge|B_2\cap R|$ which implies $k_1=k_2$. $\square$

By applying the above process we can see that if B' contains the minimum possible number of elements of R and B'' contains the maximum possible number of elements of R, then we can construct a sequence $B'=B_0, B_1, B_2, \ldots, B_l=B''$ of bases of M such that

$$|B_{i-1}\triangle B_i|=2 \quad \text{for } i=1,2,\ldots,l, \tag{2.1}$$

$$|B_i \cap R| = |B_{i-1} \cap R| + 1 \quad \text{for } i = 1, 2, \ldots, l. \tag{2.2}$$

That is, each basis in the sequence is obtained from the previous basis by adding a single element of R and removing an element of $E \setminus R$. If we only require (2.1) to hold, then the existence of such a sequence of bases is an elementary property of matroids, sometimes stated as "The basis graph of a matroid is connected." Moreover, this in itself is sufficient to prove Proposition 2.1, for if $|B_{i-1} \triangle B_i| = 2$, then $|R \cap B_{i-1}| - 1 \le |R \cap B_i| \le |R \cap B_{i-1}| + 1$, i.e., the number of elements of R can increase or decrease by at most one.

(Suppose that each edge of G has a real cost associated with it. Gabow and Tarjan [9] and Glover [10] give algorithms for finding a spanning tree of minimum cost containing a prescribed number of red edges. Glover and Klingman [11] and Gabow [8] solved this problem for the case that all red edges are incident with a single node. This is a special case of the weighted matroid intersection problem.)

Let $G = (V, E)$ be a directed graph and let $r \in V$ be a specified root node. An *aborescence* rooted at r is a set A of arcs that form a connected graph such that each node other than r has indegree one and r has indegree zero.

We now show that this same property of 'continuity' does hold for aborescences, however we no longer have the stronger monotonicity property. Let G be a directed graph with a fixed root node r. Let $\mathbf{A}$ be the set of all (spanning) arborescences rooted at r and, for any $A \in \mathbf{A}$, let $R(A) = |R \cap A|$. We call arcs of R *red arcs* and call arcs of $E \setminus R$ *blue arcs*.

Proposition 2.2. *Let $A, A' \in \mathbf{A}$. Then there exists a sequence $A = A_0, A_1, \ldots, A_l = A'$ of arborescences in $\mathbf{A}$ such that $|A_{i-1} \triangle A_i| = 2$ for $i = 1, 2, \ldots, l$.*

Proof. We prove by induction on $s = |A \triangle A'|$. If $s = 2$, we have nothing to prove, so assume $s \ge 4$. Let e' be an arc of $A' \setminus A$ such that the head v of e' is at a maximum distance in A from r. Let e be the arc of A whose head is v. Remove e from A and let W be the set of nodes thereby disconnected from r. Some arc of A' must join a node of $V \setminus W$ to a node of W. If it were any arc other than e', we would contradict our choice of e' so, in fact, e' must join a node of $V \setminus W$ to v. Therefore $A'' = A \cup \{e'\} \setminus \{e\}$ is an arborescence in $\mathbf{A}$ satisfying $|A'' \triangle A| = 2$ and $|A'' \triangle A'| = s - 2$. The result now follows by applying induction to A' and A''. $\square$

Just as for matroid bases, if $A_1, A_2 \in \mathbf{A}$ satisfy $|A_1 \triangle A_2| = 2$, then $|R(A_1) - R(A_2)| \le 1$, so we have the following:

Corollary 2.3. *Let $G = (V, E)$ be a directed graph, let $R \subseteq E$ and let A_1 and A_2 be arborescences rooted at $r \in V$. Then for any integer k between $R(A_1)$ and $R(A_2)$ there exists an arborescence A rooted at r such that $R(A) = k$.*

However, unlike in the case of matroids, it is not possible in general to find a se-

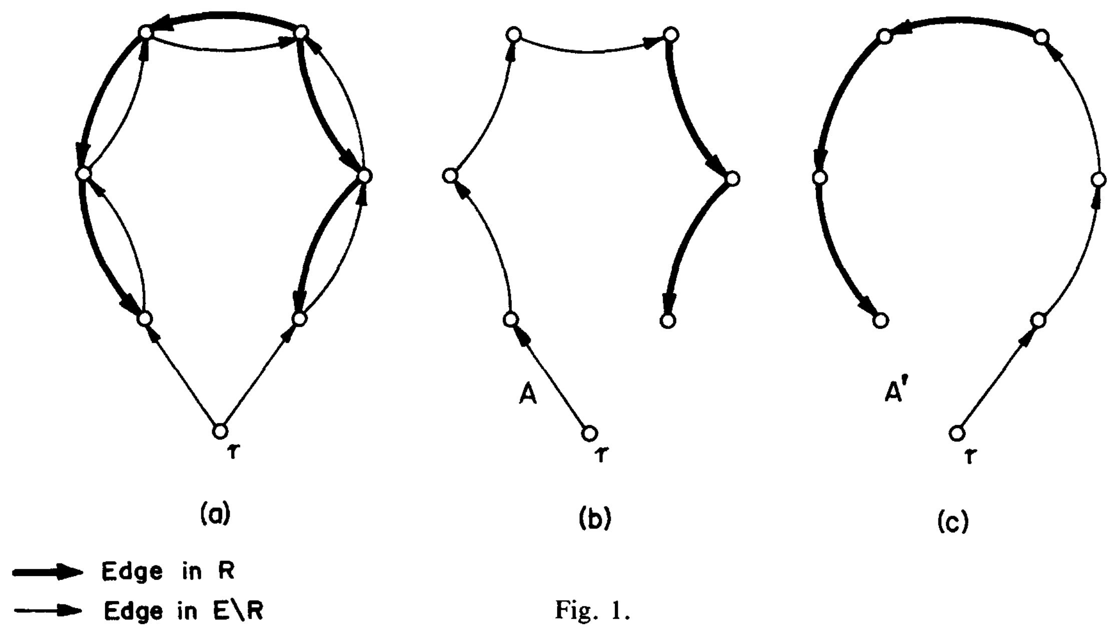

Fig. 1.

quence $A_0, A_1, \ldots, A_l$ as described in Proposition 2.2 such that $R(A_{i-1}) < R(A_i)$ or even $R(A_{i-1}) \leq R(A_i)$ for $1 \leq i \leq l$. For consider the graph of Fig. 1(a). If A and A' are the aborescences of Fig. 1(b), (c), respectively, then $R(A) = 2$, $R(A') = 3$ but any sequence of the form of Proposition 2.2 will contain an aborescence A_i for which $R(A_i) = 0$.

However, we can describe a direct algorithm which will, for any aborescence A, either find an arborescence A' such that $R(A') = R(A) + 1$ or else prove that $R(A)$ is maximum. First we introduce some terminology.

Let $V(R) = \{v \in V \setminus \{r\}$: there exists an arc $(u, v) \in R\}$. That is, $V(R)$ is the set of nodes which are the head of a red arc. For any $S \subseteq V$ we let $\bar{S} = V \setminus S$ and we let $(S, \bar{S})$ denote the set of arcs with tail in S and head in $\bar{S}$. We say that $(S, \bar{S})$ is a *blue cut* if

$$r \in S, \tag{2.3}$$

$$(S, \bar{S}) \subseteq E \setminus R, \tag{2.4}$$

$$\text{for every arc } (u, v) \in (S, \bar{S}), \quad v \in V(R). \tag{2.5}$$

Informally, a blue cut is a cut directed away from r such that all arcs in it are blue and such that the head of each such arc is also incident with a red arc.

A *blue packing* is a set $\mathbf{B}$ of pairwise disjoint blue cuts. The size of $\mathbf{B}$ (denoted by $|\mathbf{B}|$) is the number of cuts in the packing. Note that any arborescence rooted at r must contain at least $|\mathbf{B}|$ blue arcs incident with nodes of $V(R)$. Therefore the maximum number of red arcs in an aborescence is at most $|V(R)| - |\mathbf{B}|$. In fact, for a suitable arborescence and blue cut packing, equality will hold.

Theorem 2.4. *The maximum of* $\{R(A) : A \in \mathbf{A}\}$ *equals the minimum of* $\{|V(R)| - |\mathbf{B}| : \mathbf{B}$ *is a blue cut packing of* $G\}$.

We prove this by giving an algorithm for the following: Given $G = V(E)$, r, R as above and an arborescence A, either find an arborescence A' such that $R(A') = R(A) + 1$ or else find a blue cut packing $\mathbf{B}$ of size $|V(R)| - |R(A)|$.

If every node of $V(R)$ is the head of a red arc of A, then setting $\mathbf{B} = \emptyset$ gives the required packing. If not, choose $v \in V(R)$ such that the arc (u, v) of A with head r belongs to $E \setminus R$ and such that the path in A from r to v is maximal, subject to this. Let W be the set of nodes such that the path in A from r passes through v. If any arc $(w, v) \in R$ has $w \notin W$, then we can remove (u, v) and add (w, v) and get an arborescence A' as required. If not, choose some $(w, v) \in R$ with $w \in W$. (Such an arc exists because $v \in V(R)$.) Adding (w, v) to A creates a unique directed cycle C. Note that every node of $V(R)$ in C is the head of a red arc of C, by our choice of v. Let λ be the number of nodes of $V(R)$ in C.

Let $\tilde{G} = (\tilde{V}, \tilde{E})$ be obtained from G by shrinking C to form a pseudonode $\tilde{C}$. The arcs of $\tilde{G}$ have the same colours as the corresponding arcs of G, except for arcs whose heads are $\tilde{C}$. If $(s, t) \in E$ is such that t is a node of C, but s is not a node of C, then we let

$$(s, \tilde{C}) \in \tilde{R} \quad \text{if } (t \in V(R) \text{ and } (s,t) \in R)$$

$$\text{or } (t \notin V(R) \text{ and } (s,t) \in E \setminus R),$$

$$(s, \tilde{C}) \in \tilde{B} = \tilde{E} \setminus \tilde{R} \quad \text{if } t \in V(R) \text{ and } (s,t) \in E \setminus R.$$

Note that we cannot have $t \notin V(R)$ and $(s, t) \in R$.

Let $\tilde{A}$ be the arborescence in $\tilde{G}$ whose arcs correspond to the arcs of A (not in C). If we let $\tilde{R}(\tilde{A})$ denote the number of arcs of $\tilde{R}$ in $\tilde{A}$, then

$$\tilde{R}(\tilde{A}) = R(A) - (\lambda - 1). \tag{2.6}$$

Recursively apply the algorithm to $\tilde{G}, r, \tilde{R}, \tilde{A}$. There are two possible outcomes:

(1) Suppose we obtain an aborescence $\tilde{A}'$ in $\tilde{G}$ such that $\tilde{R}(\tilde{A}') = \tilde{R}(\tilde{A}) + 1$. Let $(s, \tilde{C})$ be the arc of $\tilde{A}'$ whose head is $\tilde{C}$. Let (s, t) be the corresponding arc of G. We construct an arborescence A' in G by adding to the arcs of $\tilde{A}$ all arcs of C except the one whose head is t. It can be straightforwardly verified that, in all cases, $R(A') = \tilde{R}(\tilde{A}') + \lambda - 1 = \tilde{R}(\tilde{A}) + \lambda$. Therefore, by (2.6), $R(A') = R(A) + 1$ as required.

(2) Suppose we obtain a blue cut packing $\tilde{\mathbf{B}}$ in $\tilde{G}$ and $|\tilde{\mathbf{B}}| = |V(\tilde{R})| - \tilde{R}(\tilde{A})$. If all arcs of $\tilde{G}$ with heads equal to $\tilde{C}$ belong to $\tilde{B}$, then $|V(\tilde{R})| = |V(R)| - \lambda$. Therefore $|V(R)| - R(A) = |\tilde{\mathbf{B}}| + 1$. No arc with head $\tilde{C}$ can belong to a cut in $\tilde{\mathbf{B}}$, but the set of arcs of G with heads in C and tails not in C forms a blue cut in G. We add this cut to $\tilde{\mathbf{B}}$ and obtain a blue cut packing $\mathbf{B}$ in G of the required size. If there exists an arc of $\tilde{R}$ whose head is $\tilde{C}$, then $|V(\tilde{R})| = |V(R)| + \lambda + 1$ and so $|V(R)| - R(A) = |\tilde{\mathbf{B}}|$. By the definition of $\tilde{R}$ and $\tilde{B}$ for the arcs whose head is $\tilde{C}$, it can be easily verified that the arcs of G corresponding to the arcs of the cuts of $\tilde{\mathbf{B}}$ form a blue cut packing $\mathbf{B}$ in G, and $|\mathbf{B}| = |V(R)| - R(A)$ as required. □

Theorem 2.4 can also be deduced from Edmonds' theorem [5] giving a linear system sufficient to define the convex hull of the incidence vectors of the aborescences, plus the total dual integrality of the linear system. Moreover, the algorithm is a special case of a 'primal' algorithm for maximum weight aborescences. 'Dual' algorithms for the general problem are given by Chu and Liu [4], Edmonds [5], Bock [2] and Fulkerson [7]. Gabow and Tarjan [9] present an algorithm which finds a minimum cost aborescence containing k red arcs, for the case that all the red arcs are incident with a single node.

Note that arborescences form a special case of matroid intersection. Consider the general problem: given matroids $M_1=(E,\mathbf{F}_1)$ and $M_2=(E,\mathbf{F}_2)$, $R\subseteq E$ and k, does there exist a basis B of $M_1\cap M_2$ such that $|B\cap R|=k$? This problem is open for matroids in general and even for the special case that $M_1\cap M_2$ is the set of matchings of a bipartite graph. As was noted in [1], considering an even cycle whose edges are alternately in and not in R, we can see that this matching problem does not have the continuity property possessed by matroids and arborescences. Hence neither do matroid intersection problems in general.

3. Exact arborescences and trees

We now consider the general 'exact' problem for arborescences, which will imply the results for spanning trees. That is, we have a directed graph $G=(V,E)$, a nonnegative integral weight function w and a specified root node r. We wish to find whether there exists an arborescence A such that $\sum (w(i,j):(i,j)\in A)=k$, for some prescribed k. Note that if we have negative weights, we can transform the problem into one with nonnegative weights by adding a suitable constant to the weight of each arc and adjusting k appropriately.

Let M be an $n\times n$ matrix having zero row sums whose rows and columns are indexed by V. For $i\neq j$ we denote the (i,j)-th element by $-M_{ij}$. Let $D(M)$ be the determinant of the matrix obtained by omitting the row and column corresponding to r. For each A in $\mathbf{A}$, let $Q(A)$ be the product of the M_{ij} over all directed arcs (j,i) in A. In [3] the following identity is proved:

$$D(M)=\sum_{A\in\mathbf{A}} Q(A).$$

Given a nonnegative integral weight function w and a real variable x, we define the matrix $M(x)$ as follows:

$$M_{ij}(x)=\begin{cases}-x^{w(j,i)} & \text{if } (j,i)\in E,\ i\neq j,\\ 0 & \text{if } (j,i)\notin E,\ i\neq j,\end{cases}\quad \text{for } i,j=1,2,\ldots,n;$$

$$M_{ii}(x)=\sum_{j\neq i} -M_{ij}(x)\quad \text{for } i=1,2,\ldots,n.$$

Then

$$D(M(x))=\sum a_k x^k,$$

where a_k is the number of arborescences rooted at v_1 with weight exactly k. (Note that if we replace (j,i) with (i,j) in the above, we enumerate arborescences directed 'towards' the root.)

$D(M(x))$ is a polynomial of degree at most $p=n\cdot\max\{w(e)\}$, hence its $p+1$ coefficients can be obtained by evaluating $D(M(x))$ for $p+1$ points and solving a system of $p+1$ linear equations. Using the method given in Edmonds [6], we can compute each determinant in time $O(n^3\cdot p\log p)$, where the $p\log p$ factor accounts for the time spent performing arithmetic operations. Similarly, we can solve the system of equations in time $O(p^3\cdot p\log p)$.

Let us define the problem:

EXACT ARBORESCENCE. Given a graph, a weight function and $k>0$, is there an arborescence rooted at v of weight exactly l?

We have the following theorem.

Theorem 3.1. *There exists an algorithm which solves* EXACT ARBORESCENCE *in time* $O((n^3+p^2)p^2\log p)$.

Note that p is a polynomial function of n, the number of nodes, and the *magnitude* of the largest arc cost. If the costs are encoded in unary or if the magnitudes are bounded by a constant, then p will be a polynomial function of the length of the input corresponding to an instance of the problem. Otherwise, this need not be the case.

EXACT SPANNING TREE can be solved by replacing every undirected edge by two oppositely directed arcs and solving EXACT ARBORESCENCE.

4. Exact matchings and cycles

The same idea can be applied to perfect matchings in planar graphs.

Let $G=(V,E)$ be a planar graph with $|V|=2n$. Given an orientation of the edges and $z:E\to\mathbb{R}$ we can define a matrix M as follows:

$$M_{ij}=\begin{cases} z(e) & \text{if } e=\{i,j\} \text{ and } e \text{ is oriented from } i \text{ to } j,\\ -z(e) & \text{if } e=\{i,j\} \text{ and } e \text{ is oriented from } j \text{ to } i,\\ 0 & \text{if } \{i,j\}\notin E. \end{cases}$$

Let $\mathbf{M}$ be the set of perfect matchings of G, for $A\in\mathbf{M}$ let $Q(A)$ be the product of $z(e)$ over all edges e that belong to A. Kasteleyn [12] describes an efficient algorithm for obtaining an orientation of the edges of G such that

$$\mathrm{Pf}(M)=\sum_{A\in\mathbf{M}} Q(A),$$

where $\mathrm{Pf}(M)$ denotes the pfaffian of M.

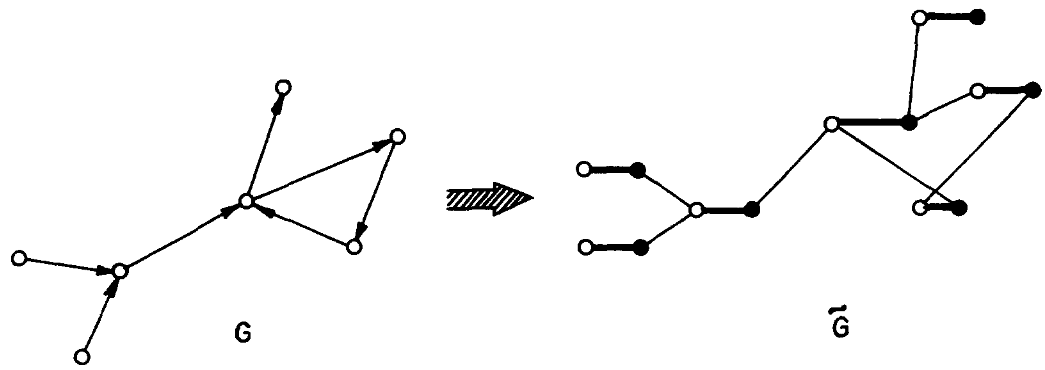

Fig. 2. Transformation of EXACT CYCLE to EXACT PERFECT MATCHING.

Given $w: E \to \mathbb{Z}$ let us define $z(e) = x^{w(e)}$. Then $\mathrm{Pf}(M) = \sum a_k x^k$, where a_k is the number of perfect matchings of weight exactly k. The pfaffian of a matrix can be computed in polynomial time in an analogous way to a determinant, see [1], or it can be computed directly from the determinant since $[\mathrm{Pf}(M)]^2 = \det(M)$.

Since $\mathrm{Pf}(M)$ is a polynomial of degree at most $p = n \cdot \max\{w(e)\}$, its $p+1$ coefficients can be obtained as has been described in the preceding section. Therefore, we have the following:

Theorem 4.1. *There exists an algorithm which solves* EXACT PERFECT MATCHING *in planar graphs in time* $O((n^3 + p^2)p^2 \log p)$.

Suppose we wish to solve EXACT CYCLE SUM for a directed graph G. The problem can be transformed into EXACT PERFECT MATCHING for a bipartite graph as follows. Delete any nodes having indegree or outdegree zero. Replace each remaining node by two new nodes, one designated as a 'head' node and the other as a 'tail' node. Construct a new edge of cost zero between each pair of new nodes. For each arc of G we construct an edge joining the appropriate hed and tail nodes. The cost of this edge is the same as the cost of the corresponding arc. This transformation is illustrated in Fig. 2.

Let $\tilde{G}$ be the resulting (bipartite) graph. Then G has a set of node disjoint directed cycles, the sum of whose edge weights is k, if and only if $\tilde{G}$ has a perfect matching, the sum of whose edge weights is k. The graph $\tilde{G}$ will be planar if and only if G has a planar embedding such that for each node i, the arcs whose heads are i form a consecutive group in the embedding. In particular, if G is cubic and planar, this will always be the case. Therefore, for these graphs we can solve EXACT CYCLE.

Finally, we mention one additional problem.

EXACT CUT. Given a graph, a weight function and $k > 0$, is there a cut of weight exactly k?

For planar and toroidal graphs this problem can be solved in polynomial time by enumerating perfect matchings, the transformations are described in [1].

References

[1] F. Barahona, Balancing signed toroidal graphs in polynomial time, Depto. de Matemáticas, Universidad de Chile, June 1981.
[2] F.C. Bock, An algorithm to construct a minimum directed spanning tree in a directed network, in: B. Avi-Itzhak, ed., Developments in Operations Research (Gordon and Breach, New York, 1971) 29-44.
[3] S. Chaiken and D.J. Kleitman, Matrix tree theorems, J. Combin. Theory (A) 24 (1978) 377-381.
[4] Chu Yoeng-Yin and Liu Tseng-hong, On the shortest aborescence of a directed graph, Scientia Sinica 4 (1965) 1396.
[5] J. Edmonds, Optimum branchings, J. Res. Nat. Bur. Standards (Sect. B) 71 (1967) 233-240.
[6] J. Edmonds, Systems of distinct representatives and linear algebra, J. Res. Nat. Bur. Standards (Sect. B) 71 (1967) 241-245.
[7] D.R. Fulkerson, Packing rooted directed cuts in a weighted directed graph, Math. Programming 6 (1974) 1-13.
[8] H.N. Gabow, A good algorithm for smallest spanning trees with a degree constraint, Networks 8 (1978), 201-208.
[9] H.N. Gabow and R.E. Tarjan, Efficient algorithms for a family of matroid intersection problems, J. Algorithms 5 (1984) 80-131.
[10] F. Glover, The generalized quasi-greedy algorithm for constrained minimum weight matroid bases Part I, MSIS report 84-5, University of Colorado, May 1984.
[11] F. Glover and D. Klingman, Finding minimum spanning trees with a fixed number of links at a node, in: B. Roy, ed., Combinatorial Programming: Methods and Applications (Reidel, Dordrecht, 1974) 111-200.
[12] P.W. Kasteleyn, Graph theory and crystal physics, in: F. Harary, ed., Graph Theory and Theoretical Physics (Academic Press, London, 1967) 43-110.
[13] C.H. Papadimitriou and M. Yannakakis, The complexity of restricted spanning tree problems, J. ACM 29 (1982) 285-309.