RC25414 (WAT1310-003) October 2, 2013
Mathematics

IBM Research Report

Maximum Weighted Induced Bipartite Subgraphs and Acyclic Subgraphs of Planar Cubic Graphs

Mourad Baïou
CNRS
and
Université Clermont II
Campus des Cézeaux BP 125
63173 Aubière Cedex
France

Francisco Barahona
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 208
Yorktown Heights, NY 10598
USA

MAXIMUM WEIGHTED INDUCED BIPARTITE SUBGRAPHS AND ACYCLIC SUBGRAPHS OF PLANAR CUBIC GRAPHS

MOURAD BAÏOU AND FRANCISCO BARAHONA

Abstract. We study the node-deletion problem consisting of finding a maximum weighted induced bipartite subgraph of a planar graph with maximum degree three. We show that this is polynomially solvable. It was shown in [4] that it is NP-complete if the maximum degree is four. We also extend these ideas to the problem of balancing signed graphs.

We also consider maximum weighted induced acyclic subgraphs of planar directed graphs. If the maximum degree is three, it is easily shown that this is polynomially solvable. We show that for planar graphs with maximum degree four it is NP-complete.

1. Introduction

Given an undirected graph $G = (V, E)$, a graph $H = (W, F)$ is said to be *induced* if $W \subseteq V$ and F is the set of edges in E having both endnodes in W. If every node u has a non-negative weight $w(u)$, the *Maximum Weighted Induced Bipartite Subgraph Problem* (MWBSP) consists of finding an induced bipartite subgraph with maximum total weight. In this paper we show that for planar graphs with degree at most three, this problem is polynomially solvable. We extend this procedure to balancing signed graphs. For planar graphs with degree at most four, it was shown in [4] that this is NP-Complete. This problem was studied in [4], and its connection to via-minimization of integrated circuits and printed circuit boards was discussed. Later in [8] they studied the application to via-minimization and to DNA sequencing. The polyhedral approach to this problem has been studied in [2], [3] and [8]. An approximation algorithm for general graphs was given in [14], and an approximation algorithm for planar graphs was presented in [15].

For a directed graph $G = (V, A)$ and induced subgraph is defined in a similar way. If every node u has a non-negative weight $w(u)$, the *Maximum Weighted Induced Acyclic Subgraph Problem* (MWASP) consists of finding and induced acyclic subgraph of maximum total weight. If a node-set induces an acyclic subgraph, its complement is called a *Directed Feedback Vertex Set* (DFVS). The DFVS problem is an NP-complete problem that appeared in the first list of NP-complete problems in Karp's seminal paper [17]. It has applications in areas such as operating systems [23], database systems [12], and circuit testing [18]. It was shown that it is NP-Complete for planar directed graphs with in-degree and out-degree at most three, see [13]. A polyhedral approach has been presented in [9], see also [2] and [3]. An approximation algorithm for general directed graphs was given in [6], and for planar directed graphs an approximation algorithm was given in [15]. See [7] for a survey on Feedback Set problems. Here we point out that it is easy to see that for planar graphs with maximum degree three, it is polynomially solvable, then we show that it is NP-complete for planar graphs with in-degree and out-degree at most two, i.e., maximum degree four.

This paper is organized as follows. In Section 2 we present some definitions and recall some classic results that will be used in the sequel. In Section 3 we study maximum weighted induced bipartite subgraphs. In Section 4 we deal with balancing signed graphs. Section 5 is devoted to maximum weighted induced acyclic subgraphs.

2. Preliminaries

In this section we give some definitions and present some classic results to be used in the following sections.

If $G = (V, E)$ is an undirected graph, the *degree* of a node is the number of edges incident to it. If S is a nod-set set we denote by $\delta(S)$ the set of edges with exactly one endnode in S. We use $\delta(v)$ instead of $\delta(\{v\})$. If e is an edge with endnodes u and v, we also use uv to denote the edge e. If $F \subseteq E$, the graph $H = (V, F)$ is called a *spanning subgraph* of G.

If $D = (V, A)$ is a directed graph, the in-degree (out-degree) of a node is the number of arcs entering (leaving) it. The degree of a node is its in-degree plus its out-degree. For a node set S we use $\delta^+(S)$ to denote the set $\{(u, v) \,|\, u \in S,\, v \notin S\}$. We use $\delta^-(S)$ to denote $\delta^+(V \setminus S)$.

Now we review two classic results in combinatorial optimization.

2.1. **The Chinese Postman Problem.** Given an undirected connected graph $G = (V, E)$ with nonnegative edge weights $w(e)$ for each edge e, this problem consists of finding a tour of minimum weight, so that every edge is visited at least once. Edmonds & Johnson [5] gave a polynomial algorithm for this. One has to find an edge-set of minimum weight that should be visited twice. This can be formulated as follows.

$$\text{minimize} \sum_{e \in E} w(e)x(e) \tag{1}$$

$$\sum_{e \in \delta(v)} x(e) \equiv \begin{cases} 1 \pmod 2 & \text{if } v \in T, \\ 0 \pmod 2 & \text{if } v \in V \setminus T, \end{cases} \tag{2}$$

$$x(e) \in \{0, 1\} \text{ for all } e \in E. \tag{3}$$

Here T denotes the set of nodes of odd degree. A solution of this corresponds to a set of paths matching the nodes in T. For this Edmonds & Johnson gave a combinatorial algorithm that solves the following linear program.

$$\text{minimize} \sum_{e \in E} w(e)x(e) \tag{4}$$

$$\sum_{e \in \delta(S)} x(e) \geq 1 \quad \text{for each node-set } S \text{ with } |S \cap T| \text{ odd}, \tag{5}$$

$$x(e) \geq 0 \quad \text{for all } e \in E. \tag{6}$$

Their algorithm shows that this linear program always has an optimal solution that is integer valued.

If T is an arbitrary set of nodes with $|T|$ even, the same results hold, and this is called the *Minimum T-join problem.* We are going to use this problem in Section 3.

Let $n = |V|$. If G is a complete graph, this problem can be solved in $O(n^3)$ time, see [10]. If the graph is planar, one can use the planar separator theorem of [20] to solve this in $O(n^{3/2} \log n)$, see [1].

2.2. **The Luchessi-Younger Theorem.** Let $G = (V, A)$ be a directed graph. An arc-set C is called a *directed cut* if there is a node set $U \subset V$ such that $C = \delta^-(U)$ and $\delta^+(U) = \emptyset$. Suppose that each arc a has a non-negative weight $w(a)$. Lucchesi & Younger [21] proved that the linear program below always has an optimal solution that is integer valued.

$$\text{minimize} \sum_{a \in A} w(a)x(a) \tag{7}$$

$$\sum_{a \in C} x(a) \geq 1 \quad \text{for every directed cut } C, \tag{8}$$

$$x(a) \geq 0 \quad \text{for all } e \in A. \tag{9}$$

Lucchesi [22] gave an $O(n^5 \log n)$ algorithm for this. Later Gabow [11] gave an $O(n^2 m)$ algorithm. Here $n = |V|$ and $m = |A|$.

3. Maximum weighted induced bipartite subgraphs

In this section we assume that the graph $G = (V, E)$ is planar, with degree at most three, and with a non-negative weight $w(u)$ for each node $u \in V$. We have to find an induced bipartite subgraph of maximum total weight. For the case when all node weights are equal to one, i.e., the maximum cardinality version, it was pointed out in [4] that this reduces to a maximum cardinality cut problem in a planar graph, that is polynomially solvable. For general non-negative weights this transformation is not valid, so this case has to be treated in a different way.

This problem is equivalent to look for a node-set of minimum weight that should be deleted to leave a bipartite graph. This can be formulated as the following linear integer program.

$$\text{minimize} \sum_{u \in V} w(u)x(u) \tag{10}$$

$$\sum_{u \in C} x(u) \geq 1, \ \text{for each odd cycle } C, \tag{11}$$

$$x(u) \in \{0, 1\}, \ \text{for all } u \in V. \tag{12}$$

Consider the linear programming relaxation obtained by replacing (12) by $x(u) \geq 0$, for all $u \in V$. Suppose for instance that G is the graph K_4, and that all weights are equal to one. If we set all variables equal to 1/3, we have a solution of value 4/3. On the other hand the optimal value of (10)-(12) is 2. This shows that this linear programming relaxation is not tight. At the end of this section we present a linear programming formulation (17)-(19), that gives the value of a minimum weight node-set to be deleted. This can be seen as an *extended formulation*, since we have three variables for each node.

We assume that G is connected, otherwise each connected component is treated independently. Starting from G, we create a *signed* graph $G' = (V', E')$, where each edge is labeled as *positive* or *negative* as follows.

- If a node u has degree one, let uv be the edge incident to u. Since the node u will appear in every maximum weighted induced bipartite subgraph, we remove the node u and the edge uv. We repeat this until every node has degree at least two.

- For each node u of degree two, let uv_1 and uv_2 be the edges incident to u. We split the node u into u_1 and u_2. We create the following edges.
  - u_1u_2, with weight $w(u)$ and labeled positive. This edge is called *artificial*.
  - u_1v_1, labeled negative.
  - u_2v_2, labeled negative.
- For each node u of degree three, let uv_1, uv_2, uv_3 be the edges incident to u. We split u into u_1, u_2, u_3 and we create the following edges.
  - The edges u_1u_2, u_2u_3, and u_3u_1, all with weight $w(u)/2$ and labeled positive. These edges are called *artificial*.
  - The edges u_1v_1, u_2v_2, u_3v_3, all labeled negative.

Notice that the graph G' is also planar. This construction is illustrated in Figure 1.

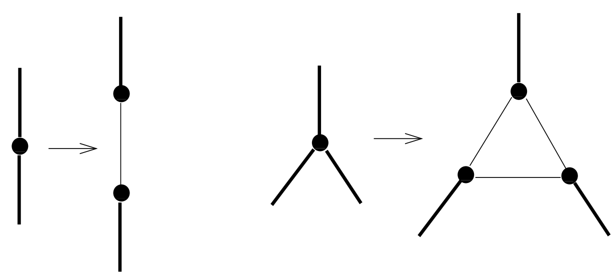

FIGURE 1. Construction of the graph G'. Negative edges appear with thick lines. Positives edges with thin lines.

We need the following lemma.

Lemma 1. *Finding a maximum weighted induced subgraph of G is equivalent to give the labels "a" and "b" to the nodes of G' in such a way that:*

(1) *The endnodes of each negative edge have different labels.*
(2) *The total weight of the positive edges whose endnodes have different labels, is minimum.*

Proof. First assume that $U \subseteq V$ induces a bipartite subgraph of G with maximum weight. Let U_1 and U_2 be the bipartition of U. We give the label a to each node in G' associated with a node in U_1, and the label b to every node in G' associated with a node in U_2. Then for each negative edge that has only one labeled endnode, we give the opposite label to the other endnode. Finally for each negative edge whose endnodes have no label, we give arbitrarily opposite labels to the endnodes.

Let $\bar{U} = V \setminus U$, and $w(\bar{U})$ the total weight of the nodes in $\bar{U}$.

Let λ be the sum of the weight of the positive edges whose endnodes have different label. Since all nodes of G' associated with a node in U have the same label, we have $\lambda \leq w(\bar{U})$.

Let $\hat{\lambda}$ be the weight of an optimal labeling satisfying (1) and (2). We have $\hat{\lambda} \leq \lambda$. Let S be the set of nodes of G whose associated nodes in G' have the same label. Clearly S induces a bipartite subgraph. Let $\bar{S} = V \setminus S$ and $w(\bar{S})$ the total weight of the nodes in $\bar{S}$. We have $w(\bar{S}) = \hat{\lambda}$, and

$$w(\bar{U}) \leq w(\bar{S}) = \hat{\lambda} \leq \lambda \leq w(\bar{U}).$$

□

Now we have to give an algorithm that finds a labeling satisfying (1) and (2) of Lemma 1. For that we need one more definition.

Definition 2. *For a signed graph, and a labeling of the nodes, we say that an edge e is violated if:*

- *e is positive and its endnodes have different labels, or*
- *e is negative and its endnodes have the same label.*

Lemma 3. *If a cycle contains and odd (resp. even) number of negative edges, then for any labeling it has an odd (resp. even) number of violated edges.*

Proof. Consider a cycle with an odd number of negative edges. Start by giving the label a to all nodes, then there is an odd number of violated edges. Now if we change the label of a node, either the number of violated edges increases by two, or decreases by two, or remains the same. Then if we keep changing the node-labels the number of violated edges is always odd.

The proof for the other case is similar □

Now we have to prove the converse.

Lemma 4. *Consider a signed graph and a set of edges marked as violated, so that for each cycle, if it has an odd (resp. even) number of negative edges then there is an odd (resp.even) number of violated edges. Then there is a set of node-labels according to Definition 2*

Proof. Start with a spanning tree T, pick any node and give it the label a, then extend the labels through T according to Definition 2.

Then pick an edge $e \notin T$, we have to see that the labels of its endnodes satisfy Definition 2. Let C be the cycle obtained when adding e to T.

Consider the case when C has an odd number of negative edges. Assume that e is positive and marked as violated. We should show that its endnodes have different labels, so assume the opposite. Based on the labels e is not violated. But $C \setminus e$ contains an even number of edges marked as violated, this contradicts Lemma 3. The proof for all other cases is similar. □

These two lemmas show that it is equivalent to work with the node labels, or with sets of violated edges satisfying the parity conditions of Lemma 3. From now on we use the second alternative. We associate to each edge e a variable $x(e)$ that should take the value 1 if e is violated and 0 otherwise. Then x should satisfy for each cycle C, the following.

$$\sum_{e \in C} x(e) \equiv \begin{cases} 1 \pmod 2 & \text{if } C \text{ has an odd number of negative edges,} \\ 0 \pmod 2 & \text{if } C \text{ has an even number of negative edges.} \end{cases} \tag{13}$$

Here we have an exponential number of equations over $GF(2)$, but we only need a maximal set that is linearly independent. Thus it is enough to impose equations (13) for the cycles in a cycle basis of G'.

Since G' is a planar graph, we can use the set of faces as a cycle basis. Let $\mathcal{F}$ be the set of faces of G', let $\mathcal{P}$ be the set of positive edges in G', $\mathcal{A}$ the set of artificial edges, and $\mathcal{N}$ the set of negative edges in G'. For each artificial edge e in G', let $\lambda(e)$ be the weight of it. Our problem can be formulated as below.

$$\text{minimize} \sum_{e\in\mathcal{A}} \lambda(e)x(e) \tag{14}$$

$$\sum_{e\in C\cap\mathcal{A}} x(e) \equiv \begin{cases} 1 \pmod 2 & \text{if } C\in\mathcal{F} \text{ and } |C\cap\mathcal{N}| \text{ is odd,} \\ 0 \pmod 2 & \text{if } C\in\mathcal{F} \text{ and } |C\cap\mathcal{N}| \text{ is even,} \end{cases} \tag{15}$$

$$x(e)\in\{0,1\} \text{ for all } e\in\mathcal{A}. \tag{16}$$

Notice that only variables associated with artificial edges have been included in problem (14)-(16). This is because edges associated with the original edges of G, should not be violated. If we work with the dual graph of G', problem (14)-(16) can be solved as a minimum T-join problem, see Section 2.

In Figure 2 we show an example of a graph G and its associated graph G'. If we look at a solution of (14)-(16) in the dual graph, we obtain a set of paths matching pairs of faces that have an odd number of negative edges, see Lemma 5. We illustrate this in Figure 3, for every violated edge we draw a perpendicular dashed line. This corresponds to an edge of the dual graph. Also we draw a square on each face having an odd number of negative edges. Also in Figure 3 we show the induced bipartite subgraph obtained after removing the nodes associated with the violated edges.

Lemma 5. *A signed planar graph has an even number of faces containing an odd number of negative edges.*

Proof. Start with all edges labeled positive. Pick one positive edge and change its label to negative. Then exactly two faces have an odd number of negative edges. After that, when we change any other label from positive to negative, either the number of faces with an odd number of negative edges increases by two, or decreases by two, or remains the same. □

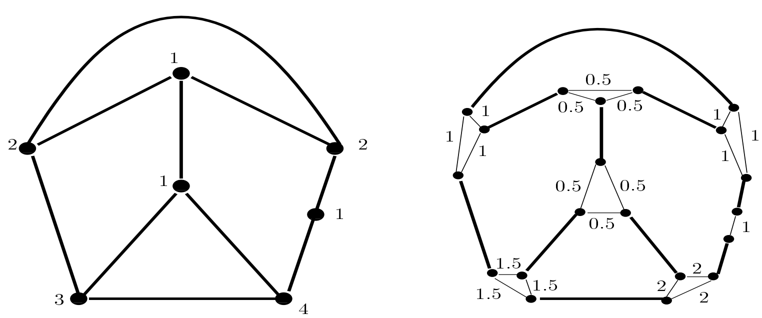

FIGURE 2. An example of the graphs G and G'. The numbers near the nodes are their weights. In G' the numbers near the positive edges are their weights.

Using planar duality, and the results of Edmonds & Johnson [5], we can see that problem (14)-(16) is equivalent to the linear program below.

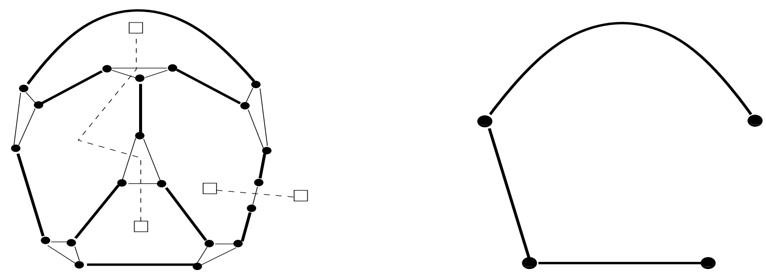

Figure 3. A solution. The violated edges are crossed with a dashed line. We also show the corresponding induced bipartite subgraph of the original graph.

$$\text{minimize } \sum_{e\in\mathcal{A}} \lambda(e)x(e) \tag{17}$$

$$\sum_{e\in C\cap\mathcal{A}} x(e) \geq 1 \quad \text{for each cycle } C \text{ with } |C\cap\mathcal{N}| \text{ odd,} \tag{18}$$

$$x(e) \geq 0 \quad \text{for all } e \in \mathcal{A}. \tag{19}$$

Now we state the main result of this section.

Theorem 6. *The problem of finding a maximum weighted induced bipartite subgraph of a planar graph with degree at most three, can be solved in $O(n^{3/2}\log n)$ time.*

4. Balancing signed graphs

A signed graph is called *balanced* if we can give the labels "a" and "b" to the nodes so that if and edge is positive, its endnodes have the same label; and if an edge is negative, its endnodes have different labels. Here we discuss how to apply the ideas of Section 3 for finding a maximum balanced subgraph.

4.1. **The node deletion case.**

Here we assume that $G = (V, E)$ is signed planar graph with non-negative weights $w(v)$, for each node $v \in V$, and with maximum degree three. We are looking for a balanced induced subgraph of maximum total weight.

We use a construction similar to the one in Section 3. The edges of the graph G' that are not artificial, keep the same labels as their associated edges in G. Then we have to solve (14)-(16) as described in Section 3. Thus we can state the following.

Theorem 7. *The problem of finding a maximum weighted balanced induced subgraph of a planar graph with degree at most three, can be solved in $O(n^{3/2}\log n)$ time.*

4.2. **The edge deletion case.**

Assume that $G = (V, E)$ is a signed planar graph with non-negative weights $w(e)$ for each edge $e \in E$. In this case we have no restriction on the node-degrees. Here we are looking for an edge set S of minimum weight so that $H = (V, E \setminus S)$ is balanced. Notice that if all edges are labeled negative, this is equivalent to the max-cut problem in a planar graph, that can be solved in polynomial time cf. [16].

We use the same reasoning as in Section 3, to formulate this as below.

$$\text{minimize} \sum_{e \in E} w(e)x(e)$$

$$\sum_{e \in C} x(e) \equiv \begin{cases} 1 \pmod 2 & \text{if } C \text{ is a cycle with an odd number of negative edges,} \\ 0 \pmod 2 & \text{if } C \text{ is a cycle with an even number of negative edges,} \end{cases}$$

$$x(e) \in \{0, 1\} \quad \text{for each edge } e \in E.$$

As before, this can be solved as a minimum T-join problem in the dual graph. Thus we have the following.

Theorem 8. *The problem of finding a maximum weighted balanced spanning subgraph of a planar graph, can be solved in $O(n^{3/2} \log n)$ time.*

5. Maximum directed induced acyclic subgraphs

Consider a planar directed graph $G = (V, A)$, with node-weights $\lambda(v) \geq 0$, for each node $v \in V$. We study the problem of finding a node set S of maximum weight that induces an acyclic subgraph. The complement of S is a directed feedback set. This problem is also known as the Directed Feedback Vertex Set Problem.

One can try a technique similar to the one in Section 3, splitting nodes, adding extra arcs keeping planarity, and using planar duality. We discuss here the limitations of this technique.

Consider the following simple transformation. Split each node v into v_1 and v_2, add the arc (v_1, v_2), replace every arc (u, v) with (u, v_1), and every arc (v, w) with (v_2, w). Let G' be this new graph. If G' is planar, we work with the dual graph as follows. Let D be the dual graph of G' regardless of the orientations of the arcs. Then for each arc a of G' let $a^\perp$ be the corresponding edge of the dual graph, we give an orientation to $a^\perp$, so that the pair $(a, a^\perp)$ forms a positively oriented basis of $\mathbb{R}^2$. Let $\vec{D}$ be the directed graph obtained with this orientation. Notice that directed cuts in $\vec{D}$ correspond to directed cycles of G'. Thus it follows from the Theorem of Lucchesi & Younger [21], that the following linear program has an optimal solution that is integer valued, moreover, this can be solved in polynomial time.

$$\text{minimize} \sum_{v \in V} \lambda(v)x(v) \tag{20}$$

$$\sum_{v \in C} x(v) \geq 1 \quad \text{for each directed cycle } C, \tag{21}$$

$$x(v) \geq 0 \quad \text{for all } v \in V. \tag{22}$$

To see this one should start with problem (7)-(9), associated with the graph $\vec{D}$. Then all variables associated with the original arcs are set to zero, and one obtains (20)-(22).

In particular, this transformation works when the degree of every node is at most three. The following theorem shows the limits of this transformation.

Theorem 9. *The minimum feedback vertex set problem is NP-Complete for planar graphs with maximum degree four, and with in-degree and out-degree at most two.*

Proof. We use a construction similar to the one used in [4] for induced bipartite subgraphs. We start with the following NP-complete problem, see [19].

Planar 3-Satisfiability (P3SAT).

Instance: A set $U = \{v_i \,|\, 1 \leq i \leq n\}$ of n boolean variables and a set $C = \{c_j \,|\, 1 \leq j \leq m\}$ of m clauses over U such that each clause contains exactly three variables or their complements. Furthermore, the following graph is planar:

$$\begin{aligned} &G_C = (V_C, E_C), \text{ where} \\ &V_C = \{c_j \,|\, 1 \leq j \leq m\} \cup \{v_i \,|\, 1 \leq i \leq n\}, \text{ and} \\ &E_C = \{c_j v_i \,|\, v_i \in c_j \text{ or } \bar{v}_i \in c_j\} \cup \{v_i v_{i+1} \,|\, 1 \leq i \leq n-1\} \cup \{v_n v_1\}. \end{aligned}$$

Question: Is there a truth assignment for U such that each clause in C is true?

Given a planar embedding of G_C we build a planar directed graph $G = (V, A)$ as follows.

- Each node v_i associated with a variable is replaced by a subgraph called *variable component*. Its nodes are $\{a^i_{j,1}, \ldots, a^i_{j,8}\}$, for $1 \leq j \leq m$. The arcs are:
  - $(a^i_{j,k}, a^i_{j,k+1})$, for $1 \leq k \leq 7$, for $1 \leq j \leq m$.
  - $(a^i_{j,8}, a^i_{j+1,1})$, for $1 \leq j \leq m$, with $a^i_{m+1,1} = a^i_{1,1}$.
  - $(a^i_{j,k+2}, a^i_{j,k})$, for $k = 1, 3, 5$, $(a^i_{j+1,1}, a^i_{j,7})$; for $1 \leq j \leq m$, with $a^i_{m+1,1} = a^i_{1,1}$.

  See Figure 4. There are $4m$ triangles, a directed cycle C_1 with $8m$ nodes and a directed cycle C_2 with $4m$ nodes. The embedding is done so that C_1 is oriented clockwise, and C_2 is oriented counter-clockwise.
- Each node c_j associated with a clause is replaced by three nodes c^1_j, c^2_j, and c^3_j. Assume that v_{i_1}, v_{i_2} and v_{i_3} are the three variables (or their complement) that appear in c_j. Assume that they appear in clockwise order in the embedding of G_C. For each variable v_{i_k} we have two cases:
  - If v_{i_k} appears in c_j, we add the arcs $(c^k_j, a^{i_k}_{j,2})$ and $(a^{i_k}_{j,4}, c^{k-1}_j)$, with $c^0_j = c^3_j$.
  - If $\bar{v}_{i_k}$ appears in c_j, we add the arcs $(c^k_j, a^{i_k}_{j,4})$ and $(a^{i_k}_{j,6}, c^{k-1}_j)$, with $c^0_j = c^3_j$.

  See Figure 5. These arcs are included in only one directed cycle D_j called a *clause cycle*.

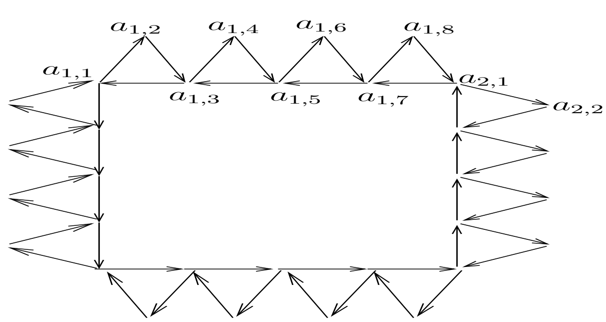

FIGURE 4. Subgraph associated with a variable v_i, for $m = 4$. The index i is not shown.

For a variable v_i consider it associated component. We need the following observations.

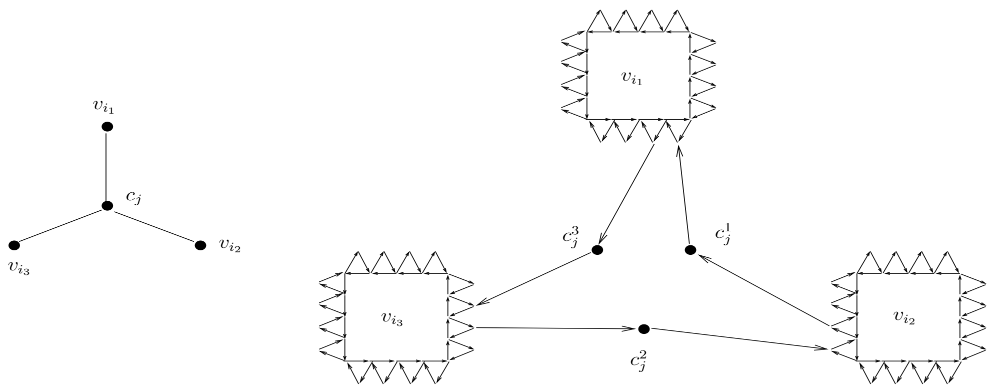

FIGURE 5. Nodes and arcs associated with a clause c_j. In this example c_j contains v_{i_1}, v_{i_2} and $\bar{v}_{i_3}$.

– A node can cover at most two triangles, and since there are $4m$ triangles, a node-set covering all triangles has cardinality at least $2m$. The triangles can be covered with the nodes $\{a^i_{j,3}, a^i_{j,7}\}$ for $1 \leq j \leq m$, or $\{a^i_{j,1}, a^i_{j,5}\}$ for $1 \leq j \leq m$. Denote by S^i_1 the first set and by S^i_2 the second set. We have $|S^i_1| = |S^i_2| = 2m$. Also any other cycle included in this component is covered by these two sets.
– Consider now any other set S of nodes covering all the triangles. Assume that S contains p nodes of degree two. Each of them covers exactly one triangle, so there are $4m-p$ triangles that should be covered with nodes of degree four. Since each of these nodes covers two triangles, we need at least $2m - p/2$ nodes of degree four. This shows that $|S| \geq 2m + p/2$.
– Consider now a node-set T containing only nodes of degree four and covering all triangles. If $|T| = 2m$, each node should cover two distinct triangles, this can only happen if $T = S^i_1$ or $T = S^i_2$. It follows that $2mn$ is a lower bound for the size of a minimum feedback set in G.

Suppose now that there is an assignment of values to the variables so that each clause is true. If a variable v_i is set to true, we pick the set S^i_1; otherwise v_i is set to false and we pick S^i_2. Thus we obtain a node set F of size $2mn$ that covers every directed cycle contained in each variable component. Now consider a clause cycle D_j corresponding to a clause c_j. Since at least one of the variables included in c_j is set to true, this cycle is covered.

On the other hand if there is a feedback set S of size $2mn$, its restriction to the subgraph associated with a variable v_i is either the set S^i_1 or S^i_2. In the first case we set v_i to true, and in the second case we set v_i to false. Since each clause cycle is covered, then each clause is set to true. □

REFERENCES

[1] F. BARAHONA, *Planar multicommodity flows, max cut and the Chinese Postman Problem*, in Polyhedral Combinatorics, DIMACS Series on Discrete Mathematics and Theoretical Computer Science No. 1, DIMACS, NJ, 1990, pp. 189–202.
[2] F. BARAHONA AND A. MAHJOUB, *Facets of the balanced (acyclic) induced subgraph polytope*, Mathematical Programming, 45 (1989), pp. 21–33.
[3] F. BARAHONA AND A. MAHJOUB, *Compositions of graphs and polyhedra I: Balanced induced subgraphs and acyclic subgraphs*, SIAM Journal on Discrete Mathematics, 7 (1994), pp. 344–358.

[4] H.-A. Choi, K. Nakajima, and C. S. Rim, *Graph bipartization and via minimization*, SIAM Journal on Discrete Mathematics, 2 (1989), pp. 38–47.

[5] J. Edmonds and E. L. Johnson, *Matching, euler tours and the chinese postman*, Mathematical programming, 5 (1973), pp. 88–124.

[6] G. Even, J. S. Naor, B. Schieber, and M. Sudan, *Approximating minimum feedback sets and multicuts in directed graphs*, Algorithmica, 20 (1998), pp. 151–174.

[7] P. Festa, P. M. Pardalos, and M. G. Resende, *Feedback set problems*, in Handbook of combinatorial optimization, Springer, 1999, pp. 209–258.

[8] P. Fouilhoux and A. Mahjoub, *Solving VLSI design and DNA sequencing problems using bipartization of graphs*, Computational Optimization and Applications, 51 (2012), pp. 749–781.

[9] M. Funke and G. Reinelt, *A polyhedral approach to the feedback vertex set problem*, in Integer programming and combinatorial optimization, Springer, 1996, pp. 445–459.

[10] H. N. Gabow, *An efficient implementation of edmonds' algorithm for maximum matching on graphs*, Journal of the ACM (JACM), 23 (1976), pp. 221–234.

[11] ———, *Centroids, representations, and submodular flows*, Journal of Algorithms, 18 (1995), pp. 586–628.

[12] G. Gardarin and S. Spaccapietra, *Integrity of data bases: A general lockout algorithm with deadlock avoidance.*, in IFIP Working Conference on Modelling in Data Base Management Systems, 1976, pp. 395–412.

[13] M. R. Garey and D. S. Johnson, *Computers and intractability: A guide to the theory of NP-completeness*, 1979.

[14] N. Garg, V. V. Vazirani, and M. Yannakakis, *Approximate max-flow min-(multi) cut theorems and their applications*, SIAM Journal on Computing, 25 (1996), pp. 235–251.

[15] M. X. Goemans and D. P. Williamson, *Primal-dual approximation algorithms for feedback problems in planar graphs*, Combinatorica, 18 (1998), pp. 37–59.

[16] F. Hadlock, *Finding a maximum cut of a planar graph in polynomial time*, SIAM Journal on Computing, 4 (1975), pp. 221–225.

[17] R. Karp, *Reducibility among combinatorial problems*, in Complexity of Computer Computations, R. Miller, J. Thatcher, and J. Bohlinger, eds., The IBM Research Symposia Series, Springer US, 1972, pp. 85–103.

[18] C. E. Leiserson and J. B. Saxe, *Retiming synchronous circuitry*, Algorithmica, 6 (1991), pp. 5–35.

[19] D. Lichtenstein, *Planar formulae and their uses*, SIAM Journal on Computing, 11 (1982), pp. 329–343.

[20] R. J. Lipton and R. E. Tarjan, *A separator theorem for planar graphs*, SIAM Journal on Applied Mathematics, 36 (1979), pp. 177–189.

[21] C. Lucchesi and D. Younger, *A minimax theorem for directed graphs*, J. London Math. Soc.(2), 17 (1978), pp. 369–374.

[22] C. L. Lucchesi, *A minimax equality for directed graphs*, PhD thesis, Thesis (Ph. D.)–University of Waterloo, 1976.

[23] A. Silberschatz, J. L. Peterson, and P. B. Galvin, *Operating system concepts*, Addison-Wesley Longman Publishing Co., Inc., 1991.

(M. Baïou) CNRS, and Université Clermont II, Campus des cézeaux BP 125, 63173 Aubière cedex, France.

E-mail address, M. Baïou: baiou@isima.fr

(F. Barahona) IBM T. J. Watson research Center, Yorktown Heights, NY 10589, USA.

E-mail address, F. Barahona: barahon@us.ibm.com