Journal of the Operations Research
Society of Japan
Vol. 43, No. 1, March 2000

A LINEAR TIME ALGORITHM FOR THE GENERALIZED STABLE SET PROBLEM ON TRIANGULATED BIDIRECTED GRAPHS

Daishin Nakamura
University of Electro-Communications

Akihisa Tamura
Kyoto University

(Received November 30, 1998; Revised May 19, 1999)

Abstract The generalized stable set problem is an extension of the maximum weight stable set problem for undirected graphs to bidirected graphs. This paper shows that the problem on triangulated bidirected graphs is solvable in linear time. We also propose an exact branch and bound algorithm for the general problem by applying the linear time algorithm.

1. Introduction

In this paper we examine 0−1 integer programming problems with two variables per inequality. Each non-redundant inequality of the integer program can be represented by one of the following forms:

$$x_i + x_j \leq 1, \qquad -x_i - x_j \leq -1, \qquad x_i - x_j \leq 0.$$

Hence, the problem can be formulated as follows: for a given finite set V, sets $P, N, I \subseteq V \times V$ and a weight vector $w \in \Re^V$ (we denote the ith element of w by w_i),

$$\text{[GSSP]} \quad \text{maximize} \quad \sum_{i \in V} w_i x_i \quad \text{subject to} \quad \begin{array}{ll} x_i + x_j \leq 1 & \text{for } (i,j) \in P, \\ -x_i - x_j \leq -1 & \text{for } (i,j) \in N, \\ x_i - x_j \leq 0 & \text{for } (i,j) \in I, \\ x_i \in \{0,1\} & \text{for } i \in V. \end{array}$$

Here we call this problem the *generalized stable set problem* (GSSP) because this is the maximum weight stable set problem (MWSSP) if $N = I = \emptyset$. The generalized set packing problem equivalent to the GSSP has also been studied in [2, 3, 7].

To deal with the GSSP, a 'bidirected' graph is useful. A *bidirected graph* $G = (V, E)$ has a set of vertices V and a set of edges E, in which each edge $e \in E$ has two vertices $i, j \in V$ as its endpoints and two associated signs (plus or minus) at i and j. The edges are classified into three types: the $(+,+)$-edges with two plus signs at their endpoints, the $(-,-)$-edges with two minus signs, and the $(+,-)$-edges (and the $(-,+)$-edges) with one plus and one minus sign. Undirected graphs may be interpreted as bidirected graphs with only $(+,+)$-edges. Given an instance of the GSSP, we obtain a bidirected graph by making $(+,+)$-edges, $(-,-)$-edges and $(+,-)$-edges for vertex-pairs of P, N and I respectively. Conversely, for a given bidirected graph with a weight vector on the vertices, by associating a variable x_i with each vertex i, we obtain the GSSP. We call a 0−1-vector satisfying the inequality system arising from a bidirected graph G, a *solution* of G, and also call a subset of vertices a solution of G, if its incidence vector is a solution of G. The GSSP is an optimization problem over the solutions of a bidirected graph.

It is well known that the MWSSP is NP-hard for general undirected graphs, and hence, the GSSP is also NP-hard. However, for several classes of undirected graphs, e.g., for perfect graphs [6] and claw-free graphs [10], the MWSSP is polynomially solvable. Particularly it can be solved in linear-time for triangulated graphs [4, 5] by applying the lexicographic breadth-first search [12]. On the other hand, there are several polynomial-time transformations from the GSSP to the MWSSP (see [13, 14]). Since these transformations preserve perfectness, the GSSP on perfect bidirected graphs can be solved in polynomial time [14]. Unfortunately, these transformations preserve neither claw-freeness nor triangulated-ness. The authors [11], however, proved that the GSSP on claw-free bidirected graphs is polynomially solvable.

In this paper, we propose a linear time algorithm for solving the GSSP on triangulated bidirected graphs. Moreover, by combining the linear time algorithm and the idea of Balas-Yu's algorithm [1] for the maximum weight clique problem, we will give an exact branch and bound algorithm for the GSSP.

2. Preliminaries

Since several distinct bidirected graphs may have the same set of solutions, it is convenient to deal with some kind of 'standard' bidirected graph. A bidirected graph is said to be *transitive*, if whenever there are edges $e_1 = (i, j)$ and $e_2 = (j, k)$ with opposite signs at j, then there is also an edge $e_3 = (i, k)$ whose signs at i and k agree with those of e_1 and e_2. Obviously, any bidirected graph and its transitive closure have the same solutions. A bidirected graph is said to be *simple* if it has no loops and if it has at most one edge for each pair of distinct vertices. Johnson and Padberg [9] showed that any transitive bidirected graph can be determined to have no solution or reduced to a simple one without essentially changing the set of solutions. They proved that a transitive bidirected graph has no solution if and only if it has a vertex with both a $(+,+)$-loop and a $(-,-)$-loop. For any bidirected graph, the associated simple and transitive bidirected graph can be constructed in time polynomial in the number of vertices. Although this construction cannot be done in linear time, we assume that a bidirected graph of any instance of the GSSP is simple and transitive because, in the application of Section 6, our linear time algorithm is applied to several triangulated subgraphs of a given instance of the GSSP, time after time.

Let $G = (V, E)$ be a simple and transitive bidirected graph and w be a weight vector on V. For a given subset $U \subseteq V$, we define the *reflection* of G at U by the transformation which reverses the signs of the u side of all edges incident to each $u \in U$ and we denote it by $G{:}U$. (For example, for G^2 in Figure 3 and $U = \{v_2, v_3\}$, $G^2{:}U$ is equal to G^1 in Figure 2.) Obviously, reflection preserves simplicity and transitivity. Let $w{:}U$ denote the vector defined by $(w{:}U)_i = -w_i$ if $i \in U$; otherwise $(w{:}U)_i = w_i$. For two subsets X and Y of V, let $X \triangle Y$ denote the symmetric difference of X and Y.

Lemma 2.1. *Let X be any solution of G. Then, $X \triangle U$ is a solution of $G{:}U$. The GSSP for (G, w) is equivalent to the GSSP for $(G{:}U, w{:}U)$.*

Proof. The first assertion is trivial from the definition of $G{:}U$. The second assertion follows from $\sum_{i \in X \triangle U}(w{:}U)_i = \sum_{i \in X \setminus U} w_i + \sum_{i \in U \setminus X}(-w_i) = \sum_{i \in X} w_i - \sum_{i \in U} w_i$, (the last term is a constant). ∎

We next define bicliques and biclique covers. A pair of disjoint sets of vertices $[C^+, C^-]$

is called a *biclique* of G if

(B1) there is an edge between any two vertices in $C^+ \cup C^-$, and
(B2) for any edge e of the vertex-induced subgraph $G[C^+ \cup C^-]$ of G by $C^+ \cup C^-$, if an endpoint i of e is in C^+ then e has a plus sign at i; otherwise e has a minus sign at i.

We call C^+ and C^- the *positive part* and *negative part* of $[C^+, C^-]$, respectively. Any solution of G satisfies the following inequality which is called the *biclique inequality*:

$$\sum_{i \in C^+} x_i + \sum_{i \in C^-} (1 - x_i) \leq 1$$

associated with a biclique $[C^+, C^-]$. On the other hand, any edge of G corresponds to a biclique and any vertex i implies two bicliques $[\{i\}, \emptyset]$ and $[\emptyset, \{i\}]$ which are associated with inequalities $x_i \leq 1$ and $x_i \geq 0$, respectively. Hence, the GSSP can be formulated as

$$\begin{array}{llll} \text{maximize} & \sum_{i \in V} w_i x_i \quad \text{subject to} & \sum_{i \in C^+} x_i + \sum_{i \in C^-} (1 - x_i) \leq 1 & \text{for } [C^+, C^-] \in \mathcal{B}, \\ & & x_i \in \{0, 1\} & \text{for } i \in V, \end{array}$$

where $\mathcal{B}$ is the set of all bicliques of G. For perfect bidirected graphs which we will define later, Sewell [13] proved that the LP-relaxation of the above formulation has an integral optimal solution for any weight vector (Guenin [7], and Ikebe and Tamura [8] also proved equivalent statements, independently). The dual problem of the LP-relaxation of the above formulation is

$$\begin{array}{llll} \text{minimize} & \sum_{C = [C^+, C^-] \in \mathcal{B}} (1 - |C^-|) \cdot y_C \quad \text{subject to} & \sum_{C^+ \ni i} y_C - \sum_{C^- \ni i} y_C = w_i & \text{for } i \in V, \\ & & y_C \geq 0 & \text{for } C \in \mathcal{B}. \end{array}$$

Since triangulated bidirected graphs are perfect as we will define later, the GSSP and the dual problem have the same optimal value. We call a feasible solution of the dual problem a *fractional biclique cover*, or shortly, a *biclique cover*. Our algorithm for the triangulated case finds a maximum weight solution and a minimum weight biclique cover having the same weight.

Here we represent a biclique cover by a set $\mathcal{C}$ including the bicliques of positive weights and a weight function $y : \mathcal{C} \to \Re$. Let $(\mathcal{C}, y)$ be a biclique cover for an instance (G, w). For any fixed subset $U \subseteq V$, let $C_U = [C^+ \triangle U_C, C^- \triangle U_C]$ and $y_U(C_U) = y(C)$ for each $C = [C^+, C^-] \in \mathcal{C}$, and $\mathcal{C}_U = \{C_U : C \in \mathcal{C}\}$ where $U_C = U \cap (C^+ \cup C^-)$.

Lemma 2.2. *$(\mathcal{C}_U, y_U)$ is a biclique cover for $(G{:}U, w{:}U)$. The difference of weights between $(\mathcal{C}_U, y_U)$ and $(\mathcal{C}, y)$ is $-\sum_{i \in U} w_i$, a constant.*

Proof. Obviously, each C_U is a biclique for $G{:}U$. The assertions follow from

$$i \notin U \implies \sum_{C_U{}^+ \ni i} y_U(C_U) - \sum_{C_U{}^- \ni i} y_U(C_U) = \sum_{C^+ \ni i} y(C) - \sum_{C^- \ni i} y(C) = w_i = (w{:}U)_i,$$

$$i \in U \implies \sum_{C_U{}^+ \ni i} y_U(C_U) - \sum_{C_U{}^- \ni i} y_U(C_U) = \sum_{C^- \ni i} y(C) - \sum_{C^+ \ni i} y(C) = -w_i = (w{:}U)_i,$$

and

$$\begin{aligned}\sum_{C_U \in \mathcal{C}_U} (1 - |C_U{}^-|) \cdot y_U(C_U) &= \sum_{C \in \mathcal{C}} (1 - |C^- \triangle U_C|) \cdot y(C) \\ &= \sum_{C \in \mathcal{C}} (1 - |C^-| + |C^- \cap U| - |C^+ \cap U|) \cdot y(C) \\ &= \sum_{C \in \mathcal{C}} (1 - |C^-|) \cdot y(C) - \sum_{i \in U} \left(\sum_{C^+ \ni i} y(C) - \sum_{C^- \ni i} y(C) \right) \\ &= \sum_{C \in \mathcal{C}} (1 - |C^-|) \cdot y(C) - \sum_{i \in U} w_i. \qquad \blacksquare\end{aligned}$$

This lemma says that a minimum weight biclique cover for $(G{:}U, w{:}U)$ can be obtained from a minimum weight biclique cover for (G, w), and vice versa.

Given a bidirected graph G, its *underlying graph*, denoted by $\underline{G}$, is defined as the undirected graph obtained from G by changing all the edges to $(+,+)$-edges. A bidirected graph is said to have a property P if it is simple and transitive and if its underlying graph has the property P. For example, a *perfect* (or *claw-free* or *triangulated*) bidirected graph is a simple and transitive bidirected graph whose underlying graph is perfect (or claw-free or triangulated). An undirected graph is called *triangulated* (or chordal) if it has no chordless cycle of length at least four, that is, every simple cycle of length at least four has an edge joining non-consecutive vertices in the cycle. Since triangulated undirected graphs are perfect, triangulated bidirected graphs are also perfect. It is known that an undirected graph $G = (V, E)$ is triangulated if and only if there is a vertex-ordering $\pi : V \to \{1, \cdots, n\}$ where $n = |V|$ such that for each $i \in \{1, \cdots, n\}$, the set consisting of $v = \pi^{-1}(i)$ and the vertices adjacent to v in $\{\pi^{-1}(i+1), \cdots, \pi^{-1}(n)\}$ forms a clique, that is, any two vertices of the set are adjacent (see, for instance, [12]). Such an ordering is called a *perfect vertex elimination scheme* (PVES) and can be found in a linear time for any triangulated undirected graph [12].

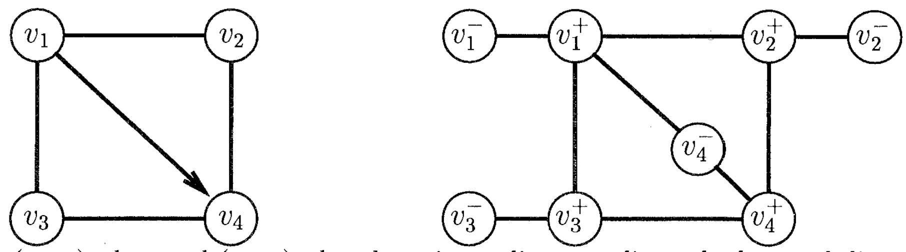
We say that a vertex is *positive* (or *negative*) if all edges incident to it have plus (or minus) signs at it, and that a vertex is *mixed* if it is neither positive nor negative. If a bidirected graph has no $(-,-)$-edge, it is said to be *pure*. We note that the negative part of any biclique of a pure bidirected graph has at most one vertex. We call a bidirected graph *canonical* if it is simple, transitive and pure, and if it has no negative vertices. For any instance (G, w) of the GSSP, we can transform it to an equivalent one whose bidirected graph is canonical as follows. Johnson and Padberg [9] proved that if G is simple and transitive, G has at least one solution $U \subseteq V$ and that a solution can be found in $O(|V|+|E|)$ time. From Lemma 2.1, $G{:}U$ has the solution $U \triangle U = \emptyset$, that is, $G{:}U$ must be pure. Let W be the set of negative vertices of $G{:}U$. Then $G{:}U{:}W$ has no negative vertex, and furthermore, it is pure because any edge (v, w) of $G{:}U$ with $w \in W$ must be a $(+,-)$-edge. Since this transformation can be done in linear time, we assume that a given bidirected graph of the GSSP is canonical in the sequel.

3. Transformations from GSSP to MWSSP

In this section, we briefly introduce transformations from the GSSP to the MWSSP.

Given a simple and transitive bidirected graph $G = (V, E)$ and a weight vector $w \in \Re^V$, let us define the undirected graph $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ and the weight vector $\widetilde{w} \in \Re^{\widetilde{V}}$ by

$$\widetilde{V} = V^+ \cup V^-, \quad V^+ = \{i^+ \mid i \in V\}, \quad V^- = \{i^- \mid i \in V\},$$



(We draw $(+,+)$-edges and $(+,-)$-edges by using ordinary undirected edges and directed edges, respectively. The head of an arrow means a minus sign and the tail means a plus sign.)

Figure 1: G is triangulated but $\widetilde{G}$ is not.

$$\begin{aligned} \widetilde{E} &= \{(i^\alpha, j^\beta) \mid (\alpha,\beta)\text{-edge incident to } i \text{ and } j \text{ in } G\} \cup \{(i^+, i^-) \mid i \in V\}, \\ \widetilde{w}_{i^+} &= \max\{w_i, 0\} \quad (i \in V) \quad \text{and} \quad \widetilde{w}_{i^-} = \max\{-w_i, 0\} \quad (i \in V). \end{aligned}$$

It is known that if a maximum weight stable set $S^+ \cup S^-$ ($S^+ \subseteq V^+, S^- \subseteq V^-$) is maximal with respect to set-inclusion among all the stable sets of $\widetilde{G}$ then S^+ is a maximum weight solution of G (see, [7, 15]). That is, the GSSP for (G, w) can be transformed to the MWSSP for $(\widetilde{G}, \widetilde{w})$. Other transformations proposed in [13, 14] are essentially equivalent to the above one since undirected graphs constructed by these are obtained from $\widetilde{G:U}$ by deleting all zero-weight vertices for some $U \subseteq V$. Since these transformations preserve perfectness [13, 14, 15], we can solve the GSSP in polynomial time on perfect bidirected graphs, rightfully on triangulated bidirected graphs, by applying the celebrated algorithm in [6]. Unfortunately, we cannot apply the linear time algorithm for the MWSSP on triangulated graphs in the above approach because all known transformations may not preserve triangulated-ness. For example, in Figure 1 the right-hand graph $\widetilde{G}$ is not triangulated even though the left-hand graph G is triangulated. If $w > 0$, then transformations in [13, 14] construct a chordless cycle of length 4. Hence, it seems difficult to develop a linear time algorithm for the GSSP on triangulated bidirected graphs by adopting the above approach.

4. A Linear Time Algorithm on Triangulated Bidirected Graphs

Let $G = (V, E)$ be a canonical triangulated bidirected graph and w be a weight vector on V. A vertex-ordering $\pi : V \to \{1, \ldots, n\}$ is said to be *topological* if $\pi(u) < \pi(v)$ for each $(-,+)$-edge (u, v).

If π is a PVES, in addition, then we call π a *topological PVES* (T-PVES). Here we suppose that we have already found a T-PVES π for G. In this section, we consider how to find a minimum weight biclique cover and a maximum weight solution of G by using π. We will describe how to find a T-PVES in the next section.

We denote $u \overset{++}{\sim} v$ if there is a $(+,+)$-edge (u, v), and $u \overset{+-}{\sim} v$ or $v \overset{-+}{\sim} u$ if there is a $(+,-)$-edge (u, v). For a T-PVES π and a vertex v, we define

$$\begin{aligned} N_\pi(v) &\overset{\text{def}}{=} \{u \in V \mid \pi(v) < \pi(u),\ v \text{ is adjacent to } u\}, \\ N_\pi^+(v) &\overset{\text{def}}{=} \{u \in N_\pi(v) \mid v \overset{++}{\sim} u\} \text{ and} \\ N_\pi^-(v) &\overset{\text{def}}{=} \{u \in N_\pi(v) \mid v \overset{-+}{\sim} u\}. \end{aligned}$$

Note that any two distinct vertices in $N_\pi(v)$ are adjacent to each other since π is a PVES, and $N_\pi(v) = N_\pi^+(v) \cup N_\pi^-(v)$ because π is topological and G has no $(-,-)$-edges. We define

$$C_\pi^+(v) \overset{\text{def}}{=} \{v\} \cup \{u \in N_\pi^+(v) \mid \text{there is no vertex } t \in N_\pi^+(v) \text{ with } u \overset{+-}{\sim} t\} \text{ and}$$

$$C_\pi^-(v) \stackrel{\text{def}}{=} \{v\} \cup \{u \in N_\pi^-(v) \mid \text{there is no vertex } t \in N_\pi^-(v) \text{ with } u \overset{+-}{\sim} t\}.$$

From the definitions, both $[C_\pi^+(v), \emptyset]$ and $[C_\pi^-(v) \setminus \{v\}, \{v\}]$ are bicliques.

Lemma 4.1. *For any vertex $u \in N_\pi^+(v) \setminus C_\pi^+(v)$, there uniquely exists $t \in C_\pi^+(v) \setminus \{v\}$ such that $u \overset{+-}{\sim} t$. For any vertex $u \in N_\pi^-(v) \setminus C_\pi^-(v)$, there uniquely exists $t \in C_\pi^-(v) \setminus \{v\}$ such that $u \overset{+-}{\sim} t$.*

Proof. For $u \in N_\pi^+(v) \setminus C_\pi^+(v)$, let $t \in N_\pi^+(v)$ be the smallest vertex in the order π such that $u \overset{+-}{\sim} t$. Suppose to the contrary that $t \in N_\pi^+(v) \setminus C_\pi^+(v)$. There is a vertex $s \in N_\pi^+(v)$ such that $t \overset{+-}{\sim} s$ by the definition of $C_\pi^+(v)$. Since π is topological and G is transitive, $\pi(s) < \pi(t)$ and $u \overset{+-}{\sim} s$, a contradiction. Therefore $t \in C_\pi^+(v) \setminus \{v\}$. For any vertex r in $C_\pi^+(v) \setminus \{v, t\}$, $r \overset{++}{\sim} t$ holds, and hence, $r \overset{++}{\sim} u$ by the transitivity. Thus such a vertex is unique. Similarly we can prove the statement for $u \in N_\pi^-(v) \setminus C_\pi^-(v)$. ∎

We will assign values to bicliques $[C_\pi^+(v), \emptyset]$ and $[C_\pi^-(v) \setminus \{v\}, \{v\}]$ by Procedure I below so that they form a biclique cover for (G, w). After Procedure I, Procedure II constructs a solution of (G, w). These are extensions of the algorithm for finding a minimum clique cover and a maximum stable set for triangulated undirected graphs [4, 5].

[Procedure I]
$\tilde{w} := w$; $\mathcal{C} := \emptyset$; $X := \emptyset$;
for $i := 1$ **to** n **do begin**
  $v := \pi^{-1}(i)$;
  if $\tilde{w}_v > 0$ **then begin**
    $y([C_\pi^+(v), \emptyset]) := \tilde{w}_v$;
    $\mathcal{C} := \mathcal{C} \cup \{[C_\pi^+(v), \emptyset]\}$;
    for $\forall u \in C_\pi^+(v) \setminus \{v\}$ **do** $\tilde{w}_u := \tilde{w}_u - \tilde{w}_v$;
    $X := X \cup \{v\}$;
  end else begin
    $y([C_\pi^-(v) \setminus \{v\}, \{v\}]) := -\tilde{w}_v$;
    $\mathcal{C} := \mathcal{C} \cup \{[C_\pi^-(v) \setminus \{v\}, \{v\}]\}$;
    for $\forall u \in C_\pi^-(v) \setminus \{v\}$ **do** $\tilde{w}_u := \tilde{w}_u + \tilde{w}_v$;
  end if
end for

[Procedure II]
for $i := n$ **downto** 1 **do begin**
  $v := \pi^{-1}(i)$;
  if $\tilde{w}_v > 0$ **then begin**
    for $\forall u \in C_\pi^+(v) \setminus \{v\}$ **do** $\tilde{w}_u := \tilde{w}_u + \tilde{w}_v$;
    if $X \cap (C_\pi^+(v) \setminus \{v\}) \neq \emptyset$ **then** $X := X \setminus \{v\}$;
  end else begin
    for $\forall u \in C_\pi^-(v) \setminus \{v\}$ **do** $\tilde{w}_u := \tilde{w}_u - \tilde{w}_v$;
    if $X \cap (C_\pi^-(v) \setminus \{v\}) \neq \emptyset$ **then** $X := X \cup \{v\}$;
  end if
end for

After executing Procedure I, $(\mathcal{C}, y)$ is a biclique cover. Its weight $\sum_{C=[C^+,C^-]\in\mathcal{C}}(1-|C^-|)\cdot y(C)$ is the sum of all values $y([C_\pi^+(v), \emptyset])$, because $[C_\pi^-(v) \setminus \{v\}, \{v\}]$ has exactly one vertex in its negative part and none of the values $y([C_\pi^-(v) \setminus \{v\}, \{v\}])$ are concerned. Hence the weight of $(\mathcal{C}, y)$ is equal to $\sum_{i\in X} \widetilde{w}_i$. On the other hand, X may not be a solution of G. Procedure II modifies X so that it forms a solution.

Lemma 4.2. *The following claims hold at the end of each iteration in Procedure II.*

- *The value $\sum_{i\in X} \widetilde{w}_i$ is preserved, i.e. it is equal to the weight of $(\mathcal{C}, y)$.*
- *$X \cap V_i$ is a solution on the subgraph $G[V_i]$ induced by V_i, where $V_i \stackrel{\text{def}}{=} \{u \mid \pi^{-1}(u) \geq i\}$.*

Proof. By induction on i. The claims hold for $i = n$. Suppose that the claims hold for some $i+1$ with $i \leq n-1$. Let $v = \pi^{-1}(i)$. We consider the case that $\widetilde{w}_v > 0$. If $|X \cap (C_\pi^+(v) \setminus \{v\})| = 0$, then $\sum_{i\in X} \widetilde{w}_i$ is preserved. If $|X \cap (C_\pi^+(v) \setminus \{v\})| = 1$, then $\sum_{i\in X} \widetilde{w}_i$ is also preserved since v is deleted from X but $\widetilde{w}_v$ is added to $\widetilde{w}_u$, where $X \cap (C_\pi^+(v) \setminus \{v\}) = \{u\}$. Otherwise let t and u be distinct vertices in $X \cap (C_\pi^+(v) \setminus \{v\})$. Then $t \overset{++}{\sim} u$ and $t, u \in X \cap V_{i+1}$. This contradicts the second claim for $i+1$. The case $w_v \leq 0$ can be proved similarly. Thus the first claim holds for i. The second claim does not hold for i only in the following four possible cases:

(Case 1) $\widetilde{w}_v > 0$, $X \cap (C_\pi^+(v) \setminus \{v\}) \neq \emptyset$ and $X \cap N_\pi^-(v) \neq \emptyset$.
(Case 2) $\widetilde{w}_v > 0$, $X \cap (C_\pi^+(v) \setminus \{v\}) = \emptyset$ and $X \cap N_\pi^+(v) \neq \emptyset$.
(Case 3) $\widetilde{w}_v \leq 0$, $X \cap (C_\pi^-(v) \setminus \{v\}) \neq \emptyset$ and $X \cap N_\pi^+(v) \neq \emptyset$.
(Case 4) $\widetilde{w}_v \leq 0$, $X \cap (C_\pi^-(v) \setminus \{v\}) = \emptyset$ and $X \cap N_\pi^-(v) \neq \emptyset$.

In Case 1, let x be any element in $X \cap N_\pi^-(v)$. Then by the transitivity of G, $x \overset{++}{\sim} y$ for all $y \in C_\pi^+(v) \setminus \{v\}$. This means that $X \cap (C_\pi^+(v) \setminus \{v\}) = \emptyset$, thus Case 1 does not occur. In Case 2, let u be any element in $(X \cap N_\pi^+(v)) \setminus C_\pi^+(v)$. Then there exists a vertex $t \in C_\pi^+(v) \setminus \{v\}$ such that $u \overset{+-}{\sim} t$ by Lemma 4.1. This means that $t \in X$, and thus Case 2 does not occur. Similarly neither Case 3 nor Case 4 occurs. Hence the second claim holds for i. ∎

Lemma 4.3. *Given a T-PVES for (G, w), a minimum weight biclique cover and a maximum weight solution can be found in linear time.*

Proof. At the end of Procedure II, X is a solution and $\sum_{i\in X} \widetilde{w}_i$ is equal to the weight of the biclique cover $(\mathcal{C}, y)$ by Lemma 4.2. Moreover, at this point, $w = \widetilde{w}$. Hence $(\mathcal{C}, y)$ is a minimum weight biclique cover and X is a maximum weight solution for (G, w).

We now consider the time complexity. We assume that $N_\pi^+(v)$ and $N_\pi^-(v)$ are sorted in the order π for each vertex v. This can be done in linear time by re-constructing adjacency lists. Let us show that $C_\pi^+(v)$ and $C_\pi^-(v)$ can be found for a given v in time proportional to $|N_\pi(v)|$, and this completes the proof.

From Lemma 4.1, we can easily show that $N_\pi^+(v) \setminus C_\pi^+(v) = \bigcup_{u\in C_\pi^+(v)} N_\pi^-(u)$ and that $N_\pi^-(t) \cap N_\pi^-(u) = \emptyset$ for any distinct vertices $t, u \in C_\pi^+(v)$. Thus the following procedure finds $C_\pi^+(v)$ in time proportional to $|N_\pi^+(v)|$.

$C := \{v\} \cup N_\pi^+(v)$;
comment $N_\pi^+(v) = \{u_1, \ldots, u_{|N_\pi^+(v)|}\}$, $\pi(u_1) < \cdots < \pi(u_{|N_\pi^+(v)|})$;
for $i := 1$ **to** $|N_\pi^+(v)|$ **do if** $u_i \in C$ **then** $C := C \setminus N_\pi^-(u_i)$;
comment $C = C_\pi^+(v)$;

Analogously, $C_\pi^-(v)$ can be found in $O(|N_\pi^-(v)|)$ time. ∎

Example 4.4. Let us consider a triangulated bidirected graph G^1 in Figure 2. The vertex order π defined by $\pi(v_i) = i$ $(i = 1, \cdots, 6)$ is a T-PVES for G^1. Let $w^1 = (2, -1, 4, 3, 2, 4)$ be a given weight vector on $\{v_1, \cdots, v_6\}$. Procedures I and II find a biclique cover and a

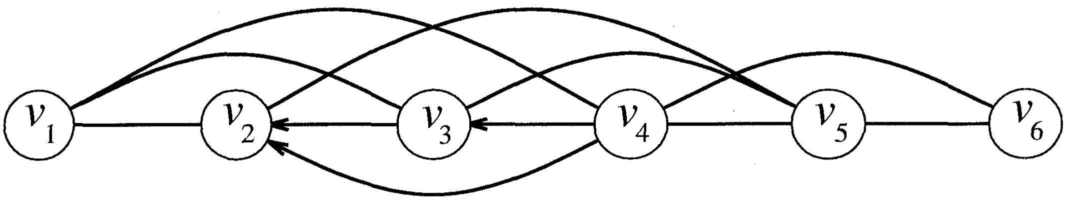


Figure 2: A triangulated bidirected graph G^1.

solution as Table 1. The biclique cover selected by Procedure I and the solution $\{v_2, v_3, v_6\}$ constructed by Procedure II have the same weight 7.

Table 1: An example

i	$\tilde{w}^1$	Procedure I $[C^+, C^-]$	Procedure I $y(C)$	Procedure I X	Procedure II X
	$(2, -1, 4, 3, 2, 4)$			$\emptyset$	$\{v_2, v_3, v_6\}$
1	$(2, -3, 4, 3, 2, 4)$	$[\{v_1, v_2\}, \emptyset]$	2	$\{v_1\}$	$\{v_1, v_2, v_3, v_6\}$
2	$(2, -3, 1, 3, 2, 4)$	$[\{v_3\}, \{v_2\}]$	3	$\{v_1\}$	$\{v_1, v_3, v_6\}$
3	$(2, -3, 1, 3, 1, 4)$	$[\{v_3, v_5\}, \emptyset]$	1	$\{v_1, v_3\}$	$\{v_1, v_3, v_6\}$
4	$(2, -3, 1, 3, -2, 1)$	$[\{v_4, v_5, v_6\}, \emptyset]$	3	$\{v_1, v_3, v_4\}$	$\{v_1, v_3, v_4, v_6\}$
5	$(2, -3, 1, 3, -2, 1)$	$[\emptyset, \{v_5\}]$	2	$\{v_1, v_3, v_4\}$	$\{v_1, v_3, v_4, v_6\}$
6	$(2, -3, 1, 3, -2, 1)$	$[\{v_6\}, \emptyset]$	1	$\{v_1, v_3, v_4, v_6\}$	$\{v_1, v_3, v_4, v_6\}$

5. Finding a T-PVES

For a vertex v in a canonical triangulated bidirected graph G, we call v *bad* if there exist distinct vertices a and b such that $a \overset{+-}{\sim} v$, $b \overset{+-}{\sim} v$, and a is not adjacent to b. Clearly, G has no T-PVES if G has a bad vertex.

We first consider the case that G has no bad vertex. At the last of this section we will examine the case that G has bad vertices.

First of all, let us review the lexicographic breadth-first search (LEX-BFS) [12] to find a PVES π for a given triangulated undirected graph.

Initially all vertices are unnumbered. During an execution of LEX-BFS, for each unnumbered vertex x, the *label* $L(x)$ of x is the set of numbered vertices which are adjacent to x. The *lexicographic order* of labels is defined by

$$L(x) \succ_{\text{lex}} L(y) \quad \overset{\text{def}}{\Longleftrightarrow} \quad \exists z \in L(x) \setminus L(y) \text{ such that } \{u \in L(x) \mid \pi(u) > \pi(z)\} = \{u \in L(y) \mid \pi(u) > \pi(z)\}.$$

At the i-th step ($i = 1, \ldots, n$), LEX-BFS selects an unnumbered vertex v which has the largest label in the lexicographic order, and let $\pi(v) = n - i + 1$. It can be proved that this algorithm finds a PVES π correctly.

A special data-structure is used to find a PVES in linear time. Unnumbered vertices are partitioned into sets $\mathcal{L} = \{S_0, \ldots, S_k\}$. Here each S_j is the non-empty set of unnumbered vertices having the same label, and $\mathcal{L}$ is a doubly-linked list sorted in the lexicographic order. The set S_0 is the set of unnumbered vertices having the largest label. At the i-th step LEX-BFS extracts a vertex $v \in S_0$, lets $\pi(v) = n - i + 1$, and modifies the structure $\mathcal{L}$. The algorithm is described as follows.

[Lexicographic breadth-first search (LEX-BFS)]
$\mathcal{L} := \{V\}$; (∗1)
for $i := n$ **downto** 1 **do begin**
  comment S_0 means the set at the head of $\mathcal{L}$;
  $v :=$ any element of S_0; (∗2)
  $\pi(v) := i$; $S_0 \leftarrow S_0 \setminus \{v\}$; **if** $S_0 = \emptyset$ **then** delete S_0 from $\mathcal{L}$;
  $\mathcal{M} := \emptyset$;
  for each unnumbered vertex u that is adjacent to v (∗3) **do begin**
    comment $S[u]$ means the set in $\mathcal{L}$ which contains u;
    if $S[u] \notin \mathcal{M}$ **then begin**
      insert $\{u\}$ into $\mathcal{L}$ just before $S[u]$; $\mathcal{M} := \mathcal{M} \cup \{S[u]\}$;
    end else begin
      comment $\widetilde{S}[u]$ means the set in $\mathcal{L}$ just before $S[u]$;
      $\widetilde{S}[u] := \widetilde{S}[u] \cup \{u\}$; (∗4)
    end if
    $S[u] := S[u] \setminus \{u\}$; **if** $S[u] = \emptyset$ **then** delete $S[u]$ from $\mathcal{L}$;
  end for
end for

Now we return to our discussion. LEX-BFS finds a PVES in linear time for a triangulated bidirected graph, but this may not be a T-PVES. Hence we slightly modify LEX-BFS to find a T-PVES in the case that G has no bad vertex.

Before an execution of LEX-BFS, find a topological order $\tau : V \rightarrow \{1, \ldots, n\}$ so that $\tau(u) < \tau(v)$ for any $u \overset{-+}{\sim} v$. Since G is simple and transitive, there is no directed cycle consisting of $(+, -)$-edges. Hence such a topological order always exists. Next sort adjacency lists in the order τ for each vertex. This can be done in linear time by re-constructing adjacency lists.

At (∗1), represent V by a doubly-linked linear list sorted in the order τ. At (∗2), select the vertex v from S_0 that is the largest in the order τ, i.e., select the tail vertex v from the doubly-linked linear list representing S_0. At (∗3), select unnumbered vertices that are adjacent to v in the order τ, i.e., in the order of the adjacency list of v. At (∗4), insert u at the tail of the doubly-linked linear list representing $\widetilde{S}[u]$. Note that u is the largest in $\widetilde{S}[u]$ in the order τ from the modification at (∗3). During an execution of LEX-BFS, each set S in $\mathcal{L}$ is sorted in the order τ.

Lemma 5.1. *The modified version of LEX-BFS finds a T-PVES in linear time if there is no bad vertex.*

Proof. Let us consider the time when v is selected at (∗2) in the modified version. Suppose to the contrary that there is an unnumbered vertex u such that $u \overset{+-}{\sim} v$. Because τ is topological, $\tau(u) > \tau(v)$. This means that $u \notin S_0$, since v is the largest vertex in S_0 in the order τ. Hence u has a smaller label than that of v, because S_0 is the set of unnumbered vertices having the largest label.

Let x be a vertex that is adjacent to v but $x \neq u$. If $v \overset{++}{\sim} x$, then $u \overset{++}{\sim} x$ from the transitivity. Similarly if $v \overset{+-}{\sim} x$, then $u \overset{+-}{\sim} x$. Otherwise, $v \overset{-+}{\sim} x$, then u is adjacent to x because v is not bad. Hence the set of numbered vertices which are adjacent to u includes the set of numbered vertices which are adjacent to v, i.e., the label of u is not smaller than the label of v, a contradiction. ∎

We finally deal with the case that G has bad vertices. We will show that bad vertices can be converted into non-bad vertices by reflection.

Lemma 5.2. *Let B be the set of all the bad vertices of a canonical triangulated bidirected graph G. Then $G{:}B$ is a canonical triangulated bidirected graph having no bad vertex.*

Proof. Let v be a vertex. If v is not bad in G, then it is not bad in $G{:}B$ because G has no $(-,-)$-edge. Then let v be a bad vertex in G, and a and b be distinct vertices such that $a \overset{+-}{\sim} v$, $b \overset{+-}{\sim} v$, and a is not adjacent to b. Assume that there are distinct vertices c and d such that ($c \overset{++}{\sim} v$ or $c \overset{-+}{\sim} v$), ($d \overset{++}{\sim} v$ or $d \overset{-+}{\sim} v$), and c is not adjacent to d. From the transitivity of G, $\{a, b, c, d\}$ induces a chordless cycle of length 4. This contradicts the fact that G is triangulated. Therefore $G{:}B$ has no bad vertex.

If $v \overset{+-}{\sim} u$ for some u in G then $a \overset{+-}{\sim} u$ and $b \overset{+-}{\sim} u$ from the transitivity, thus u is also bad. Suppose that $v \overset{++}{\sim} t$ for some t. Then $a \overset{++}{\sim} t$ and $b \overset{++}{\sim} t$ from the transitivity. In this case t is not bad, since otherwise there exists a chordless cycle of length 4 by the same discussion above. Therefore $G{:}B$ has no $(-,-)$-edge. Hence $G{:}B$ is canonical. ∎

Next we consider how to find the set of all the bad vertices B in linear time. Let $\pi : V \to \{1, \ldots, n\}$ be a PVES of G. This can be found in linear time by LEX-BFS. For a vertex v, let us define $N^-(v) \overset{\text{def}}{=} \{x \mid x \overset{+-}{\sim} v\}$. If $|N^-(v)| \leq 1$, then v is not bad. If $|N^-(v)| \geq 2$, let x be the smallest vertex in $N^-(v)$ in the order π. Then v is bad if and only if there exists a vertex $y \in N^-(v) \setminus \{x\}$ such that x is not adjacent to y, since π is a PVES. To check whether x is adjacent to y or not in constant time, we use the following procedure because we do not have the adjacency matrix.

```
[Bad-Vertices]
for ∀x ∈ V do S(x) := ∅;
for ∀v ∈ V do begin
    if |N⁻(v)| ≥ 2 then begin
        x := the smallest vertex of N⁻(v) in the order π;
        for ∀y ∈ N⁻(v) \ {x} do S(x) := S(x) ∪ {(y, v)};
        comment If x is not adjacent to y, then v is bad;
    end if
end for
B := ∅; for ∀v ∈ V do a[v] := 0;
for ∀x ∈ V do begin
    for each y that is adjacent to x do a[y] := 1;
    for each (y, v) ∈ S(x) do if a[y] = 0 then B := B ∪ {v};
    for each y that is adjacent to x do a[y] := 0;
end for
```

Theorem 5.3. *For a given canonical triangulated bidirected graph G and any weight vector w on the vertices, a minimum weight biclique cover and a maximum weight solution can be found in linear time.*

Proof. First find the set of all the bad vertices B by the procedure above. Next find a T-PVES of $G{:}B$, see Lemmas 5.1 and 5.2. Then find a minimum weight biclique cover and

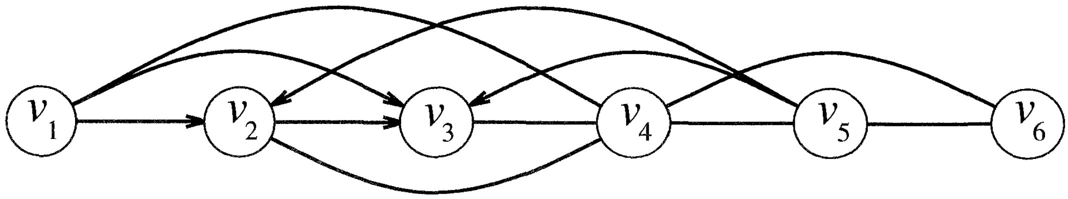


Figure 3: A triangulated bidirected graph G^2.

a maximum weight solution for $(G:B, w:B)$, see Lemma 4.3. Finally convert them into a minimum weight biclique cover and a maximum weight solution for (G, w), see Lemmas 2.1 and 2.2. All the procedures can be done in linear time. ∎

Example 5.4. Let us consider a triangulated bidirected graph G^2 in Figure 3. The vertex order π defined by $\pi(v_i) = i$ $(i = 1, \cdots, 6)$ is a PVES for G^2, but not a T-PVES. The procedure Bad-Vertices verifies that $B = \{v_2, v_3\}$ is the set of bad vertices of G^2. Then, the bidirected graph $G^2{:}B$ has a T-PVES, e.g., π. We note that $G^2{:}B$ is equivalent to G^1 in Example 4.4. Let $w^2 = (2, 1, -4, 3, 2, 4)$ be a given weight vector on the vertices $\{v_1, \cdots, v_6\}$ of G^2. As in Example 4.4, we obtain an optimal biclique cover and an optimal solution:

$$\begin{aligned} \mathcal{C}^1 &= \{[\{v_1, v_2\}, \emptyset],\ [\{v_3\}, \{v_2\}],\ [\{v_3, v_5\}, \emptyset],\ [\{v_4, v_5, v_6\}, \emptyset],\ [\emptyset, \{v_5\}],\ [\{v_6\}, \emptyset]\}, \\ y(C) &= (2, 3, 1, 3, 2, 1), \\ X^1 &= \{v_2, v_3, v_6\} \end{aligned}$$

for $(G^1, w^1) = (G^2{:}B, w^2{:}B)$. From Lemmas 2.1 and 2.2, we can easily construct an optimal biclique cover and an optimal solution for (G^2, w^2) as follows:

$$\begin{aligned} \mathcal{C}^2 &= \{[\{v_1\}, \{v_2\}],\ [\{v_2\}, \{v_3\}],\ [\{v_5\}, \{v_3\}],\ [\{v_4, v_5, v_6\}, \emptyset],\ [\emptyset, \{v_5\}],\ [\{v_6\}, \emptyset]\}, \\ y(C) &= (2, 3, 1, 3, 2, 1), \\ X^2 &= \{v_6\}. \end{aligned}$$

These have the same weight 4.

6. An Application: An Exact Algorithm for the GSSP

In this section, we extend the branch and bound algorithm of Balas and Yu [1] for the maximum clique problem to the GSSP.

1. Given an instance $(G = (V, E), w)$ of the GSSP, find a maximal triangulated induced subgraph $G[T]$ of G by using the linear time algorithm proposed in [1] which uses a lexicographic breadth-first search described in [12].
2. By using our algorithm, find a maximum weight solution X and a minimum weight biclique cover $(\mathcal{C}, y)$ in $G[T]$.
3. Let $Y = \{v \in V \setminus X \mid$ there is $x \in X$ with $x \overset{+-}{\sim} v\}$. Then $X \cup Y$ is a solution of G because of the transitivity. If there is a vertex $i \notin Y$ such that $w_i > 0$ and $X \cup Y \cup \{i\}$ is a solution of G, add i to Y; and repeat this while such a vertex exists.
4. Partition Y into two parts $Y_{\geq} = \{i \in Y \mid w_i \geq 0\}$ and $Y_{<} = \{i \in Y \mid w_i < 0\}$. For each $i \in Y_{\geq}$, add $[\{i\}, \emptyset]$ to $\mathcal{C}$ and $y([\{i\}, \emptyset]) = w_i$. Then $(\mathcal{C}, y)$ is a biclique cover of $G[T \cup Y_{\geq}]$. Since $X \cup Y_{\geq}$ and $(\mathcal{C}, y)$ have the same weight, they are a maximum weight solution and a minimum weight biclique cover of $G[T \cup Y_{\geq}]$, respectively.

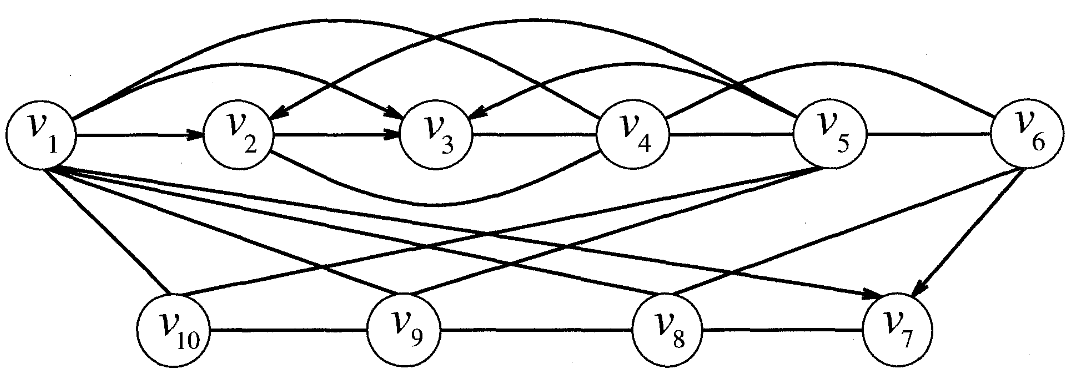


Figure 4: A bidirected graph G^3.

5. For each biclique $[C^+, C^-] \in \mathcal{C}$, if there is a vertex i such that $[C^+ \cup \{i\}, C^-]$ is a biclique of G then replace $[C^+, C^-]$ by $[C^+ \cup \{i\}, C^-]$ and $y([C^+ \cup \{i\}, C^-]) = y([C^+, C^-])$; and repeat this extension while such a vertex exists. Let $U = \{i \in V \setminus (T \cup Y_{\geq}) \mid d(i) = \sum_{C^+ \ni i} y(C) - \sum_{C^- \ni i} y(C) - w_i \geq 0\}$. For each $i \in U$, add $[\emptyset, \{i\}]$ to $\mathcal{C}$ and $y([\emptyset, \{i\}]) = d(i)$. Then $(\mathcal{C}, y)$ is a biclique cover of $G[T \cup Y_{\geq} \cup U]$. Furthermore, $X \cup Y_{\geq}$ is a maximum weight solution of $G[T \cup Y_{\geq} \cup U]$, because they have the same weight.
6. For disjoint subsets $I = \{i_1, \cdots, i_\ell\}$ and $O = \{o_1, \cdots, o_m\}$ of V, let $(G, w; I, O)$ denote the problem of finding a maximum weight solution subject to $I \subseteq X$ and $X \cap O = \emptyset$. Obviously, we can find a maximum weight solution of (G, w) by solving $(\ell+m+1)$ subproblems: $(G, w; \emptyset, \{i_1\})$, $(G, w; \{i_1\}, \{i_2\})$, $\cdots$, $(G, w; \{i_1, \cdots, i_{\ell-1}\}, \{i_\ell\})$, $(G, w; I \cup \{o_1\}, \emptyset)$, $(G, w; I \cup \{o_2\}, \{o_1\})$, $\cdots$, $(G, w; I \cup \{o_m\}, \{o_1, \cdots, o_{m-1}\})$ and $(G, w; I, O)$. If $I = Y_< \setminus U$ and $O = V \setminus (T \cup Y \cup U)$, $X \cup Y$ is an optimal solution for $(G, w; I, O)$. Hence we can find an optimal solution for (G, w) by solving another $(\ell+m)$ subproblems by recursively using the above steps.

The merit of the branch and bound algorithm is that Steps from 1 through 5 can be executed very fast. In order to decrease the number of subproblems in Step 6, Steps 3, 4 and 5 extend a current solution and biclique cover. We remark that several subproblems in Step 6 may have no solution. For solving each (G, w, I, O), we should reduce the problem by deleting vertices i whose x_i is fixed to 0 or 1.

Example 6.1. Let us consider a bidirected graph G^3 in Figure 4. Suppose that $w^3 = (2, 1, -4, 3, 2, 4, -5, 3, 2, 1)$ describes weights on the vertices $\{v_1, \cdots, v_{10}\}$ of G^3.

1. Since G^3 is not triangulated, we must find a maximal triangulated subgraph of G^3. Here we assume that $G^3[T]$ with $T = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ is found.
2. Since $(G^3[T], w^3_T)$ is equivalent to the instance (G^2, w^2) in Example 5.4, we obtain the following biclique cover $\mathcal{C}$ and solution X:

$$\begin{aligned} \mathcal{C} &= \{[\{v_1\}, \{v_2\}],\ [\{v_2\}, \{v_3\}],\ [\{v_5\}, \{v_3\}],\ [\{v_4, v_5, v_6\}, \emptyset],\ [\emptyset, \{v_5\}],\ [\{v_6\}, \emptyset]\}, \\ y(C) &= (2, 3, 1, 3, 2, 1), \\ X &= \{v_6\}. \end{aligned}$$

3. Vertex v_7 is induced by v_6. In addition, v_9 of weight 2 can be added to the current solution. Then $Y = \{v_7, v_9\}$.

4. Add $[\{v_9\}, \emptyset]$ to $\mathcal{C}$ and $y([\{v_9\}, \emptyset]) = 2$.
5. We can extend two bicliques $[\{v_6\}, \emptyset]$ and $[\{v_9\}, \emptyset]$ to $[\{v_6, v_8\}, \emptyset]$ and $[\{v_8, v_9\}, \emptyset]$. Then $\{v_8\}$ can be covered by such bicliques, and hence $U = \{v_8\}$.
6. $X \cup Y = \{v_6, v_7, v_9\}$ is an optimal solution of weight 1 for $(G^3, w^3; I{=}\{v_7\}, O{=}\{v_{10}\})$. It is enough to solve two subproblems $(G^3, w^3; \emptyset, \{v_7\})$ and $(G^3, w^3; \{v_7, v_{10}\}, \emptyset)$. In the first case, we can reduce G^3 to $G^3[\{v_2, v_3, v_4, v_5, v_8, v_9, v_{10}\}]$ because x_{v_1}, x_{v_6} must be 0. Since the bidirected graph is triangulated, in the same way as above, we obtain an optimal solution $\{v_4, v_8, v_{10}\}$ of weight 7. In the second case, we can reduce G^3 to $G^3[\{v_2, v_3, v_4, v_6\}]$, and obtain an optimal solution $\{v_6, v_7, v_{10}\}$ of weight 0. Hence $\{v_4, v_8, v_{10}\}$ is an optimal solution for (G^3, w^3).

Acknowledgements

The authors wish to thank referees for their helpful comments.

References

[1] E. Balas and C. S. Yu: Finding a maximum clique in an arbitrary graph. *SIAM Journal on Computing*, **15** (1986) 1054–1068.

[2] E. Boros and O. Čepek: On perfect $0, \pm 1$ matrices. *Discrete Mathematics*, **165/166** (1997) 81–100.

[3] M. Conforti, G. Cornuéjols and C. De Francesco: Perfect $0, \pm 1$ matrices. *Linear Algebra and its Applications*, **253** (1997) 299–309.

[4] A. Frank: Some polynomial algorithms for certain graphs and hypergraphs. In C. St. J. A. Nash-Williams and J. Sheehan (eds.): *Proceedings of the 5th British Combinatorial Conference* (Congressus Numerantium No. XV, Utilitas Mathematica, Winninpeg, 1975) 211–226.

[5] F. Gavril: Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. *SIAM Journal on Computing*, **1** (1972) 180–187.

[6] M. Grötschel, L. Lovász and A. Schrijver: The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, **1** (1981) 169–197.

[7] B. Guenin: Perfect and ideal $0, \pm 1$ matrices. *Mathematics of Operations Research*, **23** (1998) 322–338.

[8] Y. T. Ikebe and A. Tamura: Perfect bidirected graphs. *Report CSIM96-2, Department of Computer Science and Information Mathematics* (University of Electro-Communications, Tokyo, 1996).

[9] E. L. Johnson and M. W. Padberg: Degree-two inequalities, clique facets, and biperfect graphs. *Annals of Discrete Mathematics*, **16** (1982) 169–187.

[10] G. J. Minty: On maximal independent sets of vertices in claw-free graphs. *Journal of Combinatorial Theory. Series B*, **28** (1980) 284–304.

[11] D. Nakamura and A. Tamura: The generalized stable set problem for claw-free bidirected graphs. In R. E. Bixby, E. A. Boyd and R. Z. Ríos-Mercado (eds.): *Integer Programming and Combinatorial Optimization* (Lecture Notes in Computer Science 1412, Springer, 1998) 69–83.

[12] D. J. Rose, R. E. Tarjan and G. S. Lueker: Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on Computing*, **5** (1976) 266–283.

[13] E. C. Sewell: Binary integer programs with two variables per inequality. *Mathematical Programming*, **75** (1996) 467–476.

[14] A. Tamura: The generalized stable set problem for perfect bidirected graphs. *Journal of the Operations Research Society of Japan*, **40** (1997) 401–414.

[15] A. Tamura: Perfect $(0, \pm 1)$-matrices and perfect bidirected graphs. To appear in *Theoretical Computer Science A*.

Daishin NAKAMURA
Department of Computer Science
University of Electro-Communications
Chofu-shi, Tokyo, 182-8585, Japan
E-mail: `daishin@im.uec.ac.jp`