アルゴリズム 61−3
（1998. 3. 20）

クローフリーグラフに対する最大重み安定集合問題とその一般化

中村大真　田村明久

電気通信大学情報工学科

概要: 一般化安定集合問題は，無向グラフの最大重み安定集合問題の双向グラフへの拡張である．後者の問題は，クローフリー無向グラフに限定すれば多項式時間で解けることが知られている．本研究では一般化安定集合問題がクローフリー双向グラフに限定すれば多項式時間で解けることを示す．

On the maximum weight stable set problem and its extension for claw-free graphs

Daishin Nakamura and Akihisa Tamura

Department of Computer Science and Information Mathematics,
University of Electro-Communications

Abstruct: The generalized stable set problem is an extension of the maximum weight stable set problem for undirected graphs to bidirected graphs. It is known that the latter problem is polynomially solvable for claw-free undirected graphs. This paper show that the generalized stable set problem is also polynomially solvable for claw-free bidirected graphs.

1 Introduction

Let $G = (V, E)$ be an undirected graph. A subset S of V is called a *stable set* if any two elements of S are nonadjacent. Given a weight vector $w \in \Re^V$, a *maximum weight stable set* is a stable set S maximizing $w(S) = \sum_{i \in S} w_i$. The problem of finding a maximum weight stable set is called the *maximum weight stable set problem* (MWSSP). It is well known that the problem can be formulated as the following integer programming problem:

$$\text{[MWSSP]} \quad \text{maximize} \quad w \cdot x \quad \text{subject to} \quad \begin{array}{ll} x_i + x_j \leq 1 & \text{for } (i,j) \in E, \\ x_i \in \{0,1\} & \text{for } i \in V. \end{array}$$

In this paper, we consider the problem generalized as follows: for a given finite set V and for given $P, N, I \subseteq V \times V$,

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

Here we call this problem the *generalized stable set problem* (GSSP). 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. 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, we may consider the GSSP. We call a 0–1-vector satisfying the inequality system arising from a bidirected graph G a *solution* of G. We 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.

Since several distinct bidirected graphs may have the same set of solutions, we deal with some kind of 'standard' bidirected graphs. 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 loop and if it has at most one edge for each pair of distinct vertices. Johnson and Padberg [1] showed that any transitive bidirected graph can be reduced to simple one without essentially changing the set of solutions, or determined to have no solution. We note 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.

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 be *claw-free* if it is simple and transitive and if its underlying graph is claw-free (i.e., does not contain a vertex-induced subgraph which is isomorphic to the complete bipartite graph $K_{1,3}$).

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, the MWSSP is polynomially solvable. For example, Minty [2] proposed a polynomial time algorithm for the MWSSP for claw-free undirected graphs. On the other hand, there are several polynomial transformations from the GSSP to the MWSSP (see [3, 4]). Unfortunately, we cannot easily derive the polynomial solvability of the GSSP for claw-free bidirected graphs by using these transformations, because these do not preserve claw-freeness. Our aim in this paper is to verify that the GSSP for claw-free bidirected graphs is polynomially solvable.

2 Canonical bidirected graphs and their solutions

In this section, we will give several definitions and discuss basic properties of solutions of bidirected graphs. Let $G = (V, E)$ be a simple and transitive bidirected graph and w be a weight vector on V. For any subset $U \subseteq V$, we call the transformation which reverse the signs of the u side of all edges incident to each $u \in U$ the *reflection* of G at U, and we denote it by $G{:}U$. Obviously, reflection preserves simpleness 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 $X \Delta Y$ denote the symmetric difference of X and Y.

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

We say that a vertex is *positive* (or *negative*) if all edges incident 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 say that a bidirected graph is *canonical* if it is simple, transitive and pure and it has no negative vertex. For any instance (G, w) of the GSSP, we can transform it to equivalent one whose bidirected graph is canonical as follows. From the previous section, we can assume that G is simple and transitive. Johnson and Padberg [1] proved that G has at least one solution $U \subseteq V$. 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 is done in polynomial time, we assume that a given bidirected graph of the GSSP is canonical in the sequel.

For any solution X of a canonical bidirected graph G, we partition X into two parts:

$$X_B = \{i \in X \mid N_G^{-+}(i) \cap X = \emptyset\} \quad \text{and} \quad X_I = \{i \in X \mid N_G^{-+}(i) \cap X \neq \emptyset\},$$

where $N_G^{-+}(i)$ denotes the set of vertices adjacent to i by a $(-,+)$-edge incident to i with a minus sign, $N_G^{+-}(i)$ is defined analogously. Here we call X_B a *base* of X. Let

$$\text{ex}(X_B) = X_B \cup \{i \in V \mid i \in N_G^{+-}(x) \text{ for some } x \in X_B\}.$$

If $S \subseteq V$ is a stable set of $\underline{G}$, we say that S is a stable set of G. It is not difficult to show the following lemmas.

Lemma 2.2. *For any solution X of a canonical bidirected graph G, $X = \text{ex}(X_B)$, and hence, $(\text{ex}(X_B))_B = X_B$.*

Lemma 2.3. *For any solution X of a canonical bidirected graph G, its base X_B is a stable set of G.*

Lemma 2.4. *For any stable set S of a canonical bidirected graph G, $\text{ex}(S)$ is a solution of G.*

Thus there is a one-to-one correspondence between the solutions and the stable sets of G.

For any subset U of V, let $G[U]$ denote the subgraph induced by U. We call $H \subseteq V$ a connected component of G if H induces a connected component of $\underline{G}$.

Lemma 2.5. *Let X and Y be solutions of a canonical bidirected graph G. For any connected component H of $G[X_B \triangle Y_B]$, $X_B \triangle H$ and $Y_B \triangle H$ are bases of certain solutions of G.*

Let X be a specified solution of G. For any solution Y of G, let $H_1, \ldots, H_\ell$ be the connected components of $G[X_B \triangle Y_B]$. We define the weight of H_i, denoted by $\delta^X(H_i)$ or simply $\delta(H_i)$, by

$$\delta^X(H_i) = w(\text{ex}(X_B \triangle H_i)) - w(X).$$

We remark that the equation $w(Y) - w(X) = \sum \delta(H_i)$ may not hold because there may exist a vertex v such that $N_G^{-+}(v)$ contains several vertices of distinct connected components, that is, w_v may be doubly counted. In order to avoid this obstacle, we require some additional conditions.

Lemma 2.6. *For any solution X of G, there exists $U \subseteq V$ such that $G' = G{:}U$ and $X' = X \triangle U$ satisfy*

(a) *G' is canonical,*

(b) *X' is a stable set of G', i.e., $X' = (X')_B$,*

(c) *for each mixed vertex $v \notin X'$, there is a vertex $u \in X'$ adjacent to v.*

We note that a subset U having the conditions of Lemma 2.6 can be found in polynomial time. The conditions of Lemma 2.6 overcome the above obstacle.

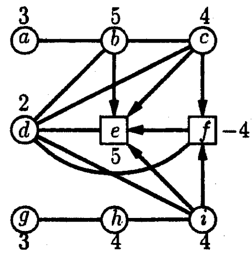
Lemma 2.7. *Let G be a canonical bidirected graph and X be a solution of G satisfying the conditions of Lemma 2.6. For any solution Y, let $H_1, \ldots, H_\ell$ be the connected components of $G[X_B \triangle Y_B]$. Then,*

$$w(Y) - w(X) = \sum_{i=1}^{\ell} \delta^X(H_i) = \sum_{i=1}^{\ell} \{w(\text{ex}(X_B \triangle H_i)) - w(X)\}.$$

3 A basic idea for finding an optimal solution of the GSSP

Given an instance (G, w) of the GSSP, for each $i = 0, 1, \ldots, |V|$, let

$$\mathcal{S}_i = \{X \subseteq V \mid X \text{ is a solution of } G \text{ and has exactly } i \text{ positive vertices }\},$$



i	w^i	solution
0	5	$\{e\}$
1	10	$\{b, e\}$
2	14	$\{b, e, h\}$
3	13	$\{b, e, f, g, i\}$
4	15	$\{a, c, e, f, g, i\}$

Figure 1:

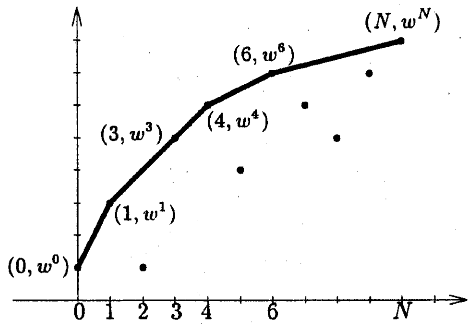


Figure 2:

$$\begin{aligned} w^i &= \max_{X \in \mathcal{S}_i} w(X), \\ \mathcal{S}_i^* &= \{X \in \mathcal{S}_i \mid w(X) = w^i\}. \end{aligned}$$

Suppose that N denotes the smallest number j satisfying $w^j = \max_i w^i$. Minty [2] showed that if a given undirected graph is claw-free, then $w^0 < \cdots < w^N$. More precisely, $(0, w^0), \ldots, (N, w^N)$ lie on an increasing concave curve. Minty's algorithm for solving the MWSSP for claw-free undirected graphs finds an optimal solution by tracing (i, w^i) one by one. However, even if a given bidirected graph is claw-free, this fact does not hold as an example in Figure 1 where $(+,+)$-edges are drawn by lines and $(+,-)$-edges by arrows whose heads mean minus signs. Thus, it seems to be difficult to trace (i, w^i) one by one for the GSSP. We will use a technique of the fractional programming. Let us consider the upper envelope of the convex hull of the set of pairs $(0, w^0), (1, w^1), \ldots, (N, w^N)$ as in Figure 2. We call (i, w^i) a *Pareto-optimal pair* if it lies on the envelope, and their solutions *Pareto-optimal solutions.* Obviously, $(0, w^0)$ and (N, w^N) are always Pareto-optimal. In Figure 2, $(0, w^0)$, $(1, w^1)$, $(3, w^3)$, $(4, w^4)$, $(6, w^6)$ and (N, w^N) are Pareto-optimal.

Let X^i be a Pareto-optimal solution with $X^i \in \mathcal{S}_i$. Suppose that $\mathcal{F}$ is a subset of all the solutions of G such that $X^i \in \mathcal{F}$ and $\mathcal{F}$ is defined independently to the weight vector w. Let us also consider the Pareto-optimal solutions for the restriction on $\mathcal{F}$. Obviously, X^i is also Pareto-optimal in $\mathcal{F}$. We consider the following two problems

$$[\mathrm{MAX}\delta] \quad \max_{Y \in \mathcal{F}} \{\delta(Y) = w(Y) - w(X^i)\}, \qquad \text{and} \qquad [\mathrm{MAX}\rho] \quad \max_{Y \in \mathcal{F}} \left\{\rho(Y) = \frac{\delta(Y)}{\nu(Y)} \mid \delta(Y) > 0\right\},$$

where $\nu(Y)$ denotes the difference of the numbers of all the positive vertices of Y and X^i. We denote $\rho(\cdot)$ and $\delta(\cdot)$ for a weight vector $\bar{w}$ by $\rho_{\bar{w}}(\cdot)$ and $\delta_{\bar{w}}(\cdot)$ explicitly. Suppose that X^i is not optimal in $\mathcal{F}$. Let Y^1 be an optimal solution of the MAXδ for $\bar{w}^0 = w$. We set $r = \rho_{\bar{w}^0}(Y^1)$ and consider the new weight vector $\bar{w}^1$ defined by

$$\bar{w}^1_i = \begin{cases} \bar{w}^0_i - r & \text{if } i \text{ is a positive vertex,} \\ \bar{w}^0_i & \text{otherwise.} \end{cases} \quad (1)$$

Then, $\delta_{\bar{w}^1}(Y^1) = 0$. For any solution $Y \in \mathcal{F}$,

$$\rho_{\bar{w}^1}(Y) = \frac{\delta_{\bar{w}^0}(Y) - r \cdot \nu(Y)}{\nu(Y)} = \rho_{\bar{w}^0}(Y) - r.$$

Thus, X^i is Pareto-optimal in $\mathcal{F}$ for $\bar{w}^1$. We now assume that there is a solution Y^* with $\rho_{\bar{w}^0}(Y^*) > \rho_{\bar{w}^0}(Y^1)$ and $\delta_{\bar{w}^0}(Y^*) > 0$. Then, evidently, $0 < \nu(Y^*) < \nu(Y^1)$. We also have $\delta_{\bar{w}^1}(Y^*) = [\delta_{\bar{w}^0}(Y^*) - r \cdot \nu(Y^*)] = \nu(Y^*)[\rho_{\bar{w}^0}(Y^*) - \rho_{\bar{w}^0}(Y^1)] > 0$. Conversely, if $\delta_{\bar{w}^1}(Y^*) > 0$ then $\rho_{\bar{w}^0}(Y^*) > \rho_{\bar{w}^0}(Y^1)$ and $\delta_{\bar{w}^0}(Y^*) > 0$. Summing up the above discussion, for an optimal solution Y^2 of the MAXδ for $\bar{w}^1$, if $\delta_{\bar{w}^1}(Y^2) = 0$ then Y^1 is an optimal solution of the MAXρ for w; otherwise, by repeating the above process at most $|V|$ times, the MAXρ for w can be solved, because of the fact that $\nu(Y^1) > \nu(Y^2) > \cdots > 0$.

From the above discussion, for each Pareto-optimal solution $X^i \in \mathcal{S}^*_i$, if we can easily define a subset $\mathcal{F}$ such that

(A1) $X^i \in \mathcal{F}$ and $\mathcal{S}^*_j \cap \mathcal{F} \neq \emptyset$ where (j, w^j) is the next Pareto-optimal pair, and
(A2) the MAXδ for $\mathcal{F}$ and for any w can be solved in time polynomial in the number of vertices of G,

then we can either determine X^i is optimal or find a Pareto-optimal solution $X^k \in \mathcal{S}^*_k$ with $i < k \leq N$ in polynomial time. (We may find $(4, w^4)$ from $(1, w^1)$ in Figure 2.) In addition, if $X^0 \in \mathcal{S}^*_0$ can be found in polynomial time, the GSSP for (G, w) can be solved in polynomial time. In fact, this initialization is not so difficult if we can apply the above technique for any vertex-induced subgraph of G, because it is sufficient to solve the GSSP for the bidirected graph obtained from the current one by deleting all the positive vertices, recursively.

Finally we introduce a tool in order to trace Pareto-optimal pairs. Let X^i be a Pareto-optimal solution with $i < N$. Without loss of generality, we assume that X^i and G satisfy the conditions of Lemma 2.6. We say that $H \subseteq V$ is an *alternating set* for X^i if H is connected in G and if $X^i \Delta H$ is a stable set of G. We define the weight $\delta(H)$ of an alternating set H with respect to w by $w(\text{ex}(X^i \Delta H)) - w(X^i)$.

Lemma 3.1. *Let (j, w^j) be the next Pareto-optimal pair of (i, w^i). Then, for any $X^j \in \mathcal{S}^*_j$, there is a connected component H of $G[X^i_B \Delta X^j_B]$ such that $\text{ex}(X^i_B \Delta H)$ is a Pareto-optimal solution with more positive vertices than X^i.*

Lemma 3.1 says that we can trance Pareto-optimal solutions by using alternating sets.

4 Finding a next Pareto-optimal solution

Let G, w and X be a given claw-free bidirected graph, a given weight vector on the vertices and a Pareto-optimal solution with respect to w. Without loss of generality, we assume that G and X satisfy the conditions of Lemma 2.6. In this section, we explain how to find a next Pareto-optimal solution.

We first give several definitions. We call the vertices of X *black* and the other vertices *white.* Any white vertex is adjacent to at most two black vertices, since otherwise G must have a claw. A white vertex is said to be *bounded* if it is adjacent to two black vertices, *free* if it is adjacent to exactly one black vertex and otherwise *super free.* A cycle (or path) is called an *alternating cycle* (or *path*) if white and black vertices appear alternately, and its white vertices form a stable set. An alternating path is called *free* if its endpoints are either black or free or super free. Alternating cycles and free alternating paths are alternating sets, and vice versa in claw-free cases. Thus, Lemma 3.1 guarantees that we deal with only alternating cycles and free alternating paths in order to find a next Pareto-optimal solution. An alternating cycle or a free alternating path is called an *augmenting cycle* or an *augmenting path* respectively if it has a positive weight. For two distinct black vertices x and y, let W denote the set of all the bounded vertices adjacent to both x and y. If W is not empty, W is called a *wing* adjacent to x (and y). A black vertex is called *regular* if it is adjacent to three or more wings, *irregular* if it is adjacent to exactly two wings, and otherwise *useless*. An alternating cycle is said to be *small* if it has at most two regular vertices; otherwise *large*. Here we call $C_1, \ldots, C_k$ a *large augmenting cycle family* if each C_i is a large augmenting cycle and each vertex in C_i is adjacent to no vertex in C_j for $1 \leq i < j \leq k$. From Lemma 2.7, $\delta(C_1 \cup \cdots \cup C_k) = \delta(C_1) + \cdots + \delta(C_k)$ holds.

Our algorithm for finding a next Pareto-optimal solution is described by using the technique discussed in the previous section:

(0) $w^0 \leftarrow w$ and $i \leftarrow 0$;

(1) Find a small augmenting cycle A_{i+1} of the maximum weight for w^i if it exists, otherwise go to (2) ; Construct the new weight w^{i+1} by applying (1), $i \leftarrow i+1$ and repeat (1) ;

(2) Find a large augmenting cycle family A_{i+1} of the maximum weight for w^i if it exists, otherwise go to (3) ;
Construct the new weight w^{i+1} by applying (1), $i \leftarrow i+1$ and repeat (2) ;

(3) Find an augmenting path A_{i+1} of the maximum weight for w^i if it exists, otherwise go to (4) ; Construct the new weight w^{i+1} by applying (1), $i \leftarrow i+1$ and repeat (3) ;

(4) If $i = 0$ then X is optimal, otherwise $\mathrm{ex}(X \triangle A_i)$ is a next Pareto-optimal solution.

Note that in (2) there is no small augmenting cycle since these are eliminated in (1), and that in (3) there is no augmenting cycle since these are eliminated in (1) and (2). These facts are important in the following sense.

Theorem 4.1. *For any weight vector,*

- *a maximum weight small augmenting cycle can be found in polynomial time,*
- *a maximum weight large augmenting cycle family can be found in polynomial time if no small augmenting cycle exists,*
- *a maximum weight augmenting path can be found in polynomial time if no augmenting cycle exists.*

By Lemma 3.1 and Theorem 4.1, our algorithm find a next Pareto-optimal solution in polynomial time. Summing up the above discussions, we obtain our main theorem.

Theorem 4.2. *The GSSP for claw-free bidirected graphs is polynomially solvable.*

In the rest of the section, we briefly explain a proof of Theorem 4.1. Our approach is an extension of Minty's algorithm for undirected claw-free graphs. This, however, does not seem a straightforward extension because we must overcome several problems. A significant problem is how to deal with 'induced weights'. Let A be an alternating cycle or a free alternating path. Then its weight is expressed as

$$\delta(A) = w(A - X) - w(X \cap A) + \sum\{w(v) \mid v \text{ is mixed and } N_G^{-+}(v) \cap (A - X) \neq \emptyset\}.$$

We call the $\sum$ term the *induced weight*, which appears in the bidirected case but not in the undirected case.

We first consider cycles. Let $x_1, \ldots, x_k$ with $k \geq 3$ be distinct black vertices and $W_1, \ldots, W_k, W_{k+1} = W_1$ be wings such that x_i is adjacent to W_i and W_{i+1} for $i = 1, \ldots, k$. Then $(W_1, x_1, W_2, \ldots, W_k, x_k, W_1)$ is called a *cycle of wings*. It is easy to show the following:

Lemma 4.3 ([2]): *Let $(W_1, x_1, W_2, \ldots, W_k, x_k, W_1)$ with $k \geq 3$ be a cycle of wings and $y_i \in W_i$ for $i = 1, \ldots, k$. Then $(y_1, x_1, y_2, \ldots, y_k, x_k, y_{k+1} = y_1)$ is an alternating cycle if and only if y_i is not adjacent to y_{i+1} for $i = 1, \ldots, k$.*

Lemma 4.4. *Let v be a mixed vertex such that $N_G^{-+}(v)$ has a bounded vertex but is not included in a wing. Then there uniquely exists a black vertex x such that [$x = v$ or x is adjacent to v] and all the vertices in $N_G^{-+}(v)$ are adjacent to x.*

Lemma 4.5. *Let $C = (W_1, x_1, W_2, \ldots, W_k, x_k, W_1)$ be a cycle of wings ($k \geq 3$). Then a maximum weight alternating cycle included in C can be found in polynomial time.*

Lemma 4.6. *A maximum weight small augmenting cycle can be found in polynomial time.*

Unfortunately, a maximum weight large augmenting cycle cannot be found in polynomial time in the same way because the number of the cycles of wings having three or more regular vertices cannot be polynomially bounded. Before considering the step (2) in our algorithm, we introduce a useful property relative to wings around regular vertices. For convenience, we will use some notations as below:

- $v_1 \sim v_2$ means that v_1 and v_2 are adjacent, and $v_1 \not\sim v_2$ means v_1 and v_2 are not adjacent.
- $v_1 \overset{+-}{\sim} v_2$ says there is an edge having plus and minus sings at v_1 and v_2 respectively, and $v_1 \overset{+-}{\not\sim} v_2$ is its negation.
- $v_1 \overset{+}{\sim} v_2$ denotes either $v_1 \overset{++}{\sim} v_2$ or $v_1 \overset{+-}{\sim} v_2$, and $v_1 \overset{+}{\not\sim} v_2$ is the negation of $v_1 \overset{+}{\sim} v_2$.
- $v_1 \diamond v_2$ says that v_1 and v_2 are contained in the same wing, and $v_1 \not\diamond v_2$ is its negation.

Lemma 4.7 ([2]): *Given a regular vertex x, let $B(x) = \{v \mid v \sim x \text{ and } v \text{ is bounded}\}$. Then there exists a partition of $B(x)$, namely $[N^1(x), N^2(x)]$, such that for any $v_1, v_2 \in B(x)$ with $v_1 \not\diamond v_2$,*

$$v_1 \sim v_2 \iff [v_1, v_2 \in N^1(v) \text{ or } v_1, v_2 \in N^2(v)].$$

Moreover this partition is uniquely determined, and hence, it can be found in polynomial time.

This is the key lemma of Minty's algorithm. If a large alternating cycle or a free alternating path passes through $v_1 \in N^1(v)$ and a regular vertex v, then it must pass through a vertex v_2 such that $v_2 \in N^2(v)$ and $v_2 \not\diamond v_1$. From this property Minty showed that by constructing a graph called the "Edmonds' graph" and by finding a maximum weight perfect matching of it, a maximum weight augmenting path for any Pareto-optimal stable set can be found in polynomial time. To deal with induced weights, we require an additional property of the partition of vertices adjacent to a regular vertex.

Lemma 4.8. *For a regular vertex x and a vertex v such that $v = x$ or $v \sim x$, we define*

$$N^1(x) \succ_v N^2(x) \overset{\text{def}}{\iff} \exists a \in N^1(x), \exists b \in N^2(x) \text{ such that } a \not\sim b,\ a \overset{+-}{\sim} v \text{ and } b \overset{+-}{\not\sim} v,$$

$$N^2(x) \succ_v N^1(x) \overset{\text{def}}{\iff} \exists c \in N^2(x), \exists d \in N^1(x) \text{ such that } c \not\sim d,\ c \overset{+-}{\sim} v \text{ and } d \overset{+-}{\not\sim} v.$$

Then at most one of $N^1(x) \succ_v N^2(x)$ and $N^2(x) \succ_v N^1(x)$ holds.

We add the induced weight of an alternating cycle or a free alternating path to weights of appropriate vertices in it. We define $\tilde{w} : (V \cup (V \times V)) \to \Re$ by the following procedure: let $\tilde{w} \leftarrow 0$ and for each mixed vertex v,

- if $B^{-+}(v) = \{u \mid u \text{ is bounded}, v \overset{-+}{\sim} u\}$ is empty or included in a wing, $\tilde{w}(u) \leftarrow \tilde{w}(u) + w(v)$ for each $u \in B^{-+}(v)$,
- otherwise there uniquely exists a black vertex x such that $x = v$ or $x \sim v$, from Lemma 4.4,
  - ⋆ if x is regular, then
    - if $N^2(x) \succ_v N^1(x)$, then $\tilde{w}(u) \leftarrow \tilde{w}(u) + w(v)$ for each $u \in B^{-+}(v) \cap N^2(x)$,
    - otherwise $\tilde{w}(u) \leftarrow \tilde{w}(u) + w(v)$ for each $u \in B^{-+}(v) \cap N^1(x)$,
  - ⋆ otherwise x must be irregular, and $\tilde{w}(t, u) \leftarrow \tilde{w}(t, u) + w(v)$ for each pair of vertices $t, u \in B(x)$ such that $t \not\sim u$, $t \not\approx u$ and $[t \overset{+-}{\sim} v$ or $u \overset{+-}{\sim} v]$.

By combining Lemmas 4.7 and 4.8, we can prove the next lemma.

Lemma 4.9. *Let $C = (y_1, x_1, y_2, x_2, \ldots, y_k, x_k, y_{k+1} = y_1)$ be a large alternating cycle with white vertices $y_1, \ldots, y_k$ and black vertices $x_1, \ldots, x_k$. Then $\delta(C) = \sum_{i=1}^{k} w(y_i) - \sum_{i=1}^{k} w(x_i) + \sum_{i=1}^{k} \tilde{w}(y_i) + \sum_{i=1}^{k} \tilde{w}(y_i, y_{i+1})$.*

If there is no small augmenting cycle, by using Lemma 4.9, we can construct the Edmonds' graph $\hat{G}$ such that

- each edge of $\hat{G}$ is colored black or white, and it has a weight $\hat{w}$,
- all the black edges form a perfect matching M of $\hat{G}$,
- if M is a maximum weight perfect matching of $\hat{G}$ then there is no large augmenting cycle family in G,
- if $\hat{w}(M) < \hat{w}(M^*)$ for a maximum weight perfect matching M^* of $\hat{G}$, let $\hat{C}_1, \ldots, \hat{C}_k$ be all the augmenting cycles in $M^* \triangle M$; then $\hat{C}_1, \ldots, \hat{C}_k$ correspond to a maximum weight large augmenting cycle family $C_1, \ldots, C_k$ in G.

Although we omit the details about the Edmonds' graph, we can construct it in polynomial time. Hence the step (2) in our algorithm can be done in polynomial time. Analogously, if there is no augmenting cycle, for any pair of vertices a and b, we can find a maximum weight augmenting path whose endpoints are a and b, if it exists, by constructing the Edmonds' graph and by finding a maximum weight perfect matching in it. Now we can find a maximum weight augmenting path by trying all the pairs of vertices a and b.

References

[1] Johnson, E. L. and Padberg, M. W. (1982), "Degree-two inequalities, clique facets, and biperfect graphs," *Ann. Discrete Math.*, **16**, 169–187.

[2] Minty, G. J. (1980), "On maximal independent sets of vertices in claw-free graphs," *J. Combin. Theory Ser. B*, **28**, 284-304.

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

[4] Tamura, A. (1997), "The generalized stable set problem for perfect bidirected graphs," *J. Oper. Res. Soc. Japan*, **40**, 401–414.