Tractable Feedback Vertex Sets in Restricted Bipartite Graphs*

Wei Jiang[1], Tian Liu[1], and Ke Xu[2]

[1] Key Laboratory of High Confidence Software Technologies, Ministry of Education, Institute of Software, School of Electronic Engineering and Computer Science, Peking University, Beijing 100871, China
lt@pku.edu.cn

[2] National Laboratory of Software Development Environment, Beihang University, Beijing 100191, China
kexu@nlsde.buaa.edu.cn

Abstract. A feedback vertex set (FVS) is a subset of vertices whose removal renders the remaining graph a forest. We show that finding the minimum FVS is tractable in the so-called triad convex bipartite graphs.

1 Introduction

A *feedback vertex set* (FVS), sometimes also called *loop cutset*, in a graph is a subset of vertices whose removal renders the remaining graph a forest. In weighted graphs, the weight of a FVS is the summation of the weight of each vertex in the FVS, and the weight of each vertex is a positive integer. For non-weighted graphs, we just assume that each vertex has a weight one. Finding the minimum FVS (MFVS) even in non-weighted graphs is a classical $\mathcal{NP}$-complete problem. In fact, it was among the twenty-one $\mathcal{NP}$-complete problems in Karp's list [19]. MFVS has applications in deadlock prevention and recovery in operating systems [28], information security [15], VLSI chip design [10], artificial intelligence [1,30], etc.. Many algorithms have been developed for MFVS, which are approximate [3,1,4,5,27,33], randomized [2], parameterized [8,9,6], exact [12], polynomial in restricted graphs [22,24,21,29,14,20], to enumerate and count the number of MFVS [11], and to estimate the size of MFVS [26]. In this paper, we show a tractable result about MFVS in some restricted bipartite graphs. Namely, we show that MFVS is polynomial in restricted bipartite graphs which are called triad convex.

A Bipartite graph $G = (A, B, E)$ is called *tree convex*, if there is a tree $T = (A, F)$, such that for all vertex b in B, the subset $N(b) = \{a \in A|(a, b) \in E\}$ (i.e. the neighborhood of b in A) is a connected subtree in T. When T is a star (i.e. a bipartite complete graph $K_{1,|A|-1}$), G is called *star convex*. When T is a path, G is called *path convex* or just *convex*. When T is a triad (i.e. three paths with a common end), G is called *triad convex*. It was known that finding

* To whom the correspondence should be addressed: Tian Liu (lt@pku.edu.cn) and Ke Xu (kexu@nlsde.buaa.edu.cn).

W. Wang, X. Zhu, and D.-Z. Du (Eds.): COCOA 2011, LNCS 6831, pp. 424–434, 2011.

MFVS in non-weighted bipartite graphs is also $\mathcal{NP}$-hard [31], but tractable in weighted convex bipartite graphs and cocomparable graphs [22], interval graphs [24], circle graphs [14], mesh and butterfly [7,25], hypercube [13], star graph [29], rotator graph [16,20], and shuffle-based interconnection networks [21], etc..

Recently, we generalized the notion of convex bipartite graphs to the notion of tree convex bipartite graphs, and showed that finding MFVS in non-weighted star convex bipartite graphs is also $\mathcal{NP}$-hard [17]. Thus the tractability result of MFVS in convex bipartite graphs in [22] is unlikely extensible to all tree convex bipartite graphs. Here in contrast to [17], we show that finding MFVS in weighted triad convex bipartite graphs is tractable. Therefore, the tractability result in [22] on convex bipartite graphs is at least extensible to triad convex bipartite graphs. In summary, the results in [17] and here have refined the complexity classification of MFVS in [31,22] on various restricted bipartite graphs.

This paper is organized as follows. In Section 2, we recall some necessary definitions and notations along with some known results on FVS and tree convex bipartite graphs. In Section 3, we describe the details of an algorithm for finding MFVS in weighted triad convex bipartite graphs, and also show its correctness. In Section 4, we give an explicit polynomial time bound for the algorithm and remark some possible extension and limit of the algorithm. Finally, we discuss some open problems.

2 Definitions

In this section, we recall some definitions and notations along with some known results to be used in subsequent sections about FVS and convex bipartite graphs as in [22] and about tree convex bipartite graphs as in [17].

A Bipartite graph $G = (A, B, E)$ is called *tree convex*, if there is a tree $T = (A, F)$, such that for all vertex b in B, the subset $N(b) = \{a \in A | (a, b) \in E\}$ (i.e. the neighborhood of b in A) is a connected subtree in T [17]. When T is a triad (i.e. three paths with a common end), G is called *triad convex* [17]. When T is a path, G is called *path convex* or just *convex* [23]. For convex bipartite graphs, we can equivalently assume that there is a linear ordering $<$ on A, such that each neighborhood of some b is an interval of A under $<$, see e.g. [22].

Although the notion of tree convex bipartite graphs was recently introduced by us in [17], from the results in [32] we can know that both convex bipartite graphs and tree convex bipartite graphs are recognizable in linear time, and the associated linear ordering $<$ on A in convex bipartite graphs and the associated tree T on A in tree convex bipartite graphs are both constructible in linear time. Thus we can safely assume that they are parts of the inputs.

Assume a convex bipartite graph $G = (A, B, E)$, $A = \{a_1, \ldots, a_n\}$, $B = \{b_1, \ldots, b_m\}$, with a linear ordering $a_i < a_j$ for $i < j$ on A. For convenience, we add two more vertices a_0 and a_{n+1} to A [22]. The following notations are also from [22]:

- For a vertex v in a graph, $N(v)$ denotes the set of all vertices adjacent to v.
- For every $b \in B$ in G, let $L[b]$ and $R[b]$ denote the smallest and largest vertices, respectively, of A connected to b. Let $B = \{b_1, \ldots, b_m\}$ with $R[b_i] < R[b_j]$ for $1 \leq i < j \leq m$.
- For $a_i \in A$ where $0 \leq i \leq n+1$, let $A_i = \{a_h : a_h \in A \text{ and } 0 \leq h \leq i\}$ and $B_i = \{b : b \in B \text{ and } R[b] < a_i\}$.
- For $a_i, a_j, a_k \in A$ where $0 \leq i < j < k \leq n+1$, define $B_{i,j} = \{b : b \in B \text{ and } a_i < L[b] \leq R[b] < a_j\}$, and $B_{i,j,k} = \{b : a_i < L[b] \text{ and } a_j \leq R[b] < a_k\}$.
- For $a_i, a_j \in A$ where $i < j$, $M(i,j)$ denotes a MCFS in $A_j + B_j$ containing $\{a_i, a_j\}$(i.e. $a_i, a_j \in M(i,j)$), where a_i and a_j are the largest two vertices in $A \cap M(i,j)$.
- For $a_i, a_j, a_k \in A$ where $0 \leq i < j < k \leq n+1$, $M(i,j,k)$ denotes a MCFS in $A_k + B_k$ containing $\{a_i, a_j, a_k\}$, where a_i, a_j, a_k are the largest three vertices in $A \cap M(i,j,k)$.
- For $a_i, a_j \in A$ and $b_k \in B$ where $0 < i < j < n+1$ and $b_k \in N(a_i) \cap N(a_j)$, $M_b(i,j,k)$ denotes a MCFS in $A_j + B_j + \{b_k\}$ containing $\{a_i, a_j, b_k\}$, where a_i and a_j are the largest two vertices in $A \cap M_b(i,j,k)$.

In [22], it was shown that all the $M(i,j)$, $M(i,j,k)$ and $M_b(i,j,k)$ can be computed in $O(|A|^3 + |A|^2|E|)$ time. A convex bipartite graph with the added vertices a_0 and a_{n+1} is shown in figure 1.

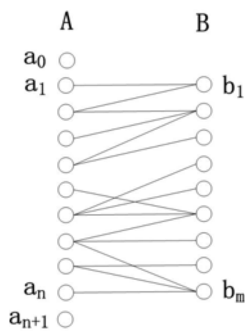

Fig. 1. A convex bipartite graph

Assume a triad convex bipartite graph $G = (A, B, E)$ with a triad $T = (A, F)$ defined on A, such that every $N(b)$ is a subtree of T for $b \in B$ [17]. To be specific, let $A = \{a_0\} \cup A_1 \cup A_2 \cup A_3$, such that for $1 \leq i \leq 3$, $A_i = \{a_{i,1}, \ldots, a_{i,n_i}\}$ and $a_0 \rightarrow a_{i,1} \rightarrow \ldots \rightarrow a_{i,n_i}$ are three paths with a common end a_0. The vertices of a triad convex bipartite graph (without edges) are shown in figure 2.

We can assume that the graphs are weighted with positive integers on their vertices. For non-weighted graphs, these weights are assumed to be one. Recall that removing a FVS renders the remaining graph a forest, and the weight of a FVS is the sum of weights of vertices in the FVS. Thus finding a minimum FVS

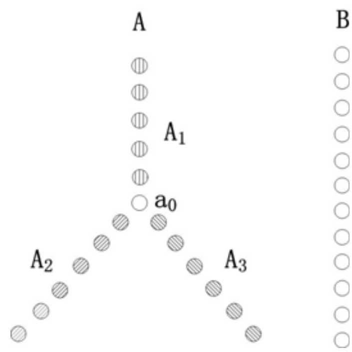

Fig. 2. A triad convex bipartite graph (without edges)

(MFVS) in a graph is equivalent to finding a maximum cycle-free set (MCFS) in the same graph [22]. Below we will give an algorithm to find a MCFS instead of a MFVS in weighted triad convex bipartite graphs as in [22].

3 The Algorithm

In this section, we will describe an algorithm for MFVS in weighted triad convex bipartite graphs. This algorithm is a natural extension of the known algorithm in [22]. Along with the description, we will also show the correctness of the algorithm.

Let U be the MCFS of G, we will consider the following two cases: $a_0 \notin U$ and $a_0 \in U$.

Case 1: $a_0 \notin U$.
Let $a_{1,i}, a_{2,j}, a_{3,k}$ be the nearest vertices to a_0 in U on each path A_1, A_2, A_3 respectively. Some of them may not exist at all, that is, the corresponding whole path may not intersect with U. we can divide A into four parts S_0, S_1, S_2, S_3 as follows:

$S_0 = \{a_0, a_{1,1}, \ldots, a_{1,i-1}, a_{2,1}, \ldots, a_{2,j-1}, a_{3,1}, \ldots, a_{3,k-1}\}$,
$S_1 = \{a_{1,i}, \ldots, a_{1,n_1}\}$,
$S_2 = \{a_{2,j}, \ldots, a_{2,n_2}\}$,
$S_3 = \{a_{3,k}, \ldots, a_{3,n_3}\}$.

An example is shown in figure 3. Note that if some vertices in $\{a_{1,i}, a_{2,j}, a_{3,k}\}$ do not exist, then the corresponding sets in $\{S_1, S_2, S_3\}$ will be empty, however this will cause no harms to our algorithm but only helps.

We can put each $v \in B$ into one of the following six sets $B_1, \ldots, B_6$ according to the situation of v's neighborhood:

$B_1 = \{v \mid a_{1,i}, a_{2,j} \in N(v), a_{3,k} \notin N(v)\}$.
$B_2 = \{v \mid a_{1,i}, a_{3,k} \in N(v), a_{2,j} \notin N(v)\}$.
$B_3 = \{v \mid a_{2,j}, a_{3,k} \in N(v), a_{1,i} \notin N(v)\}$.

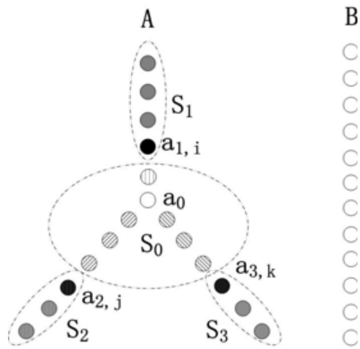

Fig. 3. Divide A into four parts

$B_4 = \{v \mid a_{1,i}, a_{2,j}, a_{3,k} \in N(v)\}$.
$B_5 = \{v \mid a_{1,i}, a_{2,j}, a_{3,k} \notin N(v), N(v) \subseteq S_0\}$.
$B_6 = \{v \mid |\{a_{1,i}, a_{2,j}, a_{3,k}\} \cap N(v)| \leq 1$, and $N(v) \cap (S_1 \cup S_2 \cup S_3) \neq \emptyset\}$.

The following observations are easily seen.

(1) At most one vertex from B_1 is selected into U (similarly for B_2,B_3,B_4). Otherwise, if both $b_p, b_q \in U$ are from B_1, then U will contain a cycle $b_p \to a_{1,i} \to b_q \to a_{2,j} \to b_p$. Let b_1, b_2, b_3, b_4 be the vertices possibly selected into U from B_1, B_2, B_3, B_4 respectively. Some of them may not exist, since they are not selected into U at all, see below.

(2) At most two vertices in $\{b_1, b_2, b_3\}$ or at most one vertex in $\{b_4\}$ are available in any MCFS, since $\{b_1, b_2, b_3\}$ will form a cycle $b_1 \to a_{1,i} \to b_2 \to a_{3,k} \to b_3 \to a_{2,j} \to b_1$, and for example $\{b_1, b_4\}$ will form a cycle $b_1 \to a_{1,i} \to b_4 \to a_{2,j} \to b_1$.

(3) All the vertices from B_5 should be in U, since obviously they are not contained in any cycle.

(4) All the edges between B_6 and S_0 are removable, since they do not change the result of MCFS.

Obviously, the selection of $a_{1,i}$,$a_{2,j}$,$a_{3,k}$ is done in at most $O(|A|^3)$ time. According to the above observations (1) and (2), we can enumerate the vertices b_1, b_2, b_3, b_4 in $O(|B|^2)$ time.

After selecting $a_{1,i}$,$a_{2,j}$,$a_{3,k}$ and b_1, b_2, b_3, b_4 as above, by the above observation (1), we can remove all vertices in $B_i \setminus \{b_i\}$ from G. After the removing, we define four induced subgraphs G_0,G_1,G_2,G_3 of G and find MCFS in G_1,G_2,G_3 as follows (also see figure 4). For any bipartite graph $G = (A, B, E)$ and any subsets $C \subseteq A$ and $D \subseteq B$, denote by $G[C, D]$ the subgraph of G induced by C and D, which is the bipartite graph $(C, D, E \cap (C\&D))$. Here $C\&D = \{(c, d)|c \in C$ and $d \in D\}$ is the set of all unordered pairs of vertices in C and D. Let $N(C) = \{b|b \in B$ and b has a neighbor $c \in C\}$. Then the graphs G_0, G_1, G_2, G_3 are defined as $G[S_0, N(S_0)], G[S_1, N(S_1)], G[S_2, N(S_2)], G[S_3, N(S_3)]$ respectively. An example of the vertex sets of subgraphs G_0,G_1,G_2,G_3 is shown in figure 4.

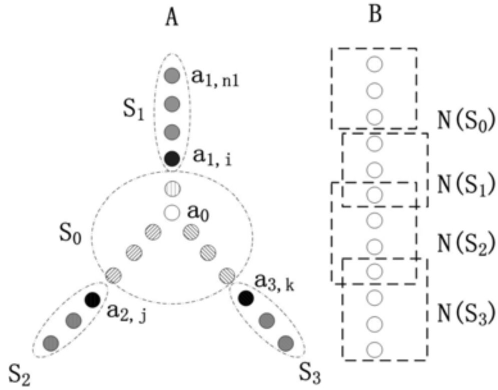

Fig. 4. The vertex sets (without edges) of subgraphs G_0, G_1, G_2, G_3

Let U_1,U_2,U_3 be MCFS of G_1,G_2,G_3 respectively. It is obvious that $U = U_1 \cup U_2 \cup U_3 \cup B_5$. Let $X_1 = \{v|v \in \{b_1, b_2, b_4\}, v$ is actually selected into $U\}$. By above observation (2), we know that $|X_1| \leq 2$. Below we will show how to find the MCFS U_1 in G_1 conditioned on X_1 (we can find U_2, U_3 in G_2, G_3 similarly).

Assume that $|S_1| = n'_1$. For convenience, we add two isolated vertices $a'_0, a'_{n'_1+1}$ into S_1 and make an ordering on $N(S_1)$ as in [22]. For the purposes to deal with U_2, U_3 in a similar way to U_1 and to use the algorithm in [22], we rename the vertices in S_1 as $a'_0, a'_1, \cdots, a'_{n'_1+1}$ and rename $S_1, N(S_1), G_1, U_1, X_1, n'_1$ as A, B, H, U, X, n respectively. Also for convenience, we will omit the primes in a'_i's and b'_j's, and use a_i's and b_j's instead in the following lemma. This renaming procedure is partially depicted in figure 5.

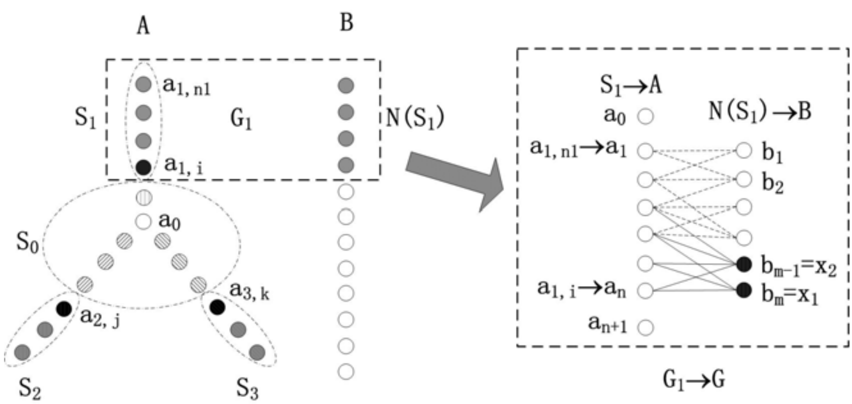

Fig. 5. Renaming vertices and finding U_1 in G_1

Lemma 1. *The MCFS U of H is computed in the following three cases accordingly:*

1. *If $X = \emptyset$, $U = M(n, n+1) \setminus \{a_0, a_{n+1}\}$.*
2. *If $X = \{x_1\}$, U is one of the following candidates, whichever has maximum weight:*
   (a) $M_b(i, n, x_1) \cup B_{i,n,n+1} \setminus \{a_0\}$, for some $i, a_i \in N(x_1)$.
   (b) $M(i, n, n+1) \setminus \{a_0\}$, for some $i, a_i \in A \setminus N(x_1)$.
3. *If $X = \{x_1, x_2\}$ and $|N(x_2) \cap A| \geq |N(x_1) \cap A|$, U is one of the following candidates, whichever has maximum weight:*
   (a) $M_b(i, n, x_2) \cup B_{i,n,n+1} \setminus \{a_0\}$, for some $i, a_i \in (N(x_2) \setminus N(x_1))$.
   (b) $M(i, n, n+1) \setminus \{a_0\}$, for some $i, a_i \in A \setminus N(x_2)$.

Proof. Consider the following three cases.

1. $X = \emptyset$.
   U_1 is obviously $M(n, n+1) \setminus \{a_0, a_{n+1}\}$ according to the definition of $M(i, j)$.
2. $X = \{x_1\}$.
   If the largest three vertices is a_i, a_n, a_{n+1}, both $M_b(i, n, x_1) \cup B_{i,n,n+1} \setminus \{a_0\}$ (when $a_i \in N(x_1)$) and $M(i, n, n+1) \setminus \{a_0\}$ (when $a_i \in A \setminus N(x_1)$) are cycle-free, and one of them is the MCFS of U according to the definitions of $M(i, j, k)$, $M_b(i, j, k)$ and $B_{i,j,k}$.
3. $X = \{x_1, x_2\}$ and $|N(x_2) \cap A| \geq |N(x_1) \cap A|$.
   If the largest three vertices is a_i, a_n, a_{n+1}, $a_i \notin |N(x_1) \cap A|$, otherwise U contains a cycle $x_1 \to a_i \to x_2 \to a_n \to x_1$. Both $M_b(i, n, x_2) \cup B_{i,n,n+1} \setminus \{a_0\}$ (when $a_i \in N(x_2) \setminus N(x_1)$) and $M(i, n, n+1) - \{a_0\}$ (when $a_i \in A \setminus N(x_2)$) are cycle-free, and one of them is the MCFS of U_1 according to the definitions of $M(i, j, k)$, $M_b(i, j, k)$ and $B_{i,j,k}$.

This finishes the proof. □

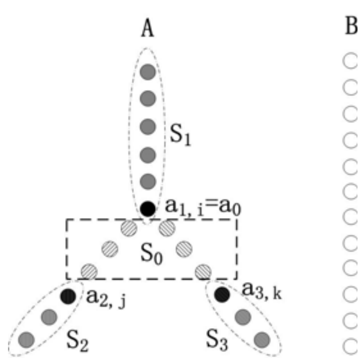

Fig. 6. Set S of Case 2

Case 2: $a_0 \in U$.
Similar to Case 1, let $a_{1,i} = a_0$ and $a_{2,j}, a_{3,k}$ be the nearest vertices to a_0 in U on each path A_2, A_3 respectively. Some of $a_{2,j}, a_{3,k}$ may not exist at all, that is, the corresponding whole path may not intersect with U. We can also divide A into four parts S_0, S_1, S_2, S_3 as follows:

$S_0 = \{a_{2,1}, \ldots, a_{2,j-1}, a_{3,1}, \ldots, a_{3,k-1}\}$,
$S_1 = \{a_0, a_{1,1}, \ldots, a_{1,n_1}\}$,
$S_2 = \{a_{2,j}, \ldots, a_{2,n_2}\}$,
$S_3 = \{a_{3,k}, \ldots, a_{3,n_3}\}$.

An example is shown in figure 6. Also note that some of S_i's maybe empty.

We can put each $v \in B$ into one of the following five sets $B_1, \ldots, B_5$ according to the situation of v's neighborhood:

$B_1 = \{v \mid a_0, a_{2,j} \in N(v), a_{3,k} \notin N(v)\}$.
$B_2 = \{v \mid a_0, a_{3,k} \in N(v), a_{2,j} \notin N(v)\}$.
$B_3 = \{v \mid a_0, a_{1,i}, a_{2,j} \in N(v)\}$.
$B_4 = \{v \mid a_{1,i}, a_{2,j}, a_{3,k} \notin N(v), N(v) \subseteq S_0\}$.
$B_5 = \{v \mid |\{a_0, a_{2,j}, a_{3,k}\} \cap N(v)| \leq 1, \text{ and } N(v) \cap (S_1 \cup S_2 \cup S_3) \neq \emptyset\}$

Then we will immediately get the similar observations and lemmas as in Case 1, so we omit the details here.

The main steps of the above algorithm are summarized as follows.

Input: A triad convex bipartite graph $G = (A, B, E)$ with triad $A = \{a_0\} \cup A_1 \cup A_2 \cup A_3$
Output: A MCFS U of G

1: $U = \emptyset$.
2: Enumerate $a_{1,i} \in A_1 \cup \{a_0\}$, $a_{2,j} \in A_2$ and $a_{3,k} \in A_3$.
3: Assume that $a_{1,i}$, $a_{2,j}$ and $a_{3,k}$ are the nearest vertices to a_0 in U on paths $\{a_0\} \cup A_1, A_2, A_3$ respectively. Divide A into S_0, S_1, S_2, S_3, divide B into $B_1, \ldots, B_6$ (when $a_{1,i} \neq a_0$) or $B_1, \ldots, B_5$ (when $a_{1,i} = a_0$) accordingly.
4: Enumerate $b_1 \in B_1, b_2 \in B_2, b_3 \in B_3, b_4 \in B_4$ (when $a_{1,i} \neq a_0$) or $b_1 \in B_1, b_2 \in B_2, b_3 \in B_3$ (when $a_{1,i} = a_0$).
5: Select at most two from $\{b_1, b_2, b_3\}$ or at most one from $\{b_4\}$ into U, and remove all other vertices in $B_1, \ldots, B_4$ (when $a_{1,i} \neq a_0$) or
select at most two from $\{b_1, b_2\}$ or at most one from $\{b_3\}$ into U, and remove all other vertices in $B_1, \ldots, B_3$ (when $a_{1,i} = a_0$).
6: Put B_5 into U and remove all edges between S_0 and B_6 (when $a_{1,i} \neq a_0$) or
put B_4 into U and remove all edges between S_0 and B_5 (when $a_{1,i} = a_0$).
7: Define subgraphs G_0, G_1, G_2, G_3 accordingly and find the MCFS U_1, U_2, U_3 in G_1, G_2, G_3 respectively. Update U by $U_1 \cup U_2 \cup U_3 \cup B_5$ (when $a_{1,i} \neq a_0$) or
by $U_1 \cup U_2 \cup U_3 \cup B_4$ (when $a_{1,i} = a_0$).
8: If the enumeration in step 4 is unfinished then goto step 4, else if the enumeration in step 2 is unfinished then goto step 2.

Algorithm 1. Finding MFVS in triad convex bipartite graphs

4 The Analysis

In this section, we give an explicit polynomial time bound for the above algorithm, and remark some possible extension and limit of the algorithm here.

Theorem 1. *A MCFS in a triad convex bipartite graphs is found in* $O(|A|^3|B|^2 (|A|^3 + |A|^2|E|))$ *time.*

Proof. Enumerate $a_{1,i}, a_{2,j}, a_{3,k}$ is done in $O(|A|^3)$ and enumerate b_1, b_2, b_3, b_4 is done in $O(|B|^2)$. So the total enumeration time is $O(|A|^3|B|^2)$. According to [22], all $M(i,j), M(i,j,k)$ and $M_b(i,j,k)$ are computed in $O(|A|^3+|A|^2|E|)$ time. Therefore, a MCFS in a triad convex bipartite graphs is found in $O(|A|^3|B|^2 (|A|^3 + |A|^2|E|))$ time. □

Remark 1. A triad is a tree which is three paths with a common end. We can also consider any k paths with a common end, where k is a constant. Although we do not have a name for this kind of trees, it is not hard to see that, using a similar algorithm as above, we can find MCFS in these tree convex bipartite graphs in $O(|A|^k|B|^{k-1}(|A|^3+|A|^2|E|))$ time, when the associated tree is k paths with a common end for constant k.

Remark 2. Recently we show that the sum of degrees larger than two in the associated tree T of tree convex bipartite graphs is an important quantity. Denote this quantity as d. When d is bounded, using a similar algorithm as here, the MFVS in these tree convex bipartite graphs is solvable in $O(|A|^d|B|^{d-1}(|A|^3+|A|^2|E|))$ time. When the number of vertices of degrees larger than two is unbounded but each degrees are bounded (thus d is also unbounded), the MFVS in these tree convex bipartite graphs is still $\mathcal{NP}$-hard [18].

5 Open Problem

It is unknown whether there are better algorithms than the algorithm presented here for MFVS on restricted bipartite graphs. Also, a complete classification of the complexity of finding MFVS on every kind of tree convex bipartite graphs is unknown. What we can say is that even for bounded degrees tree T, MFVS is still $\mathcal{NP}$-hard in these tree convex bipartite graphs [18].

Acknowledgments. We thank Professor Kaile Su for his encouragement and support. We also thank the unknown reviewers whose comments are very helpful to improve our presentations. This research was partially supported by the National 973 Program of China (Grant No. 2010CB328103), the National Natural Science Foundation of China (Grant Nos. 60725207 and 60973033), the Project of Knowledge Innovation Program (PKIP) of Chinese Academy of Sciences (Grant No. KJCX2.YW.W10), and was partially done while Tian Liu and Ke Xu were visiting KITPC at the Chinese Academy of Science.

References

1. Bar-Yehuda, R., Geiger, D., Naor, J., Roth, R.: Approximation algorithms for the feedback vertex set problem with applications to constraint satisfaction and Bayesian inference. SIAM J. Comput. 27, 942–959 (1998)
2. Becker, A., Bar-Yehuda, R., Geiger, D.: Randomized Algorithms for the Loop Cutset Problem. J. Artif. Intell. Res. 12, 219–234 (2000)
3. Becker, A., Geiger, D.: Optimization of Pearls method of conditioning and greedy-like approximation algorithms for the vertex feedback set problem. Artificial Intelligence 83, 167–188 (1996)
4. Cai, M., Deng, X., Zang, W.: An approximation algorithm for feedback vertex sets in tournaments. SIAM J. Comput. 30, 1993–2007 (2001)
5. Cai, M., Deng, X., Zang, W.: A min-max theorem on feedback vertex sets. Math. Oper. Res. 27(2), 361–371 (2002)
6. Cao, Y., Chen, J., Liu, Y.: On Feedback Vertex Set: New Measure and New Structures. In: Kaplan, H. (ed.) SWAT 2010. LNCS, vol. 6139, pp. 93–104. Springer, Heidelberg (2010)
7. Caragiannis, I., Kaklamanis, C., Kanellopoulos, P.: New bounds on the size of the feedback vertex set on meshes and butterflies. Inform. Process. Lett. 83(5), 275–280 (2002)
8. Chen, J., Fomin, F., Liu, Y., Lu, S., Villanger, T.: Improved algorithms for feedback vertex set problems. J. Comput. Syst. Sci. 74(7), 1188–1198 (2008)
9. Chen, J., Liu, Y., Lu, S., O'Sullivan, B., Razgon, I.: A fixed-parameter algorithm for the directed feedback vertex set problem. J. ACM 55(5) (2008)
10. Festa, P., Pardalos, P.M., Resende, M.G.C.: Feedback set problems. In: Du, D.-Z., Pardalos, P.M. (eds.) Handbook of Combinatorial Optimization. Handbook of Combinatorial Optimization, Supplement vol. A, pp. 209–259. Kluwer Academic Publishers, Dordrecht (2000)
11. Fomin, F.V., Gaspers, S., Pyatkin, A., Razgon, I.: On the Minimum Feedback Vertex Set Problem: Exact and Enumeration Algorithms. Algorithmica 52(2), 293–307 (2008)
12. Fomin, F.V., Villanger, Y.: Finding Induced Subgraphs via Minimal Triangulations. In: Proc. of STACS, pp. 383–394 (2010)
13. Focardi, R., Luccio, F.L., Peleg, D.: Feedback vertex set in hypercubes. Inform. Process. Lett. 76(1-2), 1–5 (2000)
14. Gavril, F.: Minimum weight feedback vertex sets in circle graphs. Information Processing Letters 107, 1–6 (2008)
15. Gusfield, D.: A graph theoretic approach to statistical data security. SIAM J. Comput. 17(3), 552–571 (1998)
16. Hsu, C.C., Lin, H.R., Chang, H.C., Lin, K.K.: Feedback Vertex Sets in Rotator Graphs. In: Gavrilova, M.L., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganá, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3984, pp. 158–164. Springer, Heidelberg (2006)
17. Jiang, W., Liu, T., Ren, T., Xu, K.: Two Hardness Results on Feedback Vertex Sets. In: Zhu, B. (ed.) FAW-AAIM 2011. LNCS, vol. 6681, pp. 233–243. Springer, Heidelberg (2011)
18. Jiang, W., Liu, T., Xu, K.: Feedback Vertex Sets in Restricted Bipartite Graphs, paper in preparation
19. Karp, R.: Reducibility among combinatorial problems. In: Complexity of Computer Computations, pp. 85–103. Plenum Press, New York (1972)

20. Kuo, C., Hsu, C., Lin, H., Lin, K.: An efficient algorithm for minimum feedback vertex sets in rotator graphs. Information Processing Letters 109, 450–453 (2009)
21. Kralovic, R., Ruzicka, P.: Minimum feedback vertex sets in shufflebased interconnection networks. Inform. Process. Lett. 86(4), 191–196 (2003)
22. Liang, Y.D., Chang, M.S.: Minimum feedback vertex sets in cocomparability graphs and convex bipartite graphs. Acta Informatica 34, 337–346 (1997)
23. Lipski Jr., W., Preparata, F.P.: Efficient algorithms for finding maximum matchings in convex bipartite graphs and related problems. Acta Informatica 15(4), 329–346 (1981)
24. Lu, C., Tang, C.: A linear-time algorithm for the weighted feedback vertex problem on interval graphs. Information Processing Letters 61, 107–111 (1997)
25. Luccio, F.L.: Exact minimum feedback vertex set in meshes and butterflies. Inform. Process. Lett. 66(2), 59–64 (1998)
26. Madelaine, F.R., Stewart, I.A.: Improved upper and lower bounds on the feedback vertex numbers of grids and butterflies. Discrete Math. 308, 4144–4164 (2008)
27. Sasatte, P.: Improved approximation algorithm for the feedback set problem in a bipartite tournament. Operations Research Letters 36, 602–604 (2008)
28. Silberschatz, A., Galvin, P.B., Gagne, G.: Operating Systems Concepts, 6th edn. John Wiley and Sons, Inc., New York (2003)
29. Wang, F.H., Wang, Y.L., Chang, J.M.: Feedback vertex sets in star graphs. Inform. Process. Lett. 89(4), 203–208 (2004)
30. Williams, R., Gomes, C.P., Selman, B.: Backdoors To Typical Case Complexity. In: Proc. of IJCAI, pp. 1173–1178 (2003)
31. Yannakakis, M.: Node-deletion problem on bipartite graphs. SIAM J. Comput. 10, 310–327 (1981)
32. Zhang, Y., Bao, F.S.: A Survey of Tree Convex Sets Test. CoRR abs/0906.0205 (2009)
33. van Zuylen, A.: Linear programming based approximation algorithms for feedback set problems in bipartite tournaments. Theor. Comput. Sci. (in press)