On Feedback Sets in Tournaments

Nisheeth K. Vishnoi *

Abstract

In this report we consider the feedback vertex and arc set problems for tournaments. We first show NP-completeness of feedback vertex set for tournaments. We then give a simple 3-approximation algorithm for feedback vertex and arc sets.

1 Introduction

A *tournament* of size n is a complete graph on n vertices with edges having directions. In this article we follow the following conventions. An edge will be denoted by either (x, y) or $x \rightarrow y$. If $V \subset V(T_n)$, then $T(V)$ will represent the tournament induced by the vertices of V. Given a graph $G = (V, E)$ (directed or undirected), a feedback vertex (respectively arc) set for G is a subset X of $V(G)$ (respectively $E(G)$) if $G \backslash X$ is acyclic. For $v \in V(T_n)$, let $d(v)$ denote the out-degree of v, i.e. the number of vertices u such that (v, u) is in $E(T_n)$.

2 Feedback vertex set for tournaments is NP-Hard

It is well known [1] that the problem of finding a feedback vertex set of size at-most k is NP-complete for directed graphs. By the next result we show that the problem is NP-complete even when the digraphs are restricted to be tournaments.

Problem 1 (TFBVS). *Given a tournament T_n and a number k, does there exist a feedback vertex set of size at most k ?*

Theorem 2. *TFBVS is NP-complete.*

Proof. It is easily seen that TFBVS is in class NP. To show that it is actually NP-complete, we reduce the problem of vertex cover to it. Given a graph G=(V,E),

*College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA, email: nkv@cc.gatech.edu

and a number k, we construct the graph $G' = (V', E')$ from it in the following manner. Let $V = \{v_1, v_2, \ldots, v_n\}$. Define

$$V' = \{v_{11}, v_{12}, v_{13}, \ldots, v_{n1}, v_{n2}, v_{n3}\}.$$

$$E_1 = \cup_{i=1}^{n} \{(v_{i1}, v_{i2}) \cup (v_{i3}, v_{i1}) \cup (v_{i3}, v_{i2})\}.$$

$$E_2 = \cup_{i \text{ } adjacent \text{ } to \text{ } j \text{ } ,i<j} \{(v_{j1}, v_{i1}) \cup \{\cup_{(k,l)\neq(1,1)}(v_{ik}, v_{jl})\}\}.$$

$$E_3 = \cup_{i \text{ } not \text{ } adjacent \text{ } to \text{ } j \text{ } ,i<j} \{\cup_{(k,l)}(v_{ik}, v_{jl})\}.$$

$$E' = E_1 \cup E_2 \cup E_3.$$

Now we will show that there exist a vertex cover of size at most k in G iff there exist a feedback vertex set of size at most k in G'. Clearly a vertex cover of size at most k in G will act as a feedback vertex set of size at most k in G'. So let W' denote a feedback vertex set in G'. Let $W = \{v_i \in V : v_{ij} \in W'\}$. If this set is not a vertex cover in G, then there exist an edge, say (v_i, v_j) $(i < j)$, such that neither v_{ik} nor v_{jl} is in W'. Thus there exist a cycle $\{v_{i1}, v_{i2}, v_{j3}, v_{j1}\}$ in G', a contradiction. Hence we have shown that TFBVS is NP-complete. □

Remark 3. *If we let TFBAS be the same problem for the arc set instead of the vertex set, we do not know whether it is NP-Hard and leave it as an open problem.*

3 Approximation Algorithms for Feedback Sets in Tournaments

The best known approximation algorithm for feedback sets in general directed graphs is $O(\log n \log \log n)$ due to Seymour [3]. In this section we show that if the problem is restricted to tournaments, the approximation factor drops down considerably to 3.

Given a tournament T_n, a 3 cycle of it will be denoted by $a \to b \to c \to a$, with a, b, c being distinct vertices of T_n. We first need a simple lemma which will give us the desired approximation factor.

Lemma 4. *Let T_n be a tournament and C_k $(3 \leq k \leq n)$ be a cycle in T_n. Then there exists a 3 cycle C_3 in T_n such that $V(C_3) \subseteq V(C_k)$.*

Proof. The proof is by induction on k. If $k = 3$, there is nothing to prove. So let C_k have at-least 4 distinct vertices. Let a, b, c be any three vertices in $V(C_k)$ which occur in C_k in that order. Then if $c \to a$, we have the 3 cycle $a \to b \to c \to a$. Otherwise by induction on C_{k-1} which is C_k with b removed and the edge $a \to c$ added, we are done. □

So by the lemma above, the task of selecting vertices (resp. edges) to remove all cycles in T_n reduces to selecting vertices (resp. edges) to remove all 3 cycles in T_n. We formulate the TFBVS problem as the following integer program. Assign a 0/1 variable x_v for each $v \in V(T_n)$, the variable being 1 iff v is picked in the feedback vertex set. Let $\mathbf{C}$ denote the family of 3 cycles in T_n. Each $C \in \mathbf{C}$ is a 3 element subset of vertices of T_n. Now the TFBVS problem can be formulated as follows.

$$\text{minimize} \sum_{v \in V(T_n)} x_v$$

$$\text{subject to} \sum_{C: v \in C} x_v \geq 1, C \in \mathbf{C}$$

$$x_v \in \{0,1\}, v \in V(T_n)$$

Notice here that given a tournament T_n with out-degrees of vertices denoted by $d(v_1), \ldots, d(v_n)$, the number of 3 cycles in it are exactly $\binom{n}{3} - \sum_{i=1}^{n} \binom{d(v_i)}{2}$, which is at-most $O(n^3)$. Hence the LP has size polynomial in n.
Let the optimal to this linear program be denoted by c^{OPT}. The LP relaxation of this problem is obtained by letting $0 \leq x_v \leq 1$ for all $v \in V(T_n)$. Notice here that the upper bound of 1 is redundant as it is a minimization problem and so can be dropped while solving the relaxed problem.
Now we find an optimal solution to the LP relaxation. Let X be the set of vertices whose corresponding variables are at-least $1/3$. We set these variables to 1 and others to 0 to get a feasible solution to the integer program. Moreover the cost of the solution obtained is at-most $3 \cdot c^*$ which is at-most $3 \cdot c^{OPT}$. Hence we have a 3 approximation to TFBVS. For further details on the technique used above the reader is referred to [2].

Remark 5. *Essentially the same idea gives a* 3 *approximation to TFBAS. Moreover if the vertices or edges have weights, still the same approximation factor works. This is because each constraint in the LP will still have* 3 *variables.*

Remark 6. *If we consider a slightly general problem of finding a feedback set, which can consist of both the edges and the vertices (possibly having weights), then the same technique yields a* 6 *approximation algorithm for it. This is because each constraint in the LP corresponding to the problem will have* 6 *variables.*

References

[1] M.R.Garey and D.S.Johnson, Computers and Intractability: A guide to the theory of NP-completeness, Freeman and Company (1979).

[2] D.S.Hochbaum, Approximation algorithms for the set covering and vertex cover problems, SIAM J. Computing 11(3) 1982.

[3] P.D.Seymour, Packing directed circuits fractionally, Combinatorica 15:281-288.