Global Journal of Computer Science and Technology: G Interdisciplinary
Volume 17 Issue 2 Version 1.0 Year 2017
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

A New Ranking Algorithm for a Round-Robin Tournament

By Raghad Rowaida & Afsana Ahmed Munia
Ahsanullah University of Science & Technology

Abstract- The problem of ranking players in a round-robin tournament, in which outcome of any match is a win or a loss, is to rank players according to their performances in the tournament. In this paper, we have improved previously developed MST (Majority Spanning Tree) algorithm for solving this problem, where the number of violations has been chosen as the criterion of optimality. We have compared the performance of our algorithm with the MST algorithm and GIK algorithm.

Keywords: ranking, round-robin tournament, upset, digraph, MST, GIK.

GJCST-G Classification: I.1.2, G.1.2

A New Ranking Algorithm for a Round-Robin Tournament

Raghad Rowaida α & Afsana Ahmed Munia σ

Abstract- The problem of ranking players in a round-robin tournament, in which outcome of any match is a win or a loss, is to rank players according to their performances in the tournament. In this paper, we have improved previously developed MST (Majority Spanning Tree) algorithm for solving this problem, where the number of violations has been chosen as the criterion of optimality. We have compared the performance of our algorithm with the MST algorithm and GIK algorithm.

Keywords: ranking, round-robin tournament, upset, digraph, MST, GIK.

I. Introduction

The problem of ranking players in a tournament has been the subject of various research investigations. This tournament structure also arises in other environments like the problems of soliciting customer preferences of a set of products, establishing funding priorities of a set of projects [5], establishing searching priorities for a set of search engines in the internet. It is known that the results of a tournament can be represented in adigraph, G=(V, A) known as tournament graph, where vertices correspond to players and arcs correspond to match results. A tournament result is said to be upset (or violation) if a lowly-ranked player has defeated a highly-ranked player. Ali[1], Cook[6], Goddard[5], Poljak[3] and many others have concentrated on the problem of determining ranks based on the results of the tournament. A constructive lower bound on the tournament ranking function was obtained in [4]. In [2], a heuristic solution to optimize the number of violations has been developed. This paper presents a new version of MST algorithm which reduces the number of violations compared to MST algorithm. The problem of minimizing the number of upsets is equivalent to finding the minimum number of arcs in adigraph deletion of which results in an acyclic digraph.

This problem is knownas Minimum Feedback Arc set Problem, and is NP-hard for general digraphs [1].

Author α: Lecturer, Bangladesh University of Business & Technology (BUBT). B.Sc. Ahsanullah University of Science & Technology (AUST). e-mail: raghad.rowaida101@gmail.com

Author σ: Lecturer, Department of CSE, Bangladesh University of Business and Technology. M. Sc. Bangladesh University of Engineering and Technology (BUBT). B. Sc. Ahsanullah University of Science and Technology (AUST). e-mail: afsana.106@gmail.com

II. Preliminaries

Before describing the new algorithm, we present here a brief discussion on MST algorithm [2] and GIK algorithm [1].

MST: For ease of discussion we recapitulate some of the definitions used in MST algorithm.

1. *cutset(i, k, j) – is* the difference between the numbers of outgoing arcs from set (i, k) to set (k + 1, j) and outgoing arcs from set (k + 1, j) to set (i, j), where set (i, j) is the set of vertices corresponding to players ranked from i to j.
2. *maxwin(i, j)* – is the maximum number of wins of a player in set (i, j).
3. *pair(i, j)* – corresponds to an upset if the player ranked j defeats the player ranked i.
4. *size* – is the number of players in the tournament.

```
MST ()
    Repeat until swap = false
    swap ←false
    for i= 1 to size-1 do
        for j = i + 1 to size do
            for k=i to j-1 do
                if cutset(i,k,j)< 0
                    swap ←true
                elseif cutset(i,k,j)= 0
                    if pair(i,j) or ( i
                    - l , k + l) or
                    (k,j+ l) is
                    upset then
                          swap ← true
                          swap
                          respective
                          pair
                     else if maxwin(i,
                     k) < maxwm(k + 1,
                     j)
                           swap
                           respective
                           pair
                     endif
                endif
                if swap = true then
                     swap set ({i, k),
                     {k+ 1,j})
```

Global Journal of Computer Science and Technology (G) Volume XVII Issue II Version I Year 2017

```
            endif
        endfor (k-loop)
    endfor (j-loop)
endfor (i-loop)
```

Assuming the number of players in the tournament to be n, complexity of the MST algorithm can be derived as follows: In the k-loop, calculation of cutset value requires *O(n)* operations. Each of the i, j and k-loop will be done at most n times for a single swap, which will reduce the number of violations by 1. The amount of computation for this is at most $O(n^4)$. Since there can be at most $O(n^2)$ violations initially, the algorithm requires at most $O(n^6)$ calculations.

GIK: This algorithm is based on the IK algorithm [ ]. When applying the IK algorithm to rank a tournament, two basic steps are executed in case of a tie. The first attempts to break the tie by restoring the players, while the second (which is applied when the first step fails) randomly ranks the players involved in the tie. The GIK algorithm differs from the IK procedure in these two steps. The restoring method is different, and if this restoring method does not resolve the ties, an attempt is made to rank the players in a manner that will reduce the overall number of violations.

The GIK algorithm appears below. The following conventions are used.

a. $|S|$ denotes the cardinality of the set of players.
b. $\varnothing$ denotes the empty set.
c. If S_1, and S_2 denote sets (subsets) of players, then $S_1 \backslash S_2$, denotes those players in S_1, but not in S_2.
d. If R denotes a ranking and *P* a player, $R||P$ denotes the ranking formed by placing player *P* after the last player in the ranking *R*.
e. Given $R_1 = (P_1 > P_2 > \ldots > P_k)$ and $R_, = (Q_1 > Q_2 > \ldots > Q_J)$, then $R1||R2 , = (P_1 > P_2 > \ldots > P_k > Q_1 > Q_2 > \ldots > Q_J)$.

The GIK Algorithm

1. Let $R = \varnothing$, $A = \{P_1, P_2, \ldots, P_n\}$.
2. If $A = \varnothing$, then to go (15); otherwise determine the current scores of players in A.
3. If $A = \varnothing$, then go to (15); otherwise determine *D*, the dominant set.
4. If $|D| > 1$, then to go (6).
5. Letting *P* denote the only player in *D*, form the ranking $R = R \,||\, P$, let $A = A\backslash\{P\}$ and go to (3).
6. If from the last time of updating the current scores of A [step (2)], set A has changed, then go to(2).
7. If $|D| > 2$, then go to (9).
8. Let P_1 and P_2, denote the players in D with $P_1 > P_2$. Let $R= R||\, P_1 ||P_2$, and $A=A\backslash\{P_1, P_2\}$. *Go* to (2).
9. If $R = \varnothing$, then go to (11).
10. Arrange all players in *D* in Hamiltonian order H, i.e. $H = (P_1 > P_2 > \ldots > P_k)$. Let $R = R \,||\, H$ and $A = A\backslash\{P_1, P_2, \ldots, P_k\}$.*Go* to (2).
11. Let Q denote the last player presently in *R,* and let $\{P_1, P_2, \ldots, P_k\}$ constitute *D.* Let i = 1.
12. If $i > k$, *go* to (10).
13. If $Pi > Q$, put *Pi* in *R* ahead of Q. Let $A = A\backslash\{Pi\}$ and $D = D\backslash\{Pi\}$. If $|D| = \varnothing$, then go to (2).Otherwise go to (4).
14. Let $i=i + 1$, and go to (12).
15. Execute procedure Arrange on the ranking *R.*
16. End.

III. The New Algorithm

In this Section we propose A new version of MST algorithm that results in minimum number of upset compared to the MST algorithm and GIK algorithm for ranking players in a round-robin tournament [].

We consider only simple connected digraphs G=(V,A). Spanning trees of any digraph are denoted by T. A directed cutset(V_i,V_j) is defined as $(V_i,V_j)=\{(k,l)\,|\,k \in V_{i,}\, l \in V_j\}$

For improvement of the algorithm we introduce the following symbols and functions:

Sa — start of setA
Ea —end of setA
Sb — start of setB
Eb —end of setB
Sc — start of setC
Ec —end of setC

Cutset(A,B)- is the difference between the numbers of outgoing arcs from set A to set B and outgoing arcs from set B to set A.

Cutset(A,C)- is the difference between the numbers of outgoing arcs from set A to set C and outgoing arcs from set B to set A.

Cutset(B,C)- is the difference between the numbers of outgoing arcs from set B to set C and outgoing arcs from set C to set B.

Procedure: Improved MST

```
Repeat until swap = true
swap = false
  for Sa =0 to size – 1 do
    for Ea = Sa to size-1 do
      for Sb = Ea to size – 1 do
        for Eb =Sb to size – 1 do
          for Sc = Eb+1 to size do
            for Ec = Sc to size do
              if (cutset(A,B) +
              cutset(A,C) +
              cutset(B,C)) < 0) then
                swap = true
              else if(cutset(A,B)
              + cutset(A,C) +
```

Global Journal of Computer Science and Technology (G) Volume XVII Issue II Version I Year 2017

```
                    cutset(B,C)) = 0) then
                        if(pair(Sa-
                      1,Sa)or
                      pair(Ea,Sb)orpair(E
                      b,Sc)orpair(Ec,Ec+1
                      ) is upset )then
                           swap = true
                           swap  respective
                         pair
                        else             if
                      (maxwin(Sa,Ea)<maxw
                      in(Sc,Ec))
                           swap=true
                           swap  respective
                         pair
                        if    (swap==true)
                      then
                           swapSet(A,C)
                  break Ec-loop
                break Sc-loop
              break Eb-loop
            break Sb-loop
          break Ea-loop
        break Sa-loop
```

IV. Experimental Results

The new_MST Algorithm has been compared with MST Algorithm and the GIK algorithm on the basis of a set of randomly generated tournaments of sizes ranging from 5 to 50 players. All algorithms have been programmed in C and runs were made on a core i3 machines. We have been measured both in terms of violations and computational time. Here new_MST gives better result compared to MST and GIK with respect to number of violations.

Table 1: Comparison among new MST, MST and GIK in terms of number of violations

No of player	Initial upset	GIK	MST	New MST
5	3.66	2.66	1.66	1.66
10	24.00	13.33	9.00	8.66
15	47.33	39.33	25.66	24.66
20	89.00	38.33	25.33	22.00
25	194.33	109.66	76.33	72.33
30	106.66	94.66	67.33	61.00
40	482.00	138.66	88.66	79.00
50	585.66	515.00	439.00	418.33

Table 2: Average computational time of three algorithms in seconds

No of player	GIK	MST	New MST
5	0.0013	0.0030	0.0010
10	0.0103	0.0090	0.0110
15	0.0173	0.0093	0.0680
20	0.0226	0.0236	0.5756
25	0.0266	0.0563	88.5506
30	0.0320	0.0216	1268.37
40	0.043	0.1913	24877.110
50	0.054	4.147	63415.8188

V. Conclusion

Experimental results show that our new MST algorithm reduces the number of violations compared to GIK and MST algorithm but the drawback is new MST algorithm requires huge time compared to those algorithms.

References Références Referencias

1. Ali, W. D. Cook and M. Kress. (1986). On the minimum violations ranking of a tournament. M qmt Sci. 32, 660-674.
2. Mohammad Kaykobad, Q. N. U. Ahmed, A. T. M. Shafiqul Khalid, Rezwan-al Bakhtiar.(1995). A New Algorithm for Ranking Players of a Round Robin Tournament. Computer Ops Res. Vol. 22 no. 2, pp. 221-226.
3. S. Poljak, V. Rodl, J. Spencer. (1988). Tournament ranking with expected profit in polynomial time, SIAM Jl Disc. Math.I, No. 3.
4. S. Poljak, A. Czygrinowy, V. Rodl, Constructive quasi-ramsey numbers and tournament ranking, SIAM J. Discrete Math., Vol. 12, No.1, pp. 48-63.
5. S. T. Goddard. (1983). Ranking tournaments and group decision making. Mgmt Sci. 29, 1384-1392.
6. W. D. Cook, I. Golan and M. Kress.(1988).Heuristics for ranking players in a round-robin tournament. Computers Ops Res. 15, 135-144.

This page is intentionally left blank