

Detecting Manipulation in Cup and Round Robin Sports Competitions

Tyrel Russell
Cheriton School of Computer Science
University of Waterloo
Waterloo, Canada N2L 3G1

Peter van Beek
Cheriton School of Computer Science
University of Waterloo
Waterloo, Canada N2L 3G1
vanbeek@uwaterloo.ca

Abstract—Some of the core attributes of sports are supposed to be integrity, honesty and sportsmanship. However, there have been numerous cases where individual athletes and teams have thrown games and colluded to manipulate sports competitions. Much of the previous work has focused on proving that some types of competitions are computationally hard to manipulate and thus possibly resistant to manipulation. In contrast, in this paper we focus on detecting patterns of manipulations in sports competitions by coalitions of teams. We show that it is possible to successfully detect certain patterns of manipulations in cups and round robins, two of the most common types of sports competitions. The practical benefit of our approach is a tool that can be used by competition organizers to highlight suspicious play within the competition and so aid in the detection of cheating coalitions.

I. INTRODUCTION

It is well known that manipulation occurs in sports. Cheating has been found in sports ranging from soccer to sumo wrestling and lawn bowling [1]–[3]. One method to prevent or impede the manipulation of sports competitions is to use a competition type that is more resistant to manipulation. As a result, much previous work in the study of manipulations in sports and related work in social choice theory on the manipulation of elections has focused on proving that some types of competitions are computationally difficult or impossible to manipulate (e.g., [4]–[9]). Unfortunately, two of the most common types of sports competitions, cups and round robins, are not resistant to manipulation as it is known that for both, finding a minimal strategy for a coalition of teams to manipulate the competition is computationally easy [10].

Another method to prevent or impede the manipulation of sports competitions is to develop robust detection techniques. Previous work in the detection of manipulation in sports has focused on identifying the rigging of individual matches or events within a match [1], [2]. Such detection is often done through an analysis of betting and financial records [2]. However, cheating is known to extend beyond single teams throwing individual matches to coalitions of teams manipulating multiple matches in order to manipulate the placement of certain teams in a competition. As just one example, the 1971–72 Bundesliga scandal in German soccer involved 52 players, nine teams, and the manipulation of 18 matches with the goal of attaining the promotion or avoiding the relegation of certain teams [3].

In this paper, we address—to the best of our knowledge, for the first time—the problem of developing automated tools for detecting coalitions of teams manipulating the winner of a competition. In a competition, a certain number of games have unexpected results and are known as upsets. If a coalition was manipulating the competition, the resulting manipulations could easily be mistaken for genuine upsets. However, the specific pattern of results should reflect the coalitions plan to change the winner of the competition using as few extra manipulations as possible, as each manipulation increases the risk of detection. It is this strategic behavior by a coalition that we aim to detect.

We formalize the notion of strategic behavior and provide algorithms, for both cup and round robin competitions, for detecting such coalitions from the upsets identified once the games have been played. We experimentally evaluate our proposals on real and randomly generated sport competition instances. On these benchmarks, our techniques accurately detect the difference between instances where an embedded coalition was manipulating the result and where upsets were occurring only at random. In cases where there was a manipulating coalition, we show that one can almost always quickly identify the cheating coalition, while in the case where there was no manipulating coalition, often no (or only a few easily dismissed) false positives are generated. The practical benefit of our approach is to provide the organizers of cup and round robin competitions with a useful tool to identify when suspicious patterns of behavior have occurred. Our approach does not provide a proof of dishonest behavior but rather provides organizers with a significant starting point to investigate possible corruption in a competition.

II. BACKGROUND

Two of the most common ways of organizing a competition among a set of teams are cups and round robins. In what follows, let T denote a set of teams of size m , where the names of the teams are t_1, t_2, \dots, t_m .

In a *cup* competition, the final winner is determined by a tree-like structure called a *competition tree*. For simplicity, we assume that the number of teams in a cup is a power of 2 and thus the competition tree is a complete binary tree. The leaves of the tree are labeled with the names of the teams, called the *seeding* of the competition. Each internal node of the tree

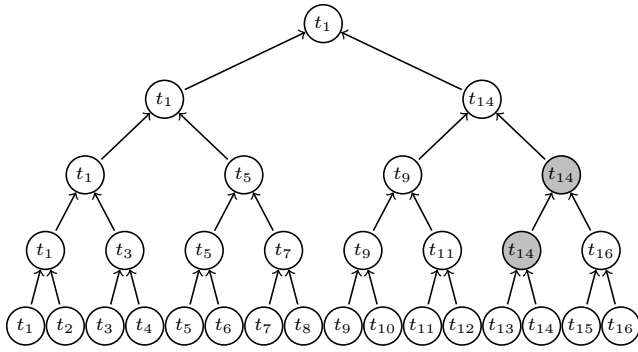


Fig. 1. The result of the cup if t_{13} and t_{16} manipulate the competition and all other matches happen as expected. Grey nodes indicate upsets.

represents a match (or game) between its two child nodes. A *round* of a cup is defined to be all of the matches that occur at an equal height k , $1 \leq k \leq \log_2 m$, from the leaves of the competition tree.

In a *round robin* competition, each team plays every other team a given number of times. We restrict ourselves to single round robin competition using the simple win-loss scoring model. In a single round robin competition, each team plays every other team exactly once. A *round* of a round robin is defined to be all the matches that occur at the same time in the schedule. If m is even, the minimal number of rounds is $m - 1$ where each team plays exactly once per round and there are $m/2$ matches per round.

A *tournament graph* is a directed graph $G = (T, E)$, where T is the set of teams and the underlying undirected graph is a complete graph. A directed edge $(t_i, t_j) \in E$ represents that t_i would be the expected winner in any fair match between team t_i and team t_j . An upset is a match where team t_i was expected to win over team t_j according to the tournament graph, but team t_j won against team t_i in the actual competition. A team t_i is said to have *caused* an upset if they lost a match that they were expected to win.

Definition 1 (Upset): Let $G = (T, E)$ be a tournament graph. An *upset* is a pair (t_j, t_i) where $(t_i, t_j) \in E$ but t_j won against t_i in some round of the actual competition. Let U denote the set of all upsets that occurred in the competition, let U^k denote only those upsets that occurred at round k , $1 \leq k \leq \log_2 m$, of the competition, and let $U_S^k, S \subseteq T$, denote only those upsets that occurred at round k that were caused by a team in S ; i.e., $U_S^k = \{(t_j, t_i) \mid (t_j, t_i) \in U^k \wedge t_i \in S\}$.

III. STRATEGICALLY OPTIMAL COALITIONS

Our focus is on a particular subset of upsets that are caused by a coalition. A *coalition* is a set of teams S , $S \subseteq T$, which conspires to manipulate the competition such that their chosen team wins the competition. A *manipulation* is an upset (t_j, t_i) , either executed or planned, that is intentional; i.e., team $t_i \in S$ threw the match or planned to throw the match.

We assume that coalitions have formed prior to the start of the competition as there is a lack of opportunity to do so during the competition. Our approach is designed for posthoc

analysis. Thus, the winner of each match and the final winner of the competition are known¹. We assume that some matches are then labeled as upsets and that the tournament graph is known. The labeling of upsets and the tournament graph could come, for example, from experts who knew the outcomes of the matches as well as the relative strengths of teams and historically how well teams have played against each other.

Let t_w denote the final winner of the competition. At each round of the competition, if t_w wins the competition as a result of manipulation by a coalition of teams, the coalition must have formulated a strategy to guarantee that t_w wins. The strategy may need to change from one round to the next as the outcomes of many of the games will not be under the coalition's control and there may be upsets caused by teams that are not in the coalition. A key point is that there is a strong incentive for a coalition to reduce the number of manipulated matches, as each manipulation increases the risk of detection while providing no benefit.

Definition 2 (Optimal Manipulation Strategy): Given a coalition S , a distinguished team t_w , and the results of past rounds $1, \dots, k-1$ of the competition, a *manipulation strategy* for S in round k is a set of manipulations that if executed ensures t_w wins the competition under the assumption that the matches not manipulated occur as expected in the tournament graph. A manipulation strategy for S in round k is *optimal* if there exists no strategy for S with fewer manipulations.

Example 1: Consider a 16 team cup where, if the outcomes of the matches follow the tournament graph (not shown), t_9 will win the cup by defeating t_1 in the final. Suppose that the coalition of t_{13} and t_{16} would like t_1 to win the cup. The coalition needs to defeat t_9 to make t_1 the winner but only t_{14} can defeat t_9 . Therefore, an optimal manipulation strategy for $S = \{t_{13}, t_{16}\}$ in round 1 is given by,

- t_{13} throws match to t_{14} in round 1,
- t_{16} throws match to t_{14} in round 2.

If t_{13} then throws the match to t_{14} in round 1 and no other upsets occur in that round, an optimal manipulation strategy for round 2 involves just advancing the plan. If t_{16} then throws the match to t_{14} in round 2 and no further upsets occur in the competition, t_{14} defeats t_9 in round 3 and t_1 wins the cup by defeating t_{14} in the final (see Figure 1).

Our goal is to recognize coalitions that may have manipulated the competition to have their team win. It is not possible to know the coalitions or their manipulation strategies but it is sometimes possible to recognize such coalitions through partial observation of their strategic behavior.

Definition 3 (Strategically Optimal Coalition): A coalition S is a *strategic coalition* if for each round k , $1 \leq k \leq \log_2 m$, the set of upsets U_S^k contains all and only the manipulations that would have been executed in round k in an optimal manipulation strategy for S in that round. A coalition S is a *strategically optimal coalition* if no proper subset of S is a strategic coalition.

¹In the case of round robins, a set of teams may be tied and the eventual winner of the competition is decided by some tie-breaking criteria, which is not discussed further in this paper.

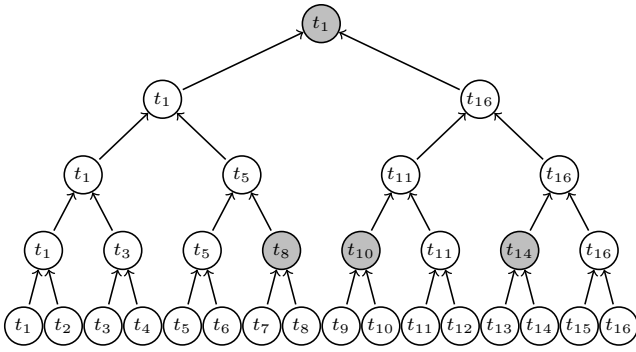


Fig. 2. The result of the cup if t_{13} and t_{16} work in a strategically optimal manner and there are additional upsets caused by non-coalition members. Grey nodes are upsets.

Example 2: Consider once again Example 1, where the coalition $S = \{t_{13}, t_{16}\}$ would like t_1 to win the cup. The optimal manipulation strategy for S in round 1 remains the same and suppose that t_{13} then throws the match to t_{14} but that there are also additional upsets in round 1 involving teams outside of the coalition (see Figure 2). Their original strategy no longer ensures that t_1 wins the cup because t_{11} will defeat both t_{14} and t_1 but would lose to t_{16} according to the tournament graph. The coalition must now reformulate their strategy by having t_{16} beat t_{14} , t_{11} and then ultimately lose to t_1 . The revised strategy for round 2 is given by,

t_{16} throws match to t_1 in round 4.

Assuming no further unexpected upsets, the strategy carries over to subsequent rounds and t_1 wins the cup.

The definition of a strategically optimal coalition can be relaxed by introducing a parameter l and requiring that a manipulation strategy for a coalition S be within l manipulations of optimal in each round of the competition. This allows coalition members to be involved in upsets that are not manipulations. We discuss this further below when we present our algorithms.

IV. DETECTING MANIPULATIONS IN CUP COMPETITIONS

In this section, we present algorithms for identifying strategically optimal coalitions within cup competition results.

Algorithm STRATEGICCOALITION verifies that a coalition S is a strategically optimal coalition. At each round k , $1 \leq k \leq \log_2 m$, the set of upsets U_S^k attributed to S must be part of an optimal manipulation strategy. The minimal number of manipulations needed to ensure that t_w is a winner for a given set S is denoted a_k . The routine MINNEEDED (not shown) for determining the minimal number of manipulations needed to establish t_r as the winner is as specified in [10, pp. 32–33]. Fixing U_S^k as played, the minimal number of manipulations possible for rounds $k+1, \dots, \log_2 m$, denoted c_k , can similarly be determined. If, for every round k , $a_k = |U_S^k| + c_k$ then S is a strategically optimal coalition since the coalition never uses more than a minimal number of manipulations given the known upsets. Note that the parameter l which relaxes the definition of strategically optimal can be incorporated easily

Algorithm:STRATEGICCOALITION(C, t_r, S, U)

input : A competition tree C , a team t_r to establish as the winner of the sub-tree, a coalition S , and the upsets U that occurred in the competition.

output: Returns true if S is a strategically optimal coalition; false otherwise.

$Upsets \leftarrow \{\}$;

for $k \leftarrow 1, \dots, \log_2 m - 1$ **do**

$a_k \leftarrow \text{MINNEEDED}(C, k, t_r, S, Upsets)$;

if $k = 1 \wedge a_k \neq |S|$ **then return false**;

$c_k \leftarrow \text{MINNEEDED}(C, k, t_r, S, Upsets \cup U_S^k)$;

if $a_k < |U_S^k| + c_k$ **then**
 return false;

$Upsets \leftarrow Upsets \cup U^k$;

return true;

into this test. In the algorithm, if a coalition can no longer generate any manipulation strategy to make t_r the winner, a_k and c_k are undefined. To avoid complicating the algorithm, these cases are not shown but it should be assumed that the algorithm returns false.

Lemma 1: Determining if a coalition S is strategically optimal can be computed in $O(m^2 \log_2 m)$ time.

Algorithm FINDALL can be used to determine all strategically optimal coalitions that ensure that a given team wins a cup competition. It returns NULL if there are no such coalitions; otherwise it returns the set of strategically optimal coalitions (possibly returning just the empty coalition to indicate that no manipulations are necessary). The algorithm is specified in a recursive, top-down manner. The initial call to the algorithm is FINDALL(C, t_w), where t_w is the final winner of the competition and C is the cup competition. The algorithm generates all strategically optimal coalitions for sub-trees and then merges them together, pruning as they are constructed. The algorithm does this by generating optimal manipulation strategies. Pruning is then based on a coalition not establishing the team t_w (it does not achieve its goal) or it uses too many manipulations to establish the team t_w (it is not optimal).

Two optimizations of the algorithm are included in the implementation used in our experiments but are not shown. First, memoization is used to avoid recomputing recursive calls. Second, decomposition is used to speed up the computation as it can be observed that each problem can be decomposed into $\log_2 m$ separate sub-problems or sub-trees, where m is the number of teams in the competition. The result is that algorithm FINDALL is efficient in practice for the sizes of competitions that occur in practice. However, the computational complexity of the algorithm and whether there exists an algorithm with an improved worst-case complexity remain open questions.

V. DETECTING MANIPULATIONS IN ROUND ROBINS

In this section, we present algorithms for identifying strategically optimal coalitions within single round robin competition results.

Algorithm:FINDALL(C, t_r)

input : A competition tree C and a team t_r to establish as the winner. Assumes that t_w , the final winner of the competition, is available.

output: Returns the set of all strategically optimal coalitions that can ensure t_r wins the subtree C ; returns NULL if there are no such coalitions.

```

if  $C$  consists of a single team then
  if  $T$  contains  $t_r$  then return  $\{\}$ ;
  else return NULL;
else
  Let  $C_r$  be the subtree of  $C$  that contains team  $t_r$  and
  let  $C_o$  be the other subtree;
   $S_r \leftarrow$  FINDALL( $C_r, t_r$ );
  if  $S_r =$  NULL then return NULL;
   $S_o \leftarrow$  NULL;
  foreach  $t_k \in C_o$  do
     $S_{temp} \leftarrow$  FINDALL( $C_o, t_k$ );
    if  $S_{temp} =$  NULL then return NULL;
    if  $t_r$  is not expected to win against  $t_k$  then
      add  $t_k$  to each coalition in the set of
      coalitions  $S_{temp}$ ;
    add the coalitions in  $S_{temp}$  to  $S_o$ ;
    remove from  $S_o$  any coalition that is a superset
    of another coalition in  $S_o$ ;
   $S_{temp} \leftarrow S_r \times S_o$ ;
  if  $t_r = t_w$  then //  $t_w$  is to win the subtree  $C$ 
    foreach  $S \in S_{temp}$  do
      if  $\neg$ STRATEGICCOALITION( $C, t_r, S$ ) then
        prune  $S$  from  $S_{temp}$ ;
  return  $S_{temp}$ ;

```

In round robins, there are two types of manipulations—we refer to them as *simple* and *complex*, respectively—that a coalition can perform to change the result of the competition. In simple manipulations, a coalition member loses its games directly to the desired winner t_w . In complex manipulations, coalition members lose games amongst themselves so that t_w needs to win fewer games in order to win the competition. We assume that a coalition member will always manipulate any game they play against t_w so that t_w wins unless t_w is already expected to win the competition.

We first consider strategies that consist of only simple manipulations. The goal of any such coalition is to lose enough games to t_w so that t_w earns more victories than the expected winner. In this context, a strategically optimal coalition would be a coalition that always manipulates games against t_w when t_w is not expected to win and does nothing otherwise. Given this goal, let w_w be the number of victories needed; i.e., the difference between the expected number of victories of the expected winner and the expected victories of t_w . There are two parts to determining the set of strategically optimal coalitions which only perform simple manipulations:

Algorithm:POSSIBLEMEMBERS(R, G, t_w)

input : A round robin competition R and a team t_w that wins the competition.

output: Returns all the possible coalition members that act optimally to establish t_w as winner.

```

 $A \leftarrow \{t_j \mid (t_j, t_w) \text{ an upset and } t_j \text{ won no extra games}\};$ 
 $B \leftarrow \{t_j \mid t_j \text{ defeated } t_w \text{ and } t_j \text{ won no extra games}\};$ 
foreach round  $r$  of the season do
   $\perp$  Identify whether  $t_w$  is expected winner in round  $r$ ;
  Remove all teams from  $A$  that played  $t_w$  in round  $r$ 
  where  $t_w$  was expected to win in round  $r$ ;
  Remove all teams from  $B$  that played  $t_w$  in round  $r$ 
  where  $t_w$  was not expected to win in round  $r$ ;
return  $A$  and  $B$ ;

```

(1) identify all the possible coalition members and (2) generate all the coalitions from the possible members.

Algorithm POSSIBLEMEMBERS shows the steps for identifying all the possible members. The algorithm (implicitly) removes all teams that lost to t_w as expected and all teams that won games they were not expected to win (since coalition members should not have won extra games by manipulation). The remaining teams are broken into two sets: (A) those teams that actually lost to t_w and (B) those teams that actually defeated t_w . A team $t_j \in A$ is optimal iff t_w is not expected to win the competition when playing t_j and a team in B is optimal iff t_w is expected to win. The teams that are not optimal are pruned from the sets of A and B .

Lemma 2: Algorithm POSSIBLEMEMBERS computes the set of possible coalition members in $O(m^3)$ time.

The set of teams A and B are then used to generate coalitions that could optimally manipulate the competition. For a coalition to be strategically optimal, the coalition must be of size w_w and, for each round, there must always be enough possible coalition members to ensure that t_w becomes the winner. Any subset of teams of size w_w from the set $A \cup B$ that satisfies the constraints is a strategically optimal coalition which only uses simple manipulations. By checking teams in $A \cup B$ which have won up to l extra games (instead of no extra games) and creating coalitions such that the total number of extra games is less than or equal to l , it is possible to incrementally relax the restriction that teams must not have won extra games as coalition members.

We next consider strategies that consist of both single and complex manipulations. Unlike manipulation strategies where only losses to t_w occur, coalitions that must self-sacrifice themselves would manipulate other games. In this case, all teams that earn extra victories and are part of the coalition must have earned the extra points from coalition members. Given a Boolean variable s_i which is true if t_i is a member of the coalition and the set of teams L_i that lose to t_i when they should have won, constraint (C2) can be derived.

The set of teams A and B can be calculated using Algorithm POSSIBLEMEMBERS, except that no check must be made to ensure they won no extra games. An additional set C is

defined that includes teams that were expected to lose to t_w and did lose to t_w while only receiving wins when t_w is not the expected winner. Since a coalition could effectively achieve their goal by only manipulating games against t_w , there must be at most $w_w - 1$ members from $A \cup B$ for complex manipulations to be necessary. Let u be the number of teams from $A \cup B$ belonging to the coalition such that $0 \leq u < w_w$. Given a value of victories that t_w could achieve, u , there is a specific set of top teams, TT_u , that must lose at most $w_w - u$ games intentionally. Since any team can be upset, a minimal manipulation strategy is one where the coalition only manipulates a minimal number of games between themselves and do not earn any wins from teams not in the coalition.

Since each TT_u set can be determined in isolation, the goal is to extend the set, if necessary, so that the coalition can achieve the goal but never manipulates more games than necessary amongst themselves. After identifying the set of teams that form the sets A , B , and C , all valid sets must be generated. The following constraint satisfaction problem is used to find all possible coalitions from A , B and C given the external constraints. Constraint (C1) ensures the coalition only attempts to have t_w earn enough games for TT_u to be valid. Constraint (C2) is the extra constraint described above that enforces that teams that are upset by coalition members are coalition members themselves. Constraint (C3) states that teams that are not a member of the possible sets of teams are not part of any coalition. Constraint (C4) ensures that all the teams in the TT_u set under consideration are members of the coalition.

$$\begin{aligned} \text{C1: } & \left(\sum_{t_i \in A \cup B} s_i \right) = u & \text{C3: } & \forall t_i \notin A \cup B \cup C \quad \neg s_i \\ \text{C2: } & \forall t_i \in T \quad s_i \Rightarrow \bigwedge_{s_j \in L_i} s_j & \text{C4: } & \forall t_i \in TT_u \quad s_i \end{aligned}$$

Given the set of possible coalitions generated by solving the constraint satisfaction problem, it must be ensured that only a minimum number of manipulations are used in each round to ensure the coalition is strategically optimal. Using the minimum cost feasible flow technique described in [10] for finding minimal manipulations in round robins, it is determined if a smaller set of teams could have manipulated the result at the start of the competition. If so, the possible coalition is discarded. Next, the remaining coalitions are checked to ensure that the manipulations used in any given round are minimal by calculating the before and after cost of using the actual manipulations, which should be the same for every round. Any possible coalition that has a minimal number of members and uses as few manipulations as possible in each round is strategically optimal. To check for coalitions which may have won a limited number of extra games, constraint (C2) is modified to not include up to l members of L_i and the minimal number of manipulations calculated must not differ by more than l additional wins.

VI. EXPERIMENTAL EVALUATION

In this section, we present the results of an empirical evaluation of the effectiveness of our algorithms for detecting

manipulations in cup and round robin competitions.

For cup competitions, we evaluated our proposals on randomly generated instances based on the NCAA Division I Basketball Championship and on real instances from Grand Slam tennis². The random instances are realistic in that they follow closely the structure of the NCAA Championship: teams are ranked and seeded using pools of 16 teams and the best-plays-worst paradigm, and tournament graphs and upsets are generated according to a distribution that was estimated from 25 past championships (1985-2009). For our experiments, we generated 1000 instances with random upsets and a further 1000 instances where a manipulating coalition was embedded in the competition. To generate a strategically optimal coalition, a team is randomly selected to be manipulated. An initial coalition is created by finding the minimal set of teams needed to make the selected team the winner of the competition. The diversity of the types of coalitions produced is then increased by randomly removing teams from the initial coalition and adding a new, viable set of teams in their place. This process is repeated recursively until no more substitutions are possible. Upsets are added randomly along with the manipulations of the coalitions, while maintaining the property that there remained at least one feasible plan for the coalition in each round.

On these cup benchmarks, the accuracy of our method is high (see Table I). For instances where there was no manipulating coalition, often no false positives are generated (from 47% to 87% of the time, depending on the size of the competition) or the number of spurious coalitions is small (from 100% having four or fewer to 95% having 12 or fewer, depending on the size of the competition). The spurious coalitions are also easy to dismiss as they involve highly ranked teams being upset by other highly ranked teams and so lack any “surprise”. For instances where there was a manipulating coalition, there were sometimes many false positives (hundreds or even thousands for the largest competitions). To deal with this problem, we propose an intuitive heuristic for sorting the coalitions. For each coalition, we define its “surprise factor” by calculating for each upset the distance between the ranks of the two teams, where of course the greater the distance the greater the surprise factor. As shown in Table I, using this heuristic the organizers could look at less than 20 coalitions and be quite confident that a coalition would be found, if it existed. The correct coalition would be detected from the spurious coalitions by examining, for example, tapes of the games or other external evidence that the teams were working together.

The second data set for cup competitions consisted of 40 Grand Slam Tennis events from 2001 to 2010. Upsets were recorded if the winner of the match was eight positions lower in ranking, where the rank was calculated by the seed value

²In both sports, there is a strong economic incentive to coalitional cheating. In NCAA basketball just reaching the “Final Four” has significant economic benefits [11] and in US Open Grand Slam tennis manipulating a single player into the quarter finals results in a payoff of \$100,000 and the payoff doubles at each subsequent round.

TABLE I

Cup. EFFECT OF COMPETITION SIZE ON ACCURACY OF DETECTING A MANIPULATING COALITION; AND PERCENTAGE OF TIMES A MANIPULATING COALITION WAS FOUND IN THE FIRST n SORTED COALITIONS, WHERE $n = 1, 10,$ AND 20 .

Size	Accuracy	Top 1	Top 10	Top 20
16	76.7	77.7%	100.0%	100.0%
32	81.2	67.8%	100.0%	100.0%
64	85.4	61.3%	99.4%	99.9%
128	89.4	49.1%	94.5%	98.4%
256	93.5	31.7%	78.1%	87.0%

TABLE II

Round robin. EFFECT OF COMPETITION SIZE ON ACCURACY OF DETECTING A MANIPULATING COALITION AND THE NUMBER OF COALITIONS GENERATED ([MIN-MAX]:AVERAGE).

Size	Accuracy		Number	
	Simple	Complex	Simple	Complex
6	88.5	98.2	[1-10]:1.6	[1-3]:1.2
12	97.5	100.0	[1-7]:1.6	[1-1]:1.0
18	99.5	98.0	[1-12]:2.0	[1-2]:1.1
24	99.5	100.0	[1-17]:2.2	[1-21]:1.5
30	99.0	100.0	[1-35]:2.1	[1-39]:1.9
36	99.5	100.0	[1-16]:1.6	[1-22]:2.7
40	99.5	100.0	[1-21]:1.5	[1-15]:2.2

assigned to the top 32 players and the World Ranking for the remaining 96 players. (A difference of eight is comparable to the definition of an upset in the NCAA tournaments as there would be eight pools of 16 in a competition with 128 opponents.) The tournament graph was constructed by generating an edge between two players t_i and t_j if t_i has a better rank than t_j or if t_i actually defeated t_j in the competition and was within eight positions of t_j . Each competition was tested to determine if the winners from the fourth round and onwards were due to coalitional manipulation. The results showed little evidence of coalitional manipulation. While some small coalitions were found, they consisted of top players being upset and allowing for the next strongest player to win and so lacked any “surprise”.

For round robin competitions, we evaluated our proposals on 100 randomly generated instances without coalitions, 100 instances with coalitions embedded that only use manipulations against t_w (simple), and 100 instances where top teams lose games to ensure that t_w wins (complex). The instances were generated as follows. First, a set of round robin schedules was generated from a canonical schedule described by de Werra [12] with the teams permuted to generate different schedules. Problems were generated for round robins of even size from 4 to 40. Forty was chosen as the maximum as it was larger than the number of teams in most professional sports leagues. Tournament graphs were generated as described for cup competitions. On these round robin benchmarks, the accuracy of our method is again high (see Table II). Unlike cup competitions, spurious coalitions are rarely generated and fewer spurious coalitions are generated than for cup competitions (see Table II). As a result, sorting based on a heuristic appears to be unnecessary.

VII. CONCLUSION AND FUTURE WORK

Previous work on manipulation in elections and sporting competitions has focused on proving that some types of competitions are computationally hard to manipulate and thus possibly resistant to manipulation. In contrast, we examined the problem from the perspective of detecting cheating. We described a reasonable set of behaviors of a coalition of teams desiring to manipulate a competition and showed that the resulting pattern of manipulations could be successfully detected in both cup and round robin competitions. We experimentally evaluated our proposals on both real and randomly generated instances. On these benchmarks, our techniques almost always accurately identified whether cheating was occurring and allowed the quick identification of the cheating coalition.

Our results constitute a step towards developing automated tools for detecting teams colluding to manipulate the winner of a competition. Below we describe two ways of further advancing this work. First, in this paper we have assumed that a coalition uses a deterministic model of the competition (i.e., a tournament graph) when planning their cheating. An alternative is a probabilistic model M , where an entry $M(t_i, t_j) = p_{ij}$ represents that team t_i would be the expected winner in any fair match between team t_i and t_j with probability p_{ij} . Open problems would then be how can teams manipulate a competition within the probabilistic model and how can automated tools be designed to recognize such manipulation. Second, the method for identify cheating competitions should take into account a cost-benefit analysis as a way of distinguishing likely coalitions from unlikely coalitions. For example, if the teams that allegedly were part of the coalitions lost more money collectively than was gained by manipulation then that manipulation strategy is unlikely.

REFERENCES

- [1] M. Duggan and S. D. Levitt, “Winning isn’t everything: Corruption in sumo wrestling,” *The Amer. Econ. Review*, vol. 92, pp. 1594–1605, 2002.
- [2] D. Hill, *The Fix*. McClelland & Stewart, 2008.
- [3] M. Maennig, “Corruption in international sports and sport management: Forms, tendencies, extent and countermeasures,” *European Sport Management Quarterly*, vol. 5, pp. 187–225, 2005.
- [4] J. J. Bartholdi, C. A. Tovey, and M. A. Trick, “The computational difficulty of manipulating an election,” *Social Choice and Welfare*, vol. 6, pp. 227–241, 1989.
- [5] V. Conitzer, T. Sandholm, and J. Lang, “When are elections with few candidates hard to manipulate?” *J. ACM*, vol. 54, pp. 1–33, 2007.
- [6] J. Lang, M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh, “Winner determination in sequential majority voting,” in *Proc. of 20th IJCAI*, 2007, pp. 1372–1377.
- [7] T. Vu, A. Altman, and Y. Shoham, “On the complexity of schedule control problems for knockout tournaments,” in *Proc. of 8th AAMAS*, 2009, pp. 225–232.
- [8] A. Altman and R. Klienbergl, “Nonmanipulable randomized tournament selections,” in *Proc. of 24th AAAI*, 2010, pp. 686–690.
- [9] T. Russell and P. van Beek, “An empirical study of seeding manipulations and their prevention,” in *Proc. of 22nd IJCAI*, 2011, pp. 350–356.
- [10] T. Russell and T. Walsh, “Manipulating tournaments in cup and round robin competitions,” in *Proc. of First International Conference on Algorithmic Decision Theory*, 2009, pp. 26–37.
- [11] B. J. Coleman, J. M. DuMond, and A. K. Lynch, “Evidence of bias in NCAA tournament selection and seeding,” *Managerial and Decision Economics*, vol. 31, pp. 431–452, 2010.
- [12] D. de Werra, “Scheduling in sports,” *Annals of Discrete Mathematics*, vol. 11, pp. 381–395, 1981.