Journal of Applied Logic • • • (• • • •) • • •–• • •

Contents lists available at ScienceDirect

Journal of Applied Logic

www.elsevier.com/locate/jal

On the complexity of bribery and manipulation in tournaments with uncertain information ☆

Nicholas Mattei [a,*,1,2], Judy Goldsmith [b,1], Andrew Klapper [b], Martin Mundhenk [c]

[a] *NICTA and University of New South Wales, Level 4, 223 Anzac Pde., Kensington, NSW, Australia*
[b] *University of Kentucky, Department of Computer Science, Lexington, KY, USA*
[c] *Institute of Computer Science, Friedrich-Schiller-Universität Jena, Germany*

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Computational social choice
Bribery
Tournaments
Ranking
Uncertainty

ABSTRACT

We study the computational complexity of bribery and manipulation schemes for sports tournaments with uncertain information. We introduce a general probabilistic model for multi-round tournaments and consider several special types of tournament: challenge (or caterpillar); cup; and round robin. In some ways, tournaments are similar to the sequential pair-wise, cup and Copeland voting rules. The complexity of bribery and manipulation are well studied for elections, usually assuming deterministic information about votes and results. We assume that for tournament entrants i and j, the probability that i beats j and the costs of lowering each probability by fixed increments are known to the manipulators. We provide complexity analyses for several problems related to manipulation and bribery for the various types of tournaments. Complexities range from probabilistic log space to $\mathrm{NP}^{\mathrm{PP}}$. This shows that the introduction of uncertainty into the reasoning process drastically increases the complexity of bribery problems in some instances.

1. Introduction

Sports competitions are common forms of entertainment and recreation around the world. In most sports contests both observers and players have some notion of which competitors are favored over others. Many individuals, including some players, wager vast sums of money on the outcomes of particular games and

☆ A preliminary version of this paper appears in the proceedings of the *Twenty-Fifth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2012)*, 2012 [35] and in Nicholas Mattei's Doctoral Dissertation [33]. Most of the work on this project occurred while the first author was a graduate student at the University of Kentucky.

* Corresponding author. Tel.: +61 02 8306 0464.
E-mail addresses: nicholas.mattei@nicta.com.au (N. Mattei), goldsmit@cs.uky.edu (J. Goldsmith), klapper@cs.uky.edu (A. Klapper), martin.mundhenk@uni-jena.de (M. Mundhenk).

[1] Supported in part by NSF grants CCF-1049360, ITR-0325063, and IIS-1107011. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

[2] NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

http://dx.doi.org/10.1016/j.jal.2015.03.004
1570-8683/

tournaments. A quick Google search reveals dozens of players, coaches, referees, and judges convicted of manipulating the outcome of sports competitions through match fixing, point shaving, and outright cheating. Additionally many websites (such as www.kenpom.com) produce and publish in-depth statistics for overall team win/loss predications, predictions for individual player stats on a per game basis, and real-time win probability graphs (which dynamically update their prediction based on in-game events). It is a world of probabilities and manipulation.

We use sports tournaments as a motivating example of domains in which bribery [14] and manipulation by coalitions [10] can undermine the integrity of competition. Using tournaments to select alternatives has received attention (and use) in a variety of fields including AI [52,56], economics [42], and operations research [50]. Tournaments and single winner elections, when the set of candidates and the set of voters are equivalent, are used in many domains including self-organization of ad hoc wireless sensor networks, where leaders are elected to delegate work or act as central routing nodes [46], and web ranking, where page rankings are sometimes determined by links from the set of pages under consideration [8]. In addition to these important applications of tournaments, there has been recent empirical research in political science and sociology revealing that, in the United States, voter preferences in political elections can be significantly affected by apparently irrelevant events, such as sports tournaments [25]. Our research sheds light on the security of tournaments to outside influence. We show that some forms of manipulation are easy and, thus, warn against using these methods for social choice without additional precautions.

The problems we consider are of an abstract nature. We define a general model of sports tournaments where we assume that we can pay an opposing team to not compete to the best of its ability. We provide a parameter to fine tune the fidelity of the model. While it may be difficult or impossible to exactly quantify the impact that say, paying a team's best player to play suboptimally, would have on the outcome probabilities of the match, there are estimations of such impacts available (e.g. http://predictionmachine.com/). Our tournament manipulation and bribery problems have a similar feel to the coalition manipulation and bribery problems often studied in social choice and voting [9,10,14]. However, as we will discuss, there are key differences that do not allow many of these results to transfer in a straightforward way.

In this paper we study general sports tournaments consisting of a series of rounds, where each round consists of a set of matches determined by the outcomes of the matches in previous rounds. Throughout the paper we will use "tournament" in its canonical sense as a sports or matchup competition and not in its strict mathematical sense as a complete directed graph [31]. We will use the equivalent and correct term *complete majority relation* when referring to the mathematical object known as a tournament. We consider several special types of sports tournaments.

General tournament: A tournament consists of a series of rounds, each round consisting of a set of matches between pairs of entrants. Which matches occur in the ith round depends on the number n of entrants and the outcomes of the matches in the preceding rounds. The winner (or set of winners) is determined by a function of the outcomes of all the rounds. Note that the outcomes of the individual matches are determined probabilistically, whereas once these outcomes have been determined, the winners of the tournament are found deterministically.

Bounded tournament: A tournament where the number of rounds is bounded from above by a constant.

Bounded history tournament: A tournament where the set of matches played in the current round depends only on the outcomes of the previous b rounds instead of all previous rounds, for some constant b.

Cup tournament: A single-elimination competition (or knockout tournament [51]) over a complete binary tree where each entrant[3] plays a sequence of matches head-to-head; the winner is the entrant that is left undefeated. The United States' men's and women's NCAA (college) Basketball Tournaments and most tennis majors fall into this category.

[3] We use the term entrant in this paper because we can imagine a tournament made up of individuals or teams.

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004

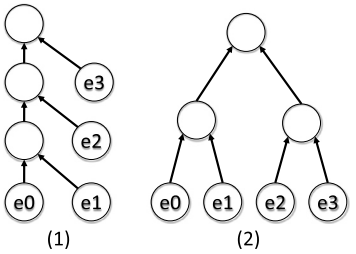

Fig. 1. Example of (1) a challenge tournament and (2) a cup tournament. The winner is the entrant who reaches the top node.

Round robin tournament: A competition where each entrant competes against every other entrant and earns a point for each victory; a winner is an entrant with the most points. The group play round of the FIFA World Cup falls into this category.

Challenge or caterpillar tournament: A series of matches where the winner of each match plays the next entrant in increasing order of rank; the winner is the entrant who wins the final match. Boxing titles and some PBA bowling competitions use this type of tournament. This is a special case of a bounded tournament.

Fig. 1 illustrates the difference between cup and challenge tournaments. In tournaments and sporting events there are several natural questions which arise, such as: "What are the odds my preferred entrant wins?" and "Does my preferred entrant have a chance of winning?"

In the next section we provide an overview of the literature in computational social choice and other settings that study tournaments. In Section 3.1 we formally describe our model of reasoning about bribery in tournaments where entrants have probabilities of winning and losing. Section 3.2 formally states the six decision problems we study in this domain for various tournament types. Section 4 details our complexity results and ends with a few observations about the change in reasoning complexity when models move from deterministic tournaments.

2. Related work

From the Gibbard–Satterthwaite Theorem we know that every reasonable voting rule can be manipulated [20,44]. The key insight of using computational complexity comes from a series of papers by Bartholdi et al. which have helped to spark a vast amount of research in the computational social choice field [6,7]. The idea of manipulation in voting schemes has been extensively studied in the deterministic case [15,17,58], under uncertain or incomplete information models [12,23,57], and in empirical experiments [32–34,36,41,54,55].

Conitzer et al. studied several manipulation problems for voting in stochastic settings, however, many of their NP-completeness results are not immediately transferable to our setting as the reductions require the ability to change the preference relation without adding candidates (entrants) [9,10]. There are also results related to manipulation under deterministic information for cup [10] and Copeland elections [16]. Hazon et al. studied the complexity of manipulating elections under an information model that includes probabilities over orderings [22,23]. They prove that it is PP-hard to evaluate the winner of a Copeland election with their model. This result does not immediately transfer to our proposed settings as we have probabilities over pairwise match-ups, which may lead to non-transitive orderings, instead of a distribution over linear orders. Likewise, the bribery problem for elections, introduced by Faliszewski et al. [14], has been well studied for voting rules under deterministic information. Again, many of these results do not transfer; the hardness reductions require the ability to introduce an unrestricted number of non-candidate voters, or our particular modeling choices of pairwise probabilities preclude direct transfer.

There has been significant work on the schedule control problem for cup or knockout tournaments. In the deterministic case Lang et al. investigate the problem of setting an agenda for a sequential majority vote

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004

(identical to scheduling match-ups in a caterpillar tournament) such that a particular outcome wins [30]. Vu et al. initiated the investigation of the question of setting the schedule, subject to various sets of constraints on fairness and interesting-ness, often finding the problems computationally hard [51–53]. Williams found that, for certain dominant entrants in a cup tournament it is easy to find a schedule such that a particular entrant will win [56]. In a series of papers, Stanton and Williams have also shown that, for deterministic tournaments where the relation between players is generated from a known random prior, it is computationally easy to find a knockout tournament such that a preferred candidate will win [47–49]. In the uncertain information case, Hazon et al. [24] find that the problem of setting an agenda that maximizes a particular candidate's probability of winning is NP-hard. However, for cup tournaments with a known, fixed complete majority relation, the problem of finding a schedule such that a preferred entrant will win was recently shown to be NP-hard by Aziz et al. [3]. An important implication of this result for us is that there is no FPRAS for counting the number of tournaments in which a particular player wins. This count could be used as a proxy for player strength in a probabilistic setting.

Russell and Walsh find that the coalitional manipulation in sports tournaments is often computationally easy [43]. In Russell and Walsh's model, entrants may be a part of a coalition of manipulators that can choose to lose their matches. There is also an extensive body of work on the elimination problem for sports tournaments. In the elimination problem, the probability that team i beats team j is known, and the problem is to find the probability that a team is eliminated given a fixed sports season or playoff schedule [21,27]. Altman et al. [2] looked at a deterministic information variant of our problem where certain teams in a coalition could throw games in a sports competition and classified certain tournament choice rules as *pairwise non-manipulable*; where no pair of entrants could obtain a better outcome.

There has been surprisingly little work on stochastic models of tournaments, though there has been interest in stochastic models for voting rules. Perhaps the closest notion is that of a *possible winner* — a notion intrinsic to reasoning under uncertainty introduced by Konczak and Lang [28], with further studies by Lang et al. [29]. Recent studies of the complexity of computing possible winners include those by Lang et al. [30], Bachrach et al. [5], Conitzer et al. [11] and Xia et al. [57]. Possible and necessary winners have also been studied in the case of partial tournaments, where some matches are already decided [4]. These papers address complexity questions for voting rules when voters are defined by their (possibly incomplete) preference profiles over a set of outcomes. While these models are similar in spirit to the one under study, none of them consider pairwise probabilities for tournaments.

3. Definitions

In this section we provide the details of our problems. We precisely define the parameters of tournaments and the model that defines our decision problem. We also include a brief overview of the computational complexity classes that we will encounter in the next section. Our definition of a *tournament* is different from that of mathematics and economics; we use tournament as it is understood in modern sports and competitions. In mathematics, a tournament is a complete, directed graph [38] that can be realized from a set of votes [37], sometimes referred to as a majority relation [31]. While these definitions are similar to ours, the problems we want to capture are not explicitly representable with the current models (both in economics and in mathematics) and we therefore begin with the definition of our model.

3.1. Model definition

We begin by defining a model with which to reason about sports tournaments and other head-to-head competitions.

Consider a tournament with n entrants $\{e_1, \ldots, e_n\}$. Note that what distinguishes a tournament from a general voting rule here is that, for a given election with separate sets of candidates and voters, we require the

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004

set of candidates to be exactly the set of voters. Let $\mathbb{Q}_{[0,1]}^{n\times n}$ be the set of $n \times n$ matrices over $[0,1] \cap \mathbb{Q}$ (the rational numbers in the interval $[0,1]$). Let $p_{i,j}$ denote the probability that entrant e_i will defeat entrant e_j. Let

$$P = [p_{i,j}] \in \mathbb{Q}_{[0,1]}^{n\times n}.$$

We require that $p_{i,j} + p_{j,i} = 1$. We choose $k \in \mathbb{N}$ to discretize the interval $[0,1]$ into steps of $1/(k+1)$, we call this the *discretization level* of the problem. That is, each

$$p_{i,j} \in \{0,1\} \cup \{1/(k+1), 2/(k+1), \ldots, k/(k+1)\} \text{ for } i,j \in \{1,\ldots,n\}.$$

Taking $k = 0$ gives us the deterministic case where each match is either won or lost with certainty. As an input to a computational problem we assume k is given in unary.

We define the *discrete price table*, C_P, to have $n^2 - n$ rows (indexed by pairs i, j, $i \neq j$) and $k+2$ columns. For each entry in the table we have a positive integer value representing the cost to lower $p_{i,j}$ to a given probability (by bribing e_i). C_P gives the cost of bribing each match in per match bribery, and gives the cost of bribing an entrant in per entrant pair bribery. We assume that all entrants will compete to the best of their abilities and we cannot increase an entrant's probability of winning a particular match other than by bribing their opponent. Therefore we designate these entries as –. We also require that $c_{i,j}(p_{i,j}) = 0$. That is, it does not cost anything to have an entrant compete at its highest level. We assume bribery is non-adaptive, carried out before the first match is played. Outside actors may not have access to the entrants after the start of the competition (i.e., the tournament is taking place in a remote or tightly controlled location) so all bribery must be done beforehand.

In general we are given a rule f for determining which matches will be played in a given round. The value of f may depend on the results of matches played in previous rounds, but does not depend on the probability matrix P nor on the cost matrix C_P. We are also given a rule g for determining the winners of the tournament, depending on the results of all matches played, but not on the probability matrix or the cost matrix. Let $Pr[e_i|P,f,g]$ denote the probability that e_i wins the tournament defined by rules f and g and probability matrix P. For cup and challenge tournaments the rule f may be represented by a binary tree T with entrants labeled on the leaves to represent the order of the matches. In this case we write $Pr[e_i|P,T]$ for $Pr[e_i|P,f,g]$. In some problems we are also given a threshold t, an integer-valued budget B or a set M of manipulating entrants. Consider Example 3.1 which will serve as a running example through the text.

Example 3.1. Consider the following example with $k = 3$ (so there are $k + 2 = 5$ possible probability configurations) and $n = 4$ entrants. We use this as a running example. These parameters for k and n are given implicitly by the probability matrix P and cost matrix C_P.

$P =$

	e_1	e_2	e_3	e_4
e_1	–	0.75	0.50	0.25
e_2	0.25	–	0.25	0.25
e_3	0.50	0.75	–	0.25
e_4	0.75	0.75	0.75	–

$C_P =$

$c_{i,j}$	0.00	0.25	0.50	0.75	1.00
$c_{1,2}$	100	40	20	0	–
$c_{1,3}$	30	25	0	–	–
$c_{1,4}$	200	0	–	–	–
$c_{2,1}$	10	0	–	–	–
$c_{2,3}$	10	0	–	–	–
$c_{2,4}$	10	0	–	–	–
$c_{3,1}$	120	75	0	–	–
$c_{3,2}$	200	100	50	0	–
$c_{3,4}$	300	0	–	–	–
$c_{4,1}$	300	200	100	0	–
$c_{4,2}$	300	200	100	0	–
$c_{4,3}$	400	300	200	0	–

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004

We next define several types of tournaments. In each case we assume that no match ends in a tie and we have a unique winner for every match. In the deterministic case a scoring model which includes ties or does not normalize to certain forms for round robin tournaments and Copeland elections can have significant effects on the complexity of evaluation, manipulation, and bribery [16,27].

We begin by defining a general model for tournaments. Let $\mathscr{P}(U)$ denote the power set of a set U. If S is a subset of the set of pairs of entrants, an *outcome* for S is a specification of a winner for each match in S. We can formalize this as follows. If $S = \{(i_1, j_1), \ldots, (i_k, j_k)\}$ is a set of matches, an outcome for S is a set $\{(i_1, j_1, t_1), \ldots, (i_k, j_k, t_k)\}$ where each $t_m \in \{i_m, j_m\}$. An outcome is *possible* if it has positive probability according to probability matrix P. We denote by $\Omega(S)$ the set of outcomes of S, and by $\Omega^p(S)$ the set of possible outcomes of S. Let $D = \cup_S \Omega(S)$ be the set of outcomes of all sets of matches.

General tournament (GT): A *general tournament* on n entrants consists of

- an integer $n > 0$ giving the number of entrants (where convenient we denote the entrants by $e_1, \ldots, e_n$);
- a number r of rounds;
- a matrix $P = [p_{i,j}]$ of probabilities of winning pairwise events;
- a discrete price matrix C_P;
- a partial function $f : \{(n, d_1, \ldots, d_{i-1}) : n \in \mathbb{Z}^+, d_j \in D\} \to \mathscr{P}(W_n)$ (the *next round function*), where $W_n = \{(i, j) : 1 \leq i < j \leq n\}$; and
- a partial function $g : \{(n, d_1, \ldots, d_r) : n \in \mathbb{Z}^+, d_j \in D\} \to \mathscr{P}(\{1, \ldots, n\})$ (the *result function*).

The value of f is the set of matches to be played at round i given the outcomes of the matches in all previous rounds. It is a partial function because not all matches can occur at every round. The domain Δ of f is defined as follows. First, $\mathbb{Z}^+ \subseteq \Delta$. Suppose $(n, d_1, \ldots, d_{i-1}) \in \Delta$ and $d_i \in \Omega(f(n, d_1, \ldots, d_{i-1}))$. If $i < r$, then $(n, d_1, \ldots, d_{i-1}, d_i) \in \Delta$. If $i = r$, then $(n, d_1, \ldots, d_{i-1}, d_i)$ is in the domain of g. The value of g is the set of winners. Note that f and g do not depend on the probability matrix P and the cost matrix C_P.

Strictly speaking, we do not need the parameter n for a single tournament. However, we want to think more broadly about a *class of tournaments*. This is an infinite family Y of tournaments parametrized in part by a subset N of the natural numbers. For each integer $n \in N$ we have a set of tournaments with n entrants. For a fixed n, individual tournaments are determined by the probability matrix P and the cost matrix C_P. All tournaments in the family Y have a common next round function and a common result function, with n as a parameter. These functions must be polynomial time computable in n and the length of the remaining input parameters. The number of rounds must be bounded by a polynomial in n.

In some tournaments each pair of participants can meet in at most one match. We call such a tournament *unitary*. Challenge tournaments, cup tournaments, and round robin tournaments are unitary, but there do exist non-unitary tournaments in the sports world (e.g., the NCAA's (college) US college baseball world series is a double elimination tournament).

Bribery can be done in several ways. Bribery is *adaptive* if the bribes in one round can depend on the outcomes of the preceding rounds. Otherwise it is *nonadaptive*. In this paper we always assume bribery is nonadaptive.

There is a question as to the effect of a bribe when two participants are involved in two or more matches. One can consider either *per match bribery*, in which a bribe affects only a single match, and *per entrant pair bribery*, in which a bribe affects all matches between a pair of entrants. Note that in unitary tournaments, per match and per entrant pair bribery models coincide.

We consider the following classes of tournaments in this paper.

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004

Bounded tournaments: We say that a class Y of tournaments is *bounded* if the number of rounds r is bounded by a polynomial in n, the cardinality of $f(n, d_1, \ldots, d_{i-1})$ is bounded by a constant b for every $(n, d_1, \ldots, d_{i-1})$ in the domain of f or g, and the discretization level k is given in unary (so the input size is $\Theta(n^2 k)$).

Bounded history tournaments: We say that a class Y of tournaments has bounded history $b \leq r$ if the functions f and g depend only on the round number and the outcomes of the most recent b rounds where b is a constant. In this case we write

$$f(n, d_1, \ldots, d_{i-1}) = f(i; n, d_{i-b}, d_{i-b+1}, \ldots, d_{i-1})$$

if $i \geq b + 1$ and

$$f(n, d_1, \ldots, d_{i-1}) = f(i; n, d_1, d_2, \ldots, d_{i-1})$$

otherwise. The point here is that the set of matches to be played in round i depends only on the outcomes of the most recent $\min(b, i-1)$ rounds rather than on all previous rounds. Similarly $g(n, d_1, \ldots, d_r) = g(n, d_{r-b+1}, \ldots, d_r)$.

Challenge tournament (CT): In a challenge tournament $n-1$ matches are played. In the first match, e_1 plays e_2. In the second match, the winner plays e_3, and so on. The winner of the last match is the unique winner of the tournament. Thus CT is a bounded class of unitary tournaments with bounded history equal to one and with $n-1$ rounds.

Cup tournament (Cup): In a cup tournament we are given a complete binary tree T with entrants labeled on the leaves. The number n of entrants must be a power of 2.[4] Each internal node is decided by the competition between the two entrants on the level below. The unique winner is the entrant who reaches the top node of the tree. Cup tournaments are unitary, have bounded history one, and have $\log(n)$ rounds.

Round robin tournament (RR): In a round robin tournament (Copeland Scoring) each entrant plays every other entrant exactly once. In each match each entrant receives 1 point for a win and 0 points for a loss. A winner of the tournament is an entrant with the maximum number of points (there may be more than one winner of the tournament). An RR tournament is a single round unitary tournament, and, thus, has bounded history.

The World Cup of Soccer is an example of a tournament that fits the model of a general tournament but not of CT, Cup, or RR. There are two phases. First, four groups of four teams each plays a round robin tournament. This is round 1 of a GT. Then the two highest scoring teams from each four-team RR play an eight-team Cup tournament, the winner of which is the overall winner. This accounts for rounds 2, 3, and 4 of a GT. The GT has history 2. Who plays in round 4 depends only on who wins in round 3 (history 1). Who plays in round 3, however, depends on who wins in rounds 1 and 2 since the positions of the entrants in the Cup tree depend on the numbers of games won in the RR phase.

Also note that with a slight generalization of a GT we can model an entire NBA (American basketball) season. The generalization is to allow two entrants to play more than one match in a round. This can be modeled by allowing the value of the next round function f to be a multiset. Now a season, including playoffs, consists of a first round consisting of 82 matches per team (a sort of multi-RR) from which 16 teams advance to a sort of hybrid Cup/best of seven tournament. Each of eight pairs of teams plays a best of seven series (each modeled by seven rounds of a GT), the winners advancing to another best of seven series. Which teams play in the second best of seven series depends on performances in the "regular season",

[4] If not, we can add sufficiently many "fake" players who lose deterministically to all "real" players, so that the number is a power of 2.

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004

so the GT has history eight. (Which teams play in third and fourth best of seven series only depends on who won the previous ones.)

3.2. Probabilistic tournament bribery problems

With this model we can now define our problem and a set of related decision problems for a class Y of tournaments specified by a number of rounds r, a next round function f, and result function g.

Given: A natural number n, a probability matrix $P \in \mathbb{Q}_{[0,1]}^{n \times n}$ describing the behavior of a tournament of class Y, a preferred entrant e^*, and (where appropriate) a cost matrix C_P, a threshold t, a budget B, or a set of manipulators $M \subseteq \{e_1, \ldots, e_n\}$.

Questions: We define the following 6 decision problems with the above input:

Evaluation: Is the probability that e^* wins the tournament (the sum of the probabilities of the futures with e^* a winner) above t?

Pos-Win: Is e^*'s probability of winning the tournament above 0?

Pos-Win-$: Can we raise e^*'s probability of winning the tournament above 0 by bribing specific entrants according to C_P and not exceeding the budget B?

Constructive coalitional manipulation (CCM): Can we raise e^*'s probability of winning the tournament above t by strategically setting all the probabilities associated with a subset M (a *coalition*) of the entrants?

Constructive bribery: Can we raise e^*'s probability of winning the tournament above t by bribing specific entrants according to C_P and not exceeding the budget B?

Exact: Can we raise e^*'s probability of winning the tournament above t by bribing specific entrants according to C_P and spending exactly B?

We note that in the evaluation, pos-win, and CCM problems the bribery table C_P can be omitted from the specification of the problem.

Example 3.2. Recall our running example matrices, shown in Example 3.1.

$P =$

	e_1	e_2	e_3	e_4
e_1	–	0.75	0.50	0.25
e_2	0.25	–	0.25	0.25
e_3	0.50	0.75	–	0.25
e_4	0.75	0.75	0.75	–

$C_P =$

$c_{i,j}$	0.00	0.25	0.50	0.75	1.00
$c_{1,2}$	100	40	20	0	–
$c_{1,3}$	30	25	0	–	–
$c_{1,4}$	200	0	–	–	–
$c_{2,1}$	10	0	–	–	–
$c_{2,3}$	10	0	–	–	–
$c_{2,4}$	10	0	–	–	–
$c_{3,1}$	120	75	0	–	–
$c_{3,2}$	200	100	50	0	–
$c_{3,4}$	300	0	–	–	–
$c_{4,1}$	300	200	100	0	–
$c_{4,2}$	300	200	100	0	–
$c_{4,3}$	400	300	200	0	–

Consider a challenge tournament with the entrants ordered on T as illustrated in Fig. 2. For this graph we can evaluate the six questions presented in above. We assume that $e^* = e^1$ and we enumerate the other necessary parameters for the questions that require them.

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004

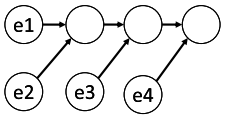

Fig. 2. Tournament graph (T) for Example 3.2.

Evaluation: For the decision problem we require a threshold; here we take $t = 0.08$. For a challenge tournament with e^* in the first position, computing the exact probability of winning is $Pr[e_1|P,T] = p_{1,2} \times p_{1,3} \times p_{1,4} = 0.75 \times 0.5 \times 0.25 = 0.09375$. Since this is greater than t this example is a "yes" instance.

Pos-Win: For the decision problem for pos-win we only need to examine the result of the evaluation procedure. Since $Pr[e_1|P,T] > 0$ then this example is a "yes" instance.

Pos-Win-$: For our running example $Pr[e_1|P,T] > 0$ without requiring any bribes and therefore is also a "yes" instance for pos-win-$.

Constructive coalitional manipulation (CCM): For this question let $M = \{e_4\}$ and $t = 0.15$. Since we are trying to raise $Pr[e_1|P,T] > 0.15$ the best thing to do is to set $p_{4,1} = 0$. This way the manipulator, e_4, loses to the preferred entrant with certainty. We can then evaluate $Pr[e_1|P,T] = p_{1,2} \times p_{1,3} \times p_{1,4} = 0.75 \times 0.5 \times 1.0 = 0.375$. Since $Pr[e_1|P,T] > 0.15$, this is a "yes" instance.

Constructive bribery: For this question let $t = 0.15$ and $B = 100$. By investigation we see there are only two meaningful options for bribes. We can bribe e_4 so that $p_{1,4} = 0.5$ or we can bribe e_2 and e_3 so that $p_{1,2} = 1.0$ and $p_{1,3} = 0.75$. Using the evaluation formula again we see that with the first option of bribes we have $Pr[e_1|P,T] = p_{1,2} \times p_{1,3} \times p_{1,4} = 0.75 \times 0.50 \times 0.50 = 0.1875$ with the second option of bribes we have $Pr[e_1|P,T] = p_{1,2} \times p_{1,3} \times p_{1,4} = 1.0 \times 0.75 \times 0.25 = 0.1875$. Both of these options give us the same value for $Pr[e_1|P,T]$ and, since both options are > 0.15, this is a "yes" instance.

Exact: For this question let $t = 0.15$ and $B = 100$. By investigation we see there are only two meaningful options for bribes and only one option that exactly spends our budget. Therefore, we bribe e_4 so that $p_{1,4} = 0.5$. Using the evaluation formula again we see that $Pr[e_1|P,T] = p_{1,2} \times p_{1,3} \times p_{1,4} = 0.75 \times 0.5 \times 0.50 = 0.1875$. Since $Pr[e_1|P,T] > 0.15$ and we have exactly spent our budget, this is a "yes" instance.

3.3. Complexity classes

Our goal in this paper is to classify the various problems related to bribery and probabilistic behavior of tournaments according to their computational complexity. As is the practice in complexity theory, we do so by locating these problems as precisely as possible in complexity classes. In this section we review the definitions of these classes. For a more complete treatment we refer the reader to [40].

P: A decision problem L is in the class P if it is decidable in polynomial time by a Turing machine.

NP: A decision problem L is in the class NP if it is decidable in polynomial time by a nondeterministic Turing machine.

PP: A decision problem L is in the class PP if there is a nondeterministic polynomial time Turing machine that on input x accepts in at least half the computation paths if $x \in L$ and accepts in fewer than half the computation paths if $x \notin L$.

PL: A decision problem L is in the class PL if there is a nondeterministic log space Turing machine that on input x accepts in at least half the computation paths if $x \in L$ and accepts in fewer than half the computation paths if $x \notin L$.

Oracle classes: Let X be a class of decision problems with a notion of adding a language L as an *oracle* for machines that recognize X. For example, if the class X is defined to be the class of problems decided

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004

Table 1
Complexity results for probabilistic tournament bribery problems. In some cases we have been unable to provide lower bounds, in these cases we note our upper bound results ($\in$).

	Challenge	Cup	Round robin
Evaluation	PL (Corollary 4.4)	P (Theorem 4.18)	$\in$ PP (Theorem 4.1)
Pos-Win	PL (Corollary 4.6)	P (Theorem 4.19)	P (Corollary 4.25)
Pos-Win-$	P (Corollary 4.8)	P (Theorem 4.20)	P (Theorem 4.24)
CCM	P (Corollary 4.10)	$\in$ NP (Theorem 4.21)	$\in \mathrm{NP}^{\mathrm{PP}}$ (Theorem 4.10)
Cons. bribery	$\in$ NP (Corollary 4.4)	$\in$ NP (Theorem 4.21)	$\in \mathrm{NP}^{\mathrm{PP}}$ (Theorem 4.27)
Exact	NP-C (Theorem 4.12)	NP-C (Theorem 4.22)	$\in \mathrm{NP}^{\mathrm{PP}}$ (Theorem 4.10)

by Turing machines with some particular constraints, then an oracle L can be added by allowing the Turing machines to make "black box" queries about membership of strings in L. If Y is a complexity class, then the class X^Y is the class of problems recognized by such oracle X-machines using an oracle $L \in Y$. If the number of oracle queries is bounded by t, then the resulting class is denoted $X^{Y[t]}$. The reader will encounter $\mathrm{NP}^{\mathrm{PP}}$ and $\mathrm{NP}^{\mathrm{PP}[1]}$ in this paper.

Completeness: If X is a class of decision problems, and $L \in X$, then L is *complete* for X if every problem in X polynomial time[5] mapping reduces to L. The class of languages that are complete for X is denoted X-C. A statement that a language is complete for a class is a statement that we have identified the complexity of the problem as tightly as possible. Thus,

$$\mathrm{PL} \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PP} \subseteq \mathrm{NP}^{\mathrm{PP}}.$$

4. Complexity results

We proceed through the results in order of tournament type. This allows us to present the results in a coherent fashion as, in many cases, the proofs and algorithms build on one another. Table 1 provides an overview of our complexity results on tournaments.

4.1. General tournaments

In this section we give complexity results that apply to all classes of tournaments.

Theorem 4.1. *Let Y be any class of general tournaments such that the number of rounds r is bounded by a polynomial in n. Then evaluation for Y is in* PP.

Proof. Suppose we're given an evaluation problem with probabilities expressed with precision k. Each probability $p_{i,j}$ can be written as $n_{i,j}/k$, where $n_{i,j}$ is an integer. Let t be the given threshold for e^*'s winning probability.

We create a nondeterministic polynomial time algorithm, M, as follows. M simulates a tournament with n entrants and r rounds.

- For each l, $1 \leq l \leq r$, suppose we have determined outcomes $d_1, \ldots, d_{l-1}$ so that $(n, d_1, \ldots, d_{l-1}) \in \Delta$. For each $(i,j) \in f(n, d_1, \ldots, d_{l-1})$, guess $n \in \{0, 1, \ldots, k-1\}$. If $n < n_{i,j}$ then i wins, otherwise j wins. This determines an outcome d_i for $f(n, d_1, \ldots, d_{l-1})$.
- At the end of the computation, if $e^* \in g(n, d_1, \ldots, d_r)$, accept; otherwise, reject.

[5] It is possible to consider other resource bounds, but here we only consider polynomial time.

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004

Let m be the number of computation paths, which we can compute as follows. At each step we have a k-way branch, so $m = k^r$. Therefore, $|m| = r \log(k)$, which is polynomial in n and the length of k. Thus, m is computable in $\log(r)$ multiplications of integers whose length is bounded by a polynomial of the input size. By a standard trick the threshold tm can be replaced by $1/2$ [45, p. 92ff].

Participant e^* wins the tournament if and only if e^* wins in $\geq tm$ of the paths. Therefore, if Y is any class of general tournaments such that the number of rounds r is bounded by a polynomial in n, evaluation for Y is in PP. □

The upper bound on the complexity of evaluation given in Theorem 4.1 can be extended immediately to the Constructive Bribery, CCM, and Exact Bribery problems as well.

Theorem 4.2. *Let Y be any class of general tournaments such that the number of rounds r is bounded by a polynomial in n. Suppose we use either per match bribery or per entrant pair bribery. Then CCM, Constructive Bribery, and Exact Bribery for Y are in* $\mathrm{NP}^{\mathrm{PP}}$.

Proof. We can construct an $\mathrm{NP}^{\mathrm{PP}}$ machine that, given a tournament T and threshold t, nondeterministically guesses a set of bribes, checks that the bribes total at most the budget, and evaluates T with a single call to a PP oracle. Since the number of rounds is polynomially bounded, a set of bribes can be guessed in polynomial time. We accept if the probability of e^* winning is $> t$. □

Note that we have actually shown these problems are in $\mathrm{NP}^{\mathrm{PP}[1]}$.

4.2. Bounded tournaments and challenge tournaments

In this section we consider tournaments with various bounds on their parameters.

Theorem 4.3. *Suppose Y is a class of bounded tournaments with bounded history. If f and g are log space computable, then evaluation for Y is in* PL.

Proof. The same proof as in Theorem 4.1 works with the observation that under the given hypotheses the space complexity is logarithmically bounded. □

In fact the same result holds if we let the length of the history be $\mathcal{O}(\log(n))$.

Corollary 4.4. *Evaluation for* CT *is in* PL.

Now that we have determined that it is computationally easy to evaluate the result of a given instance of CT, we move our attention to the possible winner problem.

Theorem 4.5. *Suppose Y is a class of bounded tournaments. Pos-Win for Y is in* PL.

Proof. Run the evaluation algorithm given in Theorem 4.3 with $t = 0$. Note that e^* is a possible winner if and only if $Pr[e^*|P, f, g] > 0$. □

Corollary 4.6. *Pos-Win for* CT *is in* PL.

Theorem 4.7. *Suppose Y is a class of bounded tournaments with bounded history. Then Pos-Win-$ for Y using per match bribery is in* P.

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004

Proof. Let b be a common bound on the history and the number of matches in each round. We provide a polynomial time dynamic programming algorithm to compute the minimum cost such that e^* has a non-zero probability of winning the tournament. For each round u and each b-tuple of subsets $S_1, \ldots S_b \subseteq \{(i,j) : 1 \leq i < j \leq n\}$ with $|S_v| \leq b$ for each v, and d_v a feasible outcome for S_v, we compute the minimum cost $min\$(i, S_1, \ldots, S_b, d_1, \ldots, d_b)$ needed to make positive the probability that S_v is the set of matches at round $u - v + 1$, $v = 1, \ldots, b$ with outcome d_v (if $u < b$ we must truncate the list of S_vs).

Recall from our definition of general tournaments that W_n is the function of next round games. For $S \subseteq W_n$ and $d \in \Omega(S)$, let $\kappa(S,d)$ be the cost of bribery to make $d \in \Omega^p(S)$. If no such bribery is possible, let $\kappa(S,d) = B + 1$. We have

$$\begin{aligned} min\$(i, S_1, \ldots, S_b, d_1, \ldots, d_b) = min\{ & min\$(i-1, S_2, \ldots, S_{b+1}, d_2, \ldots, d_{b+1}) + \kappa(S_1, d_1) : \\ & S_{b+1} \subseteq W_n, |S_{b+1}| \leq b, d_{b+1} \in \Omega(S_{b+1}), \text{ and} \\ & S_1 \in f(i; n, d_2, \ldots, d_{b+1})\}. \end{aligned}$$

This recursive formula leads directly to a dynamic programming algorithm. The time complexity of the algorithm is the number of table entries times the cost of computing each table entry.

Note that if $S \subseteq W_n$ with $|S| \leq b$, then the number $2^{|S|}$ of outcomes from matches in S is bounded. There are polynomially many choices for S_{b+1} and for each choice there is a bounded number of d_{b+1}. So we are minimizing over a polynomial size set. For each S_{b+1}, d_{b+1}, there are polynomially many ($\mathcal{O}(k^b)$) combinations of bribes to check to see whether d_{b+1} can be made possible. It follows that it takes polynomial time to update each table entry. Moreover, the number of table entries is in $\mathcal{O}(n(n^2)^{b^2} 2^{b^2})$.

Finally, if for some $S_1, \ldots, S_b, d_1, \ldots, d_b$, we have $min\$(r, S_1, \ldots, S_b, d_1, \ldots, d_b) \leq B$ and $e^* \in g(r; n, d_1, \ldots, d_b)$, then accept. Otherwise reject. □

A similar idea could be tried for per entrant pair bribery, but this seems to fail. Since a bribe of a match at an early round affects the probability distribution of the outcome of a match with the same entrants in a later round, it seems that to find $min\$(i, S_1, \ldots, S_b, d_1, \ldots, d_b)$ we must know the updated probability matrix for each $min\$(i-1, S_2, \ldots, S_{b+1}, d_2, \ldots, d_{b+1})$. There may be many patterns of bribery that all lead to the minimal cost to make $S_2, \ldots, S_{b+1}, d_2, \ldots, d_{b+1}$ possible. In fact, the minimal cost bribe that makes $S_1, \ldots, S_b, d_1, \ldots, d_b$ possible might not arise from extending a minimal cost bribe that makes $S_2, \ldots, S_{b+1}, d_2, \ldots, d_{b+1}$ possible since an earlier excessive bribe for a pair of entrants may save later bribes. It seems that we must keep track of all possible patterns of bribery that make each $S_2, \ldots, S_{b+1}, d_2, \ldots, d_{b+1}$ possible at round $i - 1$. The number of such patterns is $\Omega(k^{n^2})$, which is super polynomial.

Corollary 4.8. *Pos-Win-$ for* CT *is in* P.

Recall that in the coalitional manipulation problem we are given a set $M \subseteq E$ of members of the manipulating coalition. Observe that the coalitional manipulation problem is a special case of the bribery problem in our model. Specifically, in a coalitional manipulation instance, $B = |M|$ and the cost of bribing members of M is also 1 (unit priced). The entrants $E \setminus M$ have bribery costs equal to ∞ (so they cannot be changed in this scenario).

Theorem 4.9. *Suppose Y is a class of bounded tournaments with bounded history. Then Constructive Coalitional Manipulation for Y is in* P.

Proof. Let b be a common bound on the history and the number of matches in each round. Since there are no budget-related resource bounds in this problem we can solve this problem by maximizing e^*'s probability of winning by setting the entries for each $m \in M$ in the probability matrix P. We refer to the setting of

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004

M as a *strategy*. In this instance, C_P gives the available options of probability values for each $m \in M$ even though there are only unit prices.

We construct a strategy round by round in reverse order. The strategy for a round amounts to setting probabilities for the matches at that round. The choices for setting these probabilities depend on C_P, which participants are in M, and the initial probabilities for the matches. The effect of a strategy for a given round depends on the outcomes of the preceding b rounds since the set of matches at the current round depends on these outcomes. The algorithm can be described as follows.

Repeat for: $j = r$ downto 1

For: each $(d_{j-b}, \ldots, d_{j-1})$ with each $|d_i| \leq b$

$S_j = f(j; n, d_{j-b}, \ldots, d_{j-1})$

For: each choice of a strategy for S_j

Evaluate the resulting $r - j + 1$ round tournament

Choose the strategy that maximizes the probability that e^* wins

There are at most $(k+1)^b$ ways to choose a strategy for S_j, there are

$$2^{b^2} \binom{\frac{n(n-1)}{2}}{b}^b \in \mathcal{O}(2^{b^2} n^{2b^2})$$

ways to choose $(d_{j-b}, \ldots, d_{j-1})$ at round j, and there are r rounds. Thus the time complexity is $\mathcal{O}(k^b 2^{b^2} n^{b^2} r)$. This is polynomial in the input size since the input size is $\Theta(n^2 k)$, b is constant, and r is polynomial in n. □

Corollary 4.10. *Constructive Coalition Manipulation for* CT *is in* P.

Corollary 4.4 gives us membership in NP for the constructive bribery problem since we can verify a bribery plan in polynomial time. Theorem 4.10 shows that, for an instance of CCM, we can achieve optimal manipulations in polynomial time. However, our results for CCM do not extend in a straightforward manner to the situation of priced bribery.

The simple answer would be to apply a *bang-for-the-buck* greedy algorithm to iteratively choose the cheapest available single bribe which results in the largest change in

$$Pr[e^*|P,T] / \text{cost of bribe}.$$

However, this will not reach an optimal solution in all cases. This can be illustrated by an example of CT. Consider the following example over four entrants.

Example 4.11. Assume that $e^* = e_1$ and the challenge tournament graph, T, is the same as shown in Fig. 2 (where e_1 plays every match). Assume we have $k = 3$ and P and C_P given below with $B = 20$. For simplification, we omit the bribery cost for all matches that do not contain e^*.

$P =$

	e_1	e_2	e_3	e_4
e_1	–	0.25	0.25	0.25

$C_P =$

$c_{i,j}$	0.00	0.25	0.50	0.75	1.00
$c_{2,1}$	13	13	13	0	–
$c_{3,1}$	25	10	10	0	–
$c_{4,1}$	25	10	10	0	–

We know that when e^* is involved in every match, the winning probability is given by: $Pr[e_1|P,T] = p_{1,2} \times p_{1,3} \times p_{1,4} = 0.25 \times 0.25 \times 0.25 = 0.015625$. We set the cheapest bribe of entrant e_2 to change

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004

$p_{1,2} = 1.0$ with cost 15. This would give $Pr[e_1|P,T] = p_{1,2} \times p_{1,3} \times p_{1,4} = 1.0 \times 0.25 \times 0.25 = 0.0625$, giving a change in probability of 0.046875 for a cost of 13 or a bang-for-the-buck ratio of 0.0036058. Looking at either a bribe of e_3 or e_4 we would change $p_{1,3} = 0.75$ with a cost of 10. This would give $Pr[e_1|P,T] = p_{1,2} \times p_{1,3} \times p_{1,4} = 0.25 \times 0.75 \times 0.25 = 0.046875$, giving a change in probability of 0.03125 for a cost of 10 or a bang-for-the-buck ratio of 0.003125. This would mean that according to the bang-for-the-buck heuristic, bribing e_2 is best single action.

However, looking at our budget we see that we can afford to bribe both $p_{1,3} = p_{1,4} = 0.75$ for a combined cost of 20. This would give $Pr[e_1|P,T] = p_{1,2} \times p_{1,3} \times p_{1,4} = 0.25 \times 0.75 \times 0.75 = 0.140625$. This is, by far, the best action and shows that a simple single action greedy algorithm would make non-optimal decisions in this case.

If we require our outside manipulator to spend exactly the allocated budget then we can show NP-completeness for challenge tournaments. Situations which require an organization to spend a budget exactly occur implicitly in many large organizations and governments. For resources other than money that could be used, such as referees in sports tournaments or canvassers in political elections, there is sometimes the expectation of exact allocations.

We introduce the Subset Sum Problem for our next proof. The statement of the problem is from Garey and Johnson's book [19].

Name: Subset Sum
Given: A set $W \in \mathbb{Z}_+$, $w_1, \ldots, w_m$, and a target $S \in \mathbb{Z}_+$.
Question: Does there exist a subset $I \subseteq \{1, \ldots, m\}$ such that $\sum_{i \in I} w_i = S$?

This problem was shown to be NP-complete via a transformation from Partition by Karp [26]. Using this problem we can exactly classify the complexity of the exact bribery problem for CT.

Theorem 4.12. *Exact Bribery for* CT *is* NP*-complete.*

Proof. Membership in NP is an immediate consequence of Theorem 4.4 and a guess and check algorithm.

To show NP-hardness we provide a reduction from Subset Sum. For a given Subset Sum instance we set up a challenge tournament with S entrants such that e^* always wins (i.e., e^* has a 100% chance of winning all matches against all entrants) and we let $t = 0$, $B = S$, and $k = 0$. We then create the bribery cost matrix C_P with prices equal to the weights of $w_1, \ldots, w_m$. In this situation bribery will have no effect and we must distribute the money exactly. We have now established a polynomial time mapping such that there will be an exact bribery if and only if there is a subset such that $\sum_{i \in I} w_i = S$. □

4.2.1. Challenge the champ tournaments

In order to eliminate as many complications of the tournament graph as possible, we consider the case of a *challenge-the-champ (CTC) tournament.* This is the subclass of CT in which $e^* = e_1$, so that e^* is potentially involved in every match. We think of e^* as taking on all challengers.

Theorem 4.13. *There is a Constructive Bribery algorithm for* CTC *that runs in polynomial time with respect to B and n.*

Proof. In CTC tournaments $e^* = e_1$ and is in the first match. Our goal is to maximize the probability that e^* will win the tournament within budget through bribes to each of the other entrants in the tournament. The idea of this proof is to consider bribes on only a subset of the entrants. At each step we add an entrant $x \in \{e_2, \ldots, e_n\}$ and show a dynamic programming update that will compute the cheapest bribe for the new subset of entrants until we have considered the set of all the entrants.

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004

For an instance of CTC let F_x be the probability that e^* is still in the tournament after round $x-1$, namely that e^* has beaten entrants $e_2, \ldots e_x$. Without bribery, $F_{x+1} = F_x \cdot p_{1,x+1}$.

Let $M[u,b]$ be defined to be the maximum we can make F_u using a bribery budget of b and bribes to entrants $1, \ldots, u$. We can compute M in polynomial time by dynamic programming.

Initialize $M[1,b] = 1$ for all b.

Let $Q[x,b]$ be the maximum value $p'_{1,x}$ obtainable with bribe b to entrant x. The table Q can be computed, for all $b < B$, in time $\mathcal{O}(Bn)$ by table lookup using the cost matrix C_P.

Let $M[x+1,b] = \max_d\{M[x,b-d] \cdot Q[x+1,d]\}$. Note that d ranges from 0 to $b \leq B$, so computing each entry in M takes time $\mathcal{O}(B)$. There are $\mathcal{O}(Bn)$ entries in the table M. Thus, computing the table can be done in time $\mathcal{O}(B^2 n)$, polynomial in B and n.

Finally, observe that the maximum probability achievable for e^* to win the tournament, given budget B, is $M[n,B]$. □

Given that we are minimizing a sum (cost) and maximizing a product (joint probability), it seemed obvious that we should consider the logarithms of the probabilities, in order to reduce the problem to Knapsack, for which there are approximation schemes and pseudo-polynomial time algorithms. However, this turned out to be less than obvious: logarithms must, a priori, be truncated (unless k is a power of 2), and it was not clear to us that polynomially many bits of $\log(i/k^r)$ would suffice to give us the correct answer.

The problem is in P if the budget, B, is polynomial in the size of the input (e.g., B is input in unary). Otherwise, the algorithm is pseudo-polynomial. We can, in fact, say something stronger about the complexity of Constructive Bribery for CTC.

Theorem 4.14. *If the budget B is written in unary, Constructive Bribery for* CTC *is in PL.*

In order to prove this, we use results about the complexity of Markov decision processes (MDPs). An MDP is a tuple $M = \langle S, A, T, R\rangle$, where S is a finite set of states,[6] A is a finite set of actions, T is a probabilistic transition function from state-action pairs to states, and R is the reward/cost function, mapping from state-action pairs to real numbers. (Often, and here, we define a reward function from states to real numbers and a cost function from state-action pairs to real numbers; the function R encompasses both.) A policy for an MDP is a mapping from states to actions.

Mundhenk et al. [39] defined the *policy evaluation problem* for MDPs to be: Given an MDP M, horizon h (the number of state changes to be performed; assumed to be polynomial in $|M|$) and policy π, is the expected finite horizon value of π greater than 0? The *policy existence problem* for MDPs takes inputs MDP M and horizon h and asks if there exists a policy π for M and h with expected value greater than 0. They showed that these problems are in PL.

Proof. We give a logspace reduction from Constructive Bribery for CTC to the MDP policy existence problem.

Let T be a CTC tournament with entrants $e^* = e^1, \ldots, e^n$, probability matrix P, cost matrix C_P, target probability t, and budget B. Note that $P[1,i] = p_{1,i}$ is the probability that e^* beats e_i. Let k be the discretization parameter for P and C_P.

We create an MDP M with states $(S' \times D) \cup \{u, v\}$, where $S' = \{s_1, \ldots, s_n\}$ and $D = \{0, 1, 2 \ldots, B\}$. For $1 \leq i \leq n$, M is in state $\langle s_i, d\rangle$ if e^* has beaten $s_2, \ldots, s_i$ spending d on bribes. State u represents e^* winning the tournament, and state v represents e^* losing the tournament. The initial state is $\langle s_1, 0\rangle$.

The reward function is $R(u) = B/t$, $R(v) = 0$, and for all s_i, d, $R(\langle s_i, d\rangle) = 0$. The horizon h equals n.

[6] For our purposes, MDPs have finite state sets.

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004

There are $(n-1)(k+1)$ actions $a[i,j]$, with $1 \leq i \leq n-1$ and $0 \leq j \leq k$, one for each entry in C_P. Action $a[i,j]$ corresponds to e^* being in a match with e_{i+1}, with e_{i+1} bribed $C_P[i,j]$. For $i < n$, there is a cost of $C_P[i,j]$ for taking $a[i,j]$.

If $a[i,j]$ is taken in state $\langle s_i, d\rangle$ and $d + C_P[i,j] \leq B$, then the probability that M transitions to $\langle s_{i+1}, d + C_P[i,j]\rangle$ is the probability that e^* beats e_i, given the bribe corresponding to $C_P[i,j]$. Otherwise, M transitions to a sink state, v. Choosing $a[i,j]$ in a state $\langle s_m, d\rangle$ for $m \neq i$, or for $d + C_P[i,j] > B$, leaves the state unchanged.

There is also an action b. If action b is taken in state $\langle s_n, d\rangle$, then the next state is u. The cost is $B - d$.

Note that the total cost, c, of actions is bounded by B, and equals B if state u is reached. The total cost of reaching u is always exactly B. The expected reward for a policy π is pB/t minus the cost B, where p is the probability that M reaches state u. Thus the expected reward is $pB/t - B$. This is positive if and only if $p > t$.

Thus, there is a policy for M with expected positive reward if and only if there is a set of bribes with cost $\leq B$ such that the probability of e^* winning the tournament, given those bribes, is greater than t.

The reduction from Constructive Bribery for CTC to the MDP evaluation problem can be computed in logspace, so Constructive Bribery for CTC $\in$ PL when B is written in unary. □

Theorem 4.13, with the insights listed in this section, leads us to the conjecture that Constructive Bribery for CTC is hard when B is written in binary. Many well known NP-hard problem like Knapsack admit pseudo-polynomial time dynamic programming algorithms [19]. In fact, we require a significant modification to the tournament input in order to achieve NP-completeness results.

4.2.2. Simplified challenge tournaments

Definition 4.15. The Simplified Challenge Tournament problem (SC) is a modification of the Challenge Tournament problem where probabilities are (negative) powers of 2, bribes raise probabilities by multiplying by powers of 2 (up to 1), the costs of bribes are represented as a list of costs for raising the winning probability by factors of 2, and $e^* = e_1$.

Consider a Simplified Challenge Tournament. For each i, $2 \leq i \leq n$, the probability that $e^* = e_1$ beats e_i is 2^{-q_i} for some natural number q_i. Then the probability that e^* wins the tournament is 2^s where $s = -\sum_i q_i$.

Definition 4.16. An instance of the Knapsack Problem is a set of m items, with weights $w_1 \ldots w_m$ and values $v_1, \ldots, v_m$ and targets W and V, such that there is a set $I \subseteq \{1, \ldots, m\}$ such that $\sum_{i \in I} w_i \leq W$ and $\sum_{i \in I} v_i \geq V$.

Theorem 4.17. Knapsack $\leq_m^P$ *Constructive Bribery for* SC. *Hence Constructive Bribery for* SC *is NP-complete.*

Proof. Let $K = \langle w_1 \ldots w_m, v_1 \ldots, v_m, W, V\rangle$ be an instance of Knapsack. We construct an instance of Constructive Bribery for SC with $n = m + 1$ entrants as follows.

Let $k = \max_i\{v_i\} + 1$. Let $B = W$. Let $e^* = e_1$. For each $2 \leq i \leq m + 1$, let $p_{1,i} = 2^{-k}$, so a priori, the probability that e^* wins is 2^{-mk}. Let $t = 2^{-s}$, where $s = mk - V$.

For each i, there are two costs associated with bribing e_i: a cost of w_i for raising the probability that e^* beats e_i to 2^{-k+v_i}, and a cost of $B + 1$ for raising that probability any higher. Thus, there is at most one affordable bribe available for each entrant.

If there is a set I such that $\sum_{i \in I} w_i \leq W$ and $\sum_{i \in I} v_i \geq V$, then the cost of bribing all e_i, $i \in I$, is within budget $B = W$, and the probability that e^* wins, after bribery, is at least $t = 2^{-s}$, where $s = mk - \sum_{i \in I} v_i$.

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004

If there is a set of bribes S that allows e^* to win with probability at least t, that set can contain at most one bribe per entrant, and must bring the winning probability from 2^{-mk} to $2^{-(mk-V)}$ or higher. This means that the sum of the increases over the bribed entrants must be at least V, with cost at most W, giving a solution to K. □

4.3. Cup rule tournaments

In the definition of CUP we assume that each instance is a complete tree. We could consider more general tree based tournaments but do not do so in this paper. Note that we could try to reduce tournaments based on general trees to CUP tournaments by adding entrants who lose to all other entrants and cannot be bribed. However, complexity bounds may not be preserved since the number of entrants that must be added may be super polynomial in the number of entrants in the given tournament.

Theorem 4.18. *Evaluation for* CUP *is in* P.[7]

Proof. To prove this we construct a polynomial time algorithm that computes $Pr[e_i|P,T]$ for all entrants e_i. We construct a table of size $n \cdot \lceil \log_2(n) \rceil$ and use dynamic programming to compute for each i the probability, $L_{r,i}$, that entrant i advances to round r (with $r = 1$ being the first match e_i competes in). When we get to the last round, our table will contain the probability that each entrant won.

Let $G(r,i)$ be the set of entrants that e_i may face in round r.
for $i = 1$ to n **do**
  $L_{1,i} = 1$
end for
for $r = 2$ to $\lceil \log_2(n) \rceil$ **do**
  for $i = 1$ to n **do**
    $L_{r+1,i} = L_{r,i} \cdot \sum_{x \in G(r,i)} p_{i,x} \cdot L_{r,x}$
  end for
end for

At each stage, r, the numerators and denominators of the probabilities are in $\mathcal{O}(k^r) \subset \mathcal{O}(k^n)$, having sizes $n \log(k)$ (where k is the discretization level). Each e_i potentially competes against each e_x in exactly one round, so there are exactly $n(n-1) \in \mathcal{O}(n^2)$ multiplications in total. Thus the total time is in $\mathcal{O}(n^2(n \log(k))^2) = \mathcal{O}(n^4 \log(k)^2)$. □

Turning again to the possible winner problem variants we see that evaluating whether an entrant has a chance of winning is feasible for cup tournaments as well.

Theorem 4.19. *Pos-Win for* CUP *is in* P.

Proof. Run the evaluation algorithm given in Theorem 4.18 with $t = 0$. □

Theorem 4.20. *Pos-Win-$ for* CUP *is in* P.

Proof. We provide a polynomial time dynamic programming algorithm to compute the minimum cost such that e^* has a non-zero probability of winning the tournament. In particular, for each e we construct a vector V_e such that $V_e(L)$ is the minimum cost of guaranteeing a non-zero probability that e is still in the tournament at level L of the game tree. The answer we want is $V_{e^*}(\log_2(n))$.

[7] The same result with a slightly more optimized algorithm is shown in [51] and [24].

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004

Let the tournament graph T be a complete binary tree. Let L be a level of the binary tree starting with 0 at the bottom level. For each entrant e_i we then create a vector, V_{e_i}, of size $\log_2(n)$ with $V_{e_i}(0)$ initialized to 0 (the cost to get to the 0th level). Let $V_{e_i}(L)$ be the minimum cost for e_i to have a non-zero probability of winning its Lth contest. We construct the **minimum cost matrix**, $min\$$, an integer matrix of size n^2, as follows: $min\$_{ij} = 0$ if i starts with a non-zero probability to beat j and the minimum bribe cost $min\$_{ij} = c_{ij}(1/(k+1))$ otherwise.

for $L = 1$ to $\log(n)$ **do**
  for all $e_i \in T$ **do**
    Let $G(e_i, L)$ be the set of entrants in the sub-tree at level L containing e_i.
    $V_{e_i}(L) = V_{e_i}(L-1) + \min_{x \in G(e_i,L) - G(e_i,L-1)}(min\$_{i,x} + V_x(L-1))$
  end for
end for

At the end of execution $V_{e_i}(\log_2(n))$ will contain the minimum cost to promote e_i to the top level of T with a non-zero winning probability. We can compare this cost with B and accept if $V_{e^*}(\log_2 n) \leq B$, else reject. □

Theorem 4.18 immediately provides us with the following corollaries through standard guess and check algorithms.

Corollary 4.21. *CCM and Constructive Bribery for* Cup *are in* NP.

We have not found lower bounds for CCM or constructive bribery for Cup. We can prove NP-completeness for the exact case directly.

Theorem 4.22. *Exact Bribery for* Cup *is* NP*-complete.*

Proof. To show membership in NP we can nondeterministically guess a bribery scheme and check that it utilizes the entire budget. We can then evaluate the tournament in polynomial time by Theorem 4.18 to see that e^*'s winning probability exceeds the given threshold.

To show NP-hardness we can use a reduction similar to the one used in Theorem 4.12. Recall that S is the target sum in a Subset Sum instance. In this construction we choose a number, j, such that $j^2 \geq |S|$. We build our cup instance with j entrants. We set the added entrants' bribery prices to be $> B$ so that these entrants can be safely ignored. We then use the same mapping presented in the proof of Theorem 4.12 to show Exact Bribery is NP-complete. □

Theorem 4.23. *If $k = 0$ (deterministic version), then Constructive Bribery for* Cup *is in P via a dynamic programming algorithm.*

Proof. Let $\text{MinCost}[i, r]$ be the minimum cost in the deterministic Cup Bribery instance for e_i to win a match at round r. Note that $\text{MinCost}[i, 0] = 0$ for all i.

Let $ST(i, r)$ be the set of non-e_i entrants in e_i's subtree of height r, namely the set of entrants e_i might play at round r. Let $C(i, j)$ be the cost to bribe e_j to lose to e_i. Note that, if e_i already loses to e_j, the cost is 0. Let $H = \log_2(n+1)$ be the height of the tree, i.e., the number of rounds in the tournament.

We can compute

$$\text{MinCost}[i, r+1] = min_{j \in ST(i,r)}\{\text{MinCost}[j, r] + C(i, j)\} + \text{MinCost}[i, r].$$

Thus, it takes time $\mathcal{O}(n)$ to compute each entry of MinCost, and there are $\mathcal{O}(n)$ entries needed to find $\text{MinCost}[e^*, H]$, the minimum cost for e^* to win the entire tournament. □

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004

4.4. Round robin tournaments

Round Robin tournaments are of particular interest in the social choice community because of their close correspondence to the Copeland election system. Faliszewski et al. [16] provided a comprehensive study of bribery and manipulation of Copeland elections under deterministic scenarios. In addition, many papers on sports elimination, including those by Gusfield and Martel [21] and Kerns and Paulusma [27], study sports round robin tournaments for particular competitions (FIFA, Major League Baseball etc.).

We conjecture that the evaluation problem for round robin tournaments is PP-hard. Hazon et al. showed that the problem is PP-hard for the voting system Copeland with an imperfect information model [22]. However, as mentioned in the related work section the result from Hazon et al. does not immediately transfer. Likewise Gusfield and Martel showed a similar problem of determining the probability that a team is eliminated from a sports competition is PP-hard [21]. Their result does not hold in our context as their proof requires the ability to construct arbitrary schedules.

Both variants of the possible winner problem for round robin tournaments have polynomial time algorithms. While at first glance this may seem odd, it is because answering the question of possible winner does not require enumerating all possible outcomes of the tournament. Instead, to answer the possible winner problem we only need to find one configuration of the matches in which e^* wins.

The case of possible winner with access to bribes is somewhat more complicated for round robin tournaments. We begin with this problem because, as we show, the possible winner problem without access to bribes can leverage the same algorithmic construction as the possible winner with access to bribes problem. Recall the Minimum Cost Feasible Flow problem [1].

Name: Minimum Cost Feasible Flow
Given: A graph $G(V, E)$ with vertices V, edges E, source node s, and sink node t. Each $e \in E$ has flow, capacity, and cost. An edge labeled "$[x, y], c$" has capacity y and requires flow of at least x with cost per unit c. Cost is accumulated on a per unit flow basis (i.e., if the flow on an edge is 2 and $c = 3$ then the total cost of the edge is 6).
Question: Does there exist a flow from s to t such that the flow into each node is the same as the flow out, all minimum edge flows are satisfied, no maximum edge flows are violated, and cost is minimal?

There is a polynomial time algorithm for this problem and its solution (when the constraints are integers) is integral [1].

Using minimum cost feasible flows, Russell and Walsh provided an algorithm to compute minimal constructive manipulations for deterministic round robin tournaments when the number of matches that e^* can win is fixed [43]. Their algorithm did not allow for bribery (we do not bound the number of manipulable matches) and assumed that the number of matches that e^* can win is fixed. Likewise, Faliszewski et al. showed manipulation results for Copeland under deterministic information elections using feasible flows [16] and a similar construction was used by Faliszewski for the nonuniform bribery case (bribery where each voter may have different prices for changing their votes) [13]. While our construction is similar to theirs (and to an earlier constructions from Gusfield and Martel [21] and Ford and Fulkerson [18]) there are significant differences between the constructions. Our results are for a more flexible model, which is able to encapsulate bribery and manipulation, and works for even or odd numbers of entrants in the tournament.

Theorem 4.24. *Pos-Win-$ for* RR *is in* P.

Proof. We give a polynomial time reduction of an instance of Pos-Win-$ to a minimal cost feasible flow problem. Our reduction constructs a set of minimum cost feasible flow instances; the minimum cost bribery corresponds to the cost of the flow in one of the polynomially many graphs we construct. There is a

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004

polynomial time algorithm for finding the solution to a minimum cost feasible flow problem [1]. We build this sequence of graphs so that, in each graph, if there is a way to bribe the entrants such that e^* can be made a possible winner, the cost of the bribery scheme will be equal to the cost of the minimal cost flow in that graph. We build a graph for each possible winning score e^* could have. There will be at most $n/2$ graphs built in this proof and therefore we can answer Pos-Win-$ for RR in time polynomial in the size of the input.

We first construct the *minimum cost matrix*, $min\$_{i,j}$, as in the proof of Theorem 4.19. We also construct a domination graph D, which is a complete directed graph that has the set of entrants as nodes and each directed edge goes from a winning entrant to a losing entrant. We construct D by placing a directed edge from all entrants that are more likely to win to those less likely to win. While constructing D, we set the deterministic outcome of all matches with deterministic bribery cost involving e^* such that e^* is a winner (maximizing its possible wins). Once we set all the nondetermined matches in favor of e^* we let $t = outDeg(e^*)$. In this instance t represents the total number of matches that e^* can possibly win without making any bribes. In order for e^* to be a possible winner, we need to bribe between 0 and $n-t-1$ entrants to lose to e^* (as well as possibly influencing the outcomes of other matches).

For each i from 0 to $n-t$ we construct a feasible flow graph that tells us whether e^* can be made a possible winner by bribing exactly i matches involving e^*. We iterate over these graphs until we find a situation where e^* is a possible winner. Intuitively, the graph selects the cheapest bribes in order to make e^* a winner. Each unit of flow in the graph is thought of as a point for a winning entrant. The costs of edges correspond to minimum bribe costs in order to "fix" deterministic matches. We arrange the flow network such that, if there is a minimum cost feasible flow, then e^* can be made a winner for cost equal to the cost of the flow.

We detail the construction of the graph in three groups of edges between two interior stages. Fig. 5 illustrates an example of a completely constructed graph and we proceed through a description of the construction illustrating a scenario with five entrants. The nodes other than the source and sink are classified as either left nodes or right nodes. The left nodes are those that have edges from the source. The right nodes are those that have edges to the sink. The left and right nodes are connected by the interior edges of the graph.

In order to build the graph we begin with a bipartite graph (not including the source and sink). The nodes on the left hand side of the graph represent all the matches in the tournament. The nodes on the right hand side of the graph represent all the entrants to the tournament. Each unit of flow in this graph can be thought of, intuitively, as a "win" flowing to a particular entrant. We begin by building a source node s, a sink node t, a node for each match that will be played, and a collection node for each entrant in the tournament. The match nodes in Fig. 3 are labeled with the numbers of the entrants participating in a particular match while the collector nodes are labeled with the individual entrants.

To build the edges we first set one edge from the source to each of the left nodes with capacity and flow of 1 as illustrated in Fig. 3. These units of flow will count how many matches a particular entrant will win. Since each match is potentially worth one point we have exactly one unit of flow into each match node.

The internal edges (edges from matches to entrants) are constructed depending on the prices and probabilities of the individual matches. Fig. 4 illustrates the construction once all the internal edges have been added.

- If one entrant is sure to beat another entrant and we cannot afford any bribes with respect to these matches, then we build an edge $[1, 1], c = 0$ from the match node to the sure-to-win entrant.
- If both entrants have nonzero probabilities of winning, then we build an edge $[0, 1], c = 0$ to both entrant nodes. Since we are only attempting to find a configuration that would allow e^* to win, this construction allows us to check both possible match outcomes.

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004

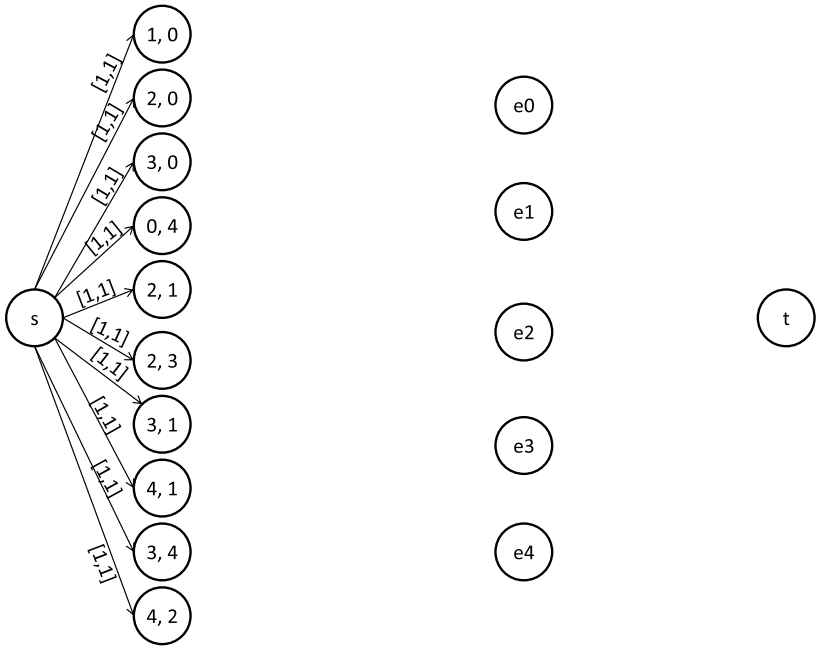

Fig. 3. We have a source, sink, match nodes for each possible match, and collector nodes for each participating entrant. We build edges from the source to all match nodes with 1 unit of flow.

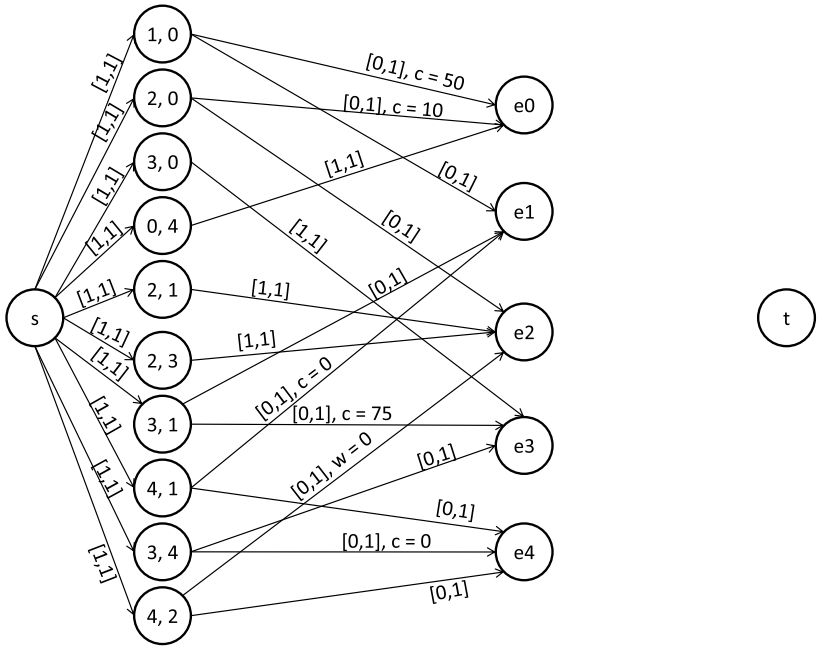

Fig. 4. To construct the internal linkage we build edges from all match nodes to their sure-to-win entrants; we build two edges in cases where either entrant is a possible winner; and we encode the cost of minimum bribes to change deterministic match outcomes.

- The final case of internal edges occurs if one entrant is sure to beat another and we can possibly afford the bribe. In this case we build an edge $[0,1], c = 0$ from the match node to the sure-to-win entrant and an edge $[0,1], c = min\$_{i,j}$ from the match node to the node of the entrant who can be made a possible winner if the bribe is made. This construction allows our flow network to change the outcome of a deterministic match with cost equal to the amount of the minimum bribe.

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004

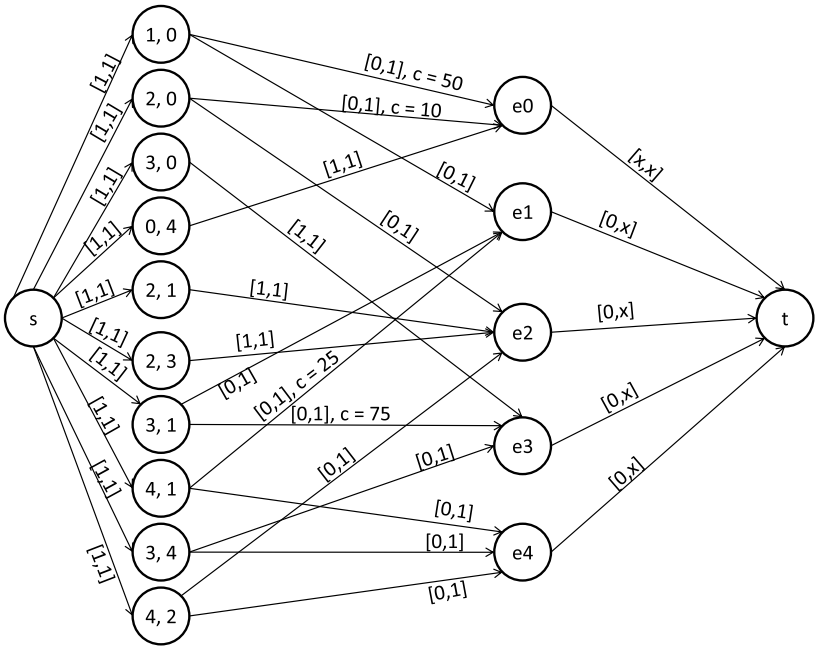

Fig. 5. A complete example of a minimal cost winner determination flow network.

We then build the final set of edges from right (entrant) nodes to the sink with capacity x, which we change between iterations of the graph. Recall that t is the total number of wins that e^* can achieve without resorting to bribery. We set $x = t + i$ and we iterate i from 0 to $n - t$ and build a graph as described above for each i. Here, x represents the most matches that e^* can win making i bribes to entrants involved in matches with e^*. We find the minimum cost feasible flow for each combination of bribes that e^* can make to achieve a given score. A complete graph construction is showing Fig. 5.

We check each graph in turn for $i = 0, \ldots, n - t$ and if we attain a feasible flow with cost $\leq B$ we accept, otherwise, reject.

This method is complete and will find the minimum cost bribery to give e^* a chance to win. Assume that there is a cheaper flow then the one found by our network. This would mean that there existed a less expensive way to grant e^* a number of wins greater than or equal to any other entrant. Since each entrant wins at most n matches in any particular graph and we check all n possible scores for e^*, this cannot happen. Therefore, this construction will find the minimal cost bribery construction.

We build at most n networks with $\mathcal{O}(n^2)$ nodes and edges. Since the minimum cost feasible flow problem can be solved in polynomial time, the overall running time of our algorithm is polynomial. Therefore pos-win-$ for RR is in P. □

We can straightforwardly extend the proof of Theorem 4.24 to the case where we do not have access to our budget. In the Pos-Win problem we cannot actually bribe anyone, we just have to choose winners of the non-deterministic matches in a way that allows e^* a way to win. The most direct way to show this is to apply the algorithm presented in Theorem 4.24 with $B = 0$. We can optimize this construction somewhat but it is sufficient to state the following corollary.

Corollary 4.25. *Pos-Win for* RR *is in* P.

Theorem 4.26. *The deterministic version of* RR Bribery *is in P.*

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004

Table 2
Complexity results for deterministic tournaments. The cup and round robin results are from Russell and Walsh [43]; the challenge results are from this paper.

	Challenge	Cup	Round-robin
Evaluation	P	P	P
CCM	P	P	P

This is based on Russell and Walsh's construction for CCM on RR tournaments [43]. We consider a min-cost max-flow network for each c from the current number of e^*'s wins up to n. To Walsh's min-cost max-flow network, we add the matches with e^* (his v_w). For all edges in bribable matches, we add the weight equal to the cost of the bribe. The capacities for edges to e^*'s matches is $[1, 1]$, and the capacity for the edge from e^* to the sink is $[c, n]$.

4.5. Reduction between tournament types

Theorem 4.27. *Constructive Bribery for* CTC $\leq_m^P$ *Constructive Bribery for* RR.

Proof. Suppose we are given a challenge-the-champ tournament with entrants $e_1, \ldots, e_n$ and $e^* = e_1$, and target probability t. We assume without loss of generality that n is odd. (If not, we add one entrant who is guaranteed to lose to e^* and cannot be bribed.) We also assume that $n \geq 7$ so that $(n-1)/2 + 1 < n - 2$.

We create a round robin tournament with entrants $e_1, \ldots, e_n$ and new distinguished entrants x and y. The budget, B, is the same as for the input challenge tournament. We have some bribable edges and some not. In particular the matches between e_is and e_js with for $i, j \neq 1$ are deterministically set so that each e_i wins exactly half its matches of this type. Furthermore, x beats all the e_i, $i \neq 1$, and x beats y. Furthermore, y beats e^*, which beats x. All the e_i, $i \neq k$, beat y. The costs of bribing the losers of these matches are $B + 1$. The probability that e^* beats e_i is as in the challenge tournament, as are the costs of bribing e_i to lower that cost. The target probability is t, as in the CTC tournament.

Note that y deterministically wins 1 match; x wins $n + 1$ matches; e_i wins $n/2 + 1$ matches; e^* beats x. The probability that e^* wins the maximum number of matches for the tournament is exactly the probability that e^* beats all the e_i. This is the same in both the challenge and round robin tournaments, as are the effects of bribes. □

Note that Evaluation for CT $\leq_m^P$ Evaluation for RR, since Evaluation for CT is in P.

5. Conclusions

Table 1 provides an overview of our complexity results on tournaments. Russell and Walsh [43] provided a thorough complexity analysis of similar problems in deterministic cases, all of which generate polynomial time solvable problems. Table 2 reveals that, in general, polynomial time evaluation procedures lead to polynomial time manipulation procedures (for deterministic systems).[8]

If we can compute a constructive manipulation in the deterministic case then we can determine a possible winner in the stochastic case. The problem of a possible winner with uncertain information is the same as constructive bribery in the deterministic setting with $B = \infty$.

In this paper we have studied the complexity of bribery and manipulation problems in sports tournaments, in which the actors seeking to affect the outcomes have access to only probabilities of how teams will perform. We don't yet have hardness proofs for some of the problems we present here. We conjecture

[8] This is also observed in [10].

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004

that the problems are, in fact, complete for the classes we mention. We observe that in some cases the introduction of uncertainty has no effect on membership in these complexity classes while in others it represents a significant change in reasoning structure and complexity.

References

[1] R. Ahuja, T. Magnanti, J. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, 1993.
[2] A. Altman, A.D. Procaccia, M. Tennenholtz, Nonmanipulable selections from a tournament, in: Proceedings of the 21st International Joint Conference on Artificial Intelligence, IJCAI 2009, 2009.
[3] H. Aziz, S. Gaspers, S. Mackenzie, N. Mattei, P. Stursberg, T. Walsh, Fixing a balanced knockout tournament, in: Proceedings of the 28th AAAI Conference on Artificial Intelligence, AAAI, 2014.
[4] H. Aziz, P. Harrenstein, M. Brill, J. Lang, F. Fischer, H. Seedig, Possible and necessary winners of partial tournaments, in: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2012, 2012.
[5] Y. Bachrach, N. Betzler, P. Faliszewski, Probabilistic possible winner determination, in: Proceedings of the 24th AAAI Conference on Artificial Intelligence, AAAI 2010, 2010.
[6] J. Bartholdi, C. Tovey, M. Trick, The computational difficulty of manipulating an election, Soc. Choice Welf. 6 (1989) 227–241.
[7] J. Bartholdi, C. Tovey, M.A. Trick, How hard is it to control an election?, Math. Comput. Model. 16 (1992) 27–40.
[8] D. Bouyssou, T. Marchant, P. Perny, M. Pirlot, A. Tsoukás, P. Vincke, Evaluation and Decision Models with Multiple Criteria: Stepping Stones for the Analyst, Springer, 2006.
[9] V. Conitzer, T. Sandholm, Complexity of manipulating elections with few candidates, in: Proceedings of the 18th AAAI Conference on Artificial Intelligence, AAAI 2002, 2002.
[10] V. Conitzer, T. Sandholm, J. Lang, When are elections with few candidates hard to manipulate?, J. ACM 54 (2007) 1–33.
[11] V. Conitzer, T. Walsh, L. Xia, Dominating manipulations in voting with partial information, in: Proceedings of the 25th American Association of Artificial Intelligence Conference, AAAI 2011, 2011.
[12] G. Erdélyi, H. Fernau, J. Goldsmith, N. Mattei, D. Raible, J. Rothe, The complexity of probabilistic lobbying, in: Proceedings of the 1st International Conference on Algorithmic Decision Theory, ADT 2009, 2009;
D. Binkele-Raible, G. Erdélyi, H. Fernau, J. Goldsmith, N. Mattei, J. Rothe, The complexity of probabilistic lobbying, Discrete Optim. 11 (2014) 1–21.
[13] P. Faliszewski, Nonuniform bribery, in: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2008, 2008.
[14] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, How hard is bribery in elections?, J. Artif. Intell. Res. 35 (2009) 485–532.
[15] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, Using complexity to protect elections, Commun. ACM 53 (2010) 74–82.
[16] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, J. Rothe, Llull and Copeland voting computationally resist bribery and constructive control, J. Artif. Intell. Res. 35 (2009) 275–341.
[17] P. Faliszewski, A. Procaccia, AI's war on manipulation: are we winning?, AI Mag. 31 (2010) 53–64.
[18] L. Ford, D. Fulkerson, Flows in Networks, Princeton University Press, 1962.
[19] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, 1979.
[20] A. Gibbard, Manipulation of voting schemes: a general result, Econometrica 41 (1973) 587–601.
[21] D. Gusfield, C. Martel, The structure and complexity of sports elimination numbers, Algorithmica 32 (2002) 73–86.
[22] N. Hazon, Y. Aumann, S. Kraus, M. Wooldridge, Evaluation of election outcomes under uncertainty, in: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2008, 2008.
[23] N. Hazon, Y. Aumann, S. Kraus, M. Wooldridge, On the evaluation of election outcomes under uncertainty, Artif. Intell. 189 (2012) 1–18.
[24] N. Hazon, P. Dunne, S. Kraus, M. Wooldridge, How to rig elections and competitions, in: Proceedings of the 2nd International Workshop on Computational Social Choice, COMSOC 2008, 2008.
[25] A. Healy, N. Malhotra, C.H. Mo, Irrelevant events affect voters' evaluations of government performance, Proc. Natl. Acad. Sci. USA 107 (2010) 12804–12809.
[26] R. Karp, Reducibility among combinatorial problems, in: Complexity of Computer Computations, vol. 43, 1972, pp. 85–103.
[27] W. Kern, D. Paulusma, The computational complexity of the elimination problem in generalized sports competitions, Discrete Optim. 1 (2004) 201–214.
[28] K. Konczak, J. Lang, Voting procedures with incomplete preferences, in: Proceedings of the 19th International Joint Conference on Artificial Intelligence, IJCAI 2005, 2005.
[29] J. Lang, M. Pini, F. Rossi, D. Salvagnin, K. Venable, T. Walsh, Winner determination in voting trees with incomplete preferences and weighted votes, Auton. Agents Multi-Agent Syst. 25 (2012) 130–157.
[30] J. Lang, M. Pini, F. Rossi, K. Venable, T. Walsh, Winner determination in sequential majority voting, in: Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI 2007, 2007.
[31] J.-F. Laslier, Tournament Solutions and Majority Voting, Springer, 1997.
[32] N. Mattei, Empirical evaluation of voting rules with strictly ordered preference data, in: Proceedings of the 2nd International Conference on Algorithmic Decision Theory, ADT 2011, 2011.
[33] N. Mattei, Decision making under uncertainty: theoretical and empirical results on social choice, manipulation, and bribery, Ph.D. thesis, University of Kentucky, 2012.

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004

[34] N. Mattei, J. Forshee, J. Goldsmith, An empirical study of voting rules and manipulation with large datasets, in: Proceedings of the 4th International Workshop on Computational Social Choice, COMSOC 2012, Springer, 2012.
[35] N. Mattei, J. Goldsmith, A. Klapper, On the complexity of bribery and manipulation in tournaments with uncertain information, in: Proceedings of the 25th International Florida Artificial Intelligence Research Society Conference, FLAIRS 2012, 2012.
[36] N. Mattei, T. Walsh, Preflib: a library of preference data, in: Proceedings of the 2nd International Conference on Algorithmic Decision Theory, ADT 2013, 2013.
[37] D.C. McGarvey, A theorem on the construction of voting paradoxes, Econometrica (1953) 608–610.
[38] H. Moulin, Choosing from a tournament, Soc. Choice Welf. 3 (1986) 271–291.
[39] M. Mundhenk, J. Goldsmith, C. Lusena, E. Allender, Complexity of finite-horizon Markov decision process problems, J. ACM 47 (2000) 681–720.
[40] C. Papadimitriou, Computational Complexity, Addison-Wesley Publishing Company, Inc., 1994.
[41] A.D. Procaccia, J.S. Rosenschein, Average-case tractability of manipulation in voting via the fraction of manipulators, in: Proceedings of 6th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2007, 2007.
[42] S. Rosen, Prizes and incentives in elimination tournaments, Am. Econ. Rev. 76 (1986) 701–715.
[43] T. Russell, T. Walsh, Manipulating tournaments in cup and round robin competitions, in: Proceedings of the 1st International Conference on Algorithmic Decision Theory, ADT 2009, 2009.
[44] M. Satterthwaite, Strategy-proofness and arrow's conditions: existence and correspondence theorems for voting procedures and social welfare functions, J. Econ. Theory 10 (1975) 187–216.
[45] J. Simon, On some central problems in computational complexity, Ph.D. thesis, Cornell University, 1975.
[46] K. Sohrabi, J. Gao, V. Ailawadhi, G.J. Pottie, Protocols for self-organization of a wireless sensor network, IEEE Pers. Commun. 7 (2000) 16–27.
[47] I. Stanton, V. Vassilevska Williams, Manipulating single-elimination tournaments in the Braverman–Mossel model, in: IJCAI Workshop on Social Choice and Artificial Intelligence, 2011.
[48] I. Stanton, V. Vassilevska Williams, Rigging tournament brackets for weaker players, in: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, IJCAI 2011, 2011.
[49] I. Stanton, V. Vassilevska Williams, Manipulating stochastically generated single-elimination tournaments for nearly all players, in: Proceedings of 7th International Workshop on Internet and Network Economics, WINE, 2011.
[50] G. Tullock, Toward a Theory of the Rent-Seeking Society, A&M University Press, Texas, 1980.
[51] T. Vu, A. Altman, Y. Shoham, On the complexity of schedule control problems for knockout tournaments, in: Proceedings of the 8th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2009, 2009.
[52] T. Vu, N. Hazon, A. Altman, S. Kraus, Y. Shoham, M. Wooldridge, On the complexity of schedule control problems for knock-out tournaments, Working paper, 2013.
[53] T. Vu, Y. Shoham, Fair seeding in knockout tournaments, ACM Trans. Intell. Syst. Technol. 3 (2011) 1–17.
[54] T. Walsh, An empirical study of the manipulability of single transferable voting, in: Proc. of the 19th European Conference on Artificial Intelligence, ECAI-2010, in: Frontiers in Artificial Intelligence and Applications, 2010.
[55] T. Walsh, Where are the hard manipulation problems?, J. Artif. Intell. Res. 42 (2011) 1–39.
[56] V. Williams, Fixing a tournament, in: Proceedings of the 24th AAAI Conference on Artificial Intelligence, AAAI 2010, 2010.
[57] L. Xia, V. Conitzer, Determining possible and necessary winners under common voting rules given partial orders, J. Artif. Intell. Res. 41 (2011) 25–67.
[58] L. Xia, M. Zuckerman, A. Procaccia, V. Conitzer, J. Rosenschein, Complexity of unweighted coalitional manipulation under some common voting rules, in: Proceedings of the 21st International Joint Conference on Artificial Intelligence, IJCAI 2009, 2009, pp. 348–353.

Please cite this article in press as: N. Mattei et al., On the complexity of bribery and manipulation in tournaments with uncertain information, Journal of Applied Logic (2015), http://dx.doi.org/10.1016/j.jal.2015.03.004