

Equilibrium Computation For Knockout Tournaments Played By Groups

Grzegorz Lisowski, M. S. Ramanujan, and Paolo Turrini

University of Warwick

United Kingdom

{Grzegorz.Lisowski,R.Maadapuzhi-Sridharan,P.Turrini}@warwick.ac.uk

ABSTRACT

In single-elimination knockout tournaments, participants face each other based on a starting seeding and progress to the next rounds by beating their direct opponents. In this paper we initiate the study of coalitional knockout tournaments, which generalise single-elimination knockout tournaments by allowing groups of players, or coalitions, to strategically select one of their members to take part in the tournament, following the starting seeding. We investigate the algorithmic properties of pure strategies Nash equilibria in these games under various setups, i.e., whether or not choices can be made at each round and whether or not tournament progression is important to the group. Despite the more complex tournament structure when compared to single-elimination, we provide (quasi-) polynomial-time algorithms for all cases. Our results can be applied to those tournaments where pre-play selection plays an important role, such as sport events or elections with run-off.

KEYWORDS

Single-elimination Tournaments; Coalitional Strategies; Equilibria

ACM Reference Format:

Grzegorz Lisowski, M. S. Ramanujan, and Paolo Turrini. 2022. Equilibrium Computation For Knockout Tournaments Played By Groups. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022)*, Online, May 9–13, 2022, IFAAMAS, 9 pages.

1 INTRODUCTION

Sport events are frequently organised as knockout tournaments among selected individuals or teams. In international competitions, for example, coaches choose from a nation’s best sportspeople to maximise their country’s winning chances, which requires studying the potential rivals at different stages of the tournament. Strategic selection is also relevant in those tournaments where progression matters, even when the chances of winning the entire tournament may be slim. For instance, clubs participating in the UEFA Champions League receive significant financial rewards for reaching each stage of the competition and so, selecting the right squad (which has to be before the beginning of the knockout stages) is critical.

Selection headaches of this type are of course not limited to sports. Many other competitive environments can be modelled as tournaments, where self-interested participants strive to make their best possible choice against the predicted choices of their potential opponents. If we consider presidential elections with run-off, for instance, we observe a similar phenomenon: parties put forward

one candidate each, with the top two in the first round competing against each other. Also in this case the right selection of a candidate heavily depends on the potential choices of the opposing parties.

Tournaments, understood as directed graphs over players (see, e.g. [23], [20], [12], [21]), were introduced in the game theory and social choice research, and have gained attention in computer science for their well-behaved computational properties (see e.g., [4]). In this paper we focus on what are possibly the simplest and best-known tournaments, knockout (or single-elimination) tournaments, where participants are initially associated with the leaves of a full binary tree, and the winner of the match played between the players at a pair of sibling nodes proceeds to the next stage, i.e., the parent of these two nodes. Knockout tournaments have been extensively analysed in the computational social choice and algorithmic game theory communities, either in connection with social choice functions (see, e.g. [24], [19], [6]) or from the point of view of an external manipulator trying to rearrange the initial seeding to guarantee a certain player to be the winner (see, e.g. [14], [26], [16], [3]).

However, the problem of analysing the strategic behaviour of groups in knockout tournaments - be they tennis players or party candidates to choose from - has been largely overlooked.

Our contribution. In this paper we address this gap by extending standard (single-elimination) tournaments to account for the strategic behaviour of groups. Before the tournament starts, we allow each of them to make an independent choice of the best selection, or *candidate*, to put forward. We study the equilibrium behaviour of groups (or *coalitions*) in such tournaments from an algorithmic point of view. Our analysis spans three axes: A) whether coalitions choose their candidates once and for all – called *one-shot* tournaments (which we model as winner-take-all knockout tournaments) – or not – *dynamic* tournaments B) whether only winning matters – what we call *win-lose* tournaments – or also tournament progression – *beyond win-lose*; C) whether we focus on computing equilibria or on verifying a given one. Despite the complex tournament structure, we show polynomial-time or quasi-polynomial-time algorithms for all these cases (Table 1).

Related literature. Our results are connected to multiple research lines in the computational analysis of tournaments. In the context of social choice theory, attention has been paid to stable solutions [4] and to subclasses displaying desirable properties [5]. The existence of well-behaved solutions has also motivated the study of the complexity of their computation (as in e.g., [7], [8], [5]).

Our approach is closely related to strategic voting (see, e.g. [22]), the growing branch of social choice theory dealing with strategic behaviour in collective decision-making and, more specifically, to strategic candidacy [9]. In our case the electorate is composed of

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online. © 2022 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

coalitions who vote for one of their members, i.e., the one to put forward in the tournament, strategising over the outcome of the tournament – our social choice function – given the choices of the opponents. Along these lines, [17] recently studied, in the context of tournament solutions, how politicians can form a coalition so that a representative of their group is elected. Further, [11] conducted a complexity analysis of how parties can win elections based on the voters’ preferences over the set of all potential candidates. In the context of voting with candidate selection on Hotelling-Downs spaces, equilibrium computation was shown to be not tractable in the general case [13].

Finally, our results are directly relevant to the problem of *tournament manipulation*, for example understanding how the seeding can be manipulated to force a particular winner, a problem extensively studied in the literature (by e.g. [27], [10], [2] [10], [26], [14], [16], [3], [18]) or whether a pair of players can reverse their comparison to make one of them the winner of the tournament (see, e.g., [1]).

Paper structure. In section 2 we provide the basic setup, notably the definitions of binomial arborescence, coalitional strategies and equilibrium solutions, together with some key observations. Section 3 analyses one-shot tournaments, providing algorithm for computing their equilibria, while Section 4 focuses on dynamic ones.

2 PRELIMINARIES

Tournaments. Let $[n] = \{1, 2, \dots, n\}$ be a set of players and let D be a digraph $([n], E)$ with $E \subseteq [n]^2$, representing the results of a round-robin tournament played between them. Here, for every pair of players, exactly one is picked to be the winner between the two, which is encoded in D as an arc directed from the winner to the loser. So, for every pair of players i, j with $i \neq j$, exactly one of $(i, j), (j, i)$ belongs to E . We shall call such a graph a *tournament digraph*. Also, if $(i, j) \in E$, we say that i *beats* j .

Let $V(D)$ denote the set of vertices of D . A *knockout tournament* or *single-elimination tournament* (henceforth, SE-tournament) on D is defined as a complete binary tree T with n leaves $L(T)$ and a bijective function $\pi : V(D) \rightarrow L(T)$ called the *seeding*, mapping the n players to the n leaves. Then, the *winner of the knockout tournament* corresponding to π is determined recursively: the winner at a leaf l is the player j with $l = \pi(j)$, and the winner of the subtree rooted at a node v is the winner of the match between the winners of the two subtournaments rooted at the children of v , as decided by the orientation of the unique arc between these two vertices. In this paper, we will only consider SE-tournaments based on perfect binary trees, implying that the number of players entering the tournament is a power of 2. An example of a SE-tournament is depicted in the left side of Figure 1.

Arborescences. The proofs in the paper will often use the technical notion of *binomial arborescence* [26], which allows for a succinct formulation of the structural properties of SE-tournaments. An *arborescence* is a rooted directed tree such that all the arcs are directed away from the root.

DEFINITION 1 ([26]). *Let D be a tournament digraph. The set of binomial arborescences over D is recursively defined as follows:*

- Each $a \in V(D)$ is a binomial arborescence rooted at a ;

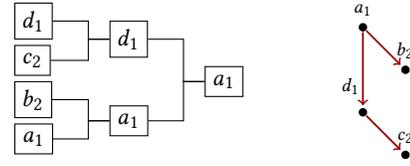


Figure 1: On the left an SE-tournament with 4 players. On the right the corresponding binomial arborescence. Notice that the first-round matches are (a_1, b_2) and (d_1, c_2) , and the only second-round match is (a_1, d_1) , which is won by a_1 .

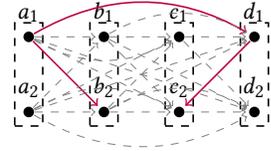


Figure 2: A coalitional structure of teams A, B, C, D , each with two members. In red, the arborescence-shaped selection.

- If, for some $r > 0$, T_a and T_b are 2^{r-1} -node binomial arborescences rooted at a and b respectively, then the tree T resulting from adding an arc from a to b is the 2^r -node binomial arborescence rooted at a .

An example of binomial arborescence is in the right of Figure 1. If a binomial arborescence T is such that $V(T) = V(D)$, then T is a *spanning binomial arborescence* (s.b.a.) of D . Intuitively, an s.b.a. can be used to compactly encode how an SE-tournament will evolve, following its tournament digraph.

As shown by [26], there is a formal connection between binomial arborescences and knockout tournaments.

PROPOSITION 1 ([26]). *Let D be a tournament digraph and let $v^* \in V(D)$. Then, there is a seeding of $V(D)$ such that the resulting knockout tournament is won by v^* iff D has an s.b.a. rooted at v^* .*

As a result of this proposition, we shall interchangeably use the terms binomial arborescence and SE-tournament, when these are clear from the context. Henceforth, we will mainly work with binomial arborescences, as this allows for neater proofs and procedures.

Subtournaments. Consider now a set of players $[2^m]$. We denote by $SE_{\pi, [2^m]}$ the spanning binomial arborescence representing the SE-tournament played by the players in $[2^m]$ following the seeding π . We call the root of $SE_{\pi, [2^m]}$ the *winner* of $SE_{\pi, [2^m]}$. Moreover, for each $r \in \{0, \dots, m\}$, we denote by $SE_{\pi, [2^m]}^r$ the binomial arborescence representing the subtournament of $SE_{\pi, [2^m]}$ played by the winners of all the r^{th} round matches of $SE_{\pi, [2^m]}$ (notice that there are exactly 2^{m-r} players who win at least r rounds). In other words, $SE_{\pi, [2^m]}^0$ is the same as the full tournament $SE_{\pi, [2^m]}$ and, for every $r \in [m]$, $SE_{\pi, [2^m]}^r$ denotes the subgraph of $SE_{\pi, [2^m]}^{r-1}$ obtained by simply deleting all its leaves. Notice that $SE_{\pi, [2^m]}^m$ contains a single vertex which corresponds to the winner of $SE_{\pi, [2^m]}$.

Furthermore, let u, v_1, v_2 be vertices of a binomial arborescence T such that u is the parent of v_1 and v_2 . For each $x \in \{1, 2\}$, let

ℓ_x denote the size (number of vertices) of the largest binomial arborescence contained in T and rooted at v_i . If $\ell_1 < \ell_2$, then we say that v_2 is a heavier child of u than v_1 . We denote this by $v_2 \succ_u^T v_1$, with the explicit references to u and T dropped when these are clear from the context. Notice that ℓ_1 and ℓ_2 can never be equal. If v_2 is the unique child of u that is heavier than all of their siblings, then we say that v_2 is the *heaviest* child of u . The *lightest* child of u is symmetrically defined. For $r \in [\log \ell]$, these notions can be extended in the natural way to r *heaviest children* (or r *lightest children*) of u . Intuitively, assuming u is not a leaf, the heaviest child of u is the *last* player beaten by u in the knockout tournament and the lightest child of u is the *first* player beaten by u , i.e., the opponent of u in the first round. Similarly, the r heaviest children of u are the last r opponents of u that lose to u and the r lightest children of u are the first r opponents of u that lose to u .

Coalitional Structures. In this paper we study *coalitional* tournaments, where players are partitioned into groups (also called *teams* or *coalitions*). Informally, each coalition will put forward one of their members in the spot allocated to them by the seeding. When each coalition makes their choice, this will give rise to a standard SE-tournament. So, for a set \mathcal{P} of players, a *coalitional structure* of \mathcal{P} is a partition $C = \{C_1, C_2, \dots, C_\ell\}$ of \mathcal{P} , such that $\ell = 2^m$ for some $m > 0$. For a set $C' \subseteq C$ of coalitions, we use $\mathcal{P}(C')$ to denote their set of players. That is, $\mathcal{P}(C') = \bigcup_{C \in C'} C$. Moreover, for a player $p \in \mathcal{P}$, we denote by $\mathcal{C}(p)$ the coalition to which p belongs.

We use D_C to denote the $|C|$ -partite coalitional structure representing the results of matches played between all pairs of players of distinct coalitions. So, D_C is a graph (\mathcal{P}, E) such that for every pair of players $p_i, p_j \in \mathcal{P}$ for which $p_i \in C_i, p_j \in C_j$ and $C_i \neq C_j$, exactly one of $(p_i, p_j), (p_j, p_i)$ belongs to E .

Figure 2 shows a 4-partite coalitional structure, with a strategic selection reflecting the SE-tournament and the binomial arborescence in Figure 1.

Input representation. In order to analyze the running time of our algorithms, we need to first establish the input representation and input size. Given the observations above, we represent a coalitional knockout tournament in space $O(n^2 \log n) + O(n \log \ell) + O(\ell \log \ell)$, where ℓ is the number of coalitions and n the total number of players. Here, $O(n^2 \log n)$ is a bound on the representation of D_C , $O(n \log \ell)$ is the space to represent the partition of players into coalitions and $O(\ell \log \ell)$ the space to represent the seeding. As the number of coalitions ℓ is at most the number of players n (we do not permit empty coalitions), the input has bit-size $O(n^2 \log n)$.

We assume that the seeding is fixed and known to all coalitions a priori. Hence, when possible, in order to keep the notation simple, we will refrain from explicitly referring to the seeding.

3 ONE-SHOT KNOCKOUT TOURNAMENTS

In this section, we study tournaments where the coalitions choose a representative once and for all, before the competition starts. We are interested in equilibrium strategies - specifically, Nash equilibria - and to study the complexity of their computation and verification.

Starting from a coalition structure C we call a selection of players by all teams a *strategy profile* (or simply, a *profile*) and formally define it as a function $s : C \rightarrow \mathcal{P}$, such that $s(C_i) \in C_i$. In other words,

a strategy profile is a selection by each coalition of one of their members, an example of which is shown in Figure 2. We note that a strategy profile s can be simply rewritten as a tuple (p_1, \dots, p_ℓ) of members of the corresponding coalitions. Also, whenever the seeding π is clear from the context, we write SE_s to denote the tournament between players in s following π , and $SE_{s[p'_i]}$ the tournament obtained from SE_s by replacing player p_i with p'_i (in the same coalition as p_i). Furthermore, we say that a coalition C *wins* tournament SE_s if the winner of SE_s belongs to C . Figure 1 features the tournament $SE_{\{a_1, b_2, c_2, d_1\}}$, with a_1 being the winner.

3.1 Win-Lose Games

The simplest type of one-shot tournaments we look at are the ones where only winning matters (i.e., *win-lose games*). When picking a player to put forward in a win-lose game, a coalition will have to reason about the possible outcomes of the tournament as a function of the opponents' choices. To encode this reasoning we define appropriate variants of the classical Nash equilibria.

DEFINITION 2 (EQUILIBRIA). A profile $s = (p_1, \dots, p_\ell)$ is a Nash Equilibrium (NE), if for all i and for all $p'_i \in C_i$: if C_i wins $SE_{s[p'_i]}$ then C_i wins SE_s .

In other words, in a NE, no coalition can improve their outcome by changing their representative unilaterally. Clearly, a NE need not exist in general, which opens important algorithmic questions regarding their existence and computation.

3.1.1 Structural and algorithmic properties of NE.

Equilibrium existence. We start by addressing the question of deciding whether a given strategy profile is a NE, what we call the problem of *recognising* a NE. To do so we introduce a technical notion and a key lemma, before we are able to show its tractability.

DEFINITION 3. Let $j \in [\ell]$ such that p_j is the root of the s.b.a. SE_s . Let j' be a coalition distinct from j and consider the p_j - $p_{j'}$ path H in SE_s . For every player p on this path, we denote by $\text{Opponent}_H^+[p, p_{j'}]$ the set defined as follows.

- If $p = p_{j'}$, $\text{Opponent}_H^+[p, p_{j'}]$ is the set of children of $p_{j'}$ in SE_s .
- Else, $\text{Opponent}_H^+[p, p_{j'}]$ is the set $\{p\} \cup \{v \mid v \text{ is a child of } p \text{ and } v \succ_p p'\}$, where p' denotes the unique child of p contained in H .

The utility of the above definition is derived from the fact that it formalises the set of future opponents that the coalition $C_{j'}$ would have to face if it were to replace $p_{j'}$ with a different player $p'_{j'}$ in the profile s - what we denote $\bigcup_{p \in V(H)} \text{Opponent}_H^+[p, p_{j'}]$. Note that this set is no larger than $\log \ell$ since that is the maximum number of opponents faced by any player in the tournament. We drop the explicit reference to H in this notation when the root p_j (the winner) is clear from the context. See Figure 3 for a visual illustration.

The following structural lemma forms the crux of our algorithmic results. Informally, we use the fact that a profile is a NE if and only if no coalition which is losing the tournament can switch their representative to one which beats all players beaten by the original representative *plus* every future opponent (that the original representative did not get to face following its loss), along with the properties of the technical notion in Definition 3.

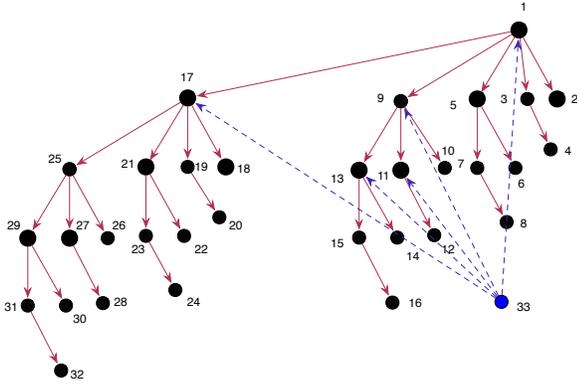


Figure 3: An example of a non-NE strategy, where players 12 and 33 belong to the same coalition (for convenience called Coalition 12), and all other coalitions consist of one player only and the index of the Coalition is the same as the index of the player (e.g., Player 14 plays for Coalition 14). There are 33 players and 32 coalitions in total. The seeding pairs up Coalition $2i + 1$ and Coalition $2i + 2$ in the first round, for each $i \in \{0, \dots, 15\}$. Consider the path $1 \rightarrow 9 \rightarrow 11 \rightarrow 12$, call it H . Notice that $\text{Opponent}^+[12, 12] = \emptyset$, $\text{Opponent}^+[11, 12] = \{11\}$, $\text{Opponent}^+[9, 12] = \{9, 13\}$, $\text{Opponent}^+[1, 12] = \{1, 17\}$. Thus, if Coalition 12 wanted to win by replacing its chosen player (Player 12) with an alternative, then it would have to beat the players chosen by Coalitions $\{11, 9, 13, 1, 17\}$, which are precisely the Coalitions indexed by $\bigcup_{p \in V(H)} \text{Opponent}_H^+[p, 12]$. Also, fixing the choices of all coalitions, player 33 improves upon player 12, beating all the potential future opponents – Players $\{11, 9, 13, 1, 17\}$.

LEMMA 1. Consider a profile $s = (p_1, \dots, p_\ell)$ and let $j \in [\ell]$ such that p_j is the root of the s.b.a. SE_s . Then, s is a NE iff there is no $j' \in [\ell] \setminus \{j\}$ and player $p_{j'} \in C_{j'}$ such that $p_{j'}$ beats all players in $\text{Opponent}^+[p, p_{j'}]$ for every player p on the p_j - $p_{j'}$ path in SE_s .

Figure 3 shows a non-NE profile based on the characterization in Lemma 1. As a consequence of this lemma, we obtain a simple algorithm for testing whether a given profile is a NE.

PROPOSITION 2. *Recognising a NE is P-time solvable.*

We note that this test can be done in $O(n^3)$ -time, implying a running time that is subquadratic in the input size.

Equilibrium computation. We are further interested in *computing* a NE, which is much more complex than recognising one. Surprisingly, we show that this can still be done in quasi-polynomial time.

To be precise, our main theorem, Theorem 1, will show the existence of an $n^{O(\log \ell)}$ -time algorithm for computing a NE. To obtain this result, we will make use of a key lemma, Lemma 2, which establishes that a NE (if one exists) can be obtained by composing specific types of strategies for various subtournaments. This lemma effectively implies that we can compute a NE (if one exists) by examining only at most $n^{O(\log \ell)}$ out of the set of possibly $n^{\Omega(\ell)}$ -many strategy profiles.

Let us start with providing several useful notions. For a set S and $q \in \mathbb{N}$, we denote by $\binom{S}{\leq q}$ the set of subsets of S of size at most q and by $[q]_{\geq 0}$ the set $\{0\} \cup [q]$. Let us denote by $C_r(C_j)$ the set of coalitions who could potentially meet Coalition C_j within the first r -rounds (over all possible digraphs D_C and profiles s) plus the Coalition C_j itself. Notice that $|C_r(C_j)| = 2^r$. Thus, $C_r(C_j) \setminus C_{r-1}(C_j)$, denotes the set of all possible opponent coalitions that C_j could face exactly in the r^{th} round. For technical reasons, we set $C_r(C_j) = \emptyset$ for every $r \in \{-1, -2\}$ and $C_j \in C$. Finally, for the seeding π we define by π_j^r the *restriction* of π to $C_r(C_j)$. Similarly, for a profile s , we denote by s_j^r , the profile $s|_{C_r(C_j)}$, which is the restriction of s to $C_r(C_j)$. Conversely, we say that s is an *extension* of s_j^r . Notice that there can be multiple possible extensions of s_j^r .

We next define a mapping ζ which allow us to reason about a NE in terms of the outcomes of specific subtournaments.

DEFINITION 4. We define the function $\zeta : C \times \mathcal{P} \times \mathcal{P} \times [\log \ell]_{\geq 0} \times \binom{\mathcal{P}}{\leq \log \ell} \rightarrow 2^S$ as follows. Let $C_j \in C, r \in [\log \ell]_{\geq 0}, p_j \in C_j, p_{j'} \in C_{j'}$ such that $C_{j'} \in C_r(C_j) \setminus C_{r-1}(C_j), Z \in \binom{\mathcal{P}}{\leq \log \ell}$ such that $Z \cap \mathcal{P}(C_r(C_j)) = \{p_j\}$. Then, $\zeta(C_j, p_j, p_{j'}, r, Z)$ denotes the set of all profiles s over $C_r(C_j)$ such that:

- (a) player $p_j \in C_j$ wins $SE_{\pi_j^r, s}$
- (b) player $p_{j'}$ is the final opponent of p_j in tournament $SE_{\pi_j^r, s}$
- (c) for every coalition $C_{j''} \in C_r(C_j) \setminus \{C_j\}$ and player $p_{j''} \in C_{j''}$, either (i) $p_{j''}$ is beaten by a player in Z or (ii) by a player in the set $\text{Opponent}^+[p, p_{j''}]$ for some player p on the p_j - $p_{j''}$ path in $SE_{\pi_j^r, s}$.

For every other choice of $C_j, p_j, p_{j'}, r, Z$, $\zeta(C_j, p_j, p_{j'}, r, Z) = \emptyset$.

So, $\zeta(C_j, p_j, p_{j'}, r, Z)$ is the set of all those profiles over the r -round tournament comprised of Coalition C_j 's first r matches, such that the player p_j (from Coalition C_j) wins these r rounds, Player $p_{j'}$ is the final opponent of p_j in the r round tournament (i.e., p_j and $p_{j'}$ play in the r^{th} round) and all players in any coalition who could potentially meet C_j within the first r rounds are beaten either by a player in the set Z or by a player in the set $\text{Opponent}^+[p, p_{j''}]$ for some player p on the p_j - $p_{j''}$ path in the s.b.a. $SE_{\pi_j^r, s}$.

An informal description of the motivation behind this definition is the following. If we take Z to denote p_j plus its “potential future opponents” from round $r + 1$ onwards, then $\zeta(C_j, p_j, p_{j'}, r, Z)$ contains precisely all those profiles over $C_r(C_j)$ such that in any profile (over all coalitions) that is an extension thereof, there is no benefit for any coalition in $C_r(C_j)$ (other than possibly C_j) to unilaterally alter its strategy because any alternative player will either fail to win the first r rounds anyway (Property (c) (ii) of Definition 4) or eventually lose to a “future opponent” in the set Z (Property (c) (i) of Definition 4). Notice that the set Z is never larger than $\log \ell$ because the number of rounds in the whole tournament is $\log \ell$. Our quasi-polynomial running time arises from the fact that we have $n^{O(\log \ell)}$ possibilities for Z and essentially go over all of these possibilities in our algorithm.

Definition 4 and Lemma 1 entail that by setting the arguments of the function ζ appropriately, one can capture all NEs. Thus:

OBSERVATION 1. *Every NE is contained in $\zeta(C_j, p_j, p_{j'}, \log \ell, \{p_j\})$ for some $C_j \in C, p_j \in C_j$ and $p_{j'} \in C_{j'}$ and conversely, for every $C_j \in$*

$C, p_j \in C_j$ and $p_{j'} \in C_{j'}$, every profile in $\zeta(C_j, p_j, p_{j'}, \log \ell, \{p_j\})$ is a NE.

Given the above observation, what naturally suffices is to compute the function ζ for all possible settings of the arguments. However, this may not be possible in quasi-polynomial time because even just listing all possible NEs could be too computationally expensive. To overcome this obstacle, we next prove a structural result that shows, intuitively, that if a strategy profile s is a NE for a given tournament, then we can reconstruct s (or an alternate NE) by going over the subtournaments at every possible level, examining the image (which is a set of profiles) of ζ by setting the arguments appropriately, and then merging these profiles. In particular, we show that if there is a NE-strategy s then, for every round $r \in [\log \ell]$, s extends some profile s' in $\zeta(C_j, p_j, p_{j'}, r, Z)$ for some $C_j, p_j \in C_j, p_{j'} \in C_{j'}$ and Z , and moreover, for any other profile $s'' \in \zeta(C_j, p_j, p_{j'}, r, Z)$, we can “cut” s' from s and “paste” s'' to obtain another NE. The main algorithmic consequence of this fact is that for any choice of the arguments $C_j, p_j, p_{j'}, r$ and Z , instead of computing all the profiles contained in $\zeta(C_j, p_j, p_{j'}, r, Z)$, it is sufficient to compute a single one because if any of them can be extended to a NE, then every one of them can be extended to a NE. This “pruning lemma” is at the heart of our algorithm.

LEMMA 2. *Let $s = (p_1, \dots, p_\ell)$ be a NE, $r \in [\log \ell]$ and let the winner of SE_s be $p_{j^*} \in C_{j^*}$. Let $p_j \in C_j$ be the winner of the binomial sub-arborescence $SE_{\pi_j^r, s_j^r}$ and let $p_{j'} \in C_{j'}$ denote the final opponent of p_j in $SE_{\pi_j^r, s_j^r}$. Let $Z = \bigcup_{p \in V(H) \setminus \{p_{j'}\}} \text{Opponent}_H^+[p, p_{j'}]$ where H denotes the p_{j^*} - $p_{j'}$ path in SE_s .*

- (1) Then, $\zeta(C_j, p_j, p_{j'}, r, Z)$ is non-empty.
- (2) Moreover, for every $p_{j_1} \in C_{j_1}$ and $p_{j_2} \in C_{j_2}$, for every $\hat{s}_1 \in \zeta(C_j, p_j, p_{j_1}, r-1, Z \cup \{p_{j'}\})$, and $\hat{s}_2 \in \zeta(C_{j'}, p_{j'}, p_{j_2}, r-1, Z \cup \{p_{j'}\})$, the composed profile $\hat{s}_1 \cdot \hat{s}_2$ is contained in $\zeta(C_j, p_j, p_{j'}, r, Z)$.

PROOF. We first show that $s_j^r \in \zeta(C_j, p_j, p_{j'}, r, Z)$. By the definition of s_j^r , p_j is the root of $SE_{\pi_j^r, s_j^r}$ and $p_{j'}$ is the heaviest child of p_j in $SE_{\pi_j^r, s_j^r}$. Hence, if $s_j^r \notin \zeta(C_j, p_j, p_{j'}, r, Z)$, then there exists $p_{j''} \in C_{j''} \in C_r(C_j) \setminus \{C_j\}$, such that $p_{j''}$ beats every player in Z and $p_{j''}$ beats every player in the set $\text{Opponent}^+[p, p_{j''}]$ for every player p on the p_j - $p_{j''}$ path in $SE_{\pi_j^r, s_j^r}$. Along with our choice of Z , this would then also imply that $p_{j''}$ beats every player in the set $\text{Opponent}^+[p, p_{j''}]$ for every player p on the p_{j^*} - $p_{j''}$ path in SE_s , contradicting our choice of s as a NE (see Lemma 1).

For the second statement, we need to prove that Properties (a), (b), and (c) in Definition 4 are satisfied by $\hat{s} = \hat{s}_1 \cdot \hat{s}_2$. The first two properties are satisfied since p_j beats $p_{j'}$, $p_j = SE_{\hat{s}_1}^{r-1}$, and $p_{j'} = SE_{\hat{s}_2}^{r-1}$. Suppose Property (c) is violated. Then, for some $p_{j''} \in C_{j''} \in C_r(C_j) \setminus \{C_j\}$, $p_{j''}$ beats all players in Z and all players in the set $\text{Opponent}^+[p, p_{j''}]$ for each player p on the p_j - $p_{j''}$ path in $SE_{\pi_j^r, s_j^r}$. However, note that if $C_{j''} \in C_{r-1}(C_j)$ then we obtain a contradiction to our assumption that $\hat{s}_1 \in \zeta(C_j, p_j, p_{j_1}, r-1, Z \cup \{p_{j'}\})$ satisfies Property (c) of Definition 4, and otherwise (if $C_{j''} \in C_r(C_j) \setminus C_{r-1}(C_j)$), this violates our assumption that $\hat{s}_2 \in \zeta(C_{j'}, p_{j'}, p_{j_2}, r-1, Z \cup \{p_{j'}\})$ satisfies Property (c) of Definition 4. \square

Lemma 2 forms the core of our algorithm to compute a NE, whose running time we establish with the following theorem.

THEOREM 1. *There is an $n^{O(\log \ell)}$ -time algorithm for computing a NE.*

PROOF. Due to Observation 1, our algorithm to compute a NE aims to identify and return an element of a non-empty $\zeta(C_j, p_j, p_{j'}, \log \ell, \{p_j\})$ if such a $C_j, p_j, C_{j'}$ and $p_{j'}$ exist. This is necessary and sufficient. We achieve this via a dynamic programming algorithm that fills a table T where the cells are indexed by tuples of the form $(C_j, p_j, p_{j'}, r, Z)$. Moreover, any non-empty cell indexed by the tuple $(C_j, p_j, p_{j'}, r, Z)$ contains a single element of $\zeta(C_j, p_j, p_{j'}, r, Z)$ and an empty cell indexed by the tuple $(C_j, p_j, p_{j'}, r, Z)$ indicates that $\zeta(C_j, p_j, p_{j'}, r, Z)$ is empty. Notice that if the table is filled correctly, then the solution (i.e., a NE strategy) can be determined by going over all possible $C_j, p_j, C_{j'}$ and $p_{j'}$ (of which there are polynomially many) and examining the entry of T indexed by the tuple $(C_j, p_j, p_{j'}, \log \ell, \{p_j\})$.

We next describe how to fill the table T . We proceed by iteratively increasing the value of r and in each iteration, filling all cells of T corresponding to the value of r in the current iteration. In the base case, $r = 1$ and so for every $C_j \in C$, it follows that $|C_r(C_j)| = 2$. Hence, for every $Z \in \binom{C}{\leq \log \ell}$, $p_j \in C_j$ and $p_{j'} \in C_{j'} \in C_1(C_j) \setminus \{C_j\}$, it is straightforward to decide whether $\zeta(C_j, p_j, p_{j'}, 1, Z) \neq \emptyset$ in polynomial time by simply trying all possible profiles. If it is non-empty, then we compute and add to $T(C_j, p_j, p_{j'}, 1, Z)$ an arbitrary element of $\zeta(C_j, p_j, p_{j'}, 1, Z)$. Otherwise we set $T(C_j, p_j, p_{j'}, 1, Z) = \emptyset$ (including for those indices which ζ maps to \emptyset by definition). Hence we may assume that we have filled the table T for all entries with $r = 1$. Note that this step takes time $n^{O(\log \ell)}$ since we have polynomially many choices for $C_j, p_j, C_{j'}, p_{j'}$ and $n^{O(\log \ell)}$ possibilities for Z and furthermore, determining the entry $T(C_j, p_j, p_{j'}, 1, Z)$ for each fixed choice of $C_j, p_j, C_{j'}, p_{j'}$ and Z as described above takes only polynomial time.

Now, suppose that $r > 1$ and inductively assume that for all $r' < r$, for all choices of $C_j, C_{j'}, p_j, p_{j'}$, and Z , we have filled the table entry $T(C_j, p_j, p_{j'}, r', Z)$ correctly. Now, fix a choice of $C_j, C_{j'}, p_j \in C_j, p_{j'} \in C_{j'}$, and $Z \in \binom{P}{\leq \log \ell}$ and we describe our procedure to fill the table entry $T(C_j, p_j, p_{j'}, r, Z)$. We check if there is a $\hat{s}_1 \in T(C_j, p_j, p_{j_1}, r-1, Z \cup \{p_{j'}\})$ and a $\hat{s}_2 \in T(C_{j'}, p_{j'}, p_{j_2}, r-1, Z \cup \{p_{j'}\})$ for some choice of p_{j_1} and p_{j_2} . If yes, then we set $T(C_j, p_j, p_{j'}, r, Z) = \hat{s}_1 \cdot \hat{s}_2$ and otherwise we set it to \emptyset . The second point of Lemma 2 indicates that composing \hat{s}_1 and \hat{s}_2 in this way indeed results in a profile that is contained in $\zeta(C_j, p_j, p_{j'}, r, Z)$, implying the correctness of our algorithm. Finally, the claimed running time bound follows from the fact that the table T has $n^{O(\log \ell)}$ entries in total, each of which is being filled in polynomial time using constant-time lookups into polynomially-many previously filled entries of T . \square

Observe that our algorithm actually achieves more than what is claimed in the statement of the theorem. Indeed, for every player p_j , our algorithm computes an equilibrium in which p_j is the winner (if one exists).

3.2 Beyond Win-Lose Games

In earlier sections we assumed coalitions to be only interested in winning the tournament. It is however natural to look at scenarios in which participants have a major incentive to progress as far as possible even if they do not win, as in the UEFA Champions League.

Here, we study coalitional games in which the teams' utility is determined by the round they reach. We call these games *beyond win/lose* (B-W/L) games. Formally, we define utility functions u_1, \dots, u_ℓ where for any $s = (p_1, \dots, p_\ell)$, $u_i(s) = k$ if and only if $p_i \in SE_S^k \setminus SE_S^{k+1}$. As with the win/lose scenario, in a NE no coalition can improve their utility by selecting different player.

DEFINITION 5 (BEYOND WIN/LOSE NE). *A profile $s = (p_1, \dots, p_\ell)$ is a: Beyond Win-Lose Nash Equilibrium (B-W/L NE), if for all $i, k \in [\log \ell]_{\geq 0}$, p'_i : if $u_i(s[p'_i]) = k$, then $u_i(s) \geq k$.*

In the remainder of this section we focus on structural results characterising B-W/L NE and algorithmic results addressing the problems of recognition and computation for each.

3.2.1 Algorithmic properties of B-W/L NE. We begin by providing a structural result which we then use for recognising and computing B-W/L NEs, capturing when a profile is a B-W/L NE.

LEMMA 3. *Consider a profile $s = (p_1, \dots, p_\ell)$ and let $j \in [\ell]$ such that p_j wins SE_S . Furthermore, suppose that $p_{j'} \in C_{j'}$ is the final opponent of p_j in SE_S . Then, s is a B-W/L NE if and only if (a) $s_1 = s|_{C_{\log \ell-1}(C_j)}$ and $s_2 = s|_{C \setminus C_{\log \ell-1}(C_j)}$ are both B-W/L NEs and (b) there is no $p'_{j'} \in C_{j'}$ which beats p_j such that $p'_{j'}$ wins $SE_{s_2[p'_{j'}]}^{\log \ell-1}$.*

PROOF. In the forward direction, suppose that s is a B-W/L NE. Observe that if Property (b) is violated then $C_{j'}$ can improve its position in $SE_{s_1[p'_{j'}]}$, so s is not a B-W/L NE. Moreover, suppose that one of s_1 or s_2 is not a B-W/L NE for the respective subtournaments. W.l.o.g., suppose that for some $C_i \in C \setminus \{C_j, C_{j'}\}$, $u_i(s_1[p'_i]) > u_i(s_1)$. That is, C_i is able to improve its position in the subtournament played by $C_{\log \ell-1}(C_j)$ by playing p'_i instead of p_i . Then, $u_i(s[p'_i]) > u_i(s)$. That is, C_i can also strictly improve its position in $s[p'_i]$, a contradiction to s being a B-W/L NE. The argument when $C_i \in C \setminus C_{\log \ell-1}(C_j)$ is identical and hence Property (a) also holds.

Conversely, suppose that Properties (a) and (b) hold and s is not a B-W/L NE. Let $p'_i \in C_i$ be such that $u_i(s[p'_i]) > u_i(s)$. Since Property (b) holds, it cannot be the case that $i \in \{j, j'\}$. Moreover, if $C_i \in C^{\log \ell-1}(C_j)$, then it contradicts our assumption that s_1 is a B-W/L NE and, otherwise, our assumption that s_2 is one. \square

This allows us to provide a P-time algorithm for recognising a B-W/L NE, which can be extended to the computation problem.

PROPOSITION 3. *Recognising a B-W/L NE is P-time solvable.*

PROOF. By Lemma 3, we have that for a given profile $s = (p_1, \dots, p_\ell)$ and $j \in [\ell]$ such that p_j is the winner of SE_S , s is a B-W/L NE if and only if there is no $j' \in [\ell] \setminus \{j\}$ and player $p'_{j'} \in C_{j'}$ such that $p'_{j'}$ beats the parent of p_j as well as all players in the set $\text{Opponent}^+[p_{j'}, p_{j'}]$. The P-time algorithm for recognising a B-W/L NE follows. \square

THEOREM 2. *Computing a B-W/L NE is P-time solvable.*

PROOF. Notice that if there is a B-W/L NE, then for $r = 1$, for every $C_j \in C$, the subtournament over $C_r(C_j)$ can be won by exactly one out of the two coalitions in this subtournament and the set of potentially winning players that can participate in a B-W/L NE can be easily computed in polynomial time. Indeed, suppose that C_j and $C_{j'}$ are the two coalitions in this subtournament. If every player in C_j loses to some player in $C_{j'}$ and every player in $C_{j'}$ loses to some player in C_j , then in any profile, at least one out of these two coalitions will be able to improve their final position by at least one place, implying the non-existence of a B-W/L NE. Hence, we identify all players in C_j ($C_{j'}$) that beats every player in $C_{j'}$ (respectively, C_j) by examining all matches between players in these two coalitions. This is clearly P-time computable.

We now inductively argue a similar property for $r > 1$. By the induction hypothesis, we have that if there is a B-W/L NE, then for every $C_j \in C$, the subtournaments over $C_r(C_j) \setminus C_{r-1}(C_j)$ and $C_{r-1}(C_j)$ can be won by exactly one coalition each (say, C' and C'' respectively) and the potentially winning players from each of these coalitions (denoted by $\mathcal{P}^*(C')$ and $\mathcal{P}^*(C'')$ respectively) that can participate in a B-W/L NE can be computed in polynomial time. Now, notice that for a B-W/L NE to exist, exactly one out of the following two cases must occur: either (i) there is a player in $\mathcal{P}^*(C')$ that beats every player in $\mathcal{P}^*(C'')$ or (ii) there is a player in $\mathcal{P}^*(C'')$ that beats every player in $\mathcal{P}^*(C')$. Moreover, the set of players satisfying (i) or (ii) can be computed in polynomial time. \square

4 DYNAMIC KNOCKOUT TOURNAMENTS

We now allow coalitions to choose players at each round of the tournament. In this model, a strategy of a coalition C consists of $\ell-1$ (not necessarily distinct) players representing, for each opposing coalition, a choice of a player of C to face said opposing coalition. We therefore model coalitional choices as dynamic strategy profiles $\sigma : C \rightarrow (C \rightarrow \mathcal{P})$. Specifically, for each distinct $i, j \in [\ell]$, $\sigma(C_i)(C_j)$ elects a member of the coalition C_i when facing coalition C_j . For every $C \in C$, let $\mathcal{S}(C)$ denote the set of all functions $C \rightarrow C$, i.e. the set of all possible responses to a given coalition. We require dynamic strategy profiles σ to be such that $\sigma(C_i) \in \mathcal{S}(C_i)$ for each $C_i \in C$. We use ρ_i to denote $\sigma(C_i)$ for every coalition $C_i \in C$. Equivalently, we represent σ as a tuple $(\rho_1, \dots, \rho_\ell)$ where $\rho_i \in \mathcal{S}(C_i)$, and denote with ρ_{ij} the player in C_i selected to play against C_j (also named $\rho_i(C_j)$). In this interpretation, for every $i \in [\ell]$, $\sigma(C_i)(C_i)$ is meaningless as a coalition does not face itself, and so we assigned an arbitrary player of C_i to be $\sigma(C_i)(C_i)$ (say the lexicographically smallest one).

Input representation. Notice that a dynamic strategy can be represented as a matrix M of the size $\ell \times \ell$, where ℓ is the number of coalitions. Then, each entry of the matrix $M[i, j]$ such that $i \neq j$ corresponds to $\sigma(C_i)(C_j)$. If $i = j$, $M[i, j]$ is empty. Given this representation, a dynamic strategy can be encoded in space $O(\ell^2 \log n)$.

While dynamic games typically warrant history-dependent strategies, the tree structure of coalitional knockout tournaments allows for a pair of coalitions to meet only at a unique round, if ever, as a function of the initial seeding. Therefore, each coalitional choice is effectively history-dependent when the initial seeding is fixed.

Mirroring the one-shot case, we represent tournaments between coalitions, as induced from a dynamic strategy profile. Let D_C be a

digraph on \mathcal{P} and σ a dynamic strategy profile over C . A *dynamic coalitional digraph* (D_C, σ) is a digraph with vertex set C where (i, j) is an arc of (D_C, σ) if and only if $(\sigma_i(C_j), \sigma_j(C_i))$ is an arc of D_C . Also, the notion of binomial arborescence SE_σ is lifted from the one-shot case in the natural manner.

Further, the solution concept is lifted from the one-shot case as expected, both for the win-lose (d-NE) and the beyond win-lose (B-W/L d-NE) games. The techniques used in the proofs of the results in this section are similar to their one-shot correspondents. Thus, we omit some of them.

4.1 Dynamic Win-Lose Games

In this section, we begin by lifting NEs to the current setting.

DEFINITION 6. A dynamic profile $\sigma = (\rho_1, \dots, \rho_\ell)$ is a dynamic Nash Equilibrium (d-NE), if for all i and for all ρ'_i : if $SE_{\pi, \sigma[\rho'_i]}^{\log \ell} = C_i$ then $SE_\sigma^{\log \ell} = C_i$.

So, a dynamic strategy profile is a dynamic NE if no losing coalition can become a winner by changing their strategy (which now corresponds to selecting a player for each opposing coalition) unilaterally. We establish a useful structural equivalence between d-NE, and NE for a one-shot coalitional tournament defined on an auxiliary tournament digraph where the players correspond to the set of the given parties' strategies. We first define the auxiliary digraph.

DEFINITION 7. Consider the digraph D_C . We define by D_C^{Dyn} the graph with vertex set $\bigcup_{C \in \mathcal{C}} \mathcal{S}(C)$ and arc set defined as follows. For every distinct $C_i, C_j \in \mathcal{C}$, for every $\rho_i \in \mathcal{S}(C_i)$ and $\rho_j \in \mathcal{S}(C_j)$, there is an arc (ρ_i, ρ_j) if $(\rho_i(C_j), \rho_j(C_i))$ is an arc in D_C and there is an arc (ρ_j, ρ_i) otherwise.

That is, D_C^{Dyn} has the arc (ρ_i, ρ_j) if and only if the player p_i elected to face C_j by C_i wins against the player C_j elected to face C_i . Observe that like D_C , D_C^{Dyn} is also a $|C|$ -partite tournament digraph with a partition for every $\mathcal{S}(C_i)$ where $C_i \in \mathcal{C}$.

This construction allows us to reason about the dynamic solution concepts in terms of one-shot tournaments, as the following characterisation shows.

LEMMA 4. Let σ be a dynamic strategy profile over the coalitions C . Then, σ is a d-NE for the dynamic SE-tournament over C using the pairwise results in D_C and seeding π iff $(\rho_1, \dots, \rho_\ell)$ is a NE for the one-shot SE-tournament over the coalitions $\mathcal{S}(C) = \{\mathcal{S}(C) \mid C \in \mathcal{C}\}$ using the pairwise results in D_C^{Dyn} and seeding π projected from the coalitions in C to those in $\mathcal{S}(C)$ in the natural way.

4.1.1 Algorithmic properties of d-NE. We now address the algorithmic questions of recognising and computing a dynamic NE. Observe that in D_C^{Dyn} , the number of vertices could be as large as $\ell \cdot m^\ell$ where m is the size of the largest coalition in D_C . As a result, although we can transfer our structural results on NEs from the one-shot setting to the dynamic setting using the graph D_C^{Dyn} , we cannot simply use the same algorithms because the running time, though polynomial in the size of D_C^{Dyn} , will no longer remain polynomial in the size of the actual input, which is linear in the size of the graph D_C . However, using appropriate queries that can

be answered in polynomial-time (in the size of D_C), we can still obtain a polynomial-time algorithm for recognising a d-NE. In the rest of the paper, polynomial-time refers to polynomial-time in the size of D_C .

Lemmata 1 and 4 entail the tractability of recognising d-NE.

PROPOSITION 4. *Recognising a d-NE is P-time solvable.*

PROOF. By invoking Lemma 1 on D_C^{Dyn} and the equivalence given by Lemma 4, we conclude that a given dynamic profile $\sigma = (\rho_1, \dots, \rho_\ell)$ won by C_j is a d-NE if and only if there is no $j' \in [\ell] \setminus \{j\}$ and strategy $\rho'_{j'} \in \mathcal{S}(C_{j'})$ such that $\rho'_{j'}$ beats all players in the set $\text{Opponent}^+[\rho, \rho_{j'}]$ for every player ρ on the ρ_j - $\rho_{j'}$ path in the b.a. SE_σ contained in the graph D_C^{Dyn} . Observe that given σ , the b.a. d- SE_σ and the set $\text{Opponent}^+[\rho, \rho_{j'}]$ for every player ρ on the ρ_j - $\rho_{j'}$ path can be computed in polynomial time by simply querying, for every pair of coalitions – “who is the winner between the two” if both played their respective strategies contained in σ . Now, for any $j' \in [\ell] \setminus \{j\}$, we can check whether there is a strategy $\rho'_{j'} \in \mathcal{S}(C_{j'})$ which beats the players selected by the respective coalitions by simply inspecting the arcs of the graph D_C . \square

Then, with reasoning similar to the one-shot case we obtain a quasi-polynomial-time algorithm for computing a d-NE if it exists.

THEOREM 3. *A d-NE can be computed in $n^{O(\log \ell)}$ -time.*

4.2 Beyond Dynamic Win-Lose Games

Let us further analyse the final case, that of tournaments where coalitions can modify their choices at each round and are interested in tournament progression. The solution studied in this section is a natural modification of that considered earlier.

We define utility functions u_1, \dots, u_ℓ where, for every profile $\sigma = (\rho_1, \dots, \rho_\ell)$, $u_i(\sigma) = k$ if and only if $C_i \in SE_\sigma^k \setminus SE_\sigma^{k+1}$. That is, Coalition C_i wins k rounds but not $k + 1$ rounds. This means that the utility of coalition i is dependent on the final round which they reach under a given strategy profile. The solution concept mirrors the one we defined previously.)

DEFINITION 8. A dynamic profile $\sigma = (\rho_1, \dots, \rho_\ell)$ is a Beyond Win-Lose Dynamic Nash Equilibrium (B-W/L d-NE), if for all $i, k \in [\log \ell]_{\geq 0}$, ρ'_i : if $u_i(\sigma[\rho'_i]) = k$, then $u_i(\sigma) \geq k$.

4.2.1 Algorithmic properties of B-W/L d-NE. Let us commence by the study of NE in the beyond win/lose, dynamic setting. We first provide a structural result on which we base the analysis of its algorithmic properties.

LEMMA 5. Consider a dynamic profile $\sigma = (\rho_1, \dots, \rho_\ell)$ and let $j \in [\ell]$ such that C_j wins SE_σ . Furthermore, suppose that $C_{j'}$ is the final opponent of C_j in SE_σ . Then, σ is a B-W/L d-NE if and only if $\sigma_1 = \sigma|_{C^{\log \ell - 1}(C_j)}$ and $\sigma_2 = \sigma|_{C \setminus C^{\log \ell - 1}(C_j)}$ are both B-W/L d-NEs and (b) $(\rho_j, \rho_{j'})$ is an NE.

PROOF. In the forward direction, consider a B-W/L d-NE $\sigma = (\rho_1, \dots, \rho_\ell)$. Observe that if Property (b) is violated then $C_{j'}$ can improve its position by choosing $\rho_{j'j} \in C_{j'}$ as a player which beats $\rho_{jj'}$, contradicting our assumption that σ is a B-W/L d-NE. Hence we conclude that Property (b) is satisfied. On the other

| | ONE-SHOT | | DYNAMIC | |
|----------|----------|------|---------|------|
| | CHECK | FIND | CHECK | FIND |
| W/L NE | P | Q-P | P | Q-P |
| B W/L NE | P | P | P | P |

Table 1: Summary of the algorithmic results in the paper.

hand, suppose that one of σ_1 or σ_2 is not a B-W/L d-NE for the respective subtournaments. Without loss of generality, suppose that for some $C_i \in C \setminus \{C_j, C_{j'}\}$, C_i is able to improve its position in the subtournament played by $C_{\log \ell - 1}(C_j)$ by choosing the strategy ρ'_i instead of ρ_i . Then, C_i can also strictly improve its position in $SE_{\sigma[\rho'_i]}$, a contradiction to σ being a B-W/L d-NE. The case when $C_i \in C \setminus C_{\log \ell - 1}(C_j)$ is identical and hence Property (a) holds.

Conversely, suppose that Properties (a) and (b) hold and suppose that σ is not a B-W/L NE. Let $\rho'_i \in \mathcal{S}(C_i)$ be such that $u_i(\sigma[\rho'_i]) > u_i(\sigma)$. Since Property (b) holds, it cannot be the case that $i \in \{j, j'\}$. Moreover, if $C_i \in C_{\log \ell - 1}(C_j)$, then it contradicts our assumption that σ_1 is a B-W/L d-NE and otherwise, it contradicts our assumption that σ_2 is a B-W/L d-NE. \square

The following algorithmic result follows from Lemma 5.

THEOREM 4. *Recognising and computing a B-W/L d-NE are P-time solvable.*

Notice that a coalitional knockout tournament can admit multiple dynamic B-W/L NEs for a fixed seeding, but these are all outcome-equivalent.

5 DISCUSSION

We introduced a model for knockout tournaments played between groups, to allow for each group to strategically select a member to take part in it. We carried out an algorithmic analysis of Nash equilibrium strategies under various setups occurring in practice, showing tractable results (polynomial-time or quasi-polynomial-time) for all cases. Table 1 gives an overview of our results.

We covered, for space reasons, only the treatment of Nash equilibria, but alternative solution concepts have also been studied. In particular, dominant strategy equilibria (DSE), where a profile $s = (p_1, \dots, p_\ell)$ is called a DSE if for all i and s' : if C_i wins $SE_{s'} \in C_i$ then C_i wins $SE_{s'}[p_i]$. In other words, no coalition can improve on their DSE strategy, irrespective of the choices of their opponents. Adapting this one-shot win-lose definition to the beyond win-lose and dynamic counterparts, analogously to what done with Nash equilibria, P-time algorithms can be shown for all cases in Table 1.

We foresee various potential research directions building on our work, which involve relaxing some of the assumptions, especially on participants knowledge and strategies, and looking at alternative solution concepts. Here we discuss a few concrete ones.

- (1) It is interesting to explore fast parameterized algorithms for finding a one-shot or dynamic W/L NE, e.g., parameterized by the number of coalitions ℓ .

- (2) All our results so far have assumed a tournament with a fixed seeding. However the choice of seeding may influence the tournament enormously, in particular some seedings may admit an equilibrium while others may not. Establishing the complexity of finding a seeding such that a given solution concept exists is therefore an important problem in this setup. This has repercussions for tournament fixing [26], as a malicious external attacker may be in a position to choose a seeding with a favourable winner in all resulting equilibria and high complexity barriers may act against it.
- (3) A third direction for future research is to establish the existential and algorithmic properties of the considered solution concepts under relaxed versions of our model. For instance, consider the scenario in which the *beating relation* or tournament digraph is relaxed to be stochastic. In this case, studying the existence and computational complexity of various equilibria based on the expected utility would be of high interest. This would bring us closer to understanding a setting that models real-world scenarios more faithfully than possible using only a “static” tournament digraph.
- (4) The combination of sequential decision-making in tournaments and the *beating relation* between players suggests novel solution concepts. Assume that coalition A can win a win-lose one-shot tournament, provided coalition B does not field player b , who defeats every player in A . We have that, every time B fields b , A is indifferent to the choice of any of their players. However there is a sense in which A *must* play a potential winner, *should something happen* to B . This suggests a trembling-hand interpretation of the *beating relation* (building on a classical solution concept [25]), which coalitions can try and exploit. Intuitively, profile s is a *trembling-hand tournament perfect equilibrium* if, for every coalition C_i , the strategy profile s is a Nash equilibrium at each sub-tournament, where C_i plays s_i as if they were able to reach that sub-tournament.
- (5) Another natural research direction concerns the possibility of cooperative and semi-cooperative behaviour among coalitions. Assume that coalition A *needs* coalition B to win the tournament, but B can never win. We envisage an interesting variant of endogenous games [15] arising in our tournaments. Before the tournament starts, A can transfer a part of the expected payoff to B , should B refrain from fielding a player *blocking* A . This should be seen as a form of manipulation, e.g., flipping results of a certain number of matches, where incentives comes from the players themselves
- (6) Finally, the extension to mixed strategies is a natural follow-up. It is then interesting to explore whether the knockout tournament structure can add any advantage in terms of (mixed-strategy) Nash equilibrium computation, as opposed to full-blown normal form games.

ACKNOWLEDGEMENTS

This work was supported by the Engineering and Physical Sciences Research Council [grant number EP/V007793/1].

REFERENCES

- [1] Alon Altman, Ariel D Procaccia, and Moshe Tennenholtz. 2009. Nonmanipulable selections from a tournament. In *Twenty-First International Joint Conference on Artificial Intelligence*. IJCAI, 686–690.
- [2] Haris Aziz, Serge Gaspers, Simon Mackenzie, Nicholas Mattei, Paul Stursberg, and Toby Walsh. 2014. Fixing a balanced knockout tournament. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAAI, 552–558.
- [3] Haris Aziz, Serge Gaspers, Simon Mackenzie, Nicholas Mattei, Paul Stursberg, and Toby Walsh. 2018. Fixing balanced knockout and double elimination tournaments. *Artificial Intelligence* 262 (2018), 1–14.
- [4] Felix Brandt, Markus Brill, and Bernhard Harrenstein. 2016. Tournament Solutions. In *Handbook of Computational Social Choice*, F Brandt, V Conitzer, U Endriss, J Lang, and A. D. Procaccia (Eds.). Cambridge University Press, 453–474.
- [5] Felix Brandt, Markus Brill, and Paul Harrenstein. 2018. Extending tournament solutions. *Social Choice and Welfare* 51, 2 (2018), 193–222.
- [6] Felix Brandt, Markus Brill, Hans Georg Seedig, and Warut Suksompong. 2018. On the structure of stable tournament solutions. *Economic Theory* 65, 2 (2018), 483–507.
- [7] Felix Brandt and Felix A. Fischer. 2008. Computing the minimal covering set. *Mathematical Social Sciences* 56, 2 (2008), 254–268.
- [8] Felix Brandt, Felix A. Fischer, Paul Harrenstein, and Maximilian Mair. 2010. A computational analysis of the tournament equilibrium set. *Social Choice and Welfare* 34, 4 (2010), 597–609.
- [9] Markus Brill and Vincent Conitzer. 2015. Strategic Voting and Strategic Candidacy. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI, 819–826.
- [10] Krishnendu Chatterjee, Rasmus Ibsen-Jensen, and Josef Tkadlec. 2016. Robust Draws in Balanced Knockout Tournaments. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. IJCAI, 172–179.
- [11] Piotr Faliszewski, Laurent Gourvès, Jérôme Lang, Julien Lesca, and Jérôme Monnot. 2016. How hard is it for a party to nominate an election winner?. In *Twenty-Fifth International Joint Conference on Artificial Intelligence*. IJCAI, 257–263.
- [12] David Fisher and Jennifer Ryan. 1995. Tournament games and positive tournaments. *Journal of Graph Theory* 19, 2 (1995), 217–236.
- [13] Paul Harrenstein, Grzegorz Lisowski, Ramanujan Sridharan, and Paolo Turrini. 2021. A Hotelling-Downs Framework for Party Nominees. In *Twentieth International Conference on Autonomous Agents and Multiagent Systems*. AAMAS, 593–601.
- [14] Noam Hazon, Paul Dunne, Sarit Kraus, and Michael Wooldridge. 2008. How to rig elections and competitions. In *Second International Workshop on Computational Social Choice*. COMSOC, 301–312.
- [15] Matthew O. Jackson and Simon Wilkie. 2005. Endogenous Games and Mechanisms: Side Payments Among Players. *Review of Economic Studies* 72, 2 (2005), 543–566.
- [16] Michael P Kim, Warut Suksompong, and Virginia Vassilevska Williams. 2017. Who can win a single-elimination tournament? *SIAM Journal on Discrete Mathematics* 31, 3 (2017), 1751–1764.
- [17] Aleksei Y. Kondratev and Vladimir V. Mazalov. 2019. Tournament solutions based on cooperative game theory. *International Journal of Game Theory* 49, 1 (2019), 119–145.
- [18] Christine Konicki and Virginia Vassilevska Williams. 2019. Bribery in Balanced Knockout Tournaments. In *Eighteenth International Conference on Autonomous Agents and Multi Agent Systems*. AAMAS, 2066–2068.
- [19] Gilbert Laffond, Jean Lainé, and Jean-François Laslier. 1996. Composition-consistent tournament solutions and social choice functions. *Social Choice and Welfare* 13, 1 (1996), 75–93.
- [20] G. Laffond, J. F. Laslier, and M. Le Breton. 1993. The bipartisan set of a tournament game. *Games and Economic Behavior* 5 (1993), 182–201. Issue 1.
- [21] Jean-Francois Laslier. 1997. *Tournament solutions and majority voting*. Springer Verlag.
- [22] Reshef Meir. 2018. *Strategic Voting*. Morgan & Claypool.
- [23] JW Moon. 1968. *Topics on tournaments*. Holt, Reinhart and Winston.
- [24] Hervé Moulin. 1986. Choosing from a tournament. *Social Choice and Welfare* 3, 4 (1986), 271–291.
- [25] Reinhard Selten. 1975. Reexamination of the perfectness concept for equilibrium points in extensive games. *International Journal of Game Theory* 4 (1975), 25–55.
- [26] Virginia Vassilevska Williams. 2010. Fixing a Tournament. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*. AAAI, 895–900.
- [27] Thuc Vu, Alon Altman, and Yoav Shoham. 2009. On the complexity of schedule control problems for knockout tournaments. In *Eighth International Conference on Autonomous Agents and Multiagent Systems*. AAMAS, 225–232.