Teams in Online Scheduling Polls: Game-Theoretic Aspects[*]

Robert Bredereck,[1] Jiehua Chen,[2] Rolf Niedermeier[2]
[1]University of Oxford, United Kingdom, robert.bredereck@cs.ox.ac.uk
[2]TU Berlin, Germany, {jiehua.chen, rolf.niedermeier}@tu-berlin.de

Svetlana Obraztsova,[3] Nimrod Talmon[4]
[3]I-CORE, Hebrew University of Jerusalem, Israel, svetlana.obraztsova@gmail.com
[4]I-CORE, Weizmann Institute of Science, Israel, nimrodtalmon77@gmail.com

Abstract

Consider an important meeting to be held in a team-based organization. Taking availability constraints into account, an online scheduling poll is being used in order to decide upon the exact time of the meeting. Decisions are to be taken during the meeting, therefore each team would like to maximize its relative attendance (i.e. the proportional number of its team members attending the meeting). We introduce a corresponding game, where each team can declare a lower total availability in the scheduling poll in order to improve its relative attendance—the pay-off. We are especially interested in situations where teams can form coalitions.

We provide an efficient algorithm that, given a coalition, finds an optimal way for each team in a coalition to improve its pay-off. In contrast, we show that deciding whether such a coalition exists is NP-hard. We also study the existence of Nash equilibria: Finding Nash equilibria for various small sizes of teams and coalitions can be done in polynomial time while it is coNP-hard if the coalition size is unbounded.

1 Introduction

An organization is going to hold a meeting, where people are to attend. Since people come from different places and have availability constraints, an open online scheduling poll is taken to decide upon the meeting time. Each individual can approve or disapprove of each of the suggested time slots. In order to have the highest possible attendance, the organization will schedule the meeting at a time slot with the maximum sum of declared availabilities. During the meeting, proposals will be discussed and decisions will be made. Usually, people have different interests in the decision making, e.g. they are from different teams who each want their own proposals to be put through. We consider people with the same interest as members of the same *team* and as a result, each team (instead of each individual) may declare the number of its members that can attend the meeting at each suggested time slot.

For a simple illustration, suppose that three teams, t_1, t_2, and t_3, are about to hold a meeting, either at 9am or at 10am. Two members from t_1, one member from t_2, and three members from t_3 are available at 9am, while exactly two members of each team are available at 10am. The availabilities of the teams can be illustrated as an integer matrix (illustrated on the right hand side):

$$A := \begin{array}{c} \\ \\ \\ \end{array}\begin{array}{cc} c_1 & c_2 \end{array} \\ A := \begin{pmatrix} 2 & 2 \\ 1 & 2 \\ 3 & 2 \end{pmatrix} \begin{array}{c} t_1 \\ t_2 \\ t_3 \end{array}$$

A time slot is a winner if it receives the maximum sum of declared availabilities. Thus, if the three teams declare their true availabilities, then both 9am and 10am co-win (since six people in total are available at 9am and 10am each), and the meeting will be scheduled at either 9am or 10am.

Now, if a team (i.e. people with the same interest) wants to influence any decision made during the meeting, then it will want to send as many of its available team members to the meeting as possible because this will maximize its *relative power*—the proportion of its own attendees. For our simple example, if the meeting is to be held at 9am, then the relative powers of teams t_1, t_2, and t_3 are $1/3$, $1/6$, and $1/2$, respectively. A team may change its availabilities declared in the poll from time to time. However, teams must not report a number which is higher than their true availability since it cannot send more members than available. Given this constraint, it is interesting to know whether any team can increase its relative attendance by misreporting its availability, possibly changing the winning time slot to one where their relative powers are maximized.

For the case where several time slots co-win, it is not clear which co-winning time slot will be used. To be on the safe side, the teams must maximize the relative power of *each* co-winning time slot. In other words, our teams are pessimistic and consider their *pay-off* as the *minimum* over all the relative powers at each co-winning time slot. In our example, this means that the pay-off of team t_2 would be $1/6$, since this is its relative power at 9am, which is smaller than its relative power, $1/3$, at 10am. The pay-offs of teams t_1 and t_3 are both $1/3$. In this case, team t_2 can be strategic by updating its availability and declare zero availability at 9am; as a result, the meeting would be held at 10am, where team t_2 has better pay-off with relative number of $1/3$.

We do not allow arbitrary deviations from the real availabilities of teams; specifically, we do not allow a team to declare as available a higher number than actually available. Further, we do not allow a team to send more team members

[*]R. Bredereck is supported by the DFG fellowship BR 5207/2. N. Talmon is supported by an I-CORE ALGO fellowship. Main work done while R. Bredereck and N. Talmon were with TU Berlin.

to the meeting than it declared as available, because this is often mandated by the circumstances. For example, the organizer might need to arrange a meeting room and specify the number of participants in the meeting up-front (similarly, if the meeting is to be carried in a restaurant, the number of chairs at the table shall be decided beforehand); or the organizer might need to obtain buses to transport the participants. Thus, the teams must send exactly the declared number of members to the meeting. For instance, it is not possible for team t_2 to declare 3 at 9am since only one of its team members is available. A formal description of the corresponding game, called *team power game* (TPG, in short), and a discussion on our example are given in Section 2.

As already remarked, to improve the pay-off, a team may lie about the number of its available members. Sometimes, teams can even form a coalition and update their availabilities strategically. In our example, after team t_2 misreported its availabilities such that each team receives a pay-off of $1/3$, teams t_1 and t_3 may collaborate: if both teams keep their declared availabilities at 9am but declare zero availability at 10am (note that team t_2 does not change its updated availabilities), then 9am will be the unique winner (with total availability of 5); as a result, t_1 and t_3 receive better pay-offs of $2/5$ and $3/5$, respectively. Such a successful deviation from the declared availabilities of the teams in a coalition (while keeping the declared availabilities of the teams not in the coalition unchanged) is called an *improvement step*.

After some teams perform an improvement step, other teams may also want to update their availabilities to improve. This iterative process leads to the question of whether there is a stable situation where improvement is impossible—a *Nash equilibrium*. Of course, when searching for equilibria, it is natural to ask how hard it is to decide whether an improvement step is possible.

In this paper, we are interested in the computational complexity of the following problems: (1) finding an improvement step (if it exists) for a specific coalition, (2) finding an improvement step (if it exists) for any coalition, and (3) finding a *t-strong Nash equilibrium* (if it exists).

Main Contributions. We show that, depending on the size of the coalition (i.e., the number of teams that could deviate from their declared availabilities), the computational complexity of finding an improvement step for a given coalition and deciding whether an improvement step exists for an arbitrary coalition ranges from being polynomial-time solvable to being NP-hard; further, deciding whether an improvement step exists for any coalition of size at most t is W[2]-hard when parameterizing by the coalition size t. We show that a 1-strong Nash equilibrium always exists for some special profiles and we provide a simple polynomial-time algorithm for finding it in these cases. Finally, we show that deciding whether a t-strong Nash equilibrium exists is coNP-hard. Our results are summarized in Table 1. Due to space constraints, many proofs are deferred to our technical report (Bredereck et al. 2016).

Table 1: Complexity results for the team power game. "Unary" (resp. "Binary") means that the input and the strategy profiles are encoded in unary (resp. binary). Variable t stands for the number of teams in a coalition, while $a_{\max}$ stands for the maximum true availability. An entry labeled with "P" means polynomial-time solvability. An entry labeled with "FPT for k" means solvability in $f(k) \cdot |I|^{O(1)}$ time, where f is a function solely depending on k and $|I|$ denotes the size of the given input. An entry labeled with "W[2]-hard for k" implies that the corresponding problem is not "FPT for k" unless $\mathsf{W[2]} = \mathsf{FPT}$ (this is considered unlikely in parameterized complexity theory).

(1)	Finding an improvement step for a given coalition	
	Unary	in P (Thm. 1)
	Binary	in FPT for t (Thm. 2)[1]
(2)	Deciding the existence of an improvement step for any coalition	
	Binary	in P for constant t (Cor. 1)
	$a_{\max} = 1$	NP-complete (Thm. 3)
	$a_{\max} = 1$	W[2]-hard for t (Thm. 3)
(3)	Finding a 1-strong Nash equilibrium	
	$a_{\max} \leq 3$	in P, always exists (Thm. 4)
	$a_{\max} \geq 4$	open (Rem. 2)
	Finding a 2-strong Nash equilibrium	
	$a_{\max} = 1$	in P, always exists (Prop. 1)
	$a_{\max} \geq 2$	open, does not always exist
	Deciding the existence of a t-strong Nash equilibrium	
	$a_{\max} = 2$	coNP-hard (Thm. 5)

[1] We conjecture it to be even in P. Strong NP-hardness is excluded by Theorem 1.

Related Work. Recently, online scheduling polls such as Doodle / Survey Monkey caught the attention of several researches. Reinecke et al. (2013) initiated empirical investigations of scheduling polls and identified influences of national culture on people's scheduling behavior, by analyzing actual Doodle polls from 211 countries. Zou, Meir, and Parkes (2015) also analyzed actual Doodle polls, and devised a model to explain their experimental findings. They observed that people participating in open polls tend to be more "cooperative" and additionally approve time slots that are very popular or unpopular; this is different to the behavior of people participating in closed polls. Obraztsova et al. (2015) formally modeled the behavior observed by Zou, Meir, and Parkes (2015) as a game, where approving additional time slots may result in pay-off increase. While the game introduced by Obraztsova et al. (2015) captures the scenario that each individual player tries to appear to be cooperative, our team power game models the perspective that each individual team (player) as a whole tries to maximize its relative power in the meeting, which means that approving more time slots is not necessarily a good strategy.

Quite different in flavor, Lee (2014) considered a computational problem from the point of view of the poll initiator, whose goal is to choose the time slots to poll over, in order to optimize a specific cost function. Finally, since scheduling polls might be modeled as approval elections, we mention the vast amount of research done on approval elections in general, e.g., (Brams and Fishburn 1978) and on iterative approval voting in particular, e.g., (Dery et al. 2015; Lev and Rosenschein 2012; Meir et al. 2010).

2 Preliminaries

We begin this section by defining the rules of the game which is of interest here. Then, we formally define the related computational problems we consider in this paper. Throughout, by $[n]$ we mean the set $\{1, 2, \dots, n\}$.

Rules of the Game. The game is called the *team power game* (TPG, in short). It consists of n players, the *teams*, $t_1, t_2, \dots, t_n$, and m possible *time slots*, $c_1, c_2, \dots, c_m$. Each team t_i is associated with a *true availability vector* $A_i = (a_i^1, a_i^2, \dots, a_i^m)$, where $a_i^j \in \mathbb{N}$ is the (true) availability of team t_i for time slot c_j. Importantly, each team is only aware of its own availability vector. During the game, each team t_i announces a *declared (availability) vector* $B_i = (b_i^1, b_i^2, \dots, b_i^m)$, where $b_i^j \leq a_i^j$ is the declared availability of team t_i for time slot c_j; using standard game-theoretic terms, we define the *strategy* of team t_i to be its declared availability vector B_i. We use A and B to denote the matrices consisting of a row for each team's true and declared availability vectors. That is, for $i \in [n]$ and $j \in [m]$, $A := (a_i^j), B := (b_i^j)$. Given a declared availability matrix B, the co-winners of the corresponding scheduling poll, denoted as $\mathsf{winners}(B)$, are the time slots with the maximum sum of declared availabilities: $\mathsf{winners}(B) := \underset{c_j \in \{c_1, c_2 \dots, c_m\}}{\arg\max} \{\sum_{i \in [n]} b_i^j\}$.

Before we define the *pay-off* of each team, we introduce the notion of *relative power*. The relative power $\mathsf{team\text{-}power}(B, t_i, c_j)$ of team t_i at time slot c_j equals the number of members from t_i who will attend the meeting at time slot c_j, divided by the total number of attendees at this time slot: $\mathsf{team\text{-}power}(B, t_i, c_j) := \frac{b_i^j}{\sum_{k \in [n]} b_k^j}$.

In order to define the pay-off of each team, we need to decide how to proceed when several time slots tie as co-winners. In this paper we consider a *maximin* version of the game, where ties are broken adversarially. That is, the pay-off of team t_i is defined to be the minimum, over all co-winners, of its *relative power*:

$$\mathsf{pay\text{-}off}(B, t_i) := \min_{c_j \in \mathsf{winners}(B)} \mathsf{team\text{-}power}(B, t_i, c_j).$$

When we refer to an *input* for TPG, we mean a true availability matrix $A \in \mathbb{N}^{n \times m}$ where each row A_i represents the true availability of a team t_i for the m time slots. When we refer to a *strategy profile* (in short, *strategy*) for input A we mean a declared availability matrix $B \in \mathbb{N}^{n \times m}$ where each row B_i represents the declared availability vector of team t_i.

Computational Problems Related to the Game. Given a *coalition*, i.e., a subset of teams, a deviation of the teams in the coalition from their current strategy profile is an *improvement step* if, by this deviation, each team in the coalition strictly improves its pay-off. Given a positive integer t, a *t-strong Nash equilibrium* for some input A is a strategy profile B such that no coalition of at most t teams has an improvement step wrt. B. We are interested in the following computational questions:

1. Given an input, a strategy profile, and a coalition of at most t teams, does this coalition admit an improvement step compared to the given strategy profile?

2. Given an input, a strategy profile, and a positive integer t, is there any coalition of at most t teams which has an improvement step compared to the given strategy profile?

3. Given an input and a positive integer t, does a *t-strong Nash equilibrium* for this input exist?

We are particularly interested in understanding the dependency of the computational complexity of the above problems on the number t of teams in a coalition. Specifically, we consider (1) t being a constant (modeling situations where not too many teams are willing to cooperate or where cooperation is costly) and (2) t being unbounded.

Illustrating Example. Consider the input matrix A given in Section 1, which specifies the true availabilities of three teams t_1, t_2, t_3 over two time slots c_1, c_2. If all teams declared their true availabilities, then both time slots win with total availability 6. The pay-offs of the teams t_1, t_2, t_3 are $1/3, 1/6, 1/3$, respectively. Team t_2 can improve its pay-off by declaring $(0, 2)$ (i,e., declaring 0 for c_1 and 2 for c_2). As a result, c_2 would become the unique winner with total availability 6 and team t_2 would receive a better pay-off: $1/3$. Thus, the profile B for A where all teams declare their true availabilities (i.e., where $B = A$) is not a 1-strong Nash equilibrium. Nevertheless, A does admit the following 1-strong Nash equilibrium:

$$B' := \begin{pmatrix} 2 & 2 \\ 1 & 2 \\ 3 & 0 \end{pmatrix}$$

The declared availability matrix B', however, is not a 2-strong Nash equilibrium, since if team t_1 and t_2 would form a coalition and declare the same availability vector $(0, 2)$, then c_2 would be the unique winner with total availability 4 and both t_1 and t_2 would have a better pay-off: $1/2$.

3 Improvement Steps

We begin with the following lemma, which basically says that, in search for an improvement step, a fixed coalition of teams needs only to focus on a single time slot.

Lemma 1. *If a coalition has an improvement step wrt. a strategy profile B, then it also has an improvement step $E = (e_i^j)$ wrt. B, where there is one time slot c_k such that each team t_i in the coalition declares zero availability for all other time slots (i.e., $e_i^j = 0$ holds for each team t_i in the coalition and each time slot $c_j \neq c_k$).*

By Lemma 1, we know that if a fixed coalition has an improvement step for a strategy profile B, then it admits an improvement step that involves only one time slot. Assume that it is time slot c_k. In order to compute an improvement step for the coalition, we first declare zero availabilities for the teams in the coalition, for all other time slots. Then, we have to declare specific availabilities for the teams in the coalition, for time slot c_k. This is where the collaboration between the teams comes into play: even though each team, in order to improve its pay-off, might wish to declare as high as possible availability for time slot c_k (i.e., its true availability), the teams shall collaboratively decide on the declared availabilities, since a too-high declared availability for one team might make it impossible for another team (even when declaring the maximum possible amount, i.e., the true availability) to improve its pay-off. It turns out that this problem is basically equivalent to the following problem (which, in our eyes, is interesting also on its own).

Relation to Horn Constraint Systems. Consider the following number problem, called t-THRESHOLD COVERING, which, given a natural number vector $(a_1, a_2, \dots, a_t) \in \mathbb{N}^t$, a rational number vector $(p_1, p_2, \dots, p_t) \in \mathbb{Q}^t$ with $\sum_{i \in [t]} p_i \leq 1$, and a natural number $p \in \mathbb{N}$, searches for a natural number vector $(x_1, x_2, \dots, x_t)$ where for each $i \in [t]$ the following holds: (1) $1 \leq x_i \leq a_i$ and (2) $x_i/(p + \sum_{i \in [t]} x_i) > p_i$.

Intuitively, the vector $(a_1, \dots, a_t)$ corresponds to the true availabilities of the teams in the coalition in time slot c_k, while the solution vector $(x_1, \dots, x_t)$ corresponds to the declared availabilities of the teams in the coalition in time slot c_k; accordingly, the first constraint makes sure that each declared availability is upper-bounded by its true availability. Further, the vector $(p_1, \dots, p_t)$ corresponds to the current pay-offs of the teams in the coalition, while p corresponds to the sum of the declared availabilities of the teams not in the coalition at time slot c_k; accordingly, the second constraint makes sure that, for each team in the coalition, the new pay-off is strictly higher than its current pay-off. More formally, we argue that the coalition $\{t_1, t_2, \dots, t_t\}$ has an improvement step compared to strategy profile B, involving only time slot c_k, if and only if the instance (A^*, P, p) for t-THRESHOLD COVERING has a solution, where $A^* := (a_1^k, \dots, a_t^k)$, $P := (\mathsf{pay\text{-}off}(B, t_1), \dots, \mathsf{pay\text{-}off}(B, t_t))$, and $p := \sum_{i \in [n] \setminus [t]} b_i^k$.

Remark 1. *Since the values p_i ($i \in [t]$) are rational numbers, we can rearrange the second constraint in the description of t-*THRESHOLD COVERING *to obtain an integer linear feasibility problem. This means that t-*THRESHOLD COVERING *is a special variant of the so-called* HORN CONSTRAINT SYSTEM *problem which, given a matrix $U = (u_{i,j}) \in \mathbb{R}^{n' \times m'}$ with each row having at most one positive element, a vector $b \in \mathbb{R}^{n'}$, and a positive integer d, decides the existence of an integer vector $x \in \{0, 1, \dots, d\}^{m'}$ such that $U \cdot x \geq b$;* HORN CONSTRAINT SYSTEM *is weakly* NP*-hard and can be solved in pseudo-polynomial time (Lagarias1985 ; Lagarias1985).*

Taking a closer look at t-THRESHOLD COVERING, we observe the following: if we would know the sum of the variables $(x_1, \dots, x_t)$, then we would be able to directly solve our problem by checking every constraint and taking the smallest feasible value (i.e., given $\sum_{i \in [t]} x_i$, we would set each x_i to be the minimum over all values satisfying $x_i/(p + \sum_{i \in [t]} x_i) > p_i$). This yields a simple polynomial-time algorithm for finding an improvement step for the likely case where all availabilities are polynomially upper-bounded in the input size; technically, this means where the input profile A is encoded in unary.

Theorem 1. *Consider an input A and a strategy profile B. Let s be the sum of all entries in A. Finding an improvement step (if it exists) for a given coalition is solvable in $O(s^2)$ time.*

Indeed, t-THRESHOLD COVERING can be reduced to finding the sum $\sum_{i \in [t]} x_i$. If the input is encoded in binary, however, then this sum might be exponentially large in the number of bits that encode our input, thus we cannot simply enumerate all possible values. If the coalition size t or a certain parameter ℓ that measures the number of “large” true availabilities is a constant, then we still have polynomial-time algorithms for which the degree of the polynomial in the running time does not depend on the specific parameter value. Specifically, by the famous Lenstra’s theorem (1983, later improved by Kannan (1987) and by Frank and Tardos (1987)), we have the following result.

Theorem 2. *Consider an input A and a strategy profile B. Let L be the length of the binary encoding of A. For each of the following times T, there is a T-time algorithm that finds an improvement step, compared to B, for a given coalition of t teams:*

1. $T = O(t^{2.5t + o(t)} \cdot L^2)$ and

2. for each constant value c, $T = f(\ell_c) \cdot t^2 \cdot L^{c^2+2}$,

where f is a computable function and $\ell_c := \max_j |\{i \in [t] \colon a_i^j > L^c\}|$ is the maximum over the numbers of teams t_i in the coalition that have true availabilities a_i^j with $a_i^j > L^c$ for the same time slot c_j.

Using Theorem 2, and checking all $\sum_{i=1}^{t} \binom{n}{i}$ possible coalitions of size at most t, we obtain the following.

Corollary 1. *Given an input and a strategy profile, we can find, in polynomial time, a coalition of a constant number of teams and, for this coalition, find an improvement step compared to the given profile.*

In general, however, deciding whether an improvement step exists is computationally intractable as the next result shows. We briefly note that, under standard complexity assumptions, a problem being W[2]-hard for parameter k presumably excludes any algorithm with running time $f(k) \cdot |I|^{O(1)}$, where f is a computable function depending only on k and $|I|$ is the size of the input.

Theorem 3. *Given an input and a strategy profile, deciding whether there is a coalition of size t that has an improvement step is* W[2]*-hard wrt. t even if all teams are of size one. It remains* NP*-complete if there is no restriction on the coalition size.*

Proof. (Sketch). To show W[2]-hardness, we provide a parameterized reduction from the SET COVER problem, which is W[2]-complete wrt. the set cover size k (Downey and Fellows 2013): Given sets $\mathcal{F} = \{S_1, \dots, S_m\}$ over a universe $U = \{u_1, \dots, u_n\}$ of elements and a positive integer k, SET COVER asks whether there is a size-k *set cover* $\mathcal{F}' \subseteq \mathcal{F}$, i.e., $|\mathcal{F}'| = k$ and $\bigcup_{S_i \in \mathcal{F}'} S_i = U$. The idea of such a parameterized reduction is, given a SET COVER instance $(\mathcal{F}, U, k)$, to produce, in $f(k) \cdot (|\mathcal{F}| + |U|)^{O(1)}$ time, an equivalent instance (A, B, t) such that $t \leq g(k)$, where f and g are two computable functions. Let $(\mathcal{F}, U, k)$ denote a SET COVER instance. For technical reasons, we assume that each set cover contains at least three sets.

Time slots. For each element $u_j \in U$, we create one *element slot* e_j. Let $E := \{e_1, \ldots, e_n\}$ denote the set containing all element slots. We create two special time slots: α (the original winner) and β (the potential new winner).

Teams and true availabilities $A = (a_i^j)$. For each set $S_i \in \mathcal{F}$, we create a *set team* t_i that has true availability 1 at time slot α, at time slot β, and at each element slot e_j with $u_j \in S_i$. We introduce several dummy teams, as follows. Intuitively, the role of these dummy teams is to allow to set specific sums of availabilities for the time slots; the crucial observation in this respect is that the dummy teams do not have any incentive to change their true availabilities, therefore we can assume that they do not participate in any coalition. For each element u_j, let $\#(u_j)$ denote the number of sets from $\mathcal{F}$ that contain u_j. For each element slot e_j, we create $(2m - 1 - \#(u_j))$ dummy teams such that each of these dummy teams has availability 1 at element slot e_j and availability 0 for all other time slots. Similarly, for time slot α, we create m additional dummy teams, each of which has availability 1 for time slot α and availability 0 for all other time slots. For time slot β, we create $2m - 1 - k$ further dummy teams, each of which has availability 1 for time slot β and availability 0 for all other time slots.

Declared availabilities $B = (b_i^j)$. Each dummy team declares availability for the time slot where it is available. Each set team declares availability for all time slots where it is available except for time slot β where all set teams declare availability 0.

We set the size of the coalition t to be k. This completes the reduction which can be computed in polynomial time. Indeed, it is also a parameterized reduction. A formal correctness proof as well as the extension to the case of unrestricted coalition sizes to show the NP-hardness result are deferred to our technical report (Bredereck et al. 2016). □

Taking a closer look at the availability matrix constructed in the proof of Theorem 3, we observe the following.

Corollary 2. *Deciding the existence of an improvement step for any coalition is* NP*-hard, even for very sparse availability matrices, i.e., even if each team has only one team member and is truly available at no more than four time slots.*

4 Nash Equilibria

We move on to consider the existence of Nash equilibria. Somewhat surprisingly, it seems that, a 1-strong Nash equilibrium always exists. Unfortunately, we can only prove this when the maximum availability $a_{\max} := \max_{i\in[n],j\in[m]} a_i^j$ is at most three. Extending our proof strategy to $a_{\max} \geq 4$ seems to require a huge case analysis.

Theorem 4. *If the maximum availability* $a_{\max}$ *is at most three, then* TPG *always admits a* 1*-strong Nash equilibrium.*

Proof. (Sketch). Let $A = (a_i^j)$ be the input profile. We begin by characterizing two simple cases for which 1-strong Nash equilibria always exist.

Safe single-team slot. Suppose that a time slot c_j exists where only one team, t_i, is available with some availability a^* (i.e., $a_i^j = a^*$), all other teams are not available in this time slot (i.e., $a_{i'}^j = 0$ for all $i' \neq i$), and no other team, $t_{i'}, i' \neq i$, is available with availability greater than a^* at any time slot. Then, we obtain a 1-strong Nash equilibrium $B = (b_i^l)$ by setting $b_i^j := a_i^j$, and, for each $i' \in [n]$ and each $j' \neq j$, setting $b_{i'}^{j'} := 0$; to see why B is a Nash equilibrium, notice that the only team (namely t_i) i.e. available at time slot c_j already receives the best possible pay-off (namely 1) and no other team can prevent c_j from being a co-winner, which would be necessary to improve their pay-off (which is 0). We call such time slot c_j a *safe single-team slot*.

Safe multiple-team slot. Suppose that a time slot c_j exists where multiple teams have non-zero true availabilities and no single team is powerful enough to prevent c_j from co-winning, by declaring zero availability. That is, for each team t_i and each time slot $c_{j'} \neq c_j$, it holds that $a_i^{j'} \leq \sum_{i' \neq i} a_{i'}^j$. Again, we obtain a 1-strong Nash equilibrium $B = (b_i^j)$ by setting $b_l^j := a_l^j$ for each team t_l and setting $b_l^{j'} := 0$ for each other time slot $c_j \neq c_{j'}$. We call such time slot c_j a *safe multiple-team slot*. For example, the following input profile contains two safe multiple-team slots, namely c_1 and c_4:

$$\begin{array}{ccc} & \begin{array}{cccc} c_1 & c_2 & c_3 & c_4 \end{array} & \\ A := & \begin{pmatrix} 1 & 2 & 0 & 0 \\ 2 & 0 & 2 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 3 \end{pmatrix} & \begin{array}{c} t_1 \\ t_2 \\ t_3 \\ t_4 \end{array} \end{array}$$

We are ready to consider instances $a_{\max} \leq 3$. We only show the case with $a_{\max} = 2$. The remaining cases are deferred our technical report (Bredereck et al. 2016).

Instances with $a_{\max} = 2$. Consider the maximum availability sum x of all time slots, i.e., the maximum column sum of the matrix A. Clearly, $x \geq a_{\max}$. We proceed by considering the different possible values of x.

Cases with $x = 2$: If x is two, then there is a time slot where only one team is available with availability $a_{\max} = 2$. Thus, there is a safe single-team slot.

Cases with $x = 3$: If x is three, then, since we have $a_{\max} = 2$, it follows that either (1) there is a safe single-team slot where only one team is available with availability $a_{\max} = 2$ or (2) there is a time slot c_j where a single team t_i has availability $a_{\max} = 2$ and another team $t_{i'}$ has availability 1. In the first case, there is a safe single-team, so let us consider the second case. To this end, let c_j be the time slot such that a single team t_i has availability $a_{\max} = 2$ and another team $t_{i'}$ has availability 1. Next, we show how to construct a 1-strong Nash equilibrium $B = (b_i^j)$. First, for each team t_i, set $b_i^j := a_i^j$ and $b_i^{j'} := 0, j' \neq j$. This makes time slot c_j the unique winner. Team t_i receives pay-off $2/3$ and team $t_{i'}$ receives pay-off $1/3$. Second, for each time slot $c_{j'} \neq c_j$, if $a_{i'}^{j'} > 0$, then set $b_{i'}^{j'} := 1$; otherwise, find any team $t_k \neq t_i$ with non-zero availability $a_k^{j'} = 1$ and set $b_k^{j'} := 1$. In this way, every time slot except c_j has total availability one (if there is at least one team with non-zero availability for this slot). Thus, c_j remains a unique winner and the declared total availabilities of other time slots make it impossible for any team to improve: First, team t_i cannot improve because it would receive the same pay-off $2/3$ for every time slot which it could make a new single winner (recall that no safe single-team slot exists). Second, team $t_{i'}$ also cannot improve because it cannot create a new single winner at all. Last, neither of the remaining teams can improve because they cannot prevent c_j from co-winning. Hence, we have a 1-strong Nash equilibrium.

Cases with $x \geq 4$: Every time slot c_j with availability sum x is a safe multiple-team slot since $\forall j'\colon a_i^{j'} \leq a_{\max} = 2$ and $\forall i\colon \sum_{i' \neq i} a_{i'}^{j} \geq x - a_{\max} \geq 2$. □

Remark 2. *We do not know any instances without* 1*-strong Nash equilibria. However, we could not generalize our proof even for instances with* $a_{\max} = 4$*. Nevertheless, some general observations from our proof hold for every* $a_{\max}$*. In particular, if there is a column with only one entry with* $a_{\max}$ *(a special case of a safe single-team slot) or if the maximum column sum is at least* $2a_{\max}$ *(a special case of a safe multiple-team slot), then a* 1*-strong Nash equilibrium exists.*

Since our proof is constructive, we obtain the following.

Corollary 3. *If the maximum availability* $a_{\max}$ *is at most three, then a* 1*-strong Nash equilibrium for* TPG *can be found in polynomial time.*

The situation where $t \geq 2$ is quite different already with only two teams. By a proof similar to the case of $t = 1$ and $a_{\max} = 2$, we can show that a 2-strong Nash equilibrium always exists for $t = 2$ and $a_{\max} = 1$:

Proposition 1. *If the maximum availability* $a_{\max}$ *is one, then a* 2*-strong Nash equilibrium for* TPG *always exists and can be found in polynomial time.*

Complementing Theorem 4, we demonstrate that 2-strong Nash equilibria do not always exist, even when $a_{\max} = 2$; to this end, consider the following example.

$$A := \begin{array}{cc} \begin{array}{cc} c_1 & c_2 \end{array} & \\ \begin{pmatrix} 2 & 0 \\ 2 & 2 \\ 0 & 2 \end{pmatrix} & \begin{array}{c} t_1 \\ t_2 \\ t_3 \end{array} \end{array}$$

The main crux of this example is that t_1 (or, symmetrically, t_3) can cooperate with t_2; in such cooperation, t_2 can choose whether to be 'in favor' of t_1 or t_3, by declaring either $b_2^1 = 2$ and $b_2^2 = 0$ (favoring t_1), or $b_2^1 = 0$ and $b_2^2 = 2$ (favoring t_3). Moreover, t_1 or t_3 can 'reward' t_2 by not declaring its true availability 2, but only 1. In such a cooperation, both t_2 and t_1 (or t_2 and t_3) strictly improve their pay-offs.

Next, we show that if the coalition size is unbounded, then finding a Nash equilibrium becomes coNP-hard.

Theorem 5. *Deciding whether a Nash equilibrium exists for a given input is* coNP*-hard.*

Proof. (Sketch). We reduce from the complement of the following NP-complete problem (Gonzalez 1984): RESTRICTED X3C, which given sets $\mathcal{F} = \{S_1, \ldots, S_{3n}\}$, each containing exactly 3 elements from $E = \{e_1, \ldots, e_{3n}\}$ such that (1) $n \geq 2$ and (2) each element e_i appears in exactly 3 sets, asks whether there is a size-n *exact cover* $\mathcal{F}' \subseteq \mathcal{F}$, i.e., $|\mathcal{F}'| = n$ and $\bigcup_{S_i \in \mathcal{F}'} S_i = U$.

Given an instance $(\mathcal{F}, E)$ of the complement of RESTRICTED X3C we construct a game. For each element e_i ($i \in [3n]$) we construct a time slot e_i. We construct one additional time slot, denoted by α. For each set S_j ($j \in [3n]$) we construct a team s_j. For a team s_j, we set its availability for time slot e_i, namely a_i^j, to be n if $e_i \in S_j$, and otherwise 0. We set the availability of all teams to be 1 in time slot α. We consider $2n$-strong Nash-equilibria; thus, we consider coalitions containing up to $2n$ teams. This finishes the description of the polynomial-time reduction. The correctness proof can be found in our technical report (Bredereck et al. 2016). □

5 Conclusion

We introduced a game considering power of teams (referred to as TPG) that is naturally motivated by online scheduling polls where teams declare and update their availabilities in a dynamic process to increase their relative power. Our work leads to several directions for future research.

Tie-breaking rules: In this paper the teams are pessimistic, i.e., in case of several co-winners, the pay-off is defined as the *minimum* of the relative number, over the co-winners. This corresponds to situations where ties are broken adversarially. We chose this tie-breaking as a standard and natural one, and as one which models teams which are pessimistic in nature, where having too low power in the team might have very bad consequences. Naturally, one might study other tie-breaking rules such as breaking ties uniformly at random or breaking ties lexicographically; we mention that most of our results seem to transfer to lexicographic tie-breaking.

More refined availability constraints: In the online scheduling polls considered in this paper, the availability constraints expressed by the participants are dichotomous: each participant can only declare either "available" or "not available" at each time slot. Sometimes, the availability constraints of people participating in scheduling polls are more fine-grained; for example, a participant might not be sure whether she is available or not for some of the suggested time slots, but can only provide a "maybe available" answer for these time slots. Correspondingly, it is interesting to study TPG when we allow participants to express more refined availability constraints, maybe even allowing them to fully rank the time slots according to their constraints.

Nash modification problem: Taking the point of view of the poll convener (who desires to reach a Nash equilibrium), we suggest to study the following problem: given an input for TPG, what is the minimum number of time slots that shall be removed so that the modified input will have a Nash equilibrium?

References

Brams, S. J., and Fishburn, P. C. 1978. Approval voting. *American Political Science Review* 72(03):831–847.

Bredereck, R.; Chen, J.; Niedermeier, R.; Obraztsova, S.; and Talmon, N. 2016. Teams in online scheduling polls: Game-theoretic aspects. Technical Report abs/1611.08892, arXiv.org.

Dery, L. N.; Obraztsova, S.; Rabinovich, Z.; and Kalech, M. 2015. Lie on the fly: Iterative voting center with manipulative voters. In *Proceedings of IJCAI-2015*, 2033–2039.

Downey, R. G., and Fellows, M. R. 2013. *Fundamentals of Parameterized Complexity*. Springer.

Frank, A., and Tardos, É. 1987. An application of simultaneous Diophantine approximation in combinatorial optimization. *Combinatorica* 7(1):49–65.

Gonzalez, T. F. 1984. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science* 38:293–306.

Kannan, R. 1987. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research* 12(3):415–440.

Lee, H. 2014. Algorithmic and game-theoretic approaches to group scheduling. In *Proceedings of AAMAS-2014*, 1709–1710.

Lenstra, H. W. 1983. Integer programming with a fixed number of variables. *Mathematics of Operations Research* 8(4):538–548.

Lev, O., and Rosenschein, J. S. 2012. Convergence of iterative voting. In *Proceedings of AAMAS-2012*, 611–618.

Meir, R.; Polukarov, M.; Rosenschein, J. S.; and Jennings, N. R. 2010. Convergence to equilibria in plurality voting. In *Proceedings of AAAI-2010*, 823–828.

Obraztsova, S.; Elkind, E.; Polukarov, M.; and Rabinovich, Z. 2015. Doodle poll games. In *Proceedings of the 1st IJCAI Workshop on AGT*.

Reinecke, K.; Nguyen, M. K.; Bernstein, A.; Näf, M.; and Gajos, K. Z. 2013. Doodle around the world: Online scheduling behavior reflects cultural differences in time perception and group decision-making. In *Proceedings of CSCW-2013*, 45–54.

Zou, J. Y.; Meir, R.; and Parkes, D. C. 2015. Strategic voting behavior in Doodle polls. In *Proceedings of CSCW-2015*, 464–472.