Group Activity Selection on Social Networks

Ayumi Igarashi, Dominik Peters, Edith Elkind
Department of Computer Science
University of Oxford, UK
{ayumi.igarashi, dominik.peters, edith.elkind}@cs.ox.ac.uk

Abstract

We propose a new variant of the group activity selection problem (GASP), where the agents are placed on a social network and activities can only be assigned to connected subgroups. We show that if multiple groups can simultaneously engage in the same activity, finding a stable outcome is easy as long as the network is acyclic. In contrast, if each activity can be assigned to a single group only, finding stable outcomes becomes intractable, even if the underlying network is very simple: the problem of determining whether a given instance of a GASP admits a Nash stable outcome turns out to be NP-hard when the social network is a path, a star, or if the size of each connected component is bounded by a constant. On the other hand, we obtain fixed-parameter tractability results for this problem with respect to the number of activities.

Introduction

Companies assign their employees to different departments, large decision-making bodies split their members into expert committees, and university faculty form research groups: division of labor, and thus group formation, is everywhere. For a given assignment of agents to activities (such as management, product development, or marketing) to be successful, two considerations are particularly important: the agents need to be capable to work on their activity, and they should be willing to cooperate with other members of their group.

Many relevant aspects of this setting are captured by the *group activity selection problem* (GASP), introduced by Darmann et al. (2012). In GASP players have preferences over pairs of the form (activity, group size). The intuition behind this formulation is that certain tasks are best performed in small or large groups, and agents may differ in their preferences over group sizes; however, they are indifferent about other group members' identities. In the analysis of GASP, desirable outcomes correspond to *stable* and/or *optimal* assignments of players to activities, i.e., assignments that are resistant to player deviations and/or maximize the total welfare. In the work of Darmann et al. (2012), players are assumed to have approval preferences, and a particular focus is placed on individually rational assignments with the maximum number of participants; subsequently, Darmann (2015) investigated a model where players submit ranked ballots.

However, the basic model of GASP ignores the relationships among the agents: Do they know each other? Are their working styles and personalities compatible? Typically, we cannot afford to ask each agent about her preferences over all pairs of the form (coalition, activity), as the number of possible coalitions grows quickly with the number of agents. A more practical alternative is to adopt the ideas of Myerson (1977) and assume that the relationships among the agents are encoded by a *social network*, i.e., an undirected graph where nodes correspond to players and edges represent communication links between them; one can then require that each group is connected with respect to this graph.

In this paper we extend the basic model of GASP to take into account the agents' social network. We formulate several notions of stability for this setting, including Nash stability and core stability, and study the complexity of computing stable outcomes in our model. These notions of stability are inspired by the hedonic games literature (Aziz and Savani 2016) and were applied in the GASP setting by Darmann et al. (2012) and Darmann (2015).

Now, hedonic games on social networks were recently considered by Igarashi and Elkind (2016), who showed that if the underlying network is acyclic, stable outcomes are guaranteed to exist and some of the problems known to be computationally hard for the unrestricted setting become polynomial-time solvable. We obtain a similar result for GASP, but only if several groups of agents can simultaneously engage in the same activity, i.e., if the activities are *copyable*. In contrast, we show that if each activity can be assigned to at most one coalition, finding a stable outcome is hard even if the underlying network is very simple. Specifically, checking the existence of Nash stable or core stable outcomes turns out to be NP-hard even for very restricted classes of graphs, including paths, stars, and graphs with constant-size connected components. We believe that this result is remarkable since, in the context of cooperative games, such restricted networks usually enable one to design efficient algorithms for computing stable solutions (see, e.g., Chalkiadakis, Greco, and Markakis 2016; Elkind 2014; Igarashi and Elkind 2016).

Given these hardness results, we switch to the fixed parameter tractability paradigm. In the context of GASP, a particularly relevant parameter is the number of activities: generally speaking, we expect the number of players to be considerably larger than the number of available activities.

		Complexity (general case)	few activities (FPT wrt p)	copyable activities
Nash stability	trees	NP-c. (Thm 6)		poly time (Thm 5)
	paths	NP-c. (Thm 6)	$4^p p^2 \cdot \mathrm{poly}(n)$ (Thm 9)	poly time (Thm 5)
	stars	NP-c. (Thm 7)	$p^{p+1} 2^p \cdot \mathrm{poly}(n)$ (Thm 11)	poly time (Thm 5)
	small components	NP-c. (Thm 8)	$p^c 8^p \cdot \mathrm{poly}(n)$ (Thm 10)	
core stability	trees	NP-c. (Thm 12)		poly time (Thm 4)
	paths	NP-c. (Thm 12)		poly time (Thm 4)
	stars	NP-c. (Thm 12)		poly time (Thm 4)
	small components	NP-c. (Thm 12)	$p^{c+1} 8^p \cdot \mathrm{poly}(n)$ (Thm 13)	

Table 1: Overview of our complexity results. Here, n is the number of players, p is the number of activities, and c is a bound on the size of the connected components. The NP-completeness results for small components hold even for $c = 4$ for Nash stability and for $c = 3$ for core stability.

We show that for the restricted classes of networks used in our hardness proofs (i.e., paths, stars, and graphs with small connected components), finding a Nash stable outcome is fixed parameter tractable (FPT) with respect to the number of activities; some of our results extend to core stability. Our results are summarized in Table 1.

Full version. An extended version with full proofs is available on arXiv (Igarashi, Peters, and Elkind 2016).

Preliminaries

For $s \in \mathbb{N}$, let $[s] = \{1, 2, \ldots, s\}$. An instance of the *Group Activity Selection Problem* (GASP) is given by a finite set of *players* $N = [n]$, a finite set of *activities* $A = A^* \cup \{a_\emptyset\}$, where $A^* = \{a_1, a_2, \ldots, a_p\}$ and $a_\emptyset$ is the *void activity*, and a *profile* $(\succeq_i)_{i \in N}$ of complete and transitive preference relations over the set of *alternatives* $X = A^* \times [n] \cup \{(a_\emptyset, 1)\}$. Intuitively, $a_\emptyset$ corresponds to staying alone and doing nothing; multiple agents can make that choice independently from each other.

We refer to subsets $S \subseteq N$ of players as *coalitions*. We say that two non-void activities a and b are *equivalent* if for every player $i \in N$ and every $\ell \in [n]$ it holds that $(a, \ell) \sim_i (b, \ell)$. A non-void activity $a \in A^*$ is called *copyable* if A^* contains at least n activities that are equivalent to a (including a itself). We say that player $i \in N$ *approves* an alternative (a, k) if $(a, k) \succ_i (a_\emptyset, 1)$.

An outcome of a GASP is an *assignment* of activities A to players N, i.e., a mapping $\pi : N \to A$. Given an assignment $\pi : N \to A$ and a non-void activity $a \in A^*$, we denote by $\pi^a = \{\, i \in N \mid \pi(i) = a \,\}$ the set of players assigned to a. Also, if $\pi(i) \neq a_\emptyset$, we denote by $\pi_i = \{i\} \cup \{\, j \in N \mid \pi(j) = \pi(i)\}$ the set of players assigned to the same activity as player $i \in N$; we set $\pi_i = \{i\}$ if $\pi(i) = a_\emptyset$. An assignment $\pi : N \to A$ for a GASP is *individually rational* (IR) if for every player $i \in N$ with $\pi(i) \neq a_\emptyset$ we have $(\pi(i), |\pi_i|) \succeq_i (a_\emptyset, 1)$. A coalition $S \subseteq N$ and an activity $a \in A^*$ *strongly block* an assignment $\pi : N \to A$ if $\pi^a \subseteq S$ and $(a, |S|) \succ_i (\pi(i), |\pi_i|)$ for all $i \in S$. An assignment $\pi : N \to A$ for a GASP is called *core stable* (CR) if it is individually rational, and there is no coalition $S \subseteq N$ and activity $a \in A^*$ such that S and a strongly block π. Given an assignment $\pi : N \to A$ for a GASP, a player $i \in N$ is said to have an *NS-deviation* to activity $a \in A^*$ if $(a, |\pi^a| + 1) \succ_i (\pi(i), |\pi_i|)$, that is, if i would prefer to join the group π^a. An assignment $\pi : N \to A$ for a GASP is called *Nash stable* (NS) if it is individually rational and no player $i \in N$ has an NS-deviation to some $a \in A^*$.

Our Model

We now define a group activity selection problem where communication links between the players are represented by an undirected graph.

Definition 1. An instance of the *Group Activity Selection Problem with graph structure* (gGASP) is given by an instance $(N, (\succeq_i)_{i \in N}, A)$ of a GASP and a set of communication links between players $L \subseteq \{\, \{i, j\} \mid i, j \in N \wedge i \neq j \,\}$.

A coalition $S \subseteq N$ is said to be *feasible* if S is connected in the graph (N, L). An outcome of a gGASP is a *feasible assignment* $\pi : N \to A$ such that π_i is a feasible coalition for every $i \in N$. We adapt the definitions of stability concepts to our setting as follows. We say that a deviation by a group of players is feasible if the deviating coalition itself is feasible; a deviation by an individual player where player i joins activity a is feasible if $\pi^a \cup \{i\}$ is feasible. We modify the definitions in the previous section by only requiring stability against feasible deviations. Note that an ordinary GASP (without graph structure) is equivalent to a gGASP where the underlying graph (N, L) is complete.

In this paper, we will be especially interested in gGASPs where (N, L) is *acyclic*. This restriction guarantees the existence of stable outcomes in many other cooperative game settings. However, this is not the case for gGASPs: here, both core and Nash stable outcomes may fail to exist, even if (N, L) is a path or a star.

Example 2. Consider a gGASP with $N = \{1, 2, 3\}$, $A^* = \{a, b\}$, $L = \{\{1, 2\}, \{2, 3\}\}$, where preferences $(\succeq_i)_{i \in N}$ are given as follows:

$$1 : (b, 2) \succ_1 (a, 3) \succ_1 (a_\emptyset, 1)$$
$$2 : (a, 2) \succ_2 (b, 2) \succ_2 (a, 3) \succ_2 (a_\emptyset, 1)$$
$$3 : (a, 3) \succ_3 (b, 1) \succ_3 (a, 2) \succ_3 (a_\emptyset, 1)$$

We will argue that each individually rational feasible assignment π admits a strongly blocking feasible coalition and activity. If all players do nothing, then player 3 and activity

b strongly block π. Now, there are only four individually rational feasible assignments where some player is engaged in a non-void activity. First, when $\pi(1) = b$, $\pi(2) = b$, $\pi(3) = a_\emptyset$, the coalition $\{2, 3\}$ together with activity a strongly blocks π. Second, when $\pi(1) = a_\emptyset$, $\pi(2) = a$, $\pi(3) = a$, the coalition $\{3\}$ together with activity b strongly blocks π. Third, when $\pi(1) = a_\emptyset$, $\pi(2) = a_\emptyset$, $\pi(3) = b$, the coalition $\{1, 2, 3\}$ together with activity a strongly blocks π. Finally, when $\pi(1) = a$, $\pi(2) = a$, and $\pi(3) = a$, the coalition $\{1, 2\}$ together with activity b strongly blocks π. □

Similarly, a Nash stable outcome is not guaranteed to exist even for gGASPs on paths and stars.

Example 3 (Stalker game)**.** Consider a two-player gGASP where player 1 is happy to participate in any activity as long as she is alone, and player 2 always wants to participate in an activity with player 1. This instance admits no Nash stable outcomes: if player 1 engages in an activity, then player 2 wants to join her coalition, causing player 1 to deviate to another (possibly void) activity. □

However, if all activities are copyable, we can effectively treat gGASP as a non-transferable utility game on a graph. In particular, we can invoke a famous result of Demange (2004) concerning the stability of non-transferable utility games on trees. Thus, requiring all activities to be copyable allows us to circumvent the non-existence result for the core (Example 2). The argument is constructive.

Theorem 4 (implicit in the work of Demange 2004)**.** *For every gGASP where each activity $a \in A^*$ is copyable and (N, L) is acyclic, a core stable feasible assignment exists and can be found in time polynomial in p and n.*

Now, the stalker game in Example 3 does not admit a Nash stable outcome even if we make all activities copyable. However, for copyable activities we can still check the existence of a Nash stable outcome in polynomial time if the social network is acyclic.

Theorem 5. *Given an instance $(N, A, (\succeq_i)_{i \in N}, L)$ of gGASP where each activity $a \in A^*$ is copyable and the graph (N, L) is acyclic, one can decide whether it admits a Nash stable outcome in time polynomial in p and n.*

Proof. If the input graph (N, L) is a forest, we can process each of its connected components separately, so we assume that (N, L) is a tree. We choose an arbitrary node as the root and construct a rooted tree by orienting the edges in L towards the leaves. We denote by $D(i)$ the set of descendants of i (including i) in the rooted tree. Then, for each player $i \in N$, each alternative $(a, k) \in X$, and $t \in [k]$ we set $f_i((a, k), t)$ to *true* if there exists a feasible assignment $\pi : N \to A$ such that $|\pi_i \cap D(i)| = t$, $\pi(i) = a$, each player in $D(i) \cap \pi_i$ likes (a, k) at least as much as any alternative she can deviate to (including the void activity), and no player in $D(i) \setminus \pi_i$ has an NS feasible deviation. Otherwise, we set $f_i((a, k), t)$ to *false*. By construction, there exists a Nash stable feasible assignment if and only if $f_r((a, k), k)$ is *true* for some alternative $(a, k) \in X$, where r is the root of the rooted tree.

For each player $i \in N$, each alternative $(a, k) \in X$, and each $t \in [k]$, we initialize $f_i((a, k), t)$ to *true* if $t = 1$ and i weakly prefers (a, k) to any alternative of size 1, and we set $f_i((a, k), t)$ to *false* otherwise. Then, for $i \in N$ from the bottom to the root, we iterate through all the children of i and update $f_i((a, k), t)$; more precisely, for each child j of i and for $t = k, \ldots, 1$, we set $f_i((a, k), t)$ to *true* if

- $t \geq 2$ and there exists an $x \in [t - 1]$ such that both $f_i((a, k), x)$ and $f_j((a, k), t - x)$ are *true*, or
- $f_i((a, k), t)$ is *true*, and players i and j can be *separated* from each other, i.e., there exists $(b, \ell) \in X$ such that (i) $f_j((b, \ell), \ell)$ is *true*, (ii) $b = a_\emptyset$ or $(a, k) \succeq_i (b, \ell + 1)$, and (iii) $a = a_\emptyset$ or $(b, \ell) \succeq_j (a, k + 1)$.

In cases where $f_r((a, k), k)$ is *true* for some alternative $(a, k) \in X$, a Nash stable feasible assignment can be found using dynamic programming. □

Hardness Results for Nash Stability

We now move on to the case where each activity can be used at most once. We will show that computing Nash stable outcomes of gGASPs is NP-complete even when the underlying network is a path, a star, or a graph with constant size connected components. Clearly, this problem is contained in NP for any social network since we can easily check whether a given assignment is Nash stable.

Our proof for paths is by reduction from a restricted version of the NP-complete problem RAINBOW MATCHING. Given a graph G and a set of colors $\mathcal{C}$, a *proper edge coloring* is a mapping $\phi : E(G) \to \mathcal{C}$ where $\phi(e) \neq \phi(e')$ for all edges e, e' such that $e \neq e'$ and $e \cap e' \neq \emptyset$. Without loss of generality, we assume that ϕ is surjective. A *properly edge-colored graph* $(G, \mathcal{C}, \phi)$ is a graph together with a set of color and a proper edge coloring. A matching M in an edge-colored graph $(G, \mathcal{C}, \phi)$ is called a *rainbow matching* if all edges of M have different colors. An instance of RAINBOW MATCHING is an edge-colored graph $(G, \mathcal{C}, \phi)$ and an integer k. It is a "yes"-instance if G admits a rainbow matching with at least k edges and a "no"-instance otherwise. Le and Pfender (2014) show that RAINBOW MATCHING remains NP-complete even for properly edge-colored paths.

Theorem 6. *Given an instance of gGASP whose underlying graph is a path, it is NP-complete to determine whether it has a Nash stable feasible assignment.*

Proof. The hardness proof proceeds by a reduction from PATH RAINBOW MATCHING. Given an instance $(G, \mathcal{C}, \phi, k)$ of PATH RAINBOW MATCHING where $|\mathcal{C}| = q$, we construct an instance of gGASP on a path as follows. We create a vertex player v for each $v \in V(G)$ and an edge player e for each $e \in E(G)$. To create the social network, we start with G and place each edge player in the middle of the respective edge, i.e., we let $N_G = V(G) \cup E(G)$ and $L_G = \{ \{v, e\} \mid v \in e \in E(G) \}$. To the right of the graph (N_G, L_G), we attach a path consisting of "garbage collectors" $\{g_1, g_2, \ldots, g_{q-k}\}$ and q copies (N_c, L_c) of the stalker game where $N_c = \{c_1, c_2\}$ and $L_c = \{\{c_1, c_2\}\}$ for each $c \in \mathcal{C}$. We introduce a color activity c for each color $c \in \mathcal{C}$. Each vertex player v approves color activities $\phi(e)$ of its adjacent edges e with size 3; each edge player e approves the color activity $\phi(e)$ of its color with size 3; each garbage collector g_i approves any color activity c with size 1; finally, for players in N_c, $c \in \mathcal{C}$, player c_1 approves its color activity c with size 1, whereas player c_2 approves c with size 2.

We show that G has a rainbow matching of size at least k if and only if there exists a Nash stable feasible assignment.

Suppose that there exists a rainbow matching M of size k. We construct a feasible assignment π where for each $e = \{u, v\} \in M$ we set $\pi(e) = \pi(u) = \pi(v) = \phi(e)$, each garbage collector g_i, $i \in [q-k]$, is arbitrarily assigned to one of the remaining $q-k$ color activities, and the remaining players are assigned to the void activity. The assignment π is Nash stable, since every garbage collector as well as every edge or vertex player assigned to a color activity are allocated their top alternative, and no remaining player has an NS feasible deviation.

Conversely, suppose that there is a Nash stable feasible assignment π. Let $M = \{\, e \in E(G) \mid \pi(e) \in \mathcal{C} \,\}$. We will show that M is a rainbow matching of size at least k. To see this, notice that π cannot allocate a color activity to a member of N_c, since otherwise no feasible assignment would be Nash stable. Further, at most $q-k$ color activities are allocated to the garbage collectors, which means that at least k color activities should be assigned to vertex and edge players. The only individually rational way to do this is to select triples of the form (u, e, v) where $e = \{u, v\} \in E(G)$ and assign to them their color activity $\phi(e)$. Thus, M is a rainbow matching of size at least k. □

For gGASPs on stars we provide a reduction from the NP-complete problem MINIMUM MAXIMAL MATCHING (MMM). An instance of MMM is a graph G and a positive integer $k \leq |E(G)|$. It is a "yes"-instance if G admits a maximal matching with at most k edges, and a "no"-instance otherwise. The problem remains NP-complete for bipartite graphs (Demange and Ekim 2008).

Theorem 7. *Given an instance of* gGASP *whose underlying graph is a star, it is* NP*-complete to determine whether it has a Nash stable feasible assignment.*

Proof. To prove NP-hardness, we reduce from MMM on bipartite graphs. Given a bipartite graph $G = (U \cup V, E)$ with vertex bipartition (U, V) and an integer k, we create a star with center c and $|V| + 1$ leaves: one leaf for each vertex player $v \in V$ plus one stalker s. We introduce an activity u for each $u \in U$, and two additional activities a and b. A player $v \in V$ approves $(u, 1)$ for each activity u such that $\{u, v\} \in E$ as well as $(a, |V| - k + 1)$ and prefers the former to the latter. That is, $(u, 1) \succ_v (a, |V| - k + 1)$ for every $u \in U$ with $\{u, v\} \in E$; v is indifferent among the activities associated with its neighbors in the graph, that is, $(u, 1) \sim_v (u', 1)$ for all $u, u' \in U$ such that $\{u, v\} \in E$ and $\{u', v\} \in E$. The center player c approves both $(a, |V| - k + 1)$ and $(b, 1)$, and prefers the former to the latter, i.e., $(a, |V| - k + 1) \succ_c (b, 1) \succ_c (a_\emptyset, 1)$. Finally, the stalker s only approves $(b, 2)$.

We now show that G admits a maximal matching M with at most k edges if and only if our instance of gGASP admits a Nash stable assignment. Suppose that G admits a maximal matching M with at most k edges. We construct a feasible assignment π by setting $\pi(v) = u$ for each $\{u, v\} \in M$, assigning $|V| - k$ vertex players and the center to a, and assigning the remaining players to the void activity. Clearly, the center c has no incentive to deviate and no vertex player in a singleton coalition wants to deviate to the coalition of the center. Further, no vertex v has an NS-deviation to an unused activity u, since if π admits such a deviation, this would mean that $M \cup \{u, v\}$ forms a matching, contradicting maximality of M. Finally, the stalker player does not deviate since the center does not engage in b. Hence, π is Nash stable.

Conversely, suppose that there exists a Nash stable feasible assignment π and let $M = \{\, \{\pi(v), v\} \mid v \in V \wedge \pi(v) \in U \,\}$. We will show that M is a maximal matching of size at most k. By Nash stability, the stalker player should not have an incentive to deviate, and hence the center player and $|V| - k$ vertex players are assigned to activity a. It follows that k vertex players are not assigned to a, and therefore $|M| \leq k$. Moreover, M is a matching since each vertex player is assigned to at most one activity, and by individual rationality each activity can be assigned to at most one player. Now suppose towards a contradiction that M is not maximal, i.e., there exists an edge $\{u, v\} \in E$ such that $M \cup \{u, v\}$ is a matching. This would mean that in π no player is assigned to u, and v is assigned to the void activity; hence, v has an NS-deviation to u, contradicting the Nash stability of π. □

In the analysis of cooperative games on social networks one can usually assume that the social network is connected: if this is not the case, each connected component can be processed separately. This is also the case for gGASP as long as all activities are copyable. However, if each activity can only be used by a single group, different connected components are no longer independent, as they have to choose from the same pool of activities. Indeed, we will now show that the problem of finding Nash stable outcomes remains NP-hard even if the size of each connected component is at most four. Our hardness proof for this problem proceeds by reduction from a restricted version of 3SAT. Specifically, we consider (3,B2)-SAT: in this version of 3SAT each clause contains exactly 3 literals, and each variable occurs exactly twice positively and twice negatively. This problem is known to be NP-complete (Berman, Karpinski, and Scott 2003).

Theorem 8. *Given an instance of* gGASP *where each connected component of the underlying graph has size at most* 4*, it is* NP*-complete to determine whether it has a Nash stable feasible assignment.*

Proof. We reduce from (3,B2)-SAT. Consider a formula ϕ with variable set X and clause set C, where for each variable $x \in X$ we write x_1 and x_2 for the two positive occurrences of x, and $\bar{x}_1$ and $\bar{x}_2$ for the two negative occurrences of x. For each $x \in X$, we introduce four players $x_1, x_2, \bar{x}_1, \bar{x}_2$, which correspond to the four occurrences of x. For each clause $c \in C$, we introduce one stalker s_c and three other players c_1, c_2, and c_3. The network (N, L) consists of one component for each clause—a star with center s_c and leaves c_1, c_2, and c_3—and of two components for each variable $x \in X$ consisting of a single edge each: $\{x_1, x_2\}$ and $\{\bar{x}_1, \bar{x}_2\}$. Thus, the size of

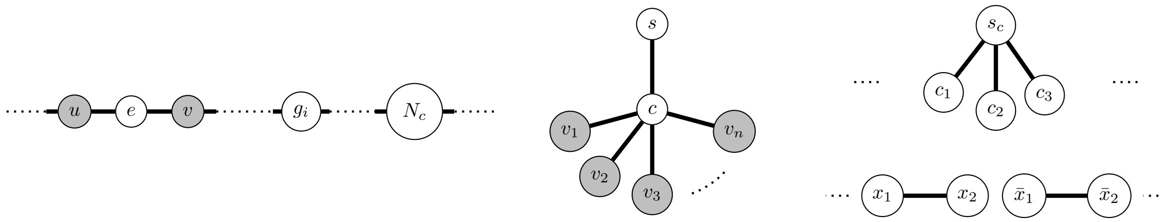

Figure 1: Graphs constructed in the proofs of Theorem 6, 7, and 8 (pictured left-to-right).

each component of this graph is at most 4. For each $x \in X$ we introduce one variable activity x, two positive literal activities x_1 and x_2, two negative literal activities $\bar{x}_1$ and $\bar{x}_2$, and two further activities a_x and $\bar{a}_x$. Also, we introduce an activity c for each clause $c \in C$. Thus,

$$A^* = \bigcup_{x \in X} \{x, x_1, x_2, \bar{x}_1, \bar{x}_2, a_x, \bar{a}_x\} \cup C.$$

For each $x \in X$ the preferences of the positive literal players x_1 and x_2 are given as follows:

$$x_1\colon (x,2) \succ (x,1) \succ (x_1,1) \succ (x_2,2) \succ (a_x,1) \succ (a_\emptyset,1),$$
$$x_2\colon (x,2) \succ (x_2,1) \succ (x_1,2) \succ (a_x,2) \succ (a_\emptyset,1).$$

If one of the positive literal players x_1 and x_2 is engaged in the void activity and the other is engaged alone in a non-void activity, this would cause the former player to deviate to another activity; thus, in a Nash stable assignment, none of the activities a_x and $a_\emptyset$ can be assigned to positive literal players. Similarly, for each $x \in X$ the preferences of the negative literal players $\bar{x}_1$ and $\bar{x}_2$ are given as follows:

$$\bar{x}_1\colon (x,2) \succ (x,1) \succ (\bar{x}_1,1) \succ (\bar{x}_2,2) \succ (\bar{a}_x,1) \succ (a_\emptyset,1),$$
$$\bar{x}_2\colon (x,2) \succ (\bar{x}_2,1) \succ (\bar{x}_1,2) \succ (\bar{a}_x,2) \succ (a_\emptyset,1).$$

As argued above, Nash stable assignments cannot allocate activities $\bar{a}_x$ and $a_\emptyset$ to negative literal players. Hence, if there exists a Nash stable assignment, there are only two possible cases: first, both players x_1 and x_2 are assigned to x, and players $\bar{x}_1$ and $\bar{x}_2$ are assigned to $\bar{x}_1$ and $\bar{x}_2$, respectively; second, both players $\bar{x}_1$ and $\bar{x}_2$ are assigned to x, and players x_1 and x_2 are assigned to x_1 and x_2, respectively.

For players in N_c where ℓ_1^c, ℓ_2^c, and ℓ_3^c are the literals in a clause c, the preferences are given by

$$c_r : \ (\ell_r^c,1) \succ (c,2) \succ (a_\emptyset,1), \qquad (r = 1,2,3)$$
$$s_c : \ (\ell_1^c,2) \sim (\ell_2^c,2) \sim (\ell_3^c,2) \sim (c,2) \succ (a_\emptyset,1).$$

That is, players c_1, c_2, and c_3 prefer to engage alone in their approved literal activity, whereas s_c wants to join one of the adjacent leaves whenever $\pi(s_c) = a_\emptyset$ and that leaf is assigned a literal activity; however, the leaf would then prefer to switch to the void activity. This means that if there exists a Nash stable outcome, at least one of the literal activities must be used outside of N_c, and some leaf and the stalker s_c must be assigned to activity c. We will show that ϕ is satisfiable if and only if there exists a Nash stable outcome.

Suppose that there exists a truth assignment that satisfies ϕ. First, for each variable x that is set to True, we assign positive literal activities x_1 and x_2 to the positive literal players x_1 and x_2, respectively, and assign x to the negative literal players $\bar{x}_1$ and $\bar{x}_2$. For each variable x that is set to False, we assign negative literal activities $\bar{x}_1$ and $\bar{x}_2$ to the negative literal players $\bar{x}_1$ and $\bar{x}_2$, respectively, and assign x to the positive literal players x_1 and x_2. Note that this procedure uses at least one of the literal activities ℓ_1^c, ℓ_2^c and ℓ_3^c of each clause $c \in C$, since the given truth assignment satisfies ϕ. Then, for each clause $c \in C$, we select a player c_j whose approved activity ℓ_j^c has been assigned to some literal player, and assign c_j and the stalker to c, and the rest of the clause players to their approved literal activity if it is not used yet, and to the void activity otherwise. It is easy to see that the resulting assignment π is Nash stable.

Conversely, suppose that there exists a Nash stable feasible assignment π. By Nash stability, for each variable $x \in X$, either a pair of positive literal players x_1 and x_2 or a pair of negative literal players $\bar{x}_1$ and $\bar{x}_2$ should be assigned to the corresponding pair of literal activities; in addition, for each clause $c \in C$, the stalker s_c and one of the players c_1, c_2, and c_3 should engage in the activity c, thereby implying that the approved literal activity of the respective leaf should be assigned to some literal players. Then, take the truth assignment that sets the variable x to True if its positive literal players x_1 and x_2 are assigned to positive literal activities x_1 and x_2; otherwise, x is set to False. This assignment can be easily seen to satisfy ϕ. □

Fixed Parameter Tractability

In the instances of gGASP that are created in our hardness proofs, the number of activities is unbounded. It is thus natural to wonder what can be said when there are few activities to be assigned. It turns out that for each of the restricted families of graphs considered in the previous section, finding Nash stable assignments in gGASP is fixed parameter tractable with respect to the number of activities.

The basic idea behind each of the three algorithms we present is that we fix a set of activities that will be assigned to the players, and for each possible subset $B \subseteq A^*$ of activities we check whether there exists a stable assignment using the activities from that subset only. Our algorithms for paths and for small components use dynamic programming, allowing us to build up the set B step-by-step.

We begin by giving the dynamic program that works for paths. Briefly, we move along the path from left to right, and, for each initial segment of the path, guess a set $B' \subseteq B$ of activities that will be used by that segment of players. For each guess, we determine whether it is possible to construct an assignment that does not admit an NS-deviation within the initial segment under consideration.

Theorem 9. *There exists an algorithm that, given an instance of* gGASP *whose underlying graph is a path, checks whether this instance has a Nash stable feasible assignment and finds one if it exists, and runs in time* $4^p p^2 \cdot \mathrm{poly}(n)$

Proof. Suppose that $N = [n]$ and $L = \{\{i, i+1\} \mid i = 1, 2, \ldots, n-1\}$. First, we guess a subset $B \subseteq A^*$ of non-void activities to be used; there are 2^p possibilities, so we try them all. For each B, we solve the problem by dynamic programming. For each $i \in [n]$, each $B' \subseteq B$, each alternative $(a, k) \in B' \times [n] \cup \{(a_\emptyset, 1)\}$, and each number $t \in [k]$, we let $f_i(B, B', (a, k), t)$ be *true* if there exists an individually rational feasible assignment $\pi : N \to A$ so that

- the set of activities assigned to $[i]$ is exactly B';
- $\pi(i) = a$, $|\pi_i| = k$, $|\pi_i \cap [i]| = t$; and
- every player in $[i]$ weakly prefers her alternative under π to engaging alone in any of the activities in $A^* \setminus B$ and has no NS feasible deviation to activities in B'.

Otherwise, we let $f_i(B, B', (a, k), t)$ be *false*.

For $i = 1$, if $B' = \{a\}$, $t = 1$, and player 1 weakly prefers (a, k) to each alternative $(b, 1)$ such that $b \in A \setminus B$, we set $f_1(B, B', (a, k), t)$ to *true* and otherwise to *false*.

For $i = 2, \ldots, n$, $f_i(B, B', (a, k), t)$ is *true* if $k - t \leq n - i$, we have $(a, k) \succeq_i (b, 1)$ for each $b \in A \setminus B$, and either

- $t = 1$, and players i and $i - 1$ can be *separated* from each other, i.e., there exists $(b, \ell) \in X$ such that (i) $f_{i-1}(B, B' \setminus \{a\}, (b, \ell), \ell)$ is true, (ii) $a = a_\emptyset$ or $(b, \ell) \succeq_{i-1} (a, k+1)$, and (iii) $b = a_\emptyset$ or $(b, \ell + 1) \succeq_i (a, k)$; or
- $t \geq 2$ and $f_{i-1}(B, B', (a, k), t-1)$ is *true*.

Otherwise $f_i(B, B', (a, k), t)$ is set to *false*. It is not difficult to see that a Nash stable assignment exists if and only if $f_n(B, B, (a, k), k)$ is *true* for some alternative $(a, k) \in X$ and some $B \subseteq A^*$. The runtime bound is immediate. □

Our algorithm for networks with small connected components is similar to the dynamic program we just discussed. We order the components, and, for each prefix of that ordering, we check if a given subset of activities can be assigned to that prefix in a Nash stable way. Within each component, we have enough time to consider all possible assignments, and each potential deviation involves at most one component. The resulting algorithm is FPT with respect to the combined parameter $p + c$, where c is a bound on the size of the components of the network.

Theorem 10. *There exists an algorithm that given an instance of* gGASP *on a graph with constant-size connected components checks whether it has a Nash stable feasible assignment, finds one if it exists, and runs in time* $p^c 8^p \cdot \mathrm{poly}(n)$*, where c is the maximum size of a connected component.*

For stars, we use a different technique to obtain an FPT result, namely (derandomized) color coding. We begin by guessing the alternative (a, k) assigned to the center player. Next, we again guess the precise set B of activities in use by the players not assigned to alternative (a, k). We then randomly color leaf players by activities in $B \cup \{a_\emptyset\}$, rejecting colorings that are infeasible or must lead to NS deviations. Crucially, the latter task reduces to straightforward counting questions, which allows this method to succeed.

Theorem 11. *There exists an algorithm that, given an instance of* gGASP *on a star checks whether it has a Nash stable feasible assignment, finds one if it exists, and runs in time* $2^p p^{p+1} \cdot \mathrm{poly}(n)$.

Proof. For each $(a, k) \in X$ and $B \subseteq A^* \setminus \{a\}$, we will check whether there exists a Nash stable assignment such that the center c and $k - 1$ leaves engage in a, exactly $|B|$ leaf players are assigned to activities in B, and the rest of the players are assigned to the void activity.

First, we will check whether the center player c strictly prefers some alternative $(b, \ell) \in B \times \{2\} \cup (A \setminus B) \times \{1\}$ to (a, k). If this is the case, there is no Nash stable outcome with the above properties, since the center player would deviate.

Next, we decide whether we can assign $|B|$ leaves to activities in B so as to obtain a Nash stable outcome. We use the color-coding technique to design a randomized algorithm: we color each leaf player using colors in B independently and uniformly at random. We say that a Nash stable assignment π that assigns exactly $|B|$ leaves to activities in B is *compatible* with a coloring χ if $\pi(i) = b$ implies $\chi(i) = b$ for each $b \in B$. If there exists a Nash stable assignment π where each of the activities in B is allocated to exactly one player, then the probability that a random coloring χ is compatible with it is $|B|^{-|B|}$: each player i with $\pi(i) \in B$ is colored 'correctly' with probability $1/|B|$. Since the success probability depends on p only, our algorithm can be derandomized by using a family of k-perfect hash functions (Alon, Yuster, and Zwick 1995).

It remains to show how to find a Nash stable outcome compatible with a given random coloring χ, or determine that no such assignment exists, in polynomial time. To this end, fix a coloring $\chi : N \setminus \{c\} \to B$. We seek to assign each player $i \in N \setminus \{c\}$ to one of the activities in $\{a, \chi(i), a_\emptyset\}$ in such a way that exactly one agent of each color engages in the color activity and k players including the center are assigned to a. For $b \in B$, let $N_b = \{i \in N \setminus \{c\} \mid \chi(i) = b\}$.

For each $b \in B$ and $\ell = 0, \ldots, |N_b| - 1$ let $f_b(\ell)$ be *true* if we can assign exactly one player in N_b to b and exactly ℓ players in N_b to a so that no player in N_b has an NS deviation. To compute these quantities, we need auxiliary variables.

Namely, for each $b \in B$, $i \in N_b$, and $\ell = 0, \ldots, |N_b| - 1$ let $f_b(i, \ell)$ be *true* if we can assign b to i, while assigning activity a to exactly ℓ players in N_b and $a_\emptyset$ to exactly $|N_b| - 1 - \ell$ players in N_b, so that no player in N_b has an NS-deviation. To compute $f_b(i, \ell)$, we first check whether player i has no incentive to deviate, i.e., whether (i) player i weakly prefers $(b, 1)$ to $(b', 1)$ for each $b' \in A \setminus B$ and (ii) $a = a_\emptyset$ or i weakly prefers $(b, 1)$ to $(a, k+1)$.

In a similar fashion, we check if the remaining players in N_b can be assigned to a and $a_\emptyset$ in the desired proportion. Specifically, let $N_b(i, a)$ be the set of players in $N_b \setminus \{i\}$ who weakly prefer (a, k) to $(b', 1)$ for each $b' \in A \setminus B$ and let $N_b(i, a_\emptyset)$ be the set of players in $N_b \setminus \{i\}$ who weakly prefer

$(a_\emptyset, 1)$ to $(b', 1)$ for each $b' \in A \setminus B$ as well as to $(a, k+1)$. We set $f_b(i, \ell)$ to *true* if and only if conditions (i) and (ii) are satisfied and we have $N_b(i, a) \cup N_b(i, a_\emptyset) = N_b \setminus \{i\}$, $|N_b(i, a)| \geq \ell$, and $|N_b(i, a_\emptyset)| \geq |N_b| - 1 - \ell$, i.e., each player in $N_b \setminus \{i\}$ can be assigned to a or $a_\emptyset$, and for each of these activities there is a sufficient number of players who would not deviate when assigned to that activity.

Having computed $f_b(i, \ell)$ for all $i \in N_b$, we can compute $f_b(\ell)$: we set $f_b(\ell)$ to *true* if $f_b(i, \ell)$ is true for some $i \in N_b$ and *false* otherwise.

We are now left with an instance of the MULTIPLE-CHOICE KNAPSACK problem: we need to check if for each $b \in B$ there is a value ℓ_b, $0 \leq \ell_b \leq |N_b| - 1$, such that $f_b(\ell_b)$ is true and $\sum_{b \in B} \ell_b = k - 1$. This problem can be solved in polynomial time by a straightforward dynamic programming algorithm. □

Core stability

By adapting the reductions for Nash stability, we can show that checking the existence of a core stable outcome is also NP-hard. This result holds for all classes of graph families that we have considered.

Theorem 12. *Given an instance of* gGASP *whose underlying graph is a path, a star, or has connected components whose size is bounded by* 3*, it is* NP*-complete to determine whether it has a core stable feasible assignment.*

Proof. To verify that a given feasible assignment is core stable, it suffices to check that for every alternative (a, k) there is no connected coalition with at least k players who strictly prefer (a, k) to the alternative of their current coalition. For the networks we consider this can be done in polynomial time, and hence our problem is in NP. The hardness reductions are similar to the respective reductions for Nash stability; essentially, we have to replace copies of the stalker game with copies of the game with an empty core. □

Our FPT result for graphs with small connected components can also be adapted to the core. In contrast, our approach for Nash stability for paths and stars does not seem to generalize to core stability, and we leave these cases for future work.

Theorem 13. *There exists an algorithm that given an instance of* gGASP *checks whether it has a core stable feasible assignment, finds one if it exists, and runs in time* $p^{c+1} 8^p \cdot \mathrm{poly}(n)$*, where c is the maximum size of the connected components.*

Conclusion

In this paper, we have initiated the study of group activity selection problems with network structure, and found that even for very simple families of graphs computing stable outcomes is NP-hard. We identified several ways to circumvent this computational intractability. For gGASPs with copyable activities, we showed that there exists a polynomial time algorithm to compute stable outcomes, and for gGASPs with few activities, we provided fixed parameter algorithms for restricted classes of networks.

We leave several interesting questions for future work. Our fixed-parameter tractability results can be extended to more general graph families, such as graphs with bounded pathwidth and graphs with a bounded number of internal nodes. However, for general graphs, the exact parameterized complexity of determining the existence of stable outcomes is unknown. When the underlying graph is complete, one can adapt techniques of Darmann et al. (2012) to show that the problem of computing Nash stable outcomes is in XP with respect to p; for other networks, including trees, it is not even clear whether our problem is in XP with respect to p. It would be also interesting to investigate the parameterized complexity of gGASPs using other parameters.

Another promising research direction is to study analogues of other solution concepts from the hedonic games literature for gGASPs; in particular, it would be interesting to understand the complexity of computing individually stable outcomes in gGASPs.

Acknowledgements We thank the anonymous reviewers for helpful comments that improved the presentation of the paper. This work was supported by the European Research Council (ERC) under grant number 639945 (ACCORD).

References

Alon, N.; Yuster, R.; and Zwick, U. 1995. Color-coding. *J. ACM* 42(4):844–856.

Aziz, H., and Savani, R. 2016. Hedonic games. In Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; and Procaccia, A. D., eds., *Handbook of Computational Social Choice*. Cambridge University Press. chapter 15.

Berman, P.; Karpinski, M.; and Scott, A. D. 2003. Approximation hardness of short symmetric instances of MAX-3SAT. Technical Report 049. http://eccc.hpi-web.de/report/2003/049/.

Chalkiadakis, G.; Greco, G.; and Markakis, E. 2016. Characteristic function games with restricted agent interactions: Core-stability and coalition structures. *Artificial Intelligence* 232:76–113.

Darmann, A.; Elkind, E.; Kurz, S.; Lang, J.; Schauer, J.; and Woeginger, G. 2012. Group activity selection problem. In WINE 2012, 156–169.

Darmann, A. 2015. Group activity selection from ordinal preferences. In ADT 2015, 35–51.

Demange, M., and Ekim, T. 2008. Minimum maximal matching is NP-hard in regular bipartite graphs. In TAMC 2008, 364–374.

Demange, G. 2004. On group stability in hierarchies and networks. *Journal of Political Economy* 112(4):754–778.

Elkind, E. 2014. Coalitional games on sparse social networks. In WINE 2014, 308–321.

Igarashi, A., and Elkind, E. 2016. Hedonic games with graph-restricted communication. In AAMAS 2016, 242–250.

A. Igarashi, D. Peters, and E. Elkind. Group activity selection on social networks. arxiv:1611.04524 [cs.GT], 2016.

Le, V. B., and Pfender, F. 2014. Complexity results for rainbow matchings. *Theoretical Computer Science* 524(C):27–33.

Myerson, R. B. 1977. Graphs and cooperation in games. *Mathematics of Operations Research* 2(3):225–229.