

Group activity selection problem with approval preferences

Andreas Darmann² · Edith Elkind¹ · Sascha Kurz³ · Jérôme Lang⁴ · Joachim Schauer² · Gerhard Woeginger⁵

Accepted: 24 September 2017 / Published online: 14 October 2017
© The Author(s) 2017. This article is an open access publication

Abstract We consider a setting where one has to organize one or several group activities for a set of agents. Each agent will participate in at most one activity, and her preferences over activities depend on the number of participants in the activity. The goal is to assign agents to activities based on their preferences in a way that is socially optimal and/or stable. We put forward a general model for this setting, which is a natural generalization of anonymous hedonic games. We then focus on a special case of our model where agents' preferences are binary, i.e., each agent classifies all pairs of the form '(activity, group size)' into ones that are acceptable and ones that are not. We formulate several solution concepts for this scenario, and study them from the computational point of view, providing hardness results for the general case as well as efficient algorithms for settings where agents' preferences satisfy certain natural constraints.

Keywords Hedonic games · GASP · Complexity · Approval preferences

✉ Edith Elkind
eelkind@gmail.com

¹ University of Oxford, Oxford, UK

² Universität Graz, Graz, Austria

³ Universität Bayreuth, Bayreuth, Germany

⁴ LAMSADE, Université Paris-Dauphine, Paris, France

⁵ RWTH Aachen, Aachen, Germany

1 Introduction

There are many real-life situations where a group of agents is faced with a choice of multiple activities, the members of the group have differing preferences over these activities, and it is feasible for the group to split into smaller subgroups, so that each subgroup can pursue its own activity. Consider, for instance, a company which, as a service to its employees, wants to provide free sports classes, and allows each employee to take at most one class (Skowron et al. 2015). Another example would be a workshop whose organizers would like to arrange one or more social activities for the free afternoon.¹ The available activities include a hike, a bus trip, and a table tennis competition. As they will take place simultaneously, each attendee can select at most one activity (or choose not to participate). If the preferences for activities do not depend on the number of participants and there are no constraints on the number of participants for each activity, then one can simply elicit the attendees' preferences over the activities, and assign each of them to their favorite activity. However, the situation becomes more complicated if one's preferences may depend on the number of other attendees who choose the same activity. For instance, the bus trip has a fixed transportation cost that has to be shared equally among its participants (while more complex cost-sharing schemes are possible, participants may view them as unfair), which implies that, typically, an attendee i is only willing to go on the bus trip if the number of other participants of the bus trip exceeds a threshold ℓ_i . Similarly, i may only be willing to play table tennis if the number of attendees who signed up for the tournament does *not* exceed a threshold u_i : as there is only one table, the more participants, the less time each individual spends playing.

Neglecting to take the number of participants of each activity into account may lead to highly undesirable outcomes, such as a bus that is shared by two persons, each of them paying a high cost, and a 48-participant table tennis tournament with one table. Adding constraints on the number of participants for each activity is a practical, but imperfect solution, as the agents' preferences over group sizes may differ: while some attendees (say, senior faculty) may be willing to go on the bus trip with just 4–5 other participants, others (say, graduate students) cannot afford it unless the number of participants exceeds 10. Thus, in this paper, we will focus on a more fine-grained approach: eliciting the agents' preferences over pairs of the form '(activity, group size)' (together with the option 'do nothing', which we refer to as the *void activity*), rather than over activities themselves, and allocating agents to activities based on this information.

Our goal is to formalize and study the problem of finding a good assignment of agents to activities based on agents' preferences over pairs of the form '(activity, group size)', which we will refer to as the Group Activity Selection Problem (GASP). We aim to put forward a model for this problem that is expressive enough to capture many real-life activity selection scenarios, yet simple enough to admit efficient procedures for finding good assignments of agents to activities. Due to our choice of motivating applications, we assume that the central authority knows (or, rather, can reliably elicit) the agents' true preferences, and its goal is to find an assignment of players to activi-

¹ Some of the co-authors of this paper had to deal with this problem when co-organizing a Dagstuhl seminar.

ties that, informally speaking, is stable and/or maximizes the overall satisfaction. We describe the basic structure of the problem, and discuss plausible constraints on the number and type of available activities and the structure of agents' preferences. We adapt various notions of stability from the hedonic games literature (such as Nash stability, individual stability and core stability) to our setting. We then focus on a simple preference model, where agents' preferences are trichotomous: each agent identifies a set of alternatives that he prefers to the void activity and a set of alternatives that he likes less than the void activity, and is indifferent among the alternatives in each set. For this simplified model, we briefly discuss the relationships among the solution concepts we define, and then focus on the algorithmic problem of finding a 'good' (socially optimal and/or stable) outcome. We show that, in general, the associated computational problems are NP-hard, but identify a number of interesting special cases that admit efficient exact or approximate algorithms.

We do not aim to provide a complete analysis of the group activity selection problem; rather, we view our work as a first step towards understanding the algorithmic and incentive issues that arise in this setting. We hope that our paper will lead to future research on this topic; to facilitate this, throughout the paper we highlight several possible extensions of our model as well as list some problems left open by our work.

1.1 Related work

The model considered in this paper is closely related to that of *anonymous hedonic games* (Banerjee et al. 2001), where, just as in our setting, players have to split into groups and each player has preferences over possible group sizes. The main difference between anonymous hedonic games and our problem is that, in our setting, the agents' preferences depend not only on the group size, but also on the activity that has been allocated to their group; thus, our model can be seen as a generalization of anonymous hedonic games. We remark that we can represent our problem as a general (i.e., non-anonymous) hedonic game (Banerjee et al. 2001; Bogomolnaia and Jackson 2002), by creating a dummy agent for each activity and endowing it with suitable preferences (see Sect. 2.2 for details). However, our setting has useful structural properties that distinguish it from a generic hedonic game: for instance, it allows for succinct representation of players' preferences, and, as we will see, has several natural special cases that admit efficient algorithms for finding good outcomes.

Our setting also has some similarities to those of congestion games (Vöcking 2006) and cost-sharing games (Jain and Mahdian 2007). Indeed, in such games agents also have preferences over the number of other agents they share resources with: specifically, in congestion games an agent's utility decreases with the number of agents who chose the same resource, and in cost-sharing games sharing the resource with a larger number of other agents lowers the cost. Our model differs from both of these settings in two important aspects: on the one hand, we do not require all activities to be of the same 'type' (i.e., some activities may behave as resources in congestion games, while others may behave as resources in cost-sharing games), and, on the other hand, we assume that agents' preferences are ordinal (and, in most of the paper, binary) rather than numeric.

Some recent proposals are also related to our model. Lee and Shoham (2015) investigate the *anonymous stable invitation* problem, where the goal is to find a set of agents

to be invited to an event, given agents' preferences over the number of invitees. This problem corresponds to the special case of GASP with only one non-void activity. Lee and Shoham focus on proving impossibility and possibility results concerning strategyproofness; they give a very simple strategyproof mechanism that finds a maximum stable invitation in the case of increasing preferences. They also consider a more general, non-anonymous version of the problem. Spradling et al. (2013) consider hedonic games with roles and teams, where agents have preferences over pairs consisting of a role (that they will play in their coalition) and the composition of roles that make up the coalition. Similarly to our setting, in this model agents have preferences over both coalitions and actions; however, in the model of Spradling et al. actions are associated with individual players rather than the coalition as a whole. Lu and Boutilier (2012) discuss a model of *cooperative group buying*, which can be embedded into GASP (more specifically, GASP with decreasing preferences, see Sect. 2). This model has a set of buyers and a set of items with volume discounts. Each buyer ranks all pairs (j, p_j) , where j is an item and p_j is a possible discounted price of this item; the discounted price of j is a function of the number of buyers who are matched to j .

After the conference version of our paper has been published, several authors explored the complexity of finding good allocations in GASP, both in the approval-based model, which is the focus of our work (Lee and Vassilevska Williams 2017), and in the ordinal model, where each agent ranks pairs of the form '(activity, group size)' (Darmann 2015). Igarashi et al. (2017a, b) and subsequently Gupta et al. (2017) considered a variant of GASP where agents are connected by a social network and members of each group have to form a connected subgraph in this network; they adapt the solution concepts that we propose to this richer setting, and put forward a number of algorithms and complexity results. A survey of this literature is provided by Darmann and Lang (2017).

2 Formal model

We start by giving a formal definition of our model.

Definition 1 An instance of the Group Activity Selection Problem (GASP) is given by a set of *agents* $N = \{1, \dots, n\}$, a set of *activities* $A = A^* \cup \{a_\emptyset\}$, where $A^* = \{a_1, \dots, a_m\}$, and a *profile* P , which consists of n votes (one for each agent): $P = (V_1, \dots, V_n)$. The vote of agent i describes his preferences over the set of *alternatives* $X = X^* \cup \{a_\emptyset\}$, where $X^* = A^* \times \{1, \dots, n\}$; alternative (a, k) , $a \in A^*$, is interpreted as "activity a with k participants", and a_\emptyset is the *void activity*.

The vote V_i of an agent $i \in N$ (also denoted by \succeq_i) is a weak order over X^* ; its induced strict preference and indifference relations are denoted by \succ_i and \sim_i , respectively. We set $S_i = \{(a, k) \in X^* \mid (a, k) \succ_i a_\emptyset\}$; we say that voter i *approves* of all alternatives in S_i , and refer to the set S_i as the *induced approval vote* of voter i . We assume that no agent can be prevented from switching from his current assignment to the void activity; as a consequence, we can assume that each agent i is indifferent among all alternatives in $\{(a, k) \in X^* \mid a_\emptyset \succ_i (a, k)\}$.

Throughout the paper we will mostly focus on a special case of our problem where no agent is indifferent between the void activity and any other alternative (i.e., for every

$i \in N$ we have $\{x \in X^* \mid x \sim_i a_\emptyset\} = \emptyset$, and each agent is indifferent between all the alternatives in S_i . In other words, preferences are *trichotomous*: the agent partitions X into three clusters S_i , $\{a_\emptyset\}$ and $X \setminus (S_i \cup \{a_\emptyset\})$, is indifferent between two alternatives in the same cluster, prefers any (a, k) in S_i to a_\emptyset , and a_\emptyset to any (a, k) in $X \setminus (S_i \cup \{a_\emptyset\})$; we denote this special case of our problem by **a-GASP**.

It will be convenient to distinguish between activities that are unique and ones that exist in multiple copies. For instance, if there is a single tennis table and two buses, then we can organize one table tennis tournament, two bus trips (we assume that there is only one potential destination for the bus trip, so these trips are identical), and an unlimited number of hikes (again, we assume that there is only one hiking route). This distinction will be useful for the purposes of complexity analysis: for instance, some of the problems we consider are easy when we have k copies of one activity, but hard when we have k distinct activities. Formally, we say that two activities a and b are *equivalent* if for every agent i and every $j \in \{1, \dots, n\}$ it holds that $(a, j) \sim_i (b, j)$. We say that an activity $a \in A^*$ is *k-copyable* if A^* contains exactly k activities that are equivalent to a (including a itself). We say that a is *simple* if it is 1-copyable; if a is k -copyable for $k \geq n$, we will say that it is ∞ -copyable (note that we would never need to organize more than n copies of any activity). If some activities in A^* are equivalent, A^* can be represented more succinctly by listing one representative of each equivalence class, together with the number of available copies. However, as long as we make the reasonable assumption that each activity exists in at most n copies, this representation is at most polynomially more succinct.

Our model can be enriched by specifying a set of *constraints* Γ . One constraint that arises frequently in practice is a *global cardinality* constraint, which specifies a bound K on the number of activities to be organized. More generally, we could also consider more complex constraints on the set of activities that can be organized simultaneously, which can be encoded, e.g., by a propositional formula or a set of linear inequalities. We leave the study of this more general setting as a topic for future work. There can also be external constraints on the number of participants for each activity: for instance, a bus can fit at most 40 people. However, these constraints can be incorporated into agents' preferences, by assuming that all agents view the alternatives that do not satisfy these constraints as unacceptable.

2.1 Special cases

We now consider some natural restrictions on agents' preferences that may simplify the problem of finding a good assignment. We first need to introduce some additional notation. Given a vote V_i and an activity $a \in A^*$, let $S_i^{\downarrow a} = \{k \mid (a, k) \in S_i\}$; the set $S_i^{\downarrow a}$ is the projection of S_i onto $\{a\} \times \{1, \dots, n\}$. Also, for integers i, j , we write $[i, j] = \{z \in \mathbb{Z} \mid i \leq z \leq j\}$; note that $[i, j] = \emptyset$ for $i > j$.

Example 1 Let $A^* = \{a, b\}$ and consider an agent i whose vote V_i is given by

$$(a, 8) \succ_i (a, 7) \sim_i (b, 4) \succ_i (a, 9) \succ_i (b, 3) \succ_i (b, 5) \succ_i (a, 6) \succ_i (b, 6) \succ_i a_\emptyset \succ_i \dots$$

Then $S_i = \{a\} \times [6, 9] \cup \{b\} \times [3, 6]$ and $S_i^{\downarrow a} = [6, 9]$.

We are now ready to define two types of restricted preferences for **a-GASP** that are directly motivated by our running example, namely, *increasing* and *decreasing* preferences. Informally, under increasing preferences an agent prefers to share each activity with as many other participants as possible (e.g., because each activity has an associated cost, which has to be split among the participants), and under decreasing preferences an agent prefers to share each activity with as few other participants as possible (e.g., because each activity involves sharing a limited resource). Of course, an agent's preferences may also be increasing with respect to some activities and decreasing with respect to others, depending on the nature of each activity. We provide a formal definition for **a-GASP** only; however, it can be extended to **GASP** in a straightforward way.

Definition 2 Consider an instance (N, A, P) of **a-GASP**. We say that the preferences of agent i are *increasing* (*INC*) with respect to an activity $a \in A^*$ if there exists a threshold $\ell_i^a \in \{1, \dots, n+1\}$ such that $S_i^{\downarrow a} = [\ell_i^a, n]$. Similarly, we say that the preferences of agent i are *decreasing* (*DEC*) with respect to an activity $a \in A^*$ if there exists a threshold $u_i^a \in \{0, \dots, n\}$ such that $S_i^{\downarrow a} = [1, u_i^a]$.

We say that an instance (N, A, P) of **a-GASP** is *increasing* (respectively, *decreasing*) if the preferences of each agent $i \in N$ are increasing (respectively, decreasing) with respect to each activity $a \in A^*$. We say that an instance (N, A, P) of **a-GASP** is *mixed increasing–decreasing* (*MIX*) if there exists a set $A^+ \subseteq A^*$ such that for each agent $i \in N$ his preferences are increasing with respect to each $a \in A^+$ and decreasing with respect to each $a \in A^- = A^* \setminus A^+$.

For some activities, an agent may have both a lower and an upper bound on the acceptable group size: e.g., one may prefer to go on a hike with at least three other people, but does not want the group to be too large (so that it can maintain a good pace). In this case, we say that an agent has *interval* (*INV*) preferences; note that *INC/DEC/MIX* preferences are a special case of interval preferences.

Definition 3 Consider an instance (N, A, P) of **a-GASP**. We say that the preferences of agent i are *interval* (*INV*) if for each $a \in A^*$ there exists a pair of thresholds $\ell_i^a, u_i^a \in \{1, \dots, n\}$ such that $S_i^{\downarrow a} = [\ell_i^a, u_i^a]$.

Other natural constraints on preferences include restricting the size of S_i (or, more liberally, that of $S_i^{\downarrow a}$ for each $a \in A^*$), or requiring agents to have similar preferences: for instance, one could limit the number of agent *types*, i.e., require that the set of agents can be split into a small number of groups so that the agents in each group have identical preferences. We will not define such constraints formally, but we will indicate if they are satisfied by the instances constructed in the hardness proofs in Sect. 4.1.

Example 2 Consider an instance of **a-GASP** with $n = 5$, $A^* = \{a, b\}$ and voters' preferences given by the following table (where $+/-$ indicates approval/disapproval):

| | (a, 1) | (a, 2) | (a, 3) | (a, 4) | (a, 5) | (b, 1) | (b, 2) | (b, 3) | (b, 4) | (b, 5) |
|---|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 1 | + | + | + | - | - | - | - | - | - | - |
| 2 | + | + | - | - | - | - | - | - | - | - |
| 3 | + | - | - | - | - | - | + | + | + | + |
| 4 | + | + | - | - | - | - | + | + | + | + |
| 5 | + | + | - | - | - | - | - | - | + | + |

This instance of **a-GASP** is mixed increasing–decreasing. Now, suppose that we add another agent whose preferences are described by the last line of the table below:

| | (a, 1) | (a, 2) | (a, 3) | (a, 4) | (a, 5) | (a, 6) | (b, 1) | (b, 2) | (b, 3) | (b, 4) | (b, 5) | (b, 6) |
|---|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 1 | + | + | + | - | - | - | - | - | - | - | - | - |
| 2 | + | + | - | - | - | - | - | - | - | - | - | - |
| 3 | + | - | - | - | - | - | - | + | + | + | + | + |
| 4 | + | + | - | - | - | - | - | + | + | + | + | + |
| 5 | + | + | - | - | - | - | - | - | - | + | + | + |
| 6 | - | - | + | + | + | + | - | - | - | - | - | - |

The resulting instance is no longer mixed increasing–decreasing.

2.2 GASP and hedonic games

Recall that a *hedonic game* (Banerjee et al. 2001; Bogomolnaia and Jackson 2002) is given by a set of agents N , and, for each agent $i \in N$, a weak order \geq_i over all coalitions (i.e., subsets of N) that include i (the associated strict preference and indifference relations are denoted by $>_i$ and $=_i$, respectively). That is, in a hedonic game each agent has preferences over coalitions that he can be a part of. A coalition $S, i \in S$, is said to be *unacceptable* for player i if $\{i\} >_i S$. A hedonic game is said to be *anonymous* if each agent is indifferent among all coalitions of the same size that include him, i.e., for every $i \in N$ and every $S, T \subseteq N \setminus \{i\}$ such that $|S| = |T|$ it holds that $S \cup \{i\} \geq_i T \cup \{i\}$ and $T \cup \{i\} \geq_i S \cup \{i\}$.

At a first glance, it may seem that the **GASP** formalism is more general than that of hedonic games, since in **GASP** the agents care not only about their coalition, but also about the activity they have been assigned to. However, we will now argue that **GASP** can be embedded into the hedonic games framework.

Given an instance of the **GASP** problem (N, A, P) with $|N| = n$, where the i -th agent’s preferences are given by a weak order \geq_i , we construct a hedonic game $H(N, A, P)$ as follows. We create $n + m$ players; the first n players correspond to agents in N , and the last m players correspond to activities in A^* . The last m players are indifferent among all coalitions. For each $i = 1, \dots, n$, player i ranks every non-singleton coalition with no activity players as unacceptable; similarly, all coalitions with two or more activity players are ranked as unacceptable. The preferences over coalitions with exactly one activity player are derived naturally from the weak orders

\succeq_i : if S, T are two coalitions involving player i , x is the unique activity player in S , and y is the unique activity player in T , then i weakly prefers S to T in $H(N, A, P)$ if and only if $(x, |S| - 1) \succeq_i (y, |T| - 1)$, and i weakly prefers S to $\{i\}$ in $H(N, A, P)$ if and only if $(x, |S| - 1) \succeq_i a_\emptyset$. We emphasize that the resulting hedonic games are not anonymous. Further, while this embedding allows us to apply the standard solution concepts for hedonic games without redefining them, the intuition behind these solution concepts is not always preserved (e.g., because activity players never want to deviate). Therefore, in Sect. 3, we will provide formal definitions of the relevant hedonic games solution concepts adapted to the setting of GASP.

We remark that when A^* consists of a single ∞ -copyable activity (i.e., there are n activities in A^* , all of them equivalent to each other), GASP become equivalent to anonymous hedonic games. Such games have been studied in detail by Ballester (2004), who provides a number of complexity results for them. In particular, he shows that finding an outcome that is core stable, Nash stable or individually stable (see Sect. 3 for the definitions of some of these concepts in the context of GASP) is NP-hard. Clearly, all these complexity results also hold for GASP. However, they do not directly imply similar hardness results for a-GASP.

3 Solution concepts

Having discussed the basic model of GASP, as well as a few of its extensions and special cases, we are ready to define what constitutes a solution to this problem.

Definition 4 An *assignment* for an instance (N, A, P) of GASP is a mapping $\pi : N \rightarrow A$; $\pi(i) = a_\emptyset$ means that agent i does not participate in any activity. Each assignment naturally partitions the agents into at most $m + 1 = |A|$ groups: we set $\pi^0 = \{i \in N \mid \pi(i) = a_\emptyset\}$ and $\pi^j = \{i \in N \mid \pi(i) = a_j\}$ for $j = 1, \dots, m$. Given an assignment π , the *coalition structure* CS_π induced by π is the coalition structure over N defined as follows:

$$CS_\pi = \left\{ \pi^j \mid j = 1, \dots, m, \pi^j \neq \emptyset \right\} \cup \left\{ \{i\} \mid i \in \pi^0 \right\}.$$

Clearly, not all assignments are equally desirable. As a minimum requirement, no agent should be assigned to a coalition that he deems unacceptable. More generally, we prefer an assignment to be stable, i.e., no agent (or group of agents) should have an incentive to change its activity. Thus, we will now define several *solution concepts*, i.e., classes of desirable assignments. We will state our definitions for the general case of GASP and then indicate how they simplify in the case of a-GASP. Given the connection to hedonic games pointed out in Sect. 2.2, we will proceed by adapting the standard hedonic game solution concepts to our setting; however, this has to be done carefully to preserve the intuition that is specific to our model.

The first solution concept that we will consider is *individual rationality*.

Definition 5 Given an instance (N, A, P) of GASP, an assignment $\pi : N \rightarrow A$ is said to be *individually rational* if for every $j = 1, \dots, m$ and every agent $i \in \pi^j$ it holds that $(a_j, |\pi^j|) \succeq_i a_\emptyset$.

In the case of **a-GASP**, this definition can be rewritten as follows: π is individually rational if for every $j = 1, \dots, m$ and every agent $i \in \pi^j$ it holds that $(a_j, |\pi^j|) \in S_i$. Clearly, if an assignment is not individually rational, there exists an agent that can benefit from abandoning his coalition in favor of the void activity. Further, an individually rational assignment always exists: for instance, we can set $\pi(i) = a_\emptyset$ for all $i \in N$. However, given an instance of **a-GASP**, a benevolent central authority would want to maximize the number of agents that are assigned to non-void activities. Formally, let $\#(\pi) = |\{i \in N \mid \pi(i) \neq a_\emptyset\}|$ denote the number of agents assigned to a non-void activity. We say that π is *maximum individually rational* if π is individually rational and $\#(\pi) \geq \#(\pi')$ for every individually rational assignment π' . Further, we say that π is *perfect*² if π is individually rational and $\#(\pi) = n$. We denote the size of a maximum individually rational assignment for an instance (N, A, P) by $\#(N, A, P)$. In Sect. 4, we study the complexity of computing a perfect or maximum individually rational assignment for **a-GASP**, both for the general model and for the special cases defined in Sect. 2.1.

Example 3 Consider the first instance of **a-GASP** from Example 2 (the one with $n = 5$). There are three maximum individually rational assignments for it:

- $\pi_1(1) = \pi_1(2) = a, \pi_1(3) = \pi_1(4) = b, \pi_1(5) = a_\emptyset$;
- $\pi_2(1) = \pi_2(5) = a, \pi_2(3) = \pi_2(4) = b, \pi_2(2) = a_\emptyset$;
- $\pi_3(2) = \pi_3(5) = a, \pi_3(3) = \pi_3(4) = b, \pi_3(1) = a_\emptyset$.

However, there is no perfect assignment. On the other hand, if agent 5 is removed, then there is a perfect assignment, namely, $\pi_4(1) = \pi_4(2) = a, \pi_4(3) = \pi_4(4) = b$.

Definition 6 Given an instance (N, A, P) of **GASP**, an assignment $\pi : N \rightarrow A$ is said to be *Nash stable* if

- (1) π is individually rational,
- (2) for every agent $i \in N$ with $\pi(i) \neq a_\emptyset$ and every $a_j \in A^* \setminus \{\pi(i)\}$ it holds that $(\pi(i), |\pi(i)|) \succeq_i (a_j, |\pi^j| + 1)$, and
- (3) for every agent $i \in N$ with $\pi(i) = a_\emptyset$ and every $a_j \in A^*$ it holds that $a_\emptyset \succeq_i (a_j, |\pi^j| + 1)$.

In the case of **a-GASP**, this definition simplifies as follows: π is Nash stable if it is individually rational and for every agent $i \in N$ such that $\pi(i) = a_\emptyset$ and every $a_j \in A^*$ it holds that $(a_j, |\pi^j| + 1) \notin S_i$. If π is not Nash stable, then there is an agent assigned to the void activity who wants to join a group that is engaged in a non-void activity, i.e., he would have approved of the size of this group and its activity choice if he was one of them. Note that in **a-GASP** an agent assigned to a non-void activity wants to deviate only if he disapproves of that activity, i.e., we do not have to worry about agents moving from one non-void activity to another one.

The requirement of Nash stability is much stronger than that of individual rationality, and there are cases where a Nash stable assignment does not exist.

² The terminological similarity with the notion of perfect partition in a hedonic game (Aziz et al. 2013) is intentional; in a hedonic game a perfect partition assigns each agent to his most preferred coalition, whereas in an **a-GASP** a perfect assignment maps each agent to one of her approved alternatives.

Proposition 1 For each $n \geq 2$, there exists an instance (N, A, P) of **a-GASP** with $|N| = n$ that does not admit a Nash stable assignment. This holds even if $|A^*| = 1$ and all agents have interval preferences.

Proof Consider an instance (N, A, P) of **a-GASP** with $A^* = \{a\}$ and induced approval votes given by $S_1 = \{(a, 1)\}$, $S_2 = \{(a, 2)\}$ and $S_i = \emptyset$ for all $i \geq 3$; note that all approved sets are intervals. The two individually rational assignments are π_1 with $\pi_1(i) = a_\emptyset$ for all $i \in N$ and π_2 with $\pi_2(1) = a$ and $\pi_2(i) = a_\emptyset$ for all $i \in N \setminus \{1\}$. Neither of these assignments is Nash stable: in π_1 agent 1 wants to engage in a , and in π_2 agent 2 wants to join agent 1. \square

Example 4 Let us consider again the instances of **a-GASP** in Example 2. For the instance with $n = 5$ there are two Nash stable assignments: π_1 and π_2 . For the instance with $n = 6$ no assignment is Nash stable.

In Definition 6 an agent is allowed to join a coalition even if the members of this coalition are opposed to this. In contrast, the notion of *individual stability* only allows a player to join a group if none of the existing group members objects.

Definition 7 Given an instance (N, A, P) of **GASP**, an assignment $\pi : N \rightarrow A$ is said to be *individually stable* if

- (1) π is individually rational,
- (2) for every agent $i \in N$ with $\pi(i) \neq a_\emptyset$ and every $a_j \in A^* \setminus \{\pi(i)\}$ such that $(a_j, |\pi^j| + 1) \succ_i (\pi(i), |\pi(i)|)$ there exists an agent $i' \in \pi^j$ with $(a_j, |\pi^j|) \succ_{i'} (a_j, |\pi^j| + 1)$, and
- (3) for every agent $i \in N$ with $\pi(i) = a_\emptyset$ and every $a_j \in A^*$ such that $(a_j, |\pi^j| + 1) \succ_i a_\emptyset$ there exists an agent $i' \in \pi^j$ with $(a_j, |\pi^j|) \succ_{i'} (a_j, |\pi^j| + 1)$.

For **a-GASP**, we obtain the following simpler notion: π is individually stable if for each agent $i \in N$ with $\pi(i) = a_\emptyset$ and each $a_j \in A^*$ it holds that $(a_j, |\pi^j| + 1) \notin S_{i'}$ for some $i' \in \pi^j \cup \{i\}$.

Proposition 2 Every instance of **a-GASP** admits an individually stable assignment; moreover, this assignment can be computed by the following simple algorithm.

1. Initialize π with $\pi(i) = a_\emptyset$ for all $i \in N$;
2. While there is $(i, a_j) \in N \times A^*$ with
 - (a) $\pi(i) = a_\emptyset$
 - (b) $(a_j, |\pi^j| + 1) \in S_i$
 - (c) $(a_j, |\pi^j| + 1) \in S_{i'}$ for all $i' \in \pi^j$
3. Do $\pi(i) \leftarrow a_j$
4. End while
5. Return π .

Proof Note first that the whenever the body of the *While* loop is executed, the number of agents assigned to a non-void activity increases by one, and no agent is ever unassigned from a non-void activity. This means, in particular, that the algorithm always terminates. Conditions (b) and (c) in Step 2 ensure that agents assigned to a non-void

activity are always happy with the number of participants assigned to that activity, i.e., for every $i \in N$ with $\pi(i) \neq a_\emptyset$ it holds that $(a_j, |\pi^j|) \in S_i$. Hence, the assignment returned by the algorithm is individually rational. Now, when the algorithm stops, either there are no agents assigned to the void activity, in which case the returned assignment π is perfect (and, *a fortiori*, individually stable), or for every agent i such that $\pi(i) = a_\emptyset$ there is no a_j such that $(a_j, |\pi^j| + 1) \in S_i$ and $(a_j, |\pi^j| + 1) \in S_{i'}$ for all $i' \in \pi^j$. Hence π is individually stable. \square

We remark that an analogue of Proposition 2 for hedonic games with dichotomous preferences has been observed by Peters (2016) (Proposition 2).

Note also that if all agents have increasing preferences, individual stability is equivalent to Nash stability: no player would object to having new members join his group.

Example 5 Consider the execution of our algorithm on the first instance of a-GASP in Example 2 (the one with $n = 5$). Initially, everyone is assigned to a_\emptyset . Then the algorithm proceeds as follows:

- 1 $\mapsto a$;
- 2 $\mapsto a$, because agent 1 does not object to that.

At this point the algorithm terminates: it cannot assign anyone to a because agent 2 would object, and it cannot assign anyone to b , because no agent approves $(b, 1)$. The resulting assignment is individually stable, even though it is far from being maximum-size: e.g., the first assignment in Example 3 is also individually stable, yet assigns four agents to non-void activities.

A related solution concept for hedonic games is *contractual individual stability*: under this concept, an agent is only allowed to move from one coalition to another if neither the members of his new coalition nor the members of his old coalition object to the move. It is not hard to adapt this concept for GASP.

Definition 8 Given an instance (N, A, P) of GASP, an assignment $\pi : N \rightarrow A$ is said to be *contractually individually stable* if

- (1) π is individually rational,
- (2) for every agent $i \in N$ with $\pi(i) \neq a_\emptyset$ and every $a_j \in A^* \setminus \{\pi(i)\}$ such that $(a_j, |\pi^j| + 1) \succ_i (\pi(i), |\pi(i)|)$ there exists an agent $i' \in \pi^j$ with $(a_j, |\pi^j|) \succ_{i'} (a_j, |\pi^j| + 1)$ or an agent $i'' \in \pi^i \setminus \{i\}$ with $(a_i, |\pi(i)|) \succ_{i''} (a_i, |\pi(i)| - 1)$, and
- (3) for every agent $i \in N$ with $\pi(i) = a_\emptyset$ and every $a_j \in A^*$ such that $(a_j, |\pi^j| + 1) \succ_i a_\emptyset$ there exists an agent $i' \in \pi^j$ with $(a_j, |\pi^j|) \succ_{i'} (a_j, |\pi^j| + 1)$.

However, in a-GASP, for individually rational assignments contractual individual stability is equivalent to individual stability. Indeed, in a-GASP no agent assigned to a non-void activity has an incentive to deviate, so we only need to consider deviations from singleton coalitions.

The solution concepts discussed so far deal with individual deviations; resistance to group deviations is captured by the notion of the *core* and its variants. One typically distinguishes between *strong* group deviations, which are beneficial for each member of the deviating group, and *weak* group deviations, where the deviation should be beneficial for at least one member of the deviating group and non-harmful for others; these notions of deviation correspond to, respectively, the *core* and the *strict core*.

Definition 9 Given an instance (N, A, P) of GASP, an assignment $\pi : N \rightarrow A$ is said to be *core stable* (or in the *core*) if it is individually rational and there is no $E \subseteq N$, $E \neq \emptyset$, such that for some $a_j \in A^*$ with $\pi^j \subseteq E$ it holds that $(a_j, |E|) \succ_i (\pi(i), |\pi(i)|)$ for all $i \in E$.

In the context of **a-GASP**, if the initial assignment is individually rational, strong group deviations amount to players in π^0 forming a coalition in order to engage in a non-void activity. Therefore, the definition of core stability simplifies as follows: an assignment π is in the core if it is individually rational and there is no $E \subseteq \pi^0$ such that for some $a_j \in A^*$ with $\pi^j = \emptyset$ it holds that $(a_j, |E|) \in S_i$ for all $i \in E$.

It is immediate from Definition 9 that every instance of **a-GASP** has a non-empty core, and an outcome in the core can be constructed by a natural algorithm given below; we note that for hedonic games with dichotomous preferences, a similar algorithm has been proposed by Aziz et al. (2016) (Proposition 8).

1. Initialization: $\pi(i) \leftarrow a_\emptyset$ for all $i \in N$; $A' \leftarrow A^*$;
2. While there is an activity $a \in A'$ and an integer k with $1 \leq k \leq n$ such that at least k agents assigned to a_\emptyset approve (a, k)
3. Do assign these k agents to a ; remove a from A'
4. End while
5. Return π .

Definition 10 Given an instance (N, A, P) of GASP, an assignment $\pi : N \rightarrow A$ is said to be *strictly core stable* (or in the *strict core*) if it is individually rational and there is no $E \subseteq N$ such that for some $a_j \in A^*$ with $\pi^j \subseteq E$ it holds that $(a_j, |E|) \succeq_i (\pi(i), |\pi(i)|)$ for all $i \in E$ and $(a_j, |E|) \succ_i (\pi(i), |\pi(i)|)$ for some $i \in E$.

For **a-GASP**, we obtain the following definition: an assignment π is strictly core stable if it is individually rational and there is no $E \subseteq N$ with $E \cap \pi^0 \neq \emptyset$ such that for some $a_j \in A^*$ with $\pi^j \subseteq E$ it holds that $(a_j, |E|) \in S_i$ for all $i \in E$. Here, the condition $E \cap \pi^0 \neq \emptyset$ ensures that at least one deviating agent strictly prefers the new outcome. Note that, in contrast to strong deviations, weak deviations may involve agents engaged in a non-void activity.

Condition $\pi^j \subseteq E$ in Definitions 9 and 10 captures that intuition that deviating agents cannot prevent other agents from engaging in their selected activity, and, to join a group that has selected some activity, the deviators require cooperation of the group members.

While the core of an **a-GASP** is always non-empty, the strict core may be empty. In addition, note that a maximum individually rational assignment is always individually stable and in the core, but may not be in the strict core; this holds even if agents have increasing preferences.

Example 6 Consider an instance (N, A, P) of **a-GASP** where $N = \{1, 2, 3\}$, $A^* = \{a, b\}$, $S_1 = \{(a, 2), (a, 3)\}$, $S_2 = \{(b, 2), (b, 3)\}$, and $S_3 = \{(a, 2), (a, 3), (b, 2), (b, 3)\}$. Both agent 1 and agent 2 want to form a coalition with agent 3 (who is happy to oblige), but have incompatible preferences over activities. Thus, the strict core of this game is empty.

By adding activities not approved by any agents and agents that do not approve any activities, we can extend Example 6 to any A^* with $|A^*| \geq 2$ and any $n \geq 3$. However, the following proposition illustrates that both of these conditions are necessary: if $|A^*| = 1$ or $n \leq 2$, the strict core of an a-GASP is guaranteed to be non-empty.

Proposition 3 *For every instance of a-GASP with $n \leq 2$ or $|A^*| = 1$ every maximum individually rational assignment is in the strict core.*

Proof If $n = 1$, our claim is immediate. Now, suppose that $n = 2$. If our instance admits a perfect assignment, then this assignment is in the strict core. Thus, suppose that this is not the case. If the unique individually rational assignment π is such that $\pi(1) = \pi(2) = a_\emptyset$, there is nothing that the agents can do, so π is in the strict core. Finally, suppose that in every maximum individually rational assignment exactly one agent is assigned to the void activity. Then any acceptable deviation that improves his welfare results in a perfect assignment, a contradiction.

If $A^* = \{a_1\}$, consider a maximum individually rational assignment π ; we have $\#\pi = |\pi^1|$. If a coalition E can successfully deviate from π , then $\pi^1 \subset E, \pi^0 \cap E \neq \emptyset$. Hence, by assigning all agents in E to a_1 we obtain an individually rational assignment π' with $\#\pi' > \#\pi$, a contradiction. \square

It is not hard to see that strict core stability implies core stability and individual stability, and Nash stability implies individual stability. However, the converse implications are not true. Moreover, Nash stability does not imply core stability or strict core stability, core stability does not imply individual stability, and neither core stability nor strict core stability imply Nash stability.

4 Computing good outcomes

In this section, we consider the computational complexity of finding a “good” assignment for a-GASP. We mostly focus on finding perfect or maximum individually rational assignments; towards the end of the section, we also consider Nash stability and strict core stability. Besides the general case of our problem, we consider special cases obtained by combining constraints on the number and type of activities (e.g., unlimited number of simple activities, a constant number of copyable activities, etc.) and constraints on voters’ preferences (INC, DEC, INV, etc.). Note that if we can find a maximum individually rational assignment, we can easily check if a perfect assignment exists, by looking at the size of our maximum individually rational assignment. Thus, we will state our hardness results for the “easier” perfect assignment problem and phrase our polynomial-time algorithms in terms of the “harder” problem of finding a maximum individually rational assignment.

4.1 Individual rationality: hardness results

We start by presenting four NP-completeness results, which show that deciding whether a perfect assignment exists is hard even under fairly strong constraints on preferences and activities. We remark that this problem is in NP (one can check in

linear time if a given assignment is individually rational and assigns each agent to a non-void activity), so in what follows we will only provide the proofs of NP-hardness.

Our first hardness result applies when all activities are simple and the agents' preferences are increasing.

Theorem 1 *It is NP-complete to decide whether a-GASP admits a perfect assignment, even when all activities in A^* are simple and all agents have increasing preferences.*

Proof We provide a reduction from EXACT COVER BY 3-SETS (X3C). Recall that an instance of X3C is a pair $\langle X, \mathcal{Y} \rangle$, where $X = \{1, \dots, 3q\}$ and $\mathcal{Y} = \{Y_1, \dots, Y_p\}$ is a collection of 3-element subsets of X ; it is a “yes”-instance if X can be covered by exactly q sets from \mathcal{Y} , and a “no”-instance otherwise. It is known that X3C remains NP-complete even when restricted to instances $\langle X, \mathcal{Y} \rangle$ in which for every pair of sets $Y_g, Y_j \in \mathcal{Y}$ with $g \neq j$ we have $|Y_g \cap Y_j| \leq 1$ (see Jiang et al. 2011). Given an instance $\langle X, \mathcal{Y} \rangle$ of X3C that satisfies this restriction, we construct an instance of a-GASP as follows. We set $N = \{1, \dots, 3q\}$ and $A^* = \{a_1, \dots, a_p\}$. For each agent i , we define his vote V_i so that the induced approval vote S_i is given by $S_i = \{(a_j, k) \mid i \in Y_j, k \geq 3\}$, and let $P = (V_1, \dots, V_n)$. Clearly, (N, A, P) is an instance of a-GASP with increasing preferences and the description size of this instance is polynomial in q and p . The condition $|Y_g \cap Y_j| \leq 1$ for $g \neq j$ ensures that all activities are simple. We claim that $\langle X, \mathcal{Y} \rangle$ is a “yes”-instance of X3C if and only if (N, A, P) admits a perfect assignment.

Indeed, suppose that there is a set of indices $I \subseteq \{1, \dots, p\}$ such that $|I| = q$ and $\cup_{i \in I} Y_i = X$. Define the assignment π by $\pi(i) = a_j$, where j is such that (1) $i \in Y_j$ and (2) $j \in I$. As $\{Y_i\}_{i \in I}$ is a partition of X , this assignment is well-defined and each agent is assigned to a non-void activity. Moreover, for each agent $i \in N$ we have $(\pi(i), 3) \in S_i$ and the number of agents assigned to $\pi(i)$ in π is exactly 3, so π is individually rational. Hence, π is a perfect assignment.

Conversely, assume that π is a perfect assignment for (N, A, P) . Consider an alternative $a_j \in A^*$ with $\pi^j \neq \emptyset$. Since a_j is approved by exactly 3 agents, we have $|\pi^j| \leq 3$. On the other hand, since no agent approves of $(a_j, 1)$ or $(a_j, 2)$, it has to be the case that $|\pi^j| = 3$. Now, since π is a perfect assignment, it follows that there are exactly q activities a_j with $|\pi^j| = 3$. Therefore, the collection of subsets $\{Y_j \mid |\pi^j| = 3\}$ is an exact cover for X . \square

Our second hardness result applies to simple activities and decreasing preferences, and holds even if each agent is willing to share each activity with at most one other agent. Its proof (together with proofs of Theorems 3 and 4) is delayed to appendix A.

Theorem 2 *It is NP-complete to decide whether a-GASP admits a perfect assignment, even when all activities in A^* are simple, all agents have decreasing preferences, and, moreover, for every agent $i \in N$ and every alternative $a \in A^*$ we have $S_i^{\downarrow a} \subseteq \{1, 2\}$.*

Our third hardness result also concerns simple activities and decreasing preferences. However, unlike Theorem 2, it holds even if each agent approves of at most 3 activities.

Theorem 3 *It is NP-complete to decide whether a-GASP admits a perfect assignment, even when all activities in A^* are simple, all agents have decreasing preferences, and, moreover, for every agent $i \in N$ it holds that $|\{a \mid S_i^{\downarrow a} \neq \emptyset\}| \leq 3$.*

Our fourth hardness result applies even when there is only one activity, which is ∞ -copyable, and every agent approves at most two alternatives; however, the agents' preferences constructed in our proof do not satisfy any of the structural constraints defined in Sect. 2.1.

Theorem 4 *It is NP-complete to decide whether a-GASP admits a perfect assignment, even when all activities in A^* are equivalent (i.e., A^* consists of a single ∞ -copyable activity a) and for every agent $i \in N$ we have $|S_i^{\downarrow a}| \leq 2$.*

4.2 Individual rationality: easiness results

The hardness results in Sect. 4.1 imply that if A^* contains an unbounded number of distinct activities, finding a maximum individually rational assignment is computationally hard, even under strong restrictions on agents' preferences (such as INC or DEC). Thus, we can only hope to develop an efficient algorithm for this problem if we assume that the total number of activities is small (i.e., bounded by a constant) or, more liberally, that the number of pairwise non-equivalent activities is small, and the agents' preferences satisfy additional constraints. We will now consider both of these settings, starting with the case where $m = |A^*|$ is bounded by a constant.

Theorem 5 *There exists an algorithm that given an instance of a-GASP finds a maximum individually rational assignment and runs in time $(n + 1)^m \text{poly}(n)$.*

Proof We will check, for each $r = 0, \dots, n$, if there is an individually rational assignment π with $\#(\pi) = r$, and output the maximum value of r for which this is the case. Fix an $r \in \{0, \dots, n\}$. For every vector $(n_1, \dots, n_m) \in \{0, \dots, n\}^m$ that satisfies $n_1 + \dots + n_m = r$ we will check if there exists an assignment of agents to activities such that for each $j = 1, \dots, m$ exactly n_j agents are assigned to activity a_j (with the remaining agents being assigned to the void activity), and each agent approves of the resulting assignment. Each check will take $\text{poly}(n)$ steps, and there are at most $(n + 1)^m$ vectors to be checked; this implies our bound on the running time of our algorithm.

For a fixed vector (n_1, \dots, n_m) , we construct an instance of the network flow problem as follows. Our network has a source s , a sink t , a node i for each player $i = 1, \dots, n$, and a node a_j for each $a_j \in A^*$. There is an arc of unit capacity from s to each agent, and an arc of capacity n_j from node a_j to the sink. Further, there is an arc of unit capacity from i to a_j if and only if $(a_j, n_j) \in S_i$. It is not hard to see that an integral flow F of size r in this network corresponds to an individually rational assignment of size r . It remains to observe that it can be checked in polynomial time whether a given network admits a flow of a given size. \square

Moreover, when A^* consists of a single simple activity a , a maximum individually rational assignment can be found by a straightforward algorithm.

Proposition 4 *Given an instance (N, A, P) of a-GASP with $A^* = \{a\}$, we can find a maximum individually rational assignment for (N, A, P) in time $O(s)$, where $s = \sum_{i \in N} |S_i|$.*

Proof Clearly, (N, A, P) admits an individually rational assignment π with $\#(\pi) = k$ if and only if $|\{i \mid (a, k) \in S_i\}| \geq k$. We use an integer array $C[\ell], \ell \in N$. We initialize $C[\ell] = 0$ for all $\ell \in N$. Then for each $i \in N$ we scan the set S_i ; for each $(a, j) \in S_i$ we increment $C[j]$ by 1. This procedure can be completed in time $O(s)$; at the end of this step for each $\ell \in N$ the cell $C[\ell]$ of our array stores the number of agents who approve (a, ℓ) . We then scan this array right to left to find the largest value of ℓ with $C[\ell] \geq \ell$. Once this value is found, another scan through agents' approval lists is sufficient to find a corresponding coalition. \square

Example 7 Consider the following instance of a-GASP with $n = 6$ and $A^* = \{a\}$.

| | (a, 1) | (a, 2) | (a, 3) | (a, 4) | (a, 5) | (a, 6) |
|---|--------|--------|--------|--------|--------|--------|
| 1 | – | – | – | + | + | + |
| 2 | – | + | + | + | + | + |
| 3 | – | – | + | + | – | – |
| 4 | – | + | + | – | – | – |
| 5 | – | – | – | + | + | + |
| 6 | + | + | + | + | + | + |

We obtain

$$C[6] = 4, C[5] = 4, C[4] = 5, C[3] = 4, C[2] = 3, C[1] = 1.$$

Thus, our algorithm finds $\ell = 4$ and outputs the coalition $\{1, 2, 3, 5\}$.

Now, suppose that A^* contains many activities, but most of them are equivalent to each other; for instance, A^* may consist of a single k -copyable activity, for a large value of k . Then the algorithm described in the proof of Theorem 5 is no longer efficient, but this setting still appears to be more tractable than the one with many distinct activities. Of course, by Theorem 4, in the absence of any restrictions on the agents' preferences, finding a maximum individually rational assignment is hard even for a single ∞ -copyable activity. However, we will now show that this problem becomes easy if we additionally assume that the agents' preferences are increasing or decreasing.

Observe first that for increasing preferences having multiple copies of the same activity is not useful: if there is an individually rational assignment where agents are assigned to multiple copies of an activity, we can reassign these agents to a single copy of this activity without violating individual rationality. Thus, we obtain the following easy corollary to Theorem 5.

Corollary 1 *Let (N, A, P) be an instance of a-GASP with increasing preferences where A^* contains at most K activities that are pairwise non-equivalent. Then we can find a maximum individually rational assignment for (N, A, P) in time $n^K \text{poly}(n)$.*

If all preferences are decreasing, we can simply eliminate all ∞ -copyable activities. Indeed, consider an instance (N, A, P) of **a-GASP** where some activity $a \in A^*$ is ∞ -copyable. Then we can assign each agent $i \in N$ such that $(a, 1) \in S_i$ to his own copy of a ; clearly, this will only simplify the problem of assigning the remaining agents to the activities.

It remains to consider the case where the agents' preferences are decreasing, there is a limited number of copies of each activity, and the number of distinct activities is small. While we do not have a complete solution for this case, we can show that in the case of a single k -copyable activity a natural algorithm succeeds in finding a maximum individually rational assignment.

Theorem 6 *Given a decreasing instance (N, A, P) of **a-GASP** where A^* consists of a single k -copyable activity (i.e., $A^* = \{a_1, \dots, a_k\}$, and all activities in A^* are pairwise equivalent), we can find a maximum individually rational assignment in time $O(n \log n)$.*

Proof Since all activities in A^* are pairwise equivalent, we can associate each agent $i \in N$ with a single number $u_i \in \{0, \dots, n\}$, which is the maximum size of a coalition assigned to a non-void activity that he is willing to be a part of. We will show that our problem can be solved by a simple algorithm. Specifically, we sort the agents in non-increasing order of u_i s. From now on, we will assume without loss of generality that $u_1 \geq \dots \geq u_n$. To form the first group, we find the largest value of i such that $u_i \geq i$, and assign agents $1, \dots, i$ to the first copy of the activity. In other words, we continue adding agents to the group as long as the agents are happy to join. We repeat this procedure with the remaining agents until either k groups have been formed or all agents have been assigned to one of the groups, whichever happens earlier.

Clearly, the sorting step is the bottleneck of this procedure, so the running time of our algorithm is $O(n \log n)$. It remains to argue that it produces a maximum individually rational assignment. To show this, we start with an arbitrary maximum individually rational assignment π and transform it into the one produced by our algorithm without lowering the number of agents that have been assigned to a non-void activity. We will assume without loss of generality that π assigns all k copies of the activity (even though this is not necessarily the case for the algorithm).

First, suppose that $\pi(i) = a_\emptyset, \pi(j) = a_\ell$ for some $i < j$ and some $\ell \in \{1, \dots, k\}$. Then we can modify π by setting $\pi(i) = a_\ell, \pi(j) = a_\emptyset$. Since $i < j$ implies $u_i \geq u_j$, the modified assignment is individually rational. By applying this operation repeatedly, we can assume that the set of agents assigned to a non-void activity forms a contiguous prefix of $1, \dots, n$.

Next, we will ensure that for each $\ell = 1, \dots, k$ the group of agents that are assigned to a_ℓ forms a contiguous subsequence of $1, \dots, n$. To this end, let us sort the coalitions in π according to their size, from the largest to the smallest, breaking ties arbitrarily. That is, we reassign the k copies of our activity to coalitions in π so that $\ell < r$ implies $|\pi^\ell| \geq |\pi^r|$. Now, suppose that there exist a pair of players i, j such that $i < j$, $\pi(i) = a_\ell, \pi(j) = a_r$, and $\ell > r$ (and hence $|\pi^\ell| \leq |\pi^r|$). We have $u_j \geq |\pi^r| \geq |\pi^\ell|$, $u_i \geq u_j \geq |\pi^r|$, so if we swap i and j (i.e., modify π by setting $\pi(j) = a_\ell, \pi(i) = a_r$), the resulting assignment remains individually rational. Observe that every such swap increases the quantity $\Sigma = \sum_{t=1}^k \sum_{s \in \pi^t} (s \cdot t)$ by at least 1: prior to the swap, the

contribution of i and j to Σ is $il + jr$, and after the swap it is $ir + jl > il + jr$. Since for any assignment we have $\Sigma \leq kn(n + 1)/2$, eventually we arrive to an assignment where no such pair (i, j) exists. At this point, each π^ℓ , $\ell = 1, \dots, k$, forms a contiguous subsequence of $1, \dots, n$, and, moreover, $\ell < r$ implies $i \leq j$ for all $i \in \pi^\ell, j \in \pi^r$.

Now, consider the smallest value of ℓ such that π^ℓ differs from the ℓ -th coalition constructed by the algorithm (let us denote it by γ^ℓ), and let i be the first agent in $\pi^{\ell+1}$. The description of the algorithm implies that π^ℓ is a strict subset of γ^ℓ and agent i belongs to γ^ℓ . Thus, if we modify π by moving agent i to π^ℓ , the resulting allocation remains individually rational (since i is happy in γ^ℓ). By repeating this step, we will gradually transform π into the output of the algorithm (possibly discarding some copies of the activity). This completes the proof. \square

Example 8 Consider the following instance of a-GASP with $n = 7$ and A^* consisting of a single 3-copyable activity a .

| | (a, 1) | (a, 2) | (a, 3) | (a, 4) | (a, 5) | (a, 6) | (a, 7) |
|---|--------|--------|--------|--------|--------|--------|--------|
| 1 | + | + | + | + | + | + | - |
| 2 | + | + | + | + | - | - | - |
| 3 | + | + | + | + | - | - | - |
| 4 | + | + | + | - | - | - | - |
| 5 | + | + | - | - | - | - | - |
| 6 | + | + | - | - | - | - | - |
| 7 | + | - | - | - | - | - | - |

With u_i being the maximum size of a coalition agent i is willing to be part of, we have $u_1 \geq \dots \geq u_7$. The algorithm described in the proof of Theorem 6 starts by assigning agents 1, 2, 3 to the first copy of a . It then assigns agents 4 and 5 to the second copy of a . Finally, agent 6 is assigned to the third copy of a , and agent 7 is assigned to the void activity.

The algorithm described in the proof of Theorem 6 can be extended to the case where we have one k -copyable activity a and one simple activity b , and the agents have decreasing preferences over both activities. For each $s = 1, \dots, n$ we will look for the best solution in which s players are assigned to b ; we will then pick the best of these n solutions. For a fixed s let $N_s = \{i \in N \mid (b, s) \in S_i\}$. If $|N_s| < s$, no solution for this value of s exists. Otherwise, we have to decide which size- s subset of N_s to assign to b . It is not hard to see that we should simply pick the agents in N_s that have the lowest level of tolerance for a , i.e., we order the agents in N_s by the values of $u_i^a = \max\{j \in N \mid (a, j) \in S_i\}$ from the smallest to the largest, and pick the first s agents. We then assign the remaining agents to copies of a using the algorithm from the proof of Theorem 6. Indeed, any assignment can be transformed into one of this form by swapping agents so that the individual rationality constraints are not broken. It would be interesting to see if this idea can be extended to the case where instead of a single simple activity b we have a constant number of simple activities or a single k' -copyable activity.

We conclude this section by giving an $O(\sqrt{n})$ -approximation algorithm for finding a maximum individually rational assignment in a-GASP with a single ∞ -copyable activity.

Theorem 7 *There exists a polynomial-time algorithm that given an instance (N, A, P) of a-GASP where A^* consists of a single ∞ -copyable activity a , outputs an individually rational assignment π with $\#(\pi) = \Omega(\frac{1}{\sqrt{n}})\#(N, A, P)$.*

Proof We say that an agent i is active in π if $\pi(i) \neq a_\emptyset$; a coalition of agents is said to be active if it is assigned to a single copy of a . We construct an individually rational assignment π iteratively, starting from the assignment where no agent is active. Let $N^* = \{i \mid \pi(i) = a_\emptyset\}$ be the current set of inactive agents (initially, we set $N^* = N$). At each step, we find the largest subset of N^* that can be assigned to a single fresh copy of a without breaking the individual rationality constraints, and append this assignment to π . We repeat this step until the inactive agents cannot form another coalition.

Now we compare the number of active agents in π with the number of active agents in a maximum individually rational assignment π^* . To this end, let us denote the active coalitions of π by B_1, \dots, B_s , where $|B_1| \geq \dots \geq |B_s|$. If $|B_1| \geq \sqrt{n}$, we are done, so assume that this is not the case. Note that since B_1 was chosen greedily, this implies that $|C| \leq \sqrt{n}$ for every active coalition C in π^* .

Let \mathcal{C} be the set of active coalitions in π^* . We partition \mathcal{C} into s groups by setting $\mathcal{C}^1 = \{C \in \mathcal{C} \mid C \cap B_1 \neq \emptyset\}$ and $\mathcal{C}^i = \{C \in \mathcal{C} \mid C \cap B_i \neq \emptyset, C \not\subseteq C^j \text{ for } j < i\}$ for $i = 2, \dots, s$. Note that every active coalition $C \in \pi^*$ intersects some coalition in π : otherwise we could add C to π . Therefore, each active coalition in π^* belongs to one of the sets $\mathcal{C}^1, \dots, \mathcal{C}^s$. Also, by construction, the sets $\mathcal{C}^1, \dots, \mathcal{C}^s$ are pairwise disjoint. Further, since the coalitions in \mathcal{C}^i are pairwise disjoint and each of them intersects B_i , we have $|\mathcal{C}^i| \leq |B_i|$ for each $i = 1, \dots, s$. Thus, we obtain

$$\begin{aligned} \#(\pi^*) &= \sum_{i=1, \dots, s} \sum_{C \in \mathcal{C}^i} |C| \leq \sum_{i=1, \dots, s} \sum_{C \in \mathcal{C}^i} \sqrt{n} \\ &\leq \sum_{i=1, \dots, s} |\mathcal{C}^i| \sqrt{n} \leq \sum_{i=1, \dots, s} |B_i| \sqrt{n} \leq \#(\pi) \sqrt{n}. \end{aligned}$$

Example 9 Consider the following parameterized instance of a-GASP with $n = \frac{k(k+1)}{2}$, where A^* consists of a single ∞ -copyable activity a and $k \in \mathbb{N}$. For each $i = 1, \dots, k - 1$, we set $S_i^{\downarrow a} = \{k, k - i\}$, and we set $S_k^{\downarrow a} = \{k\}$. We then split players $k + 1, \dots, n$ into consecutive groups of size $k - 1, k - 2, \dots, 1$ and for each player i in a group of size h we set $S_i^{\downarrow a} = \{h + 1\}$: formally, for each $h \in \{k - 1, \dots, 1\}$ and each i such that $1 + \sum_{j=h+1}^k j \leq i \leq \sum_{j=h}^k j$ we set $S_i^{\downarrow a} = \{h + 1\}$. For $k = 4$, i.e., $n = 10$, we obtain the following instance:

| | (a, 1) | (a, 2) | (a, 3) | (a, 4) | (a, 5) | (a, 6) | (a, 7) | (a, 8) | (a, 9) | (a, 10) |
|----|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|
| 1 | − | − | + | + | − | − | − | − | − | − |
| 2 | − | + | − | + | − | − | − | − | − | − |
| 3 | + | − | − | + | − | − | − | − | − | − |
| 4 | − | − | − | + | − | − | − | − | − | − |
| 5 | − | − | − | + | − | − | − | − | − | − |
| 6 | − | − | − | + | − | − | − | − | − | − |
| 7 | − | − | − | + | − | − | − | − | − | − |
| 8 | − | − | + | − | − | − | − | − | − | − |
| 9 | − | − | + | − | − | − | − | − | − | − |
| 10 | − | + | − | − | − | − | − | − | − | − |

We start by analyzing the behavior of the algorithm from the proof of Theorem 7 for $k = 4$. In the first step, the algorithm assigns a largest possible subset of agents—say $\{1, 2, 3, 4\}$ —to the first copy of a . But then it is easy to verify that no further agent can be assigned to a copy of a . On the other hand, in the maximum individually rational assignment each of the sets $\{4, 5, 6, 7\}$, $\{1, 8, 9\}$, $\{2, 10\}$, and $\{3\}$ gets a single copy of a .

Thus, the algorithm yields an assignment with only 4 agents assigned to a non-void activity, while the maximum individually rational assignment assigns all 10 agents to a copy of a .

In general, the algorithm assigns the agents $\{1, \dots, k\}$ to the first copy of a , and makes no further assignments. However, the maximum individually rational assignment assigns all of the $n = k(k + 1)/2$ agents to a copy of a : For each $h \in \{1, \dots, k - 2\}$ the set of agents $\{1 + \sum_{j=h+1}^k j, \dots, \sum_{j=h}^k j, k - h - 1\}$ is assigned to a different copy of a , agent $k - 1$ gets his own copy of a , and another copy of a is allocated to $\{k, \dots, 2k - 1\}$.

Example 9 shows that our analysis of the algorithm described in the proof of Theorem 7 is tight up to a constant factor. However, this does not rule out the possibility that an algorithm with a better approximation ratio may exist.

4.3 Nash stability

We have shown that **a-GASP** does not not always admit a Nash stable assignment (Proposition 1). In fact, it is difficult to determine whether a Nash stable assignment exists.

Theorem 8 *It is NP-complete to decide whether a-GASP admits a Nash stable assignment.*

Proof We reduce from the problem of deciding whether **a-GASP** admits a perfect assignment. Let (N, A, P) be an instance of **a-GASP**. We define an instance (N', A', P') of **a-GASP** by setting $N' = N \cup \{n + 1\}$, $A' = A \cup \{b\}$, and $P' = (V'_1, \dots, V'_{n+1})$, where for each $i \in N$, V'_i is constructed so that its induced approval vote S'_i is given by $S'_i = S_i \cup \{(b, 1)\}$ and $S'_{n+1} = \{(b, 2)\}$. By construction, the size of (N', A', P') is polynomial in that of (N, A, P) .

Assume that (N, A, P) admits a perfect assignment π . Define an assignment π' for (N', A', P') as follows: $\pi'(i) = \pi(i)$ for all $i \in N$, and $\pi'(n+1) = a_\emptyset$. Clearly, π' is Nash stable.

Conversely, assume that (N', A', P') admits a Nash stable assignment π' . If two or more agents are assigned to b , then π' is not individually rational (because there is only one agent who is willing to share b) and hence not Nash stable. If exactly one agent is assigned to b , by individual rationality it is an agent $i \in N$, so π' is not Nash stable because $n+1$ would like to join b . Therefore, no agent is assigned to b . Now, if there is an agent $i \in N$ with $\pi'(i) = a_\emptyset$, then π' is not Nash stable because i would like to engage in b . Therefore, each agent $i \in N$ is assigned to a non-void activity. Since π' is individually rational, this means that its restriction to N is a perfect assignment for (N, A, P) . \square

However, if agents' preferences satisfy INC, DEC, or MIX, a Nash stable assignment always exists and can be computed efficiently.

Theorem 9 *Given a mixed increasing–decreasing instance (N, A, P) of a-GASP, we can find a Nash stable assignment in polynomial time.*

Proof Suppose first that agents' preferences are increasing, i.e., $A^* = A^+$. We can then start by choosing an arbitrary individually rational assignment π (e.g., $\pi(i) = a_\emptyset$ for all $i \in N$). If π is not Nash stable, there exists an agent $i \in N$ with $\pi(i) = a_\emptyset$ and an activity $a_j \in A^*$ such that $(a, |\pi^j| + 1) \in S_i$. We can then modify π by setting $\pi(i) = a_j$; clearly, this assignment remains individually rational. If the resulting assignment is still not Nash stable, we can repeat this step. Since at each step the number of agents assigned to the void activity goes down by 1, this process stops after at most n steps.

For decreasing preferences, we proceed as follows. We consider the activities one by one; at step j , we consider activity a_j . Let $N_j \subseteq N$ be the set of agents that remain unassigned at the beginning of step j . Let $N_{j,\ell} = \{i \in N_j \mid (a_j, \ell) \in S_i\}$, and set $k = \max\{\ell \mid N_{j,\ell} \geq \ell\}$. Thus, k is the size of the largest group of currently unassigned agents that can be assigned to a_j . By our choice of k , the set N_j contains at most k agents that are willing to share a_j with $k+1$ or more other agents. We assign all these agents to a_j ; if the resulting coalition contains $\ell < k$ agents, we assign $k - \ell$ additional agents that approve of (a_j, k) to a_j (the existence of these $k - \ell$ agents is guaranteed by our choice of k). This completes the description of the j -th step. Note that no agent that remains unassigned after this step wants to be assigned to a_j : indeed, this activity is currently shared among k agents, so if he were to join, the size of the group that is assigned to a_j would increase to $k+1$, and none of the unassigned agents is willing to share a_j with $k+1$ other agents. If some agents remain unassigned after n steps, we assign them to the void activity. To see that this assignment is Nash stable, consider an agent i assigned to the void activity. For each activity a_j he did not want to join the coalition of agents assigned to a_j during step j . Since the set of agents assigned to a_j did not change after step j , this is still the case.

For mixed decreasing–increasing instances, we first remove all activities in A^+ and apply our second algorithm to the remaining instance; we then consider the unassigned agents and assign them to activities in A^+ using the first algorithm. \square

Moreover, the problem of finding a Nash stable assignment that maximizes the number of agents assigned to a non-void activity admits an efficient algorithm if A^* consists of a single simple activity.

Theorem 10 *Given an instance (N, A, P) of a-GASP with $A^* = \{a\}$ we can, in polynomial time, find a Nash stable assignment maximizing the number of agents assigned to a non-void activity among all Nash stable assignments, or decide that no Nash stable assignment exists.*

Proof For each $k = n, \dots, 0$, we will check if there exists a Nash stable assignment π with $\#(\pi) = k$, and output the largest value of k for which this is the case.

For each $i \in N$, let $S'_i = S_i^{\downarrow a}$. For $k = n$ a Nash stable assignment π with $\#(\pi) = n$ exists if and only if $n \in S'_i$ for each $i \in N$. Assigning every agent to a_\emptyset is Nash stable if and only if $1 \notin S'_i$ for each $i \in N$. Now we assume $1 \leq k \leq n - 1$ and set

$$\begin{aligned} U_1 &= \{i \in N \mid k \in S'_i, k + 1 \notin S'_i\}, \\ U_2 &= \{i \in N \mid k \notin S'_i, k + 1 \in S'_i\}, \\ U_3 &= \{i \in N \mid k \in S'_i, k + 1 \in S'_i\}. \end{aligned}$$

If $|U_1| + |U_3| < k$, there does not exist an individually rational assignment π with $\#(\pi) = k$. If $U_2 \neq \emptyset$, no Nash stable assignment π with $\#(\pi) = k$ can exist, since each agent from U_2 would want to deviate. If $|U_3| > k$, no Nash stable assignment π with $\#(\pi) = k$ can exist, since at least one agent in U_3 would not be assigned to a and thus would be unhappy. Finally, if $|U_1| + |U_3| \geq k$, $|U_3| \leq k$, $U_2 = \emptyset$, we can construct a Nash stable assignment π by assigning all agents from U_3 and $k - |U_3|$ agents from U_1 to a . Since we have $\pi(i) = a_\emptyset$ for all i with $k \notin S'_i$ and $\pi(i) \neq a_\emptyset$ for all i with $k + 1 \in S'_i$, no agent is unhappy. \square

4.4 Core and strict core stability

We have seen that an assignment in the core always exists and can be computed by a simple algorithm (see Sect. 3). We will now consider computational problems associated with the strict core.

We first observe that verifying whether a given assignment is in the core/strict core is easy. Indeed, in both cases, given an assignment π , we can iterate over all activities in A^* and all potential sizes of a deviating coalition. For a fixed $a_j \in A^*$ and $k \in N$, let $s(j, k)$ be the number of players in π^0 who approve (a_j, k) . If $s(j, k) \geq k$, then π is not in the core, and if $s(j, k) < k$ for all $a_j \in A^*$ and all $k \in N$, then π is core stable. For the strict core, we check if there are at least $k - |\pi^j| - s(j, k)$ players in $N \setminus (\pi^j \cup \pi^0)$ who approve (a_j, k) ; π is not in the strict core if and only if this condition is satisfied for some $a_j \in A^*$ and $k \in N$. This implies, in particular, that deciding whether a given assignment is in the strict core is in NP.

Next, we will show that for decreasing preferences an assignment in the strict core can be computed efficiently, while for increasing preferences this is not the case.

Proposition 5 *For every instance of a-GASP with decreasing preferences the strict core is non-empty. Moreover, given an instance (N, A, P) of a-GASP with decreasing preferences, we can find an assignment in the strict core in polynomial time.*

Proof Fix an instance (N, A, P) of a-GASP with decreasing preferences, and consider the algorithm used to construct an assignment in the core (page 11). Let π be the assignment constructed by this algorithm. Assume there is a subset $E \subseteq N$ with $E \cap \pi^0 \neq \emptyset$ and an $a_j \in A^*$ with $\pi^j \subset E$ such that $(a_j, |E|) \in S_i$ for all $i \in E$. Clearly, we have $|E| > |\pi^j|$. Let $E' = E \cap (\pi^j \cup \pi^0)$. By construction, $|\pi^j| < |E'| \leq |E|$. Since all agents have decreasing preferences, for each $i \in E'$ it holds that $(a_j, |E'|) \in S_i$. It follows that when the agents in π^j were assigned to a_j by the algorithm, it would have been possible to assign all agents in E' —and hence more than $|\pi^j|$ agents—to a_j , a contradiction. Thus, the assignment π is in the strict core. Clearly, the algorithm runs in polynomial time. \square

Theorem 11 *It is NP-complete to decide whether a-GASP admits an assignment in the strict core, even when all agents have increasing preferences and all activities are simple.*

Proof We have already argued that this problem is in NP. In order to prove NP-hardness, we provide a reduction from the problem of deciding whether a given instance of a-GASP with increasing preferences and simple activities admits a perfect assignment (this problem is NP-complete by Theorem 1).

Let (N, A, P) be an instance of the perfect assignment problem for a-GASP with increasing preferences and simple activities. Let $N' = N \cup \{n + 1, n + 2\}$ and $A' = A \cup \{b, c\}$. For each $i \in N$, let V'_i be such that

$$S'_i = S_i \cup \{(a, n + 1), (a, n + 2) \mid (a, n) \in S_i\} \cup \{(b, n + 1), (b, n + 2)\}.$$

Further, let V'_{n+1} and V'_{n+2} be such that

$$S'_{n+1} = (\{c\} \times [2, n + 2]) \cup (\{b\} \times [n + 1, n + 2]), \quad S'_{n+2} = \{c\} \times [2, n + 2].$$

Set $P' = (V'_1, \dots, V'_{n+2})$. Obviously, (N', A', P') is an instance of a-GASP with increasing preferences and simple activities, which can be constructed in polynomial time. In what follows, we will show that (N, A, P) admits a perfect assignment if and only if (N', A', P') has a non-empty strict core.

Suppose that π is a perfect assignment for (N, A, P) . Then we can set $\pi'(i) = \pi(i)$ for $i \in N$ and $\pi'(n + 1) = \pi'(n + 2) = c$ to obtain a perfect assignment π' for (N', A', P') ; clearly, a perfect assignment is in the strict core.

On the other hand, suppose that π' is in the strict core of (N', A', P') . If $\pi'(n + 2) \neq c$, then $\pi'(n + 2) = a_\emptyset$. Thus, no agent can be assigned to c , because π' is individually rational. Hence, for $E = \{n + 1, n + 2\}$ we have $\{i \in N' \mid \pi'(i) = c\} \subset E$, $(c, |E|) \in S'_i$ for $i \in E$, $E \cap \pi^0 \neq \emptyset$, which violates the strict core condition. It follows that $\pi'(n + 2) = c$.

Since π' is individually rational and $S_i^{\downarrow c} = \emptyset$ for all $i \in N$, this means that $\pi'(n + 1) = c$. Consequently, no agent can be assigned to b under π' , and hence

Table 1 Summary of results

| | PERFECT | NASH | STRICT CORE |
|----------------------|-------------------|------------------|--------------------|
| Unrestricted | NP-c (Theorem 1) | NP-c (Theorem 8) | NP-c (Theorem 11) |
| Increasing | NP-c (Theorem 1) | P (Theorem 9) | NP-c (Theorem 11) |
| Decreasing | NP-c (Theorem 2) | P (Theorem 9) | P (Proposition 5) |
| Single activity (s.) | P (Proposition 4) | P (Theorem 10) | P (Proposition 3) |
| Single activity (c.) | NP-c (Theorem 4) | ? | NP-c (Corollary 2) |

NP-c stands for “NP-complete”. For PERFECT (respectively, NASH, STRICT CORE), the problem is to decide whether a given instance admits a perfect assignment (respectively, a Nash stable assignment, a strictly core stable assignment); the easiness results in the first column extend to the problem of finding a maximum individually rational assignment. The rows “single activity (s.)” and “single activity (c.)” correspond to A^* composed of, respectively, a single simple activity and a single ∞ -copyable activity. Hardness results for increasing and decreasing preferences hold even if all activities are simple

$\pi'(i) \notin \{b, c\}$ for all $i \in N$. Now, suppose that for some $i \in N$ we have $\pi'(i) = a_\emptyset$. Then, for $\tilde{E} = N \cup \{n+1\}$ we have $\{i \in N' \mid \pi'(i) = b\} \subset \tilde{E}$, $(b, |\tilde{E}|) \in S'_i$ for all $i \in \tilde{E}$, and $\tilde{E} \cap \pi^0 \neq \emptyset$. Again, this is a contradiction with the fact that π' is in the strict core. Hence, for all $i \in N$ we must have $\pi'(i) \neq a_\emptyset$, i.e., $\pi'|_N$ is a perfect assignment for (N, A, P) . \square

For a single simple activity, Proposition 3 shows that finding an assignment in the strict core of a given instance of a-GASP is easy. In contrast, for a single ∞ -copyable activity, this problem is hard. This can be shown by adapting the reduction in the proof of Theorem 4: it can be verified that for instances of a-GASP constructed in that proof it holds that an assignment is in the strict core if and only if it is perfect. We thus obtain the following corollary.

Corollary 2 *It is NP-complete to decide whether a-GASP admits an assignment in the strict core, even when A^* consists of a single ∞ -copyable activity a and for every $i \in N$ we have $|S_i^{\downarrow a}| \leq 2$.*

5 Conclusions and future work

We have defined a new model for selecting multiple group activities, discussed its connections with hedonic games, defined several stability notions, and, for all of them, we have obtained several complexity results. A number of our results are positive: finding desirable assignments proves to be tractable for several restrictions of the problem that are meaningful in practice (Table 1).

We see our work as a starting point in investigating the group activity selection problem. There are two obvious directions in which it can be extended:

- relaxing the anonymity assumption and have agents express their preferences over pairs consisting of an activity and a coalition, as done in the work of [Anshelevich and Sekar \(2014\)](#);
- instead of focusing on dichotomous preferences, have agents express rankings of the alternatives (or utility functions).

It would also be interesting to study strategic behavior in **GASP**: Is there a rule for assigning agents to activities that would encourage the agents to report their true preferences, under reasonable restrictions on the agents' preference structure? What is the quality of assignments produced by such rules?

Acknowledgements This research was supported by National Research Foundation (Singapore) under Research Fellowship NRF2009-08, by ERC grant agreement number 639945, by the ANR Project CoCoRiCo-CoDec and by the Austrian Science Fund (P23724-G11 and P23829-N13). This project was initiated during the Dagstuhl seminar 12101 "Computation and Incentives in Social Choice", and the authors are very grateful to Dagstuhl for providing a great research environment and inspiration for this work. We thank Craig Boutilier, Vincent Iehlé and Michel Le Breton for helpful comments. Part of this work was done when the second author was visiting Université Paris-Dauphine. We thank the reviewers and the associate editor for their helpful comments.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

A Proofs for Section 4.1

Theorem 2 *It is NP-complete to decide whether a-GASP admits a perfect assignment, even when all activities in A^* are simple, all agents have decreasing preferences, and, moreover, for every agent $i \in N$ and every alternative $a \in A^*$ we have $S_i^{\downarrow a} \subseteq \{1, 2\}$.*

Proof Consider the following restricted variant of the problem of scheduling on unrelated machines. There are n jobs and m machines. An instance of the problem is given by a collection of numbers $\{p_{ij} \mid i = 1, \dots, n, j = 1, \dots, m\}$, where p_{ij} is the running time of job i on machine j , and $p_{ij} \in \{1, 2, +\infty\}$ for every $i = 1, \dots, n$ and every $j = 1, \dots, m$. It is a "yes"-instance if there is a mapping $\rho : \{1, \dots, n\} \rightarrow \{1, \dots, m\}$ assigning jobs to machines so that the makespan is at most 2, i.e., for each $j = 1, \dots, m$ it holds that $\sum_{i:\rho(i)=j} p_{ij} \leq 2$. This problem is known to be NP-hard [see the proof of Theorem 5 in [Lenstra et al. \(1990\)](#)].

Given an instance $\{p_{ij} \mid i = 1, \dots, n, j = 1, \dots, m\}$ of this problem, we construct an instance of **a-GASP** as follows. We set $N = \{1, \dots, n\}$, $A^* = \{a_1, \dots, a_m\}$. Further, for each agent $i \in N$ we construct a vote V_i so that the induced approval vote S_i satisfies $S_i^{\downarrow a_j} = \{1\}$ if $p_{ij} = 2$, $S_i^{\downarrow a_j} = \{1, 2\}$ if $p_{ij} = 1$, and $S_i^{\downarrow a_j} = \emptyset$ if $p_{ij} = +\infty$. Clearly, these preferences satisfy the constraints in the statement of the theorem, and our reduction can be performed in polynomial time.

Suppose there is a mapping ρ that assigns jobs to machines so that the makespan is at most 2, and consider an assignment π given by $\pi(i) = a_j$ if and only if $\rho(i) = j$. Clearly, under this assignment each agent is assigned to a non-void activity, and, moreover, $\pi(i) = a_j$ implies $S_i^{\downarrow a_j} \neq \emptyset$ (since otherwise the completion time of machine j under ρ would be $+\infty$). Now, consider an agent i with $\pi(i) = a_j$. Since under ρ each machine can be assigned at most 2 jobs, we have $|\pi^j| \leq 2$. Thus, if $S_i^{\downarrow a_j} = \{1, 2\}$, agent i is clearly satisfied. Further, if $S_i^{\downarrow a_j} = \{1\}$, we have $p_{ij} = 2$. Therefore, under ρ no job other than i can be assigned to machine j , which, in turn,

means that under π no agent other than i is assigned to activity a_j . Thus, in this case i is satisfied as well. Since this holds for each $i \in N$, it follows that π is a perfect assignment.

Conversely, let π be a perfect assignment for (N, A, P) , and consider a mapping ρ given by $\rho(i) = j$ if and only if $\pi(i) = a_j$. Since under π each agent is assigned to a non-void activity, ρ assigns each job to some machine. Further, since π is individually rational, the completion time of each machine is finite, and each machine is assigned at most 2 jobs. Now, consider a machine j , $j \in \{1, \dots, m\}$. If $p_{ij} = 1$ for each job i with $\rho(i) = j$, the completion time of machine j is at most 2. On the other hand, if $p_{ij} = 2$ for some job i with $\rho(i) = j$, then $S_i^{\downarrow a_j} = \{1\}$, which means that under π no agent other than i can be assigned to activity a_j . This, in turn, means that i is the only job assigned to machine j under ρ , and hence in this case the completion time of machine j is at most 2 as well. Since this holds for every machine, the makespan of ρ is at most 2. \square

Theorem 3 *It is NP-complete to decide whether a-GASP admits a perfect assignment, even when all activities in A^* are simple, all agents have decreasing preferences, and, moreover, for every agent $i \in N$ it holds that $|\{a \mid S_i^{\downarrow a} \neq \emptyset\}| \leq 3$.*

Proof We reduce MONOTONE 3-SAT to our problem. Recall that an instance of MONOTONE 3-SAT is given by a set of variables $X = \{x_1, \dots, x_\ell\}$ and a set of three-literal clauses $\mathcal{C} = \{C_1, \dots, C_q\}$ over X , where each clause in \mathcal{C} contains positive literals only or negative literals only. It is a “yes”-instance if there exists a truth assignment to variables in X that satisfies all clauses and a “no”-instance otherwise. For a literal x , let $\text{oc}(x)$ be the number of occurrences of x in \mathcal{C} . Without loss of generality, we can assume that the clauses C_1, \dots, C_p contain positive literals only, while the clauses C_{p+1}, \dots, C_q contain negative literals only, and that $\text{oc}(x_i) > 0$ and $\text{oc}(\bar{x}_i) > 0$ for each $x_i \in X$.

We will now define an instance of restricted network flow problem \mathcal{F} (for an illustration, see Fig. 1). Later, we will show that it corresponds to an instance of a-GASP with simple activities and decreasing preferences where each agent approves of at most three activities.

- We introduce a source s and a sink t .
- For each variable x_i
 - we introduce a node v_i and an arc of unlimited capacity from s to v_i ;
 - we introduce two nodes x_i and \bar{x}_i , an arc labelled w_i from v_i to x_i , and an arc labelled \bar{w}_i from v_i to \bar{x}_i .
- For each $i = 1, \dots, \ell$, the capacity of the arc w_i is set to $\text{oc}(x_i)$ and the capacity of the arc \bar{w}_i is set to $\text{oc}(\bar{x}_i) \cdot (q\ell + 1)$.
- For each “positive” clause C_j , $1 \leq j \leq p$, we introduce a node with the same label.
- For each “negative” clause C_j , $p + 1 \leq j \leq q$, we introduce $q\ell + 1$ nodes with the respective labels $C_j^{(1)}, \dots, C_j^{(q\ell+1)}$.
- For each $i = 1, \dots, \ell$ and each $j = 1, \dots, p$ we introduce unit capacity arcs from x_i to C_j if x_i appears in C_j .

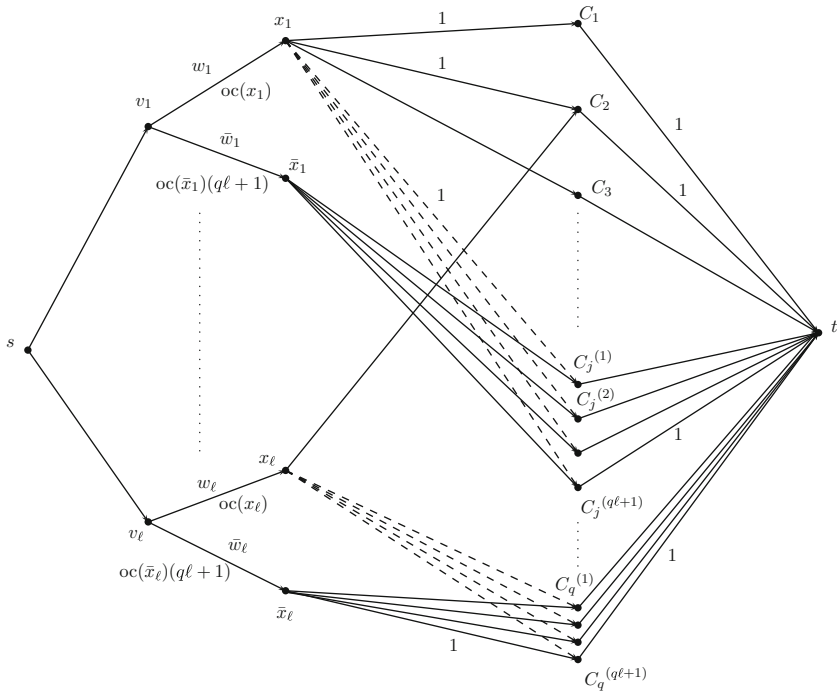


Fig. 1 Flow instance \mathcal{F}

- For each $i = 1, \dots, \ell$ and each $j = p + 1, \dots, q$ we introduce unit capacity arcs from x_i and \bar{x}_i to $C_j^{(1)}, \dots, C_j^{(q\ell+1)}$ if \bar{x}_i appears in C_j .
- Finally, we connect each $C_1, \dots, C_p, C_{p+1}^{(1)}, \dots, C_q^{(q\ell+1)}$ to t by a unit capacity arc.

We will now show that there is a satisfying truth assignment for X if and only if there is a flow of size $(q\ell + 1)(q - p) + p$ in \mathcal{F} that passes through at most one of $\{x_i, \bar{x}_i\}$ for each $i = 1, \dots, \ell$. The “if” direction is obvious, so let us consider the “only if” direction. Observe that a flow f of size $(q\ell + 1)(q - p) + p$ that uses at most one of $\{x_i, \bar{x}_i\}$ for each $i = 1, \dots, \ell$ is a maximum flow in our network. Therefore it has to send a unit of flow through each of the nodes $C_1, \dots, C_p, C_{p+1}^{(1)}, \dots, C_q^{(q\ell+1)}$. Now suppose that there is a $j \in \{p + 1, \dots, q\}$ such that all of the nodes $C_j^{(1)}, \dots, C_j^{(q\ell+1)}$ are reached via positive literals only. Then we must have $\sum_{i=1}^{\ell} oc(x_i) \geq q\ell + 1$, which is impossible. Thus, for each $j \in \{p + 1, \dots, q\}$, f sends flow through at least one of $C_j^{(1)}, \dots, C_j^{(q\ell+1)}$ via at least one negative literal \bar{x}_i . By construction, the capacity of \bar{w}_i is large enough that we can modify f so that through all of $C_j^{(1)}, \dots, C_j^{(q\ell+1)}$ the flow is sent via negative literals only. Recall that by construction, each “positive” clause C_1, \dots, C_p is reachable only via nodes that are contained in the clause. Thus, there exists a flow that passes through at most one of $\{x_i, \bar{x}_i\}$ for each $i = 1, \dots, \ell$, sends flow from positive literals only to positive clauses and from negative literals

only to (copies of) negative clauses, and passes through all clauses. Clearly, such a flow corresponds to a satisfying truth assignment.

Finally, we will now argue that the restricted flow problem described above can be interpreted as an instance of **a-GASP**. Indeed, we can view each node v_j as an activity, and each node $C_1, \dots, C_p, C_{p+1}^{(1)}, \dots, C_q^{(q\ell+1)}$ as an agent. Each agent approves of the three activities the agent is reachable from. In more detail, for each agent $i \in \{C_1, \dots, C_p\}$ we define his vote V_i so that the induced approval vote S_i is given by

$$S_i = \bigcup_{j:i \text{ is reachable from } v_j} \{v_j\} \times [1, \text{oc}(x_j)],$$

and for each agent $i \in \{C_{p+1}^{(1)}, \dots, C_q^{(q\ell+1)}\}$ we define his vote V_i so that the induced approval vote S_i is given by

$$S_i = \bigcup_{j:i \text{ is reachable from } v_j} \{v_j\} \times [1, \text{oc}(\bar{x}_j)(q\ell + 1)].$$

Note that we always have $\text{oc}(x_j) < \text{oc}(\bar{x}_j)(q\ell + 1)$. By construction, an agent is connected to x_j if she accepts up to $\text{oc}(x_j)$ participants in activity v_j ; if she accepts up to $\text{oc}(\bar{x}_j)(q\ell + 1)$ participants, she is connected to \bar{x}_j as well. Thus, a flow of size $(q\ell + 1)(q - p) + p$ corresponds to a perfect assignment. It remains to observe that our reduction can be performed in polynomial time □

Theorem 4 *It is NP-complete to decide whether a-GASP admits a perfect assignment, even when all activities in A^* are equivalent (i.e., A^* consists of a single ∞ -copyable activity a) and for every $i \in N$ we have $|S_i^{\downarrow a}| \leq 2$.*

Proof Note that this variant of **a-GASP** can be viewed as an anonymous hedonic game, and a perfect assignment corresponds to a partition of players into coalitions so that each agent approves of the size of his coalition. Therefore, when constructing an instance of **a-GASP** in our NP-hardness reduction, we will only describe the coalition sizes that each agent approves of, without mentioning the activity explicitly.

We reduce from a restricted version of X3C (see the proof of Theorem 1), where it is assumed that every element of the ground set X appears in at most three sets in \mathcal{Y} ; X3C is known to remain NP-hard under this restriction (Garey and Johnson 1979). Given an (X, \mathcal{Y}) instance of X3C, for each $x \in X$ let $t(x)$ denote the number of subsets in \mathcal{Y} that contain x ; note that $0 \leq t(x) \leq 3$. We pick $|X| + |\mathcal{Y}|$ pairwise distinct positive integers that are multiples of 4: for every subset $Y \in \mathcal{Y}$ we pick an integer $z(Y) \geq 4$, and for every element $x \in X$ we pick an integer $z(x) \geq 4$. We will now construct an instance of **a-GASP** as follows:

- for every element $x \in X$, we introduce a group G_x of $z(x) - t(x) + 1$ corresponding element-players and set $S_i^{\downarrow a} = \{z(x)\}$ for all $i \in G_x$;
- for every subset $Y \in \mathcal{Y}$, we introduce a group G_Y of $z(Y) - 3$ corresponding subset-players and set $S_i^{\downarrow a} = \{z(Y) - 3, z(Y)\}$ for all $i \in G_Y$;

- for every element $x \in X$ and every subset $Y \in \mathcal{Y}$ with $x \in Y$, we introduce a corresponding choice-player $P(x, Y)$ and set $S_{P(x, Y)}^{\downarrow a} = \{z(x), z(Y)\}$.

Observe that this reduction can be performed in polynomial time. We claim that (X, \mathcal{Y}) is a “yes”-instance of X3C if and only if the constructed instance of a-GASP admits a perfect assignment.

First, suppose that (X, \mathcal{Y}) is a “yes”-instance of X3C, which is certified by a collection of subsets $\mathcal{Y}' \subseteq \mathcal{Y}$.

- For every subset $Y \in \mathcal{Y}'$ with $Y = \{x, y, z\}$, we form a coalition consisting of the $z(Y) - 3$ subset-players in G_Y and of the three choice-players $P(x, Y)$, $P(y, Y)$, $P(z, Y)$. The coalition size is $z(Y)$, and all these players are happy.
- For every subset $Y \in \mathcal{Y} \setminus \mathcal{Y}'$, the players in G_Y form their own coalition of size $z(Y) - 3$; all these players are happy.
- For every element $x \in X$, the $z(x) - t(x) + 1$ element-players in G_x form a coalition together with the remaining $t(x) - 1$ choice-players $P(x, Y)$ with $Y \in \mathcal{Y} \setminus \mathcal{Y}'$. The coalition size is $z(x)$, and all these players are happy.

Conversely, suppose that our instance of a-GASP admits a perfect assignment π . Consider an element $x \in X$, and consider the corresponding $z(x) - t(x) + 1$ element-players in G_x . All these players must be in a coalition of size $z(x)$, and for this they need exactly $t(x) - 1$ choice-players $P(x, Y)$ with $x \in Y$ to join. Hence exactly one of the choice-players $P(x, Y)$ with $x \in Y$ remains unassigned, and must look for another coalition. The only option for him is a coalition of size $z(Y)$; hence he must be together with the $z(Y) - 3$ subset-players in G_Y and with the two other choice-players $P(y, Y)$ and $P(z, Y)$ for this subset Y . Then we say that the corresponding subset Y is *selected* by its three choice-players. It is easy to verify that there are exactly $|X|/3$ subsets selected, and that they form an exact cover. \square

References

- Anshelevich E, Sekar S (2014) Approximate equilibrium and incentivizing social coordination. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI'14), pp 508–514
- Aziz H, Brandt F, Harrenstein P (2013) Pareto optimality in coalition formation. *Games Econ Behav* 82:562–581
- Aziz H, Harrenstein P, Lang J, Wooldridge M (2016) Boolean hedonic games. In: Proceedings of the 15th international conference on principles of knowledge representation and reasoning (KR'16), pp 166–175
- Ballester C (2004) NP-completeness in hedonic games. *Games Econ Behav* 1(49):1–30
- Banerjee S, Konishi H, Sönmez T (2001) Core in a simple coalition formation game. *Soc Choice Welf* 18:135–153
- Bogomolnaia A, Jackson MO (2002) The stability of hedonic coalition structures. *Games Econ Behav* 38:201–230
- Darmann A, Lang J (2017) Group activity selection problems. In: Endriss U (ed) Trends in computational social choice, AI Access, pp 385–410
- Darmann A (2015) Group activity selection from ordinal preferences. In: Proceedings of the 4th international conference on algorithmic decision theory (ADT'15), pp 35–51
- Garey MR, Johnson DS (1979) *Computers and intractability*. W. H. Freeman and Company, New York
- Gupta S, Roy S, Saurabh S, Zehavi M (2017) Group activity selection on graphs: parameterized analysis. In: Proceedings of the 10th international symposium on algorithmic game theory (SAGT'17), pp 106–118

- Igarashi A, Bredereck R, Elkind E (2017) On parameterized complexity of group activity selection problems on social networks. In: Proceedings of the 16th conference on autonomous agents and multiagent systems (AAMAS'17), pp 1575–1577
- Igarashi A, Peters D, Elkind E (2017) Group activity selection on social networks. In: Proceedings of the 31st AAAI conference on artificial intelligence (AAAI'17), pp 565–571
- Jain K, Mahdian M (2007) Cost sharing. In: Nisan N, Roughgarden T, Tardos E, Vazirani VV (eds) Algorithmic game theory. Cambridge University Press, Cambridge, pp 385–410
- Jiang H, Zhong F, Zhu B (2011) Filling scaffolds with gene repetitions: maximizing the number of adjacencies. In: Proceedings of the 22nd annual symposium on combinatorial pattern matching (CPM'11), pp 55–64
- Lee H, Shoham Y (2015) Stable invitations. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15), pages 965–971
- Lee H, Vassilevska Williams V (2017) Parameterized complexity of group activity selection. In Proceedings of the 16th Conference on Autonomous Agents and Multiagent Systems (AAMAS'17), pages 353–361
- Lenstra J, Shmoys D, Tardos É (1990) Approximation algorithms for scheduling unrelated parallel machines. *Math Program* 46:259–271
- Lu T, Boutilier C (2012) Matching models for preference-sensitive group purchasing. In: Proceedings of the 13th ACM conference on electronic commerce (ACM EC'12), pp 723–740
- Peters D (2016) Complexity of hedonic games with dichotomous preferences. In: Proceedings of the 30th AAAI conference on artificial intelligence (AAAI'16), pp 579–585
- Skowron P, Faliszewski P, Slinko A (2015) Achieving fully proportional representation: approximability results. *Artif Intell* 222:67–103
- Spradling M, Goldsmith J, Liu X, Dadi C, Li Z (2013) Roles and teams hedonic game. In: Proceedings of the 3rd international conference on algorithmic decision theory (ADT'13), pp 351–362
- Vöcking B (2006) Congestion games: Optimization in competition. In: Proceedings of the 2nd Durham workshop on algorithms and complexity (ACID'06), pp 9–20