Parameterized Complexity of Group Activity Selection

Hooyeon Lee
Moloco, Inc.
haden.lee@molocoads.com

Virginia Vassilevska Williams
Massachusetts Institute of Technology
virgi@mit.edu

ABSTRACT

We consider the Group Activity Selection Problem (GASP) in which a group of agents need to be assigned to activities, subject to agent preferences and stability conditions. In GASP, the agents announce dichotomic preferences on which (activity, number-of-participant) pairs are acceptable to them. We consider five solution concepts of assignments: (1) individual rationality (everyone who is assigned to an activity is willing to participate), (2) (Nash) stability (no agent wants to deviate from the assignment), (3) envy-freeness (no agent is envious of someone else's assignment), (4) stability and envy-freeness, and (5) perfection (everyone is assigned and willing to participate). It is known that finding an assignment of a given size with any of these properties is NP-complete. We study the complexity of GASPon a finer scale, through the lens of parameterized complexity. We show that the solution concepts above differ substantially, when parameterized by the size of the solution (the number of assigned agents or the number of used activities). In particular, finding an individually rational assignment is fixed parameter tractable, yet other solutions concepts are less tractable (W[1]- and W[2]-hard) even under very natural restrictions on inputs.

Keywords

Parameterized Complexity; Fixed parameter tractability; W-hierarchy; Individual rationality; Stability; Envy-free

1. INTRODUCTION AND RELATED WORK

Imagine an event in which several activities are to take place concurrently. A group of agents are willing to participate, subject to their preferences. In many settings, the agent preferences include not only which activities the agent is willing to participate in, but also the number(s) of participants in each activity that are acceptable to the agent. For example, agents may wish to have enough participants in certain activities (such as a group bus tour) to split the cost associated with it, whereas they may wish to have just few participants in activities with limited resources (such as a showcase with a limited number of devices). Given the preferences of agents, the organizer wishes to find a "good" assignment subject to certain rationality and/or stability conditions. The first condition is *individual rationality*: everyone assigned to some activity is willing to participate. In addition to individual rationality, the organizer may want to ensure that agents who are not assigned to any activity do not prefer to deviate from their assignment by joining an activity (*Nash stability*). Other concepts include *envy-freeness* that asserts that no unassigned agent would prefer to take the place of an assigned agent, and *perfection* in which all agents must be assigned.

Appears in: *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

To model this setting, Darmann et al. [6] proposed the Group Activity Selection Problem (GASP) and defined three solution concepts (individual rationality, stability, and perfection). In addition, they also provided many NP-hardness results even under various restrictions on inputs. These hardness results essentially argue that it is hard to find an assignment that maximizes the number of participants in activities. Suppose, however, that we are satisfied if we can assign a small number k out of the n participants to some (up to k) of the p activities while satisfying our rationality criteria. For instance, consider a VR company that is hosting a hackathon event in which n developers wish to participate and develop prototype software for the company's latest VR devices. The company can supply up to p VR devices of different models, while the venue has space for k developers in total (due to safety reasons and limited number of workstations). As developers may have different preferences on which model of VR devices they want to use and with how many others they are willing to share the workstation, this is a natural instance of GASP in which small-size solutions matter.

In general, we can find small-size solutions in a brute-force manner: try all possible $O(p^k)$ ordered choices of k activities, all $O(k^k)$ choices for the number of participants in each activity, and all $O(n^k)$ ordered choices of k participants; then check whether the desired criterion (such as individual rationality or stability) is satisfied by the induced assignment. This runs in $O((pnk)^k)$ time,[1] which is polynomial for any fixed k but is not very desirable. A much better running time would be one of the form $O(f(k)\cdot(p+n))$ - such a runtime would be *linear*, regardless of the constant k, and the function f. More generally, on input size n, one would like a fixed parameter tractable (FPT) algorithm with runtime $f(k)\cdot n^c$, where c is independent of k. The problems that admit such algorithms are said to be in the class FPT. Developing FPT algorithms, especially linear time ones, greatly mitigates the NP-hardness of problems as it shows that the problems are actually quite tractable.

The field of parameterized complexity aims to classify NP-hard problems by determining their membership within a hierarchy of complexity classes under parameterization, called W-hierarchy. Because complexity is analyzed in terms of both the input size and an additional parmeter, it analyzes NP-hard problems on a finer scale than classic complexity. The W-hierarchy contains classes such as FPT, W[1], W[2], etc. Hierarchy theorems show that FPT is contained in W[1] which is contained in W[2], and so on (see [7] for more details). It is believed that FPT $\neq$ W[1] $\neq$ W[2], so that the problems in W[2] are believed to be harder than those in W[1] that are themselves believed to be harder than the FPT problems. Lastly,

[1] This shows that the problems we consider are in the class XP.

hardness assumptions such as the Exponential Time Hypothesis (ETH) of Impagliazzo and Paturi [11] can often be used to show that particular W[1]-hard problems cannot be solved in $n^{o(k)}$ time, giving concrete runtime lower bounds.

In this work, we investigate tractability of GASP under different solution concepts and different restrictions on inputs when the size of the solution is parameterized. We place different NP-hard versions of GASP under this parameterization into different parts of the W-hierarchy. Our classification is nearly complete, as seen in Table 1, Section 3. We show that GASP for individual rationality is in FPT, whereas for the other solution concepts the problem is W[1]-hard or W[2]-hard even if all agents have increasing (decreasing) preferences, i.e., wanting more (fewer) participants for each activity. Surprisingly to us, the case of decreasing preferences is more tractable than that of increasing preferences. Lastly, we consider the special case in which all activities are equivalent. Here the preferences of the agents can vary but for each particular agent the preferences are the same for all copies of the activity. Even though the problem is NP-hard, all parameterized versions of it (except possibly perfection) are FPT.[2]

Related Work. Computational social choice is an interdisciplinary research area involving economics, social science, and computer science including artificial intelligence and multi-agent systems. Much work has been devoted to investigating both classical and parameterized complexity of social choice problems that range from winner determination [16, 14, 3], control problems in voting rule [9, 10, 8], coalition games [15, 4], and more. This work studies parameterized complexity of a social choice problem under five different solution concepts and restrictions on inputs.

Most closely related work to ours is that of Darmann et al. [6], in which the authors defined GASP, and provided a number of classical complexity results for individual rationality, stability, and perfection. In this work, we adopt their definitions, but we also consider the new solution concept of envy-freeness. It is worth noting that GASP is closely related to Hedonic Games (see Section 2.2 of [6] and Section 2 of [12] for more details); in fact, GASP can be viewed as a class of hedonic coalition games with concise representation of preferences of agents. Ballester [2] provides a number of computational complexity results (in fact, hardness results) for finding a core-stable, Nash-stable, or individually rational outcome in hedonic games and anonymous hedonic games, but these results do not apply to GASP because of the concise representation of an input to GASP. Recently, Darmann [5] considered a different setting of GASP where agents are assumed to have strict, ordinal preferences over the outcomes, whereas both our work and [6] assume that agents are indifferent among all outcomes that they approve of. It is an interesting future problem to consider how our results in this work can be extended to the ordinal setting that Darmann [5] considered. Lee and Williams [13] recently analyzed parameterized comlexity of the Stable Invitations Problem (first studied by Lee and Shoham [12]) that generalizes GASP with social relationships (friends and enemies) while assuming only one activity. Lastly, Lee and Shoham [12] showed that incentive compatibility and solution concepts being considered in this work are not compatible in GASP, yet we do not consider strategic agents in this work.

2. DEFINITIONS AND KNOWN RESULTS

To make this work self-contained, we begin by introducing the formal definitions proposed by Darmann et al. [6], with minor modifications to notation for readability and consistency.

[2]We note that Darmann et al. [6] defined increasing/decreasing preferences as well as equivalent activities in their work.

Definition 1. In the Group Activity Selection Problem (GASP), we are given a set of agents $N = \{1,2,\dots,n\}$, a set of non-void activities $A^* = \{a_1,a_2,\dots,a_p\}$, and the *void activity* ($a_\emptyset$) which refers to the case when an agent does not participate in any of the activities in A^*. An *outcome* is a pair $(a_j,x) \in A^* \times [n]$ which is interpreted as x agents participating in non-void activity a_j. For each agent i we are given a set S_i of outcomes (called *approval set*) such that the outcomes in S_i are equally liked and strictly preferred to $a_\emptyset$, where $S_i \subseteq A^* \times [1,n]$. We write $S_i(a_j) = \{x : (a_j,x) \in S_i\}$ to refer to the set of sizes which agent i approves for activity a_j.

Similarly to Darmann et al. [6], we assume that each agent is indifferent among the outcomes in S_i; that is, the void-activity ($a_\emptyset$) draws the line between which outcomes are approved and which ones are not by the agent. While this is a simplifying assumption, note that hardness results immediately imply the same hardness for the general case without this assumption.

Example 1. Consider $N = \{1,2,3\}$ and $A^* = \{a_1,a_2\}$, and the six outcomes, $A^* \times [3]$. Suppose $S_1 = \{(a_1,1),(a_1,2),(a_1,3)\}$, $S_2 = \{(a_1,2),(a_2,2),(a_2,3)\}$, and $S_3 = \{(a_1,1),(a_2,1),(a_2,2)\}$. That is, agent 1 approves a_1 for any size (i.e., unconditional approval) while she does not approve a_2 for any size (i.e., unconditional refusal). Using our notation, $S_1(a_1) = \{1,2,3\}$ and $S_1(a_2) = \emptyset$. If we assign all agents to a_1, then $(a_1,3)$ is the outcome realized by all agents – notice that only agent 1 approves it (and thus is willing to participate) while agents 2 and 3 do not (and thus are unwilling to participate). Naturally, this assignment induces instability.

Definition 2. An assignment in GASP is a mapping $\pi : N \to A^* \cup \{a_\emptyset\}$ where $\pi(i) = a_\emptyset$ means that agent i is not assigned to any non-void activity. An assignment naturally partitions the agents into at most $|A^*|+1$ groups. We define $\pi^0 = \{i : \pi(i) = a_\emptyset\}$ and $\pi^j = \{i : \pi(i) = a_j\}$ for $j = 1,\dots p$, so that $|\pi^j|$ refers to the number of agents assigned by π to a specific activity. Let us define the size of an assignment, denoted by $|\pi|$, as the number of agents that are assigned to non-void activities; that is, $|\pi| = \sum_{j=1}^{p} |\pi^j|$.

π induces an outcome for each agent: If $\pi(i) = a_\emptyset$, then the special outcome $a_\emptyset$ is induced as i does not participate, and if $\pi(i) = a_j \in A^*$, then $(a_j, |\pi^j|)$ is the induced outcome for agent i.

We define solution concepts with different levels of stability.

Definition 3. Let π be any assignment in GASP.

π is *individually rational* (IR) if $\forall j \in [p]$ and $\forall i \in \pi^j$, it holds that $(a_j, |\pi^j|) \in S_i$.

π is *(Nash) stable* if it is IR, and $\forall i \in N$ such that $\pi(i) = a_\emptyset$ and $\forall a_j \in A^*$ it holds that $(a_j, |\pi^j|+1) \notin S_i$.

π is *envy-free* (EF) if it is IR, and $\forall i \in N$ such that $\pi(i) = a_\emptyset$ and $\forall i' \in N$ such that $\pi(i') = a_j \in A^*$, it holds that $(\pi(i'), |\pi^j|) \notin S_i$.

π is *stable-EF* if it is both stable and envy-free.

π is *perfect* if it is IR and $\pi(i) \neq a_\emptyset$ for all $i \in N$.

IR requires every agent assigned to an activity be unwilling to deviate. Stability further requires that every unassigned agent be unwilling to deviate (unilaterally, without permission of other agents). EF requires that every unassigned agent be not envious of someone else assigned to an activity. Stability and EF together define a stronger solution concept than the two. Lastly, a perfect assignment is the strongest solution concept which implies all others.

Darmann et al. [6] showed that finding a solution in GASP is NP-hard even under some restrictions on inputs – when agents have restricted domains of preferences and when activities are of the same type (i.e., all activities are equivalent).

Definition 4. (Darmann et al. [6]) Agent i has an *increasing* preference for activity a_j if there exists a threshold $l_i(a_j) \in \{1,2,\dots,n+1\}$ such that $S_i(a_j) = [l_i(a_j), n]$ (where $[n+1,n] = \emptyset$). Similarly, Agent i has a *decreasing* preference for activity a_j if there exists a threshold $u_i(a_j) \in \{0, 1, \dots, n\}$ such that $S_i(a_j) = [1, u_i(a_j)]$. Lastly, we say that two activities a_j and $a_{j'}$ are equivalent if for every agent $i \in N$, $S_i(a_j) = S_i(a_{j'})$.

Let us re-visit the problem instance from Example 1, and relate it to various definitions and concepts we have defined in this section.

Example 2. In Example 1, agent 2 has an increasing preference for a_2 with $l_2(a_2) = 2$ while agent 3 has a decreasing preference for a_2 with $u_3(a_2) = 2$. Agent 1 has (degenerate) increasing/decreasing preferences for both a_1 and a_2 with $l_1(a_1) = 1, u_1(a_2) = 3$ and $l_1(a_2) = 4, u_1(a_2) = 0$. Now consider an assignment π with $\pi(1) = \pi(2) = a_1$ and $\pi(3) = a_2$; under the assignment π, agents 1,2 realize the outcome $(a_1, 2)$ and agent 3 realizes $(a_2, 1)$. It is easy to check that π is prefect (and thus IR, stable, and EF). Consider another assignment π' with $\pi'(1) = a_1$ and $\pi'(2) = \pi'(3) = a_\emptyset$; under π', agent 1 realizes the outcome $(a_1, 1)$ and agents 2,3 realize $a_\emptyset$. π' is individually rational (as $(a_1, 1) \in S_1$), but it is not stable (as $(a_1, 2) \in S_2$) or envy-free (as $(a_1, 1) \in S_3$).

Darmann et al. [6] proved many hardness results of `GASP`, and we mention their most relevant results to this work.

Theorem 1. *Finding a perfect assignment is NP-hard, even if all agents have increasing preferences for all activities, if all agents have decreasing preferences for all activities, or if all activities are equivalent (Theorems 4.1, 4.2, 4.3, and 4.4 of [6]).*

As corollaries, finding an assignment of size k under IR, stability, EF, or stable-EF is NP-hard; we refer to these problems as k-IR-GASP, k-Stable-GASP, k-EF-GASP, and k-Stable-EF-GASP. Also as a corollary, finding a perfect assignment using k activities is NP-hard, to which we refer as k-Perfect-GASP. At the end of the paper we also consider the related problems of finding an IR/Stable/EF/Stable-EF solution of size *at least* k, and also a Perfect-GASP solution using *at most* k activities.

3. PARAMETERIZED COMPLEXITY

Our main technical results are summarized in Table 1. "General" refers to the case of arbitrary preferences of agents, "Increasing" ("Decreasing") refers to the case where all agents have increasing (decreasing) preferences for all activities, and "Equivalent" refers to the case where all activities are equivalent. All problems are known to be NP-complete due to Darmann et al. [6].

Input	k-GASP Solution Concepts				
	IR	Stable	EF	Stable & EF	Perfect
General	FPT	$W[1]$-hard	$W[1]$-C	$W[1]$-hard	$W[2]$-hard
Increasing	FPT	$W[1]$-C	$W[1]$-C	$W[1]$-hard	$W[2]$-hard
Decreasing	FPT	FPT	$W[1]$-C	FPT	$W[2]$-hard
Equivalent	FPT	FPT	FPT	FPT	Unknown

Table 1: Complexity of the Group Activity Selection Problem.

3.1 Complexity of IR-GASP

We show that k-IR-GASP can be solved in $(\exp(k) np \log n)$ time. The largest input size to `GASP` is $\Theta(n^2 p)$ as the number of possible outcomes is np and each agent needs to specify an approval set of size $O(np)$. As we seek solutions of size k, only the outcomes with the number of participants being at most k matter. We can prune the input to size $\Theta(nkp)$ (in about that much time, assuming random access to preferences). Hence our algorithm runs in sublinearithmic time in input size.

Theorem 2. *k-IR-GASP can be solved in time $2^{O(k)}(np \log n)$ where $n = |N|$ and $p = |A^*|$ (hence it is in FPT).*

Proof. We use "Color Coding" to design a randomized (Monte Carlo) algorithm, which can easily be de-randomized using a family of k-perfect hash functions as shown in the work [1].

We first "color" the agents using k colors independently and uniformly at random. We seek to assign exactly one agent of each color to some activity such that the resulting assignment is IR and of size k. Let $c(i)$ denote the color of agent i where $c(i) \in [k]$. For each activity $a_j \in A^*$ and every subset C of colors (i.e., $C \subseteq [k]$), we will first determine whether it is possible to assign to activity a_j exactly $|C|$ agents with distinct colors specified by C while satisfying the IR constraint; we refer to this subproblem by $T(C, j)$. For any fixed a_j and C, we check for every color $d \in C$ whether there exists an agent i with $c(i) = d$ and $(a_j, |C|) \in S_i$ in time $O(n)$ by iterating over agents and look up their approval sets. If the test is affirmative, we can assign exactly $|C|$ agents with distinct colors specified by C to activity a_j. We solve $T(C, j)$ for every $a_j \in A^*$ and every subset of colors, which can be done in time $O(n \cdot p \cdot 2^k)$ overall.

Next, we solve another set of subproblems (which we call $R(C, j)$) to check if it is possible to assign $|C|$ agents of distinct colors in C to activities in $A_j = \{a_1, a_2, \dots, a_j\}$ for every $j \le p$ and $C \subseteq [k]$. When $j = 1$, $R(C, j)$ is equivalent to $T(R, j)$. When $j > 1$, we enumerate over every subset $C' \subseteq C$, and solve $R(C', j-1)$ and lookup the result of $T(C \setminus C', j)$. If both $R(C', j-1)$ and $T(C \setminus C', j)$ are affirmative for some $C' \subset C$, then we conclude that $R(C, j)$ is also affirmative. Finally, if $R([k], p)$ is affirmative, then we can find an IR assignment of size k with distinct colors of agents. There are at most $O(2^k \cdot p)$ subproblems $R(C, j)$'s, and each subproblem can be solved in time $O(2^k)$ (as we enumerate over all subsets of C).

The overall runtime of this algorithm is $O(4^k \cdot p + 2^k \cdot (np)) = 2^{O(k)}(np)$. This algorithm is a Monte Carlo algorithm; even if there exists a solution, there is a chance that the algorithm does not find it due to coloring. The probability that random coloring yields distinct colors of the k agents of any fixed solution is at least $k!/k^k > 1/e^k$, which is exponentially small only in k. We can repeat this algorithm $e^k \ln n$ times to increase the probability of success to $1 - 1/n$ (with overall runtime $2^{O(k)}(np \log n)$). To de-randomize the algorithm, we use a k-perfect family of hash functions from N to $[k]$. If we have a list of colorings of agents such that for every subset $N' \subseteq N$ of size k there exists a coloring in the list that colors each agent in N' distinctly, then we can enumerate over this list of colorings in lieu of random coloring. This is precisely what a k-perfect family of hash functions offers, and the list can be specified using $2^{O(k)} \log n$ bits (for details, see [1]). This leads to a deterministic FPT algorithm with $2^{O(k)}(np \log n)$ runtime. □

3.2 Complexity of Stable-GASP

Stability is a stronger solution concept than individual rationality. This relationship is not apparent under classic complexity as both problems are NP-complete. However, under parameterization, k-IR-GASP is FPT whereas k-Stable-GASP is $W[1]$-hard.

Theorem 3. *k-Stable-GASP is $W[1]$-hard, even if each agent approves at most one size per activity. Assuming the ETH [11], k-Stable-GASP cannot be solved in time $(np)^{o(\sqrt{k})}$.*

Proof. We reduce from k-Clique[3]. The result based on ETH follows as we increase the parameter from k to $O(k^2)$.

Construction of `GASP` *instance.* Consider an instance of the k-Clique problem, $G = (V, E)$ and a parameter k where $V = \{v_1, \dots, v_n\}$. Let us create an instance of `GASP` as follows: Let $N = V \cup \{w_{i,x} : (1 \le i \le n) \wedge (1 \le x \le k-1)\}$; that is, we create n node-agents v_i's (by abusing notation) and $(k-1)$ copies of neighbor-agents ($w_{i,x}$'s) for each v_i. The neighbor-agents will be used to "select" the $k-1$ edges incident to each node if the node is to be included in a clique

[3] k-Clique is to find a clique of size k in given graph, known to be W[1]-hard.

we are seeking. Let $A^* = \{a_1, \dots, a_k\} \cup \{e_{i,j} : 1 \le i < j \le n\}$; we create k clique-activities (which are used to determine membership of a node in a clique) and $\binom{n}{2}$ edge-activities $e_{i,j}$ (where $i < j$). For each node-agent v_i, we set its approval set $S_{v_i} = \{(a_j, 1) : 1 \le j \le k\} \cup \{(e_{i,j}, 3) : i \ne j\}$. For each neighbor-agent $w_{i,x}$, we set its approval set $S_{w_{i,x}} = \{(e_{i,j}, 2) : (v_i, v_j) \in E\}$. Finally we set the parameter k' of GASP (to distinguish from k in the Clique problem) to $k' = k + 2\binom{k}{2}$. This is a valid FPT-reduction as k' depends only on k but not on n, and the size of our instance of GASP is polynomially bounded in n, k as there are $O(nk)$ agents and $O(n^2)$ activities.

Let us describe how cliques and stable assignments are related in this reduction. A node-agent is assigned to a clique-activity if and only if its corresponding node belongs to a (corresponding) clique. For each node-agent, there exists $k-1$ neighbor-agents, and these neighbor-agents must be assigned properly to edge-activities in order to ensure that the resulting set of nodes is indeed a clique.

Proof of equivalence between instances. We claim that a clique of size k exists if and only if a stable assignment of size k' exists. Without loss of generality, suppose that $C = \{v_1, v_2, \dots, v_k\}$ forms a clique in G. Consider the following assignment π:

$$\pi(v_i) = \begin{cases} a_i & i \le k \\ a_\emptyset & i > k \end{cases} \quad \text{and} \quad \pi(w_{i,x}) = \begin{cases} e_{i,x+1} & i \le k \land i \le x \\ e_{x,i} & i \le k \land i > x \\ a_\emptyset & i > k \end{cases}.$$

That is, node-agents are assigned to clique-activities and their associated neighbor-agents are assigned to edge-activities; all other agents are assigned to the void activity. Clearly π assigns exactly $k + 2\binom{k}{2} = k'$ agents to non-void activities. It is easy to verify that π is indeed a stable assignment (we omit details due to space limit).

Conversely, suppose there is a stable assignment π of size $k' = k + 2\binom{k}{2}$. First notice that for each edge-activity $e_{i,j}$ there are precisely two agents who approve the outcome $(e_{i,j}, 3)$ – namely, v_i and v_j. Therefore if π is stable, it cannot assign any node-agents (of the form v_i) to any edge-activity (of the form $e_{i,j}$). In other words, for each v_i, $\pi(v_i) \in \{a_\emptyset\} \cup \{a_1, \dots, a_k\}$. Let $C = \{v_i : \pi(v_i) \ne a_\emptyset\}$; since there are k clique-activities, $|C| \le k$. We claim that $|C| = k$ if π is stable; if $|C| < k$, then there exists some a_j such that no agent is assigned to it; since $k \le n$, there must be some v_i such that $\pi(v_i) = a_\emptyset$. This implies that π is not stable because $(a_j, 1) \in S_{v_i}$ while $\pi(v_i) = a_\emptyset$; hence $|C| = k$ must hold. By re-labeling, assume $C = \{v_1, v_2, \dots, v_k\}$ (i.e., $\pi(v_i) = a_i$ if $i \le k$ and $\pi(v_i) = a_\emptyset$ if $i > k$).

We argued earlier that π never assigns node-agents to any edge-activities if it is stable. This implies that, if π assigns any agent to an edge-activity, it must be the case that π assigns exactly two neighbor-agents (of the form $w_{i,x}$) to it (due to the construction of $S_{w_{i,x}}$'s). If π is stable, then π must assign no neighbor-agents to $e_{i,j}$ if $i > k$ or $j > k$ and exactly two neighbor-agents to $e_{i,j}$ if $i \le k$ and $j \le k$. To prove the first claim, suppose that π assigns two neighbor-agents to $e_{i,j}$ where $i > k$ (and recall that $\pi(v_i) = a_\emptyset$ when $i > k$). Then π is not stable because $(e_{i,j}, 3) \in S_{v_i}$, and thus v_i wishes to join $e_{i,j}$, and this is a contradiction. Similarly one can prove the claim in the case where $j > k$. To prove the second part, recall that $|\pi| = k' = k + 2\binom{k}{2}$. Since π assigns exactly k node-agents to non-void activities, it must assign $k(k-1) = 2\binom{k}{2}$ neighbor-agents to $\binom{k}{2}$ edge-activities from $\{e_{i,j} : i, j \le k\}$. Therefore, π must assign two agents to each of the edge-activities in $\{e_{i,j} : i, j \le k\}$. This implies that there is an edge between v_i and v_j in the original instance if $i, j \le k$. Otherwise, if $(v_i, v_j) \notin E$ where $i, j \le k$, then there is no neighbor-agents who can be assigned to $e_{i,j}$, which contradicts the assumption that π is of size k'. □

Let us consider the restricted case when all agents have increasing preferences for all activities. That is, all agents prefer no less participants for all activities that they approve.

Theorem 4. *k-Stable-GASP is $W[1]$-complete when all agents have increasing preferences for all activities.*

Proof. We reduce from the k-Clique problem to show W[1]-hardness. We omit proof of completeness due to space.

Construction of GASP instance. Let $G = (V, E)$ be a graph instance of the k-Clique problem. For each vertex $v_i \in V$, we create k^2 copies of v_i as agents (call them copies of v_i) and create an activity a_i; this creates $k^2|V|$ agents and $|V|$ activities. For each edge $e_{i,j} = (v_i, v_j) \in E$, we create two copies of $e_{i,j}$ as agents (call them copies of $e_{i,j}$) and create an activity $w_{i,j}$; this creates $2|E|$ agents and $|E|$ activities. Let $k' = k^3 + k^2 - k$, and we create $k' + 1$ copies of dummy agents (call them copies of z). For each of the k^2 copies of v_i agents, we set its approval set such that $l_{v_i}(a_i) = k^2$ (i.e., approves any outcome with a_i and size k^2 or larger) and $l_{v_i}(w_{i,j}) = 3$ if $(v_i, v_j) \in E$ and $l_{v_i}(\cdot) = n + 1$ (effectively, $+\infty$) for all other activities (where $n = k^2|V| + 2|E| + k' + 1$ is the total number of agents we create). For each of the two copies of $e_{i,j}$ agents, we set its approval set such that $l_{e_{i,j}}(w_{i,j}) = 2$. For each of the $k' + 1$ copies of z agents, we set its approval set such that $l_z(w_{i,j}) = 4$ for all (i, j) where $(v_i, v_j) \in E$. We claim that a clique of size k exists in G if and only if a stable assignment of size k' exists in our GASP instance.

Proof of equivalence between instances. Without loss of generality, suppose that $C = \{v_1, v_2, \dots, v_k\}$ is a clique of size k in G. We can construct a stable assignment of size k' as follows: (a) For k^2 copies of v_i, we assign them to a_i if $v_i \in C$ and to $a_\emptyset$ otherwise, (b) for two copies of $e_{i,j}$, we assign them to $w_{i,j}$ if $v_i \in C$ and $v_j \in C$ and to $a_\emptyset$ otherwise, and (c) copies of z are assigned to $a_\emptyset$. Note that this assignment assigns exactly $k^3 + 2\binom{k}{2} = k^3 + k(k-1) = k'$ agents to non-void activities. It is easy to verify that π is IR and stable, proof of which is omitted due to space.

Conversely, now suppose that π is a stable assignment of size k', and we show that there exists a clique of size k in G. If π assigns three or more agents to any $w_{i,j}$, then π must assign all copies of z to some activity (possibly $w_{i,j}$) or π would not be stable; yet we know that π is of size k' and there are $k' + 1$ copies of z, and therefore π can only assign two or fewer agents to each $w_{i,j}$. If π assigns two agents to some $w_{i,j}$, then those two agents must be the two copies of $e_{i,j}$ because no other agent approves the outcome $(w_{i,j}, 2)$. Furthermore, if π assigns the two copies of $e_{i,j}$ to $w_{i,j}$, then π must assign all k^2 copies of v_i to a_i and all k^2 copies of v_j to a_j – otherwise, π would not be stable. Let W be the set of activities of the form $w_{i,j}$ such that π assigns exactly two agents to $w_{i,j}$; if $|W| > \binom{k}{2}$, then there must be at least $k + 1$ indices that appear in elements of W, which implies that π must assign agents to at least $k + 1$ activities of the form a_i. This is a contradiction because π is of size k' but $(k+1)k^2 > k'$. Therefore, $|W| \le \binom{k}{2}$. Now suppose $|W| < \binom{k}{2}$ instead. As argued earlier, π can assign to at most k activities of the form a_i, but $k^3 + 2|W| < k'$, which implies that π cannot be of size k' if $|W| < \binom{k}{2}$. Lastly, suppose $|W| = \binom{k}{2}$ (and from previous arguments, it is clear that the number of the indices that appear in the elements of W must be exactly k); without loss of generality, assume $W = \{w_{i,j} : 1 \le i < j \le k\}$ (by re-labeling) – this implies that π assigns k^2 copies of v_l to a_l if $1 \le l \le k$, but more importantly, it implies that $(v_i, v_j) \in E$ because we create $w_{i,j}$ if and only if there is an edge between v_i and v_j. That is, $C = \{v_1, v_2, \dots, v_k\}$ is a clique in G. This shows W[1]-hardness of the problem. □

Unlike the case of increasing preferences, if all agents have decreasing preferences the problem admits an FPT algorithm.

Theorem 5. *k-Stable-GASP is in FPT when all agents have decreasing preferences for all activities.*

Proof. We use Color Coding to reduce this problem to a variant of the Vertex Cover problem. With probability which is exponentially small only in k, we color agents and activities "properly", and given a proper coloring we can find a stable assignment of size k in polynomial time in n, p yet exponential only in k.

Preliminaries. Suppose that a stable assignment of size k exists, and without loss of generality we know that it assigns k agents to l distinct activities (where $l \in [1,k]$), which can be done by checking every value in $[1,k]$. We first color agents and activities using l colors 1 through l, uniformly and independently at random (let $c(i)$ denote the color of agent i and $c(a_j)$ the color of activity a_j), and then fix the value of k_d for each $d \in [1,l]$ such that $\sum_{d\in[1,l]} k_d = k$. We say that the coloring c (together with l and k_d's) is compatible with a stable assignment π of size k (using l activities) if π assigns exactly k_d agents to an activity of color d for every $d \in [1,l]$. Given some coloring c, our algorithm either finds a stable assignment compatible with c or determines that no such solution exists. Clearly, any stable assignment (of size k) has at least one compatible coloring. With probability at least $(1/l)^{l+k}$, our randomized coloring is a compatible coloring of some stable assignment of size k (if it exists); it can be de-randomized using a family of k-perfect hash functions as shown in the work [1].

FPT Algorithm. We now proceed with fixed values of l and k_d's as well as some coloring c as described earlier. We will use the special color $l+1$ to mark the agents and activities that cannot be assigned/used in any stable assignment that is compatible with the given coloring c. Define $N_d = \{i \in N : c(i) = d\}$ and $A^*_d = \{a_j \in A^* : c(a_j) = d\}$ where $d \in [1, l+1]$; these subsets naturally partition N and A^* into $l+1$ subsets by their colors (at first N_{l+1} and A^*_{l+1} are empty, but we may re-color some agents and activities during the course of the algorithm). Let $N(a_j) = \{i \in N_{c(a_j)} : u_i(a_j) \geq k_{c(a_j)}\}$, which is the set of agents who have the same color as a_j and approve the size $k_{c(a_j)}$ for activity a_j (recall that agents have decreasing preferences, so we only need to check their upper-bound $u_i(a_j)$ for a given activity a_j). If $|N(a_j)| > k_{c(a_j)}$, then we label the activity a_j as "popular" because any compatible assignment must assign $k_{c(a_j)}$ agents of the same color to a_j, but more than $k_{c(a_j)}$ agents approve a_j for size $k_{c(a_j)}$. If any color $d \in [1,l]$ contains two or more popular activities, we reject the coloring because there is no stable assignment compatible with this coloring. To see why, if no agents are assigned to a popular activity of some color d, then due to compatibility there must exist at least one agent of the same color who is assigned to the void activity but approves the popular activity for size 1. Therefore, any stable, compatible assignment must assign k_d agents to a popular activity for color d (if any), but if there exist multiple popular activities of the same color, then no compatible assignment is stable. Without loss of generality (by re-coloring) let us assume that "popular" colors $[1,q]$ contain exactly one popular activity and "unpopular" colors $[q+1,l]$ contain non-popular activities (it is possible that $q = 0$ or $q = l$).

Let us now examine each color to decide whether we should reject the coloring or whether we can exclude some agents and/or activities from consideration (by re-coloring them as the special color, $l+1$). First, for each popular color $d \in [1,q]$ with a popular activity a_{j_d} (recall that there is exactly one popular activity for each popular color), we re-color all agents in $(N_d \setminus N(a_{j_d}))$ and all activities in $(A^*_d \setminus \{a_{j_d}\})$ as the special color $(l+1)$ because they cannot be assigned/used in any stable assignment compatible with c. Next, we check for each unpopular color $d \in [q+1,l]$. If $a_j \in A^*_d$ and $|N(a_j)| < k_d$ then we re-color a_j as $l+1$ as assigning k_d agents to a_j violates IR; if A^*_d becomes empty, we reject the coloring. Then, if there exist two distinct activities $a_j, a_{j'} \in A^*_d$ such that $N(a_j) \neq N(a_{j'})$, then we reject the coloring; any compatible assignment must assign no agents to at least one of these two activities (assume that a_j is such activity), but at least one agent in $N(a_j)$ approves $(a_j, 1)$ (due to decreasing preferences) while she must be assigned to the void activity, which implies instability of the assignment. If the coloring is not rejected after these conditions are checked, then we have $N(a_j) = N(a_{j'})$ for all $a_j, a_{j'} \in A^*_d$ where $d \in [q+1,l]$. Let us re-color all agents in $N_d \setminus N(a_j)$ as $l+1$ where a_j is any activity in A^*_d for all $d \in [q+1,l]$. We then check for another condition for each unpopular color $d \in [q+1,l]$. Let $A'_d = \{a_j \in A^*_d : \exists i \in N_{l+1}, u_i(a_j) \geq 1\}$. If A'_d contains two or more activities, it is clear that the coloring must be rejected because agent i (who cannot be assigned to any activity under the given coloring) approves size 1 for the activities in A'_d but the assignment can only choose one activity from A^*_d. Therefore, if $|A'_d| \geq 2$ then we reject the coloring; otherwise, if $|A'_d| = 1$, then we re-color all activities in $A^*_d \setminus A'_d$ as $l+1$ (because the only one in A'_d must be used for color d). If $|A'_d| = 0$, this step has no effect for this color.

Lastly, we consider the agents of color $l+1$ (who must be assigned to the void activity by any stable assignment compatible with the coloring). Let us define $k_{l+1} = 0$ for convenience (i.e., we do not assign any agents of color $l+1$ to any activities). For each color $d \in [1, l+1]$, if there exists some activity $a_j \in A^*_d$ and some agent $i \in N_{l+1}$ with $u_i(a_j) \geq k_d + 1$, then we reject the coloring because agent i is to be assigned to the void-activity, but she approves the outcome (a_j, k_d+1) as well as $(a_j, 1)$ (due to decreasing preferences), which means that regardless of whether a_j is used or not, no assignment would not be stable and compatible at the same time due to agent i. If the coloring has not been rejected, then we can now safely ignore all agents in N_{l+1} (as if they are non-existent) because stability constraint would not be violated by those agents.

We now proceed with the assumption that the coloring has not been rejected by our algorithm. Recall that we need to choose k_d agents among N_d where $d \in [1,q]$ to be assigned to the popular activity a_{j_d} while we know exactly which $k_{d'}$ agents must be assigned to one of the activities in $A^*_{d'}$ where $d' \in [q+1,l]$. For each popular color $d \in [1,q]$ define $N'_d = \{i \in N_d : \exists d' \in [1,l+1], u_i(a_j) \geq k_{d'} + 1 \text{ where } a_j \in A_{d'}\} \cup \{i \in N_d : \exists d' \in [q+1,l], |\{a_j \in A_{d'} : u_i(a_j) \geq 1\}| \geq 2\}$. Any stable assignment compatible with c must assign all agents in N'_d to the popular activity a_{j_d}. If some agent i in N'_d is assigned to the void-activity instead, then the resulting assignment cannot be stable; if i is contained in the first set (on the right-hand-side of definition of N'_d) above, then i approves sizes of both $k_{d'}+1$ and 1 for some activity a_j, which implies that regardless of whether a_j is used or not, i would wish to join a_j instead of $a_\emptyset$, while if i is contained in the second set (on the right-hand-side of definition of N'_d), then i approves size 1 for at least two non-popular activities of the same color which implies that i would wish to join one of them that is not used. Therefore, if $|N'_d| > k_d$ for some $d \in [1,q]$ we must reject the coloring, and otherwise we must assign all agents in N'_d to a_{j_d}. Without loss of generality we can assume that $N'_d = \emptyset$ for all $d \in [1,q]$ (provided that the coloring is not rejected yet) by assigning all such agents to the appropriate popular activity and then decreasing k_d by $|N'_d|$ before we proceed further.

Now suppose that for some color $d \in [1,q]$ and agent $i \in N_d$ and some color $d' \in [q+1,l]$ and some activity $a_j \in A^*_{d'}$, we have $u_i(a_j) \geq 1$. If i is assigned to $a_\emptyset$ (instead of a_{j_d}) and a_j is not used (no agent is assigned to it), then the assignment cannot be stable as i approves $(a_j, 1)$. That is, any compatible, stable assignment must assign i to a_{j_d} and/or use activity a_j. If we consider agents in $X = \cup_{d\in[1,q]} N_d$ and activities in $Y = \cup_{d'\in[q+1,l]} A^*_{d'}$ as vertices and there is an edge between (i, a_j) if and only if $u_i(a_j) \geq 1$ where $i \in X$ and $a_j \in Y$ (as a bipartite graph), finding a compatible assignment is equivalent to finding a vertex cover such that it chooses exactly

k_d vertices from each N_d with $d \in [1,q]$ and exactly 1 vertex from each $A^*_{d'}$ with $d' \in [q+1,l]$. Because the total number of vertices to be selected is bounded above by $k+l$, one can use a bounded search tree to determine whether a vertex cover of a small size exists or not in FPT time (i.e., exponential only in k but polynomial in n, p). If vertex i from X is chosen then we assign i to the popular activity of the same color and if vertex a_j from Y is chosen then we assign the agents of the same color to it. It is easy to verify that a compatible, stable assignment exists if and only if a vertex cover (with the aforementioned constraints) exists in this bipartite graph.

We omit the proof that our algorithm would not reject any coloring c which is compatible with at least one stable assignment. □

Lastly, we consider another special case of GASP when all activities are (pairwise) equivalent.

Theorem 6. *k-Stable-GASP with equivalent activities is in FPT.*

Proof. Let $p = |A^*|$ be the number of non-void activities which we assume are all equivalent. We use Color Coding to design a randomized FPT algorithm, which can easily be de-randomized using a family of k-perfect hash functions [1] as mentioned earlier.

We can assume that $p \leq k+1$ because k agents can be assigned to at most k copies and having more than one extra copy to which no agent is assigned does not change the problem (this is because we are seeking a solution of size exactly k). Due to space we only prove the claim when $p = k+1$, but it is easy to extend our proof.

Preliminaries. Let $N = \{1,2,\ldots,n\}$ be the set of n agents and $A^* = \{a_1,a_2,\ldots,a_p\}$ be the set of p copies of the only activity when $p = k+1$. Recall that by definition of equivalent activities, every agent i has $S_i(a_j) = S_i(a_{j'})$ for all j, j'. We first fix l (the number of copies of the activity to be used by a stable assignment) which must be between 1 and k, and $k_1,k_2,\ldots,k_l$ which is the number of agents assigned to each of the l copies; for convenience we define $k_{l+1} = 0$ as there is at least one extra copy that would not be used by the assignment. The total number of possible values for l and k_d's are bounded above by $O(k^k)$, which is exponential only in k. After we fix l, we color all agents uniformly and independently at random using colors 1 through l; let $c(\cdot)$ be this coloring scheme and $c(i)$ denote the color of agent i. We say that coloring c (together with l and k_d's) and a stable assignment π of size k are *compatible* if π assigns exactly k_d agents of color d to activity a_d for $d \in [1,l]$.

FPT Algorithm. Our algorithm will find a stable assignment compatible with c or determine that no such solution exists. Any stable assignment of size k has at least one compatible coloring, and the probability that a random coloring is compatible with some fixed stable assignment of size k is at least $(1/l)^k$. Our algorithm first partitions agents of each color into several subsets, and checks several necessary conditions for the coloring to be compatible with at least one stable assignment; if any of the conditions is not met, the coloring will be rejected by the algorithm.

For each color $d \in [1,l]$, the algorithm computes three subsets: $N_d = \{i \in N : c(i) = d\}$, $N_d^{\text{IR}} = \{i \in N_d : (a_d,k_d) \in S_i\}$, and $N_d^{\text{IN}} = \{i \in N_d : \exists d' \in [1,l+1] \text{ s.t. } (a_{d'},k_{d'}+1) \in S_i\}$ (recall $k_{l+1} = 0$). If $|N_d^{\text{IR}}| < k_d$ for any $d \in [1,l]$, no stable assignment is compatible with c because assigning k_d agents to a_d would not be individually rational (i.e., not enough agents approve the outcome), so the coloring should be rejected in this case. If for some $d \in [1,l]$ the set $N_d^{\text{IN}} - N_d^{\text{IR}}$ is not empty but contains some agent i, then no stable assignment is compatible with c because a stable assignment cannot assign i to a_d (because $i \notin N_d^{\text{IR}}$) but i would wish to join $a_{d'}$ for some $d' \in [1,l+1]$ which would make the assignment not stable; hence the coloring should be rejected in this case. If $|N_d^{\text{IN}}| > k_d$, then at least one agent i in N_d^{IN} should be assigned to the void activity, but i would wish to join $a_{d'}$ for some $d' \in [1,l+1]$ which would make the assignment not stable; hence the coloring should be rejected. If the coloring is not rejected by any of the cases mentioned earlier, then we have the following three conditions for every color $d \in [1,l]$: (a) $|N_d^{\text{IR}}| \geq k_d$, (b) $|N_d^{\text{IN}}| \leq k_d$, and (c) $N_d^{\text{IN}} \subseteq N_d^{\text{IR}}$. Let us define X_d for each $d \in [1,l]$ as follows: X_d contains an arbitrary set of k_d agents from N_d^{IR} such that every agent in N_d^{IN} is contained in X_d. Note that this is always possible due to the three conditions mentioned above. We claim that an assignment π which assigns agents in X_d to a_d and all other agents to $a_\emptyset$ is a stable assignment compatible with c. To prove compatibility, all agents in X_d are by definition of color d and $|X_d| = k_d$ for all $d \in [1,l]$. To prove stability, first consider any agent i who is assigned to the void activity and suppose $d = c(i)$. Since $i \notin X_d$, we know that $i \notin N_d^{\text{IN}}$ by definition, and therefore there is no $d' \in [1,l+1]$ such that $(a_{d'},k_{d'}+1) \in S_i$. Now consider any agent i who is assigned to a_d by π (thus $c(i) = d$). By definition $i \in X_d$ and thus $i \in N_d^{\text{IR}}$, which implies that $(a_d,k_d) \in S_i$. Therefore π is a stable assignment of size k, compatible with c.

Let us now prove that if there is at least one stable assignment that is compatible with c, then the algorithm does not reject the coloring. Let π be one such assignment and let X_d be the set of agents assigned to a_d by π. Due to compatibility we have $|X_d| = k_d$ and $c(i) = d$ for all $i \in X_d$ for all $d \in [1,l]$; in particular, by definition $X_d \subseteq N^{\text{IR}_d}$ and thus the first condition (a) above holds for all $d \in [1,l]$. Due to stability of π, every agent i with $\pi(i) = a_\emptyset$ satisfies that $\nexists d' \in [1,l+1]$ such that $(a_{d'},k_{d'}+1) \in S_i$. Therefore the conditions (b) and (c) above hold for all $d \in [1,l]$, which proves that the coloring c would not be rejected by the algorithm.

We have shown that if we begin with a coloring c (together with l and k_d's) that is compatible with at least one stable assignment, then our algorithm would find a stable assignment compatible with the coloring and that if no such assignment exists the coloring would be rejected. This is a Monte Carlo algorithm with probability of success at least $(1/k)^k$ and runtime bounded by $O((k^k)nk)$ (as our algorithm must enumerate all possible values of l and k_d's), which is polynomial in n but exponential only in k. □

3.3 Complexity of EF-GASP

Envy-freeness (EF) is a stronger solution concept than individual rationality. This relationship is not apparent under the classic complexity, but they differ under parameterization.

Theorem 7. *k-EF-GASP is $W[1]$-complete. It remains to be $W[1]$-complete even if each agent approves at most one size per activity.*

Proof. We reduce from the k-Clique problem.

Construction of GASP instance. Let $G = (V,E)$ be a graph instance of the k-Clique problem. Without loss of generality, assume $k \geq 2$. Let $V = \{v_1,v_2,\ldots,v_n\}$, and for each vertex $v_i \in V$, we create activity a_i and k^2 copies of v_i as (vertex) agents. For each edge $(v_i,v_j) \in E$, we create an activity $e_{i,j}$ and an (edge) agent $w_{i,j}$. This creates $|V|+|E|$ activities and $k^2|V|+|E|$ agents overall. For each copy of v_i, we define $S_{v_i} = \{(a_i,k^2)\} \cup \{(e_{i,j},1) : (v_i,v_j) \in E\}$ and for each agent $w_{i,j}$ we define $S_{w_{i,j}} = \{(e_{i,j},1)\}$. Let $k' = k^3 + \binom{k}{2} = k^3 + k(k-1)/2$. We claim that a clique of size k exists in G if and only if an EF assignment of size k' exists in our GASP instance.

Proof of equivalence between instances. Suppose that π is an EF assignment of size k'. If π assigns any copy of v_i to some activity $e_{i,j}$, then π cannot be EF because $w_{i,j}$ wishes to be assigned to $e_{i,j}$ in place of the copy of v_i; furthermore, π cannot assign more than one agent to any $e_{i,j}$ as no other agent approves the activity with any size other than 1. If π assigns any copy of v_i to a_i, then it must assign all k^2 copies of v_i to a_i as those agents only approve a_i with size k^2. Because π is of size k', it is clear that π can only assign agents to at most k different activities of the form a_i. Now suppose

π assigns some $w_{i,j}$ to $e_{i,j}$; due to EF, all k^2 copies of v_i must be assigned to a_i and all k^2 copies of v_j must be assigned to a_j; this implies that π can assign at most $\binom{k}{2}$ agents of the form $w_{i,j}$ to activities of the form $e_{i,j}$ (otherwise, π cannot be of size k' because $k^2(k+1) > k'$). Therefore, we conclude that π assigns k^3 vertex agents to k activities of the form a_i (without loss of generality, assume $\{a_1, a_2, \dots, a_k\}$) and that π assigns $w_{i,j}$ to $e_{i,j}$ if and only if $1 \le i, j \le k$ (all other agents are assigned to the void activity, because any unassigned vertex agent v_l ensures that no other agents are assigned to v_l's edge activities due to stability). This implies that the original instance contains a clique $C = \{v_1, v_2, \dots, v_k\}$ as there is an edge (v_i, v_j) if $1 \le i, j \le k$.

To prove the converse, suppose that $C = \{v_1, v_2, \dots, v_k\}$ is a clique in the original instance. Let π be an assignment such that π assigns k^2 copies of v_i to a_i if $i \le k$ and $w_{i,j}$ to $e_{i,j}$ if $1 \le i, j \le k$ and assigns all other agents to the void activity. Clearly π is individually rational by the construction of approval sets; π is also EF because no copy of v_i with $i > k$ or $w_{i,j}$ with $i > k$ or $j > k$ wishes to replace any other agent who is assigned to a non-void activity. This completes the proof of $W[1]$-hardness. To show completeness, one can reduce k-EF-GASP to the colored k-clique problem (known to be W[1]-complete), but we omit this proof due to space.

In our reduction each agent approves at most one size per activity, proving the second statement in the theorem. □

Unlike k-Stable-GASP, k-EF-GASP remains to be $W[1]$-complete even if all agents have increasing or decreasing preferences. Due to space, we only provide proof sketches.

Theorem 8. *k-EF-GASP is $W[1]$-complete even if all agents have increasing preferences.*

Proof sketch. We modify the reduction from our proof of Theorem 7. We construct the same `GASP` instance, but change approval sets such that if agent i approves an outcome (a,x), then we let the agent approve all outcomes (a,x') with $x < x' \le |N|$, to ensure increasing preferences. We also create $k'+1$ copies of a dummy agent z such that z approves all outcomes $(e_{i,j}, x)$ with $2 \le x \le |N|$ for all activities $e_{i,j}$ we create. Dummy agents ensure that any EF assignment of size k' assigns all copies of z to the void activity. □

Theorem 9. *k-EF-GASP is $W[1]$-complete even if all agents have decreasing preferences.*

Proof sketch. We modify the reduction from our proof of Theorem 7. We construct the same `GASP` instance, but we change approval sets such that if agent i approves an outcome (a,x), then we let the agent approve all outcomes (a,x') with $1 \le x' < x$, which ensures that all agents have decreasing preferences. □

Theorem 10. *k-EF-GASP is in FPT if all activities are equivalent.*

Proof. We use Color Coding to design a randomized FPT algorithm, which can easily be de-randomized using a family of k-perfect hash functions [1] as mentioned earlier.

Because there are only n agents we can assume that $p \le k$ because k agents can be assigned to at most k copies (unlike the case of stability, $k+1$ copies has no effect). Due to space we only prove the claim when $p = k$, but it can be easily extended.

Preliminaries. Let $N = \{1, \dots, n\}$ be the set of n agents and $A^* = \{a_1, \dots, a_k\}$ be the set of $p = k$ activities. By definition of equivalent activities, every agent i has $S_i(a_j) = S_i(a_{j'})$ for all j, j'. We first fix l (the number of activities to be used by an EF assignment) which must be between 1 and k, and $k_1, k_2, \dots, k_l$ which is the number of agents assigned to each of the l copies. The total number of possible values for l and k_d's are bounded above by $O(k^k)$. After we fix l, we color all agents uniformly and independently at random using colors 1 through l; let c be this coloring scheme and $c(i)$ denote the color of agent i. We say that coloring c (together with l and k_d's) and an EF assignment π of size k are *compatible* if π assigns exactly k_d agents of color d to activity a_d for $d \in [1,l]$.

FPT algorithm. Our algorithm either finds an EF assignment compatible with c or determines that no such solution exists. Any EF assignment of size k has at least one compatible coloring, and the probability that a random coloring is compatible with some fixed EF assignment of size k is at least $(1/k)^k$.

The algorithm first partitions agents of each color into subsets, and check several necessary conditions for the coloring to be compatible with at least one EF assignment; if any of the conditions is not met, the coloring will be rejected by the algorithm (as there is no EF assignment compatible with the given coloring). For each color $d \in [1,l]$, the algorithm computes three subsets: $N_d = \{i \in N : c(i) = d\}$, $N_d^{\mathrm{IR}} = \{i \in N_d : (a_d, k_d) \in S_i\}$, and $N_d^{\mathrm{IN}} = \{i \in N_d : \exists d' \in [1,l] \text{ s.t. } (a_{d'}, k_{d'}) \in S_i\}$. If $|N_d^{\mathrm{IR}}| < k_d$ for any $d \in [1,l]$, no EF assignment is compatible with c because assigning k_d agents to a_d would not be individually rational (i.e., not enough agents approve the outcome), so the coloring should be rejected in this case. If for some $d \in [1,l]$ the set $N_d^{\mathrm{IN}} - N_d^{\mathrm{IR}}$ is not empty but contains some agent i, then no EF assignment is compatible with c because an EF assignment cannot assign i to a_d (because $i \notin N_d^{\mathrm{IR}}$) but i would wish to join $a_{d'}$ for some $d' \in [1,l]$ which would make the assignment not envy-free; hence the coloring should be rejected in this case. If $|N_d^{\mathrm{IN}}| > k_d$, then at least one agent i in N_d^{IN} should be assigned to the void activity, but i would wish to join $a_{d'}$ for some $d' \in [1,l]$ which would make the assignment not envy-free; hence the coloring should be rejected.

If the coloring is not rejected by any of the cases mentioned earlier, then we have the following three conditions for every color $d \in [1,l]$: (a) $|N_d^{\mathrm{IR}}| \ge k_d$, (b) $|N_d^{\mathrm{IN}}| \le k_d$, and (c) $N_d^{\mathrm{IN}} \subseteq N_d^{\mathrm{IR}}$. Let us define X_d for each $d \in [1,l]$ as follows: X_d contains an arbitrary set of k_d agents from N_d^{IR} such that every agent in N_d^{IN} is contained in X_d. Note that this is always possible due to the three conditions mentioned above.

We claim that π which assigns agents in X_d to a_d and all other agents to $a_\emptyset$ is an EF assignment compatible with c. To prove compatibility, all agents in X_d are by definition of color d and $|X_d| = k_d$ for all $d \in [1,l]$. To prove stability, first consider any agent i who is assigned to the void activity and suppose $d = c(i)$. Since $i \notin X_d$, we know that $i \notin N_d^{\mathrm{IN}}$ by definition, and therefore there is no $d' \in [1,l]$ such that $(a_{d'}, k_{d'}) \in S_i$. Now consider any agent i who is assigned to a_d by π (thus $c(i) = d$). By definition $i \in X_d$ and thus $i \in N_d^{\mathrm{IR}}$, which implies that $(a_d, k_d) \in S_i$. This proves the claim.

Let us now prove that if there is at least one EF assignment that is compatible with c, then the algorithm does not reject the coloring. Let π be one such assignment and let X_d be the set of agents assigned to a_d by π. Due to compatibility we have $|X_d| = k_d$ and $c(i) = d$ for all $i \in X_d$ for all $d \in [1,l]$; in particular, by definition $X_d \subseteq N^{\mathrm{IR}_d}$ and thus the first condition (a) above holds for all $d \in [1,l]$. Due to stability of π, every agent i with $\pi(i) = a_\emptyset$ satisfies that $\nexists d' \in [1,l]$ such that $(a_{d'}, k_{d'}) \in S_i$. Therefore the conditions (b) and (c) above hold for all $d \in [1,l]$, which proves the claim.

We have shown that if we begin with a coloring c (together with l and k_d's) that is compatible with at least one EF assignment, then our algorithm would find an EF assignment compatible with the coloring and that if no such assignment exists the coloring would be rejected. This is a Monte Carlo algorithm with probability of success at least $(1/k)^k$ and runtime bounded by $O((k^k)nk)$ (as our algorithm must enumerate all possible values of l and k_d's), which is polynomial in n but exponential only in k. □

3.4 Complexity of Stable-EF-GASP

Stability and envy-freeness are not the same, but they are not exclusive, either. They together define another solution concept that is stronger than the two. k-Stable-EF-GASP is $W[1]$-hard under increasing preferences, but it is in FPT under decreasing preferences or equivalent activities. Due to space we omit proofs, but the following theorems can be proved in a similar way we proved Theorems 4, 5 and 6, respectively in this order, with some modifications.

Theorem 11. *k-Stable-EF-GASP is $W[1]$-hard even if all agents have increasing preferences for all activities.*

Theorem 12. *k-Stable-EF-GASP is in FPT if all agents have decreasing preferences for all activities.*

Theorem 13. *k-Stable-EF-GASP is in FPT if all activities are equivalent.*

3.5 Complexity of Perfect-GASP

Recall that k-Perfect-GASP is the problem of finding a perfect assignment that uses k activities out of p activities. Under this parameterization, `GASP` is W[2]-hard as the following theorem shows.

Theorem 14. *k-Perfect-GASP is $W[2]$-hard. It remains to be $W[2]$-hard even if all agents have increasing preferences or all agents have decreasing preferences.*

Proof. We reduce from the Dominating Set problem which is known to be $W[2]$-complete; recall that $D \subset V$ is a dominating set if $\forall v \in V \setminus D$, v has a neighbor in D.

Given $G = (V,E)$ and parameter k, let us create an instance of `GASP` as follows: Let $N = \{1,2,\dots,n\}$ and $A^* = \{a_1,a_2,\dots,a_n\}$ where $n = |V|$. For each agent i, define $S_i = \{(a_j,x) : ((v_i,v_j) \in E) \wedge (1 \le x \le n)\} \cup \{(a_i,x) : 1 \le x \le n\}$. Note that in this instance agents only care about activities. We set $k' = k$, and seek a perfect assignment using k activities. Let D be a dominating set of size k in G, and we can construct a perfect assignment π as follows: For each agent i, if $v_i \in D$, then let $\pi(i) = a_i$; if $v_i \notin D$, then there exists some $v_j \in D$ such that $(v_i,v_j) \in E$ because D is a dominating set, and let $\pi(i) = a_j$. Clearly π is a perfect assignment that uses only k activities. Conversely, suppose that a perfect assignment π exists which uses exactly k activities. Let A' be the set of k activities to which at least one agent is assigned under π (note that $|A'| = k$). Let $D = \{v_i : a_i \in A'\}$, and we claim that D is a dominating set in G. For any node $v_i \notin D$, we know that π assigns agent i to some activity $a_j \in A'$ where $a_j \neq a_i$, and thus $v_j \in D$. Since π is individually rational, it implies that $(v_i,v_j) \in E$, and therefore D is a dominating set. This reduction also proves the second statement of the theorem as all agents have increasing and decreasing preferences. ☐

We do not know the exact complexity of k-Perfect-GASP when all activities are equivalent (besides NP-hardness).

3.6 Solutions of size at least or at most k

We have only considered finding a solution of size exactly k. Yet one may wish to find a solution of size at least k (for maximization problems like k-GASP under IR, stable, EF, or stable-EF) or at most k (for minimization problems like k-Perfect-GASP). Unlike for most studied maximization problems (e.g. Clique, k-Path, etc.) the existence of an IR, stable, or EF assignment of size k *does not* guarantee existence of a solution of size $k-1$ or smaller (whereas a clique of size k always contains cliques of smaller sizes). Therefore we are essentially solving a different problem when we seek a solution of size at least k for these three solution concepts.

For k-IR-GASP, finding a solution of size at least k is also in FPT.

Theorem 15. *At-least-k-IR-GASP is in FPT.*

Proof. For readability, we write a "solution" to refer to an IR assignment of size at least k. Among all solutions, first suppose that there is some π such that $|\pi^j| \ge k$ for some $j \in [1,p]$ (recall the notation from Definition 2). Let π' be an assignment such that it assigns all agents in π^j to a_j while all other agents to $a_\emptyset$; clearly, π' is also a solution. We can find in $O(n^2 p)$ time this kind of solutions as follows: For each integer $x \in [k,n]$ and activity a_j, we count the number of agents approving (a_j,x), and if this number exceeds x, then assigning any x agents of them to a_j results in an IR assignment of size at least k. Now suppose instead that such solutions do not exist, and every solution π satisfies that $\forall j \in [1,p]$, $|\pi^j| < k$. We claim that, then, a solution of size between k and $2k-1$ (inclusive) exists. If this claim is true, then we can use our FPT algorithm from Theorem 2 to find a solution of size $k, k+1, \dots, 2k-1$, which would run in FPT time overall. To see why the claim is true, suppose that π is a solution of size at least $2k$ and let $j \in [1,p]$ be any index such that $|\pi^j| > 0$. If π' is the same as π except that π' assigns all agents in π^j to $a_\emptyset$ (while other agents are assigned in the same manner as π), then π' is also a solution whose size is exactly $|\pi^j|$ smaller than the size of π. As we keep applying this process, we eventually end up with a solution of size at most $2k-1$. Hence, in all cases, we can find a solution in FPT time. ☐

For k-Stable-GASP, Darmann et al. [6] show that finding a perfect assignment can be reduced to finding a stable assignment of size at least one (i.e., $k = 1$), which rules out FPT algorithms for "at least k"-Stable-GASP unless P=NP. The same reduction also applies for k-Stable-EF-GASP. For k-EF-GASP, we conjecture that a similar reduction may exist, and leave it as an open problem. For k-Perfect-GASP, finding a perfect assignment using *at most* k activities remains to be W[2]-hard: a solution using k activities implies the existence of a solution using at most $k+1$ activities.

4. DISCUSSION AND FUTURE WORK

We investigated the parameterized complexity of the Group Activity Selection Problem (`GASP`) for various solution concepts and restrictions on inputs. Our results indicate that the computational complexity of `GASP` varies when its input is restricted (due to preferences of agents or uniformity of activities) or the solution concept changes, which is not distinguishable under classic complexity. In particular, decreasing preferences make the problems more tractable, but increasing preferences do not. This is intriguing because Lee and Shoham [12], on the other hand, showed that increasing preferences lead to a strategy-proof, optimal mechanism but decreasing preferences lead to an impossibility result, under strategic settings with just one activity.

Our work leaves a few interesting open problems for future work. First, we do not know the exact complexity of k-Perfect-GASP with equivalent activities besides its NP-completeness. It would be intriguing if the problem is FPT. Second, the focus in this paper is to exhibit any FPT algorithm; we have not tried hard to optimize the dependence on k. It would be interesting to show conditional lower bounds on how the runtime should depend on k, especially in the case of k-IR-GASP. Third, obtaining completeness results to match our W[1]- and W[2]-hardness results is a natural follow-up problem. Lastly, one can consider a different setting where agents have a strict ordering over the set of outcomes (complexity results of which are recently shown by Darmann [5]), instead of dichotomic preferences on outcomes.

Acknowledgements

This work was done while both authors were at Stanford University. This work was funded in part by the National Science Foundation (grant IIS-1347214), AFOSR MURI, and the Kwanjeong Educational Foundation.

REFERENCES

[1] N. Alon, R. Yuster, and U. Zwick. Color-coding. *Journal of the ACM (JACM)*, 42(4):844–856, 1995.

[2] C. Ballester. Np-completeness in hedonic games. *Games and Economic Behavior*, 49(1):1–30, 2004.

[3] N. Betzler, J. Guo, and R. Niedermeier. Parameterized computational complexity of dodgson and young elections. *Information and Computation*, 208(2):165–177, 2010.

[4] R. H. Chitnis, M. Hajiaghayi, and V. Liaghat. Parameterized complexity of problems in coalitional resource games. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[5] A. Darmann. Group activity selection from ordinal preferences. In T. Walsh, editor, *Algorithmic Decision Theory*, volume 9346 of *Lecture Notes in Computer Science*, pages 35–51. Springer International Publishing, 2015.

[6] A. Darmann, E. Elkind, S. Kurz, J. Lang, J. Schauer, and G. Woeginger. Group activity selection problem. In P. Goldberg, editor, *Internet and Network Economics*, volume 7695 of *Lecture Notes in Computer Science*, pages 156–169. Springer Berlin Heidelberg, 2012.

[7] R. G. Downey and M. R. Fellows. *Fundamentals of parameterized complexity*, volume 4. Springer, 2013.

[8] U. Endriss, R. De Haan, and S. Szeider. Parameterized complexity results for agenda safety in judgment aggregation. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 127–136. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

[9] G. Erdélyi and M. Fellows. Parameterized control complexity in bucklin voting and in fallback voting. In *Proceedings of the 3rd International Workshop on Computational Social Choice*, pages 163–174. Universität Düsseldorf, 2010.

[10] L. A. Hemaspaandra, R. Lavaee, and C. Menton. Schulze and ranked-pairs voting are fixed-parameter tractable to bribe, manipulate, and control. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 1345–1346. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[11] R. Impagliazzo and R. Paturi. Complexity of k-sat. In *Computational Complexity, 1999. Proceedings. Fourteenth Annual IEEE Conference on*, pages 237–240. IEEE, 1999.

[12] H. Lee and Y. Shoham. Stable invitations. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 965–971, 2015.

[13] H. Lee and V. V. Williams. Complexity of the stable invitations problem. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-10, 2017, San Francisco, California, USA.*, 2017.

[14] H. Liu and J. Guo. Parameterized complexity of winner determination in minimax committee elections. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 341–349. International Foundation for Autonomous Agents and Multiagent Systems, 2016.

[15] T. Shrot, Y. Aumann, and S. Kraus. Easy and hard coalition resource game formation problems: a parameterized complexity analysis. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 433–440. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

[16] L. Xia. Fixed-parameter tractability of integer generalized scoring rules. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1599–1600. International Foundation for Autonomous Agents and Multiagent Systems, 2014.