Complexity of the Stable Invitations Problem

Hooyeon Lee, Virginia Vassilevska Williams
Computer Science Department
Stanford University
{haden.lee,virgi}@cs.stanford.edu

Abstract

We study the Stable Invitations Problem (SIP) in which an event organizer is to invite a subset of agents (from a group of agents) to an event, subject to certain rationality criteria. In SIP, the agents have friends, enemies, and preferences on the number of attendees at the event; an agent is willing to attend the event if all friends of the agent attend, no enemy of the agent attends, and the number of attendees is acceptable to the agent. We consider two solution concepts: (1) individual rationality (everyone who is invited is willing to attend) and (2) (Nash) stability (no agent wants to deviate from the given invitation). It is known that finding an invitation of given size for either concept is NP-complete. In this work, we study the complexity of SIP on a finer scale, through the lense of parameterized complexity. For the two solution concepts and the special cases where the number of friends and/or enemies is bounded above by a constant, we show that the problems belong to different complexity classes when parameterized by the size of solutions. For instance finding an individually rational invitation of size k is W[1]-complete, yet finding a stable invitation is W[2]-complete. Moreover, when all friend and enemy relations are symmetric, finding a solution of either of the concepts becomes fixed-parameter tractable unless agents have unbounded number(s) of enemies.

1 Introduction and Related Work

Imagine an event organizer who is convening an event for a group of agents who may have friends and/or enemies. Naturally, agents wish to attend with their friends and prefer that their enemies do not attend the same event. Some agents may have preferences on the number of attendees as well – some may like the event be crowded while others may prefer a small number of attendees. Given this information, the organizer wishes to invite as many agents to the event as possible, subject to certain rationality and/or stability conditions. The first condition is *individual rationality*, which requires that everyone who is invited be willing to attend. In addition to individual rationality, the organizer may want to ensure that agents who are not invited will not wish to attend without permission (*Nash stability*). To model this setting, Lee and Shoham (2015) studied the Stable Invitations Problem (`SIP`) for different solution concepts including individual rationality and (Nash) stability. They also provided many computational complexity results for `SIP` under various restrictions on agent preferences – for instance, when agents only have a small number of friends and/or enemies.

These hardness results essentially argue that it is hard to find a largest invitation (subject to rationality or stability). Suppose, however, that we are satisfied with a small solution of size k that satisfies our rationality criteria. We can try a brute-force solution: try all possible k-subsets of n agents, and then check whether the desired criterion is satisfied. This brute-force solution runs in $O(n^k)$ time, which is polynomial for any fixed k but is not desirable. A much better running time would be one of the form $O(f(k) \cdot n)$ - such a runtime would be *linear*, regardless of the constant k and the function f. More generally, on input size n, one would like an algorithm with runtime $f(k) \cdot n^c$, where c is independent of k. Such an algorithm is known as *fixed parameter tractable* (FPT). Developing FPT algorithms, especially linear time ones, greatly mitigates the NP-hardness of problems as it shows that these problems are actually quite tractable.

The field of parameterized complexity strives to classify NP-hard problems on a finer scale by analyzing time complexity in terms of both the input size and an additional parameter. The FPT problems are viewed as the most tractable problems in NP (of course, all polytime problems are in FPT). The *W-hierarchy* represents a hierarchy of complexity classes under parameterization, including 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 (Downey and Fellows 2012). Further, it is believed that FPT $\neq$ W[1] and W[2] $\neq$ W[1], so that the NP-hard parameterized 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. Popular hardness assumptions such as the Exponential Time Hypothesis (ETH) of Impagliazzo and Paturi (1999) 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.

This work investigates how different solution concepts and different restrictions on inputs influence the difficulty of `SIP` under parameterization. We consider the cases where the number of friends and enemies each agent has is bounded above by some constant, as well as the cases where friend-and-enemy relationship is symmetric. Our classification is complete, as shown in Tables 1 (for non-symmetric cases) and 2 (for symmetric cases) in Section 3.

Related Work. Lee and Shoham (2015) proposed the Stable Invitations Problem (SIP), and provided many complexity results of SIP. Our work is an extension to their work. We analyze SIP on a finer scale than the classical complexity, and further consider an interesting special case in which friend and enemy relationship is symmetric, and show that the symmetry plays a crucial role in complexity of the problems. Lee and Shoham (2015) in turn stated that SIP was motivated by the Group Activity Selection Problem (GASP) (Darmann et al. 2012), in which there are multiple activities being organized, and agents have preferences over activities as well as the number of participants in each activity; yet agents have anonymous preferences in that they do not have friend-and-enemy relationships. In the Group Activity Selection Problem, the goal is to assign as many agents to activities as possible, subject to rationality criteria similar to what we defined earlier. Darmann et al. (2012) provided many complexity results (most of which are NP-hard results) for the Group Activity Selection Problem, even under various restrictions on preferences of agents.

Both GASP and SIP can be viewed as hedonic coalition games with concise representation of preferences of agents (linear in the size of input, as opposed to exponential). More details on how these two problems are related to Hedonic Games can be found in Section 2.2 of the work by Darmann et al. (2012) and Section 2 of the work by Lee and Shoham (2015). Much work has been devoted to analyzing solution concepts in hedonic coalition games such as stability and Pareto-optimality (Bogomolnaia and Jackson 2002; Dreze and Greenberg 1980; Aziz and Brandl 2012). In this work, we focus on analyzing computational complexity of finding solutions under two of these concepts – namely, individual rationality and stability. Ballester (2004) provides a number of classical complexity results (in fact, hardness results) for finding a core-stable, Nash-stable, or individually rational outcome in hedonic games and anonymous hedonic games. However, as Lee and Shoham (2015) mention in their work, these hardness results do not imply similar hardness results for SIP; furthermore, our results go beyond NP-hardness as we place variations of SIP into the W-hierarchy.

2 Definitions and Known Results

We begin by introducing the definitions proposed by Lee and Shoham (2015), yet we make modifications to notation for readability and consistency in this paper.

Definition 1. An instance of the Stable Invitations Problem (SIP) is given by a set of agents $N = \{a_1, a_2, \dots, a_n\}$, and an *approval set* $S_i \subseteq [1, n]$, a *friend set* $F_i \subseteq N$, and an *enemy set* $E_i \subseteq N$ for each agent $a_i \in N$. It is interpreted that agent a_i is willing to attend if all friends in F_i attend, no one in E_i attends, and the number of attendees (including a_i) is acceptable (i.e., the number is contained in S_i).

Definition 2. An invitation I in SIP is a subset of agents.

We say that an invitation I is *individually rational* (IR) if for every agent $a_i \in I$, $|I| \in S_i$, $F_i \subseteq I$, and $E_i \cap I = \emptyset$.

We say that an invitation I is *(Nash) stable* if it is individually rational, and if for every agent $a_j \notin I$, $|I'_j| \notin S_j$, $F_j \not\subseteq I'_j$, or $E_j \cap I'_j \neq \emptyset$ where $I'_j = I \cup \{a_j\}$.

Individual rationality (IR) requires that every invited agent is willing to attend. Stability further requires that those who are not invited are not willing to participate (without permission of others) because not all of her friends are attending, some of her enemies are attending, or the number of attendees would be unacceptable. We consider the following two problems of finding invitations of size k:

- k-IR-Invitation: $\exists$ IR invitation of size k?
- k-Stable-Invitation: $\exists$ stable invitation of size k?

We first consider restrictions on inputs by limiting the size of largest friend-sets and enemy-sets, respectively. For integer constants α and β, if an instance of SIP satisfies $|F_i| \leq \alpha$ and $|E_i| \leq \beta$ for all $a_i \in N$, we call it an (α, β)-instance of SIP. Lee and Shoham (2015) showed that SIP can be solved in polytime only if α and β are small enough, but the problems are NP-hard in general. We will consider the same restrictions in this work, and provide our complete analysis of parameterized complexity of SIP.

In addition to these restrictions, we consider the special case where agents have symmetric relationships.

Definition 3. Given an instance of SIP, we say that agents have *symmetric social relationships* if $a_j \in F_i$ if and only if $a_i \in F_j$ and $a_l \in E_i$ if and only if $a_i \in E_l$ for every a_i.

Symmetric relationships are present in many settings. In the political world, countries are either allies or enemies or neutral, and this relationship is symmetric. Another example is alliances of airlines – airlines that serve similar regions are competing directly with each other while wish to partner with airlines from other regions to complement their routes. When a new alliance is being formed, some members only want to be in an alliance if they are together and others would not want to join if their enemies are there. Furthermore, the symmetric relationships can model settings with quite versatile structure. For instance, friendship and enemy relationships are not transitive: a friend of a friend can be a friend or an enemy (Hawaiian Airlines is a partner with American, Delta, and United Airlines, which are direct competitors of one another) and an enemy of an enemy can be an enemy or a friend (American is an enemy of Delta which is an enemy of United but American is not a friend of United).

Theorem 1 summarizes the most relevant results of Darmann et al. (2012) and Lee and Shoham (2015) to this work.[1]

Theorem 1. *k-IR-Invitation and k-Stable-Invitation can be solved in polynomial time if* $(\max_{a_i \in N} |F_i|) + (\max_{a_i \in N} |E_i|) \leq 1$ *(i.e., $\alpha + \beta \leq 1$). In other cases, both problems are NP-hard.*

Note that k-IR-Invitation and k-Stable-Invitation are of the same classical complexity, even though stability is a stronger solution concept. Under parameterization, however, these two problems are contained in different complexity classes in the W-hierarchy (see Table 1). In what follows, we show that the parameterized complexity of these problems varies with different solution concepts and under different restrictions on inputs to SIP.

[1] Darmann et al. (2012) showed easiness when $\alpha = \beta = 0$, while Lee and Shoham (2015) proved the other results.

	k-IR-Invitations				k-Stable-Invitations			
	$\beta = 0$	$\beta = 1$	$2 \leq \beta \leq f(k)$	unbounded β	$\beta = 0$	$\beta = 1$	$2 \leq \beta \leq f(k)$	unbounded β
$\alpha = 0$	P	P	FPT	W[1]-C	P	P	FPT	W[2]-C
$\alpha = 1$	P	FPT	FPT	W[1]-C	P	W[1]-C	W[1]-C	W[2]-C
$\alpha \geq 2$	W[1]-C	W[1]-C	W[1]-C	W[1]-C	W[1]-C	W[1]-C	W[1]-C	W[2]-C

Table 1: Complexity of k-IR-Invitation and k-Stable-Invitation. $f(k)$ can be an arbitrary function of k that only depends on k. All entries other than "P" imply NP-completeness. "W[1]-C" and "W[2]-C" mean W[1]-completeness and W[2]-completeness, respectively. Note that P and NP-completeness results were known prior to this work as summarized in Theorem 1, but all other results are original.

3 Parameterized Complexity

In this section, we study parameterized complexity of k-IR-Invitation and k-Stable-Invitation. Our main contributions are summarized in Table 1. For instance, finding an IR invitation of size k is in FPT when $\alpha = 1$ and β is a positive constant (bounded above by some function of k), but finding a stable invitation in the same cases is W[1]-complete.

3.1 k-IR-Invitation

Recall that k-IR-Invitation is the problem of finding an IR invitation of size k. When $\alpha + \beta > 1$, the problem is known to be NP-hard (Theorem 1). We first present easiness results: k-IR-Invitation is in W[1] in general, and it is in FPT if $\alpha \leq 1$ and β is bounded by some function $f(k)$ of k. We then present hardness results by showing that k-IR-Invitation is W[1]-hard when $\alpha \geq 2$ and/or β is unbounded.

Theorem 2. *k-IR-Invitation is in W[1].*

Proof sketch. We reduce the problem to the weighted circuit SAT [2] of constant depth and of weft at most 1. □

Theorem 3. *k-IR-Invitation is in FPT if $\alpha \leq 1$ and $\beta \leq f(k)$ where $f(k)$ can be an arbitrary function of k.*

Proof. Without loss of generality, assume that $k \in S_i$ for all $a_i \in N$. Otherwise, we can remove a_i from the input instance as no IR invitation of size k can contain a_i. If a_i is removed, and there is some a_j with $a_i \in F_j$, we remove a_j as well for the same reason. We repeat this removal process until no such agent remains (this can be done in linear time).

Let $\mathcal{A}$ be some polytime algorithm that solves k-IR-Invitation if $\alpha \leq 1$ and $\beta = 0$ (it exists due to Theorem 1). We will use $\mathcal{A}$ as a sub-routine in our FPT algorithm. Consider any coloring c which colors agents using two colors $\{0, 1\}$; let $c(i) \in \{0, 1\}$ be the color of agent a_i. We say that coloring c and IR invitation I of size k are *compatible*, if for every agent $a_i \in I$, $c(i) = 1$ and for every agent $a_j \in \cup_{i:i \in I} E_i$, $c(j) = 0$. Coloring c may be compatible with any number of solutions of size k (possibly none), and any solution of size k may be compatible with many colorings (but it is compatible with at least one coloring).

Given some arbitrary coloring c, we can find an IR invitation of size k that is compatible with c or determine that no compatible IR invitation exists in FPT time as follows. First, we re-color every agent a_i with $c(i) = 1$ to color 0 such that $\exists a_j \in F_i$ with $c(j) = 0$ or $\exists a_l \in E_i$ with $c(l) = 1$ (order in which we re-color agents does not matter). This process does not re-color any agent $a_i \in I$ if I is compatible with c. After the re-coloring step, let $N_1 = \{a_i \in N : c(i) = 1\}$, and we run the algorithm $\mathcal{A}$ on N_1 as input. Suppose that $\mathcal{A}$ finds an IR invitation I of size k given N_1. I is individually rational because its friend constraints are satisfied (due to correctness of $\mathcal{A}$) and its enemy constraints are satisfied because no agent with color 0 is included in N_1 (enforced by coloring). Now suppose that $\mathcal{A}$ reports that no IR invitation I of size k exists among the agents in N_1. Then there is no IR invitation of size k that is compatible with c; if such invitation $I' \subseteq N_1$ exists, then I' satisfies the friend constraints (because it is IR) and therefore $\mathcal{A}$ should find it, which is a contradiction. Thus if our algorithm begins with coloring compatible with some IR invitation(s), it will find one.

If we color agents uniformly and independently at random, then the probability of success of our algorithm is at least $1/2^{(k+1)\beta}$ (because, with respect to some fixed IR invitation I^*, we must color all agents in I^* as 1 and the union enemies of agents in I^* as 0, to start with compatible coloring). If we run this algorithm $2^{(k+1)\beta} \ln n$ times, the probability of success is at least $1 - 1/n$. Our FPT algorithm's runtime depends on the runtime of $\mathcal{A}$. The algorithm can be de-randomized using a family of k-perfect hash functions as shown in the work by Alon et al. (1995). □

Theorem 4. *k-IR-Invitation is W[1]-complete if β is not bounded above by any function $f(k)$.*

Proof. We reduce from k-Independent-Set. Given an arbitrary graph $G = (V, E)$ and a parameter k, we create agents $N = V = \{v_1, v_2, \ldots, v_n\}$. For each v_i, define $S_{v_i} = \{k\}$, $F_{v_i} = \emptyset$, and $E_{v_i} = \{v_j : (v_i, v_j) \in E\}$ (hence β is equal to the max-degree of nodes in G). If $I \subset V$ is an independent set of size k, then I is an IR invitation in the instance we created: For all $v_i \in I$, we have $|I| = k \in S_{v_i}$, $F_{v_i} = \emptyset \subset I$, and $E_{v_i} \cap I = \emptyset$ because I is an independent set in the original graph. Conversely, suppose I is an IR invitation of size k in the instance we created. Then I is an independent set of size k because no two agents in I are enemies of each other, and thus their corresponding nodes in the graph are not neighbors of each others. This proves W[1]-hardness, and completenes follows from Theorem 2. □

Theorem 5. *k-IR-Invitation is W[1]-complete if $\alpha \geq 2$.*

Proof Sketch. We reduce from k-Clique. Given an arbitrary graph $G = (V, E)$ and a parameter k, we create a set of

[2] For details the WCSAT problem see the work by Creignou and Vollmer (2015).

agents N as follows. For each node $v_i \in V$, we create k^2 node-agents that are labeled as $w_{i,x}$ where $x \in [k^2]$. For each node-agent $w_{i,x}$ we define $F_{w_{i,x}} = \{w_{i,x+1}\}$ (where w_{i,k^2+1} is understood as $w_{i,1}$) and $E_{w_{i,x}} = \emptyset$. Note that an IR invitation must include all or none of the $w_{i,x}$'s for each i because of their friend sets. Next, for each edge $(v_i, v_j) \in E$, we create an edge-agent $e_{i,j}$ with $F_{e_{i,j}} = \{w_{i,1}, w_{j,1}\}$ and $E_{e_{i,j}} = \emptyset$. Note that if an IR invitation includes $e_{i,j}$, then it must also include all $2k^2$ node-agents of the form $w_{i,x}$ and $w_{j,x}$ with $x \in [k^2]$ (due to friend sets).

Finally, define $k' = k^3 + \binom{k}{2}$ to be the parameter for the k-IR-Invitations we created, and define approval sets of all agents to contain k'. Clearly the instance we created satisfies $\alpha = 2$ and $\beta = 0$. The number of agents we created is $k^2|V| + |E|$, polynomial in the size of the original instance. Proof of correctness of the reduction is omitted due to space. W[1]-completenes follows from Theorem 2. □

3.2 k-Stable-Invitation

k-IR-Invitation and k-Stable-Invitation have the same classical complexity for all values of α and β, but parameterization indicates that k-Stable-Invitation is a more difficult problem than k-IR-Invitation. This is not surprising because a stable invitation requires that everyone (whether invited or not) be satisfied with the invitation.

Theorem 6. *k-Stable-Invitation is in W[2]. When β is bounded above by some function $f(k)$, it is in W[1].*

Proof sketch. We reduce k-Stable-Invitation to the weighted circuit SAT (WCSAT) of constant depth and of weft at most 2; if β is bounded, then weft can be reduced to 1. □

Theorem 7. *k-Stable-Invitation is in FPT when $\alpha = 0$ and $\beta \leq f(k)$ where $f(k)$ can be an arbitrary function of k.*

Proof Sketch. The main idea is similar to the algorithm for Theorem 3. However, finding a stable invitation is considerably more difficult due to uninvited agents. We first color all agents using two colors $\{0, 1\}$ uniformly and independently at random; let c be this coloring such that $c(i)$ is the color of agent a_i. If there is some a_i with $c(i) = 1$ such that $k \notin S_i$ or $\exists a_j \in R_i$, then we re-color a_i to $c(i) = 0$. We repeat this until no such agent remains. Now we will find k agents of color 1 that form a stable invitation. Agents of color 0 will be uninvited for sure, but we need to ensure stability – we must invite at least one enemy of every agent of color 0. This can be done in a brute-force manner in FPT time: The depth of search tree is at most k (as we can invite up to k agents) and the branching factor is $f(k)$ (because each agent has at most $f(k)$ enemies), and thus search space is bounded above by $O((f(k))^k)$. After choosing (at most) k agents to be included in the solution, the rest of the algorithm is similar to the algorithm for Theorem 3. □

Theorem 8. *k-Stable-Invitation is W[1]-complete if $\alpha, \beta \geq 1$ and β is bounded above by some function $f(k)$.*

Proof sketch. We reduce from k-Clique. Let $G = (V, E)$ be an arbitrary graph for the k-Clique problem with parameter k. Let us define $k' = 2(k^3 + \binom{k}{2})$ which is the parameter for k-Stable-Invitation. For each node $v_i \in V$, we first create a group of $2k^2$ node-agents (call them G_i) such that $G_i = \{w_{i,x} : x \in [2k^2]\}$, and define $F_{w_{i,x}} = \{w_{i,x+1}\}$ (where $w_{i,2k^2+1}$ is understood as $w_{i,1}$) and $S_{w_{i,x}} = \{k'\}$. For each edge $(v_i, v_j) \in E$, we create four edge-agents $e_{i,j}, e'_{i,j}, f_{i,j}$, and $f'_{i,j}$. Define $F_{e_{i,j}} = \{w_{i,1}\}$, $F_{e'_{i,j}} = \{w_{j,1}\}$, and $S_{e_{i,j}} = S_{e'_{i,j}} = \{k'\}$. Define $F_{f_{i,j}} = \{e_{i,j}\}$, $E_{f_{i,j}} = \{e'_{i,j}\}$, and $S_{f_{i,j}} = \{k' + 1\}$. Define $F_{f'_{i,j}} = \{e'_{i,j}\}$, $E_{f'_{i,j}} = \{e_{i,j}\}$, and $S_{f'_{i,j}} = \{k' + 1\}$. This creates $2k^2 n$ node-agents and $4|E|$ edge-agents, whose size is polynomial in n, k, and each agent has at most one friend and one enemy (satisfying $\alpha = \beta = 1$). The rest of the proof is omitted due to space, and W[1]-completenes follows from Theorem 6. □

Theorem 9. *k-Stable-Invitation is W[1]-complete if $\alpha \geq 2$ and β is bounded above by some function $f(k)$.*

Proof sketch. We reduce from k-Independent-Set. Let $G = (V, E)$ be an arbitrary graph, with parameter k. For each node $v \in V$ we create a node-agent v with approval set $S_v = \{k\}$ and friend set $F_v = \emptyset$. For each edge $(v, w) \in E$ we create an edge-agent $e_{v,w}$ with friend set $F_{e_{v,w}} = \{v, w\}$ and approval set $S_{e_{v,w}} = \{k + 1\}$. The rest of the proof is omitted due to space, and W[1]-completenes follows from Theorem 6. □

Theorem 10. *k-Stable-Invitation is W[2]-complete if β is not bounded above by any function of k.*

Proof sketch. We reduce from k-Dominating-Set. Given $G = (V, E)$ and a parameter k, we create $2n$ node-agents by creating x_i and y_i for each $v_i \in V$. We define $S_{x_i} = \{k\}, F_{x_i} = \emptyset, E_{x_i} = \{y_i\} \cup \{y_j : (v_i, v_j) \in E\}$ and $S_{y_i} = \{k\}, F_{y_i} = \emptyset, E_{y_i} = \{x_i\} \cup \{x_j : (v_i, v_j) \in E\}$. A stable invitation cannot contain any of y_i's because of their approval sets.[3] The rest of the proof is omitted due to space, and W[2]-completenes follows from Theorem 6. □

4 Symmetric Social Relationship

Recall the definition of "symmetric social relationships" from Definition 3. Under symmetric social relationships, both k-IR-Invitation and k-Stable-Invitation admit efficient FPT algorithms when β is bounded, as shown in Table 2, although their complexity does not change when β is unbounded (W[1]-complete and W[2]-complete, respectively).

When we compare the results in Table 1 and Table 2, two interesting observations can be made. First, k-IR-Invitations and k-Stable-Invitations have the same classical complexity even under symmetric social relationships. Second, both problems now admit efficient FPT algorithms for broader domains of inputs – as long as β is bounded. Note that our classical complexity results are original (i.e., not implied by Theorem 1) because Lee and Shoham (2015) did not consider the special case of symmetric social relationships.

We shall utilize "friend graph" and "enemy graph" in our reductions throughout this section. In a friend (enemy)

[3] This reduction also proves Theorem 16 for the symmetric case.

	k-IR-Invitations (symmetric social relationships)					k-Stable-Invitations (symmetric social relationships)				
	$\beta = 0$	$\beta = 1$	$\beta = 2$	$3 \leq \beta \leq f(k)$	unbounded β	$\beta = 0$	$\beta = 1$	$\beta = 2$	$3 \leq \beta \leq f(k)$	unbounded β
$\alpha = 0$	P	P	P	FPT	W[1]-C	P	P	P	FPT	W[2]-C
$\alpha = 1$	P	P	FPT	FPT	W[1]-C	P	P	FPT	FPT	W[2]-C
$\alpha \geq 2$	P	FPT	FPT	FPT	W[1]-C	P	FPT	FPT	FPT	W[2]-C

Table 2: Complexity of SIP with symmetric social relationships. $f(k)$ can be an arbitrary function of k that only depends on k. All entries other than "P" imply NP-completeness. "W[1]-C" and "W[2]-C" mean W[1]-completeness and W[2]-completeness, respectively. All results are original (including classical complexity results).

graph, nodes represent agents and there is an edge between nodes if they are friends (enemies).

4.1 Symmetric k-IR-Invitations

We first present classical complexity results for k-IR-Invitations under symmetric social relationships, followed by parameterized complexity results.

Theorem 11. *When agents have symmetric social relationships, k-IR-Invitations can be solved in polynomial time if (i) $\beta = 0$, (ii) $\beta = 1$ and $\alpha \leq 1$, or (iii) $\beta = 2$ and $\alpha = 0$. Otherwise, the problem is NP-hard.*

Proof sketch. Let us consider case (ii) in the statement. As before, without loss of generality assume that all agents accept the size k (i.e., $k \in S_i$ for all $a_i \in N$). We first construct an *enemy graph* in which nodes represent agents, and we create an edge between two nodes if their corresponding agents are enemies of each other. For every pair of friends, we merge their nodes in this graph into a meta-node of weight 2 (if they are also enemies of each other, then we simply remove them from the graph); let us call the resulting graph a *friend graph*. Now finding an IR invitation of size k is equivalent to finding an independent set of total weight k in the friend graph. Although finding an independent set (of given size) is NP-hard, all nodes in the friend graph have at most two edges, and thus each connected component in the friend graph is either a path or a cycle. A dynamic programming algorithm can solve this problem in polytime.

Let us now prove that the problem is NP-hard if none of the three conditions in the statement holds. It is known that the Independent Set problem is NP-hard even if every node has degree at most 3 (Garey, Johnson, and Stockmeyer 1976). Given an instance of this problem, we can create an instance of k-IR-Invitations as follows. For each node, we create an agent a_i with $S_i = \{k\}$ (agent only approves size k). If there is an edge between two nodes, we make their corresponding agents enemies of each other. The resulting instance is a valid instance (with symmetric social relationships) of k-IR-Invitations with $\alpha = 0$ and $\beta = 3$ (because each node in the original instance as at most three neighbors). This shows NP-hardness of k-IR-Invitations with symmetric social relationships when $\alpha = 0$ and $\beta \geq 3$. Other cases require some modifications to our reduction, which we omit due to space. □

Theorem 12. *When agents have symmetric social relationships, k-IR-Invitations is in FPT if $\beta \leq f(k)$ where $f(k)$ can be an arbitrary function of k.*

Proof. As before, without loss of generality, assume $k \in S_i$ for all $a_i \in N$. We first create a *friend graph* in which nodes represent agents, and we create an edge between two nodes if their corresponding agents are friends. Clearly, subsets of nodes in this graph and invitations have one-to-one correspondence. In the friend graph, it is clear that all or none agents in each component should be chosen to form an IR invitation. Thus, if any connected component contains two nodes whose corresponding agents are enemies of each other, then we can safely remove the component from the graph (as it cannot be included in any IR invitation). Likewise, if any component contains more than k agents, we can remove the component as well.

We then create an *enemy graph* in which nodes represent connected components in the friend graph. Each node in the enemy graph has a weight that is equal to the size of the component it represents, and we create an edge between two nodes if their corresponding components contain a pair of enemies (one agent in each component). Because each agent has at most β enemies, each node in the enemy graph has at most $k \cdot \beta$ edges. Notice that an independent set in the enemy graph represents an IR invitation in the original instance.

Similarly to the FPT algorithm given in proof of Theorem 3, we use Color Coding to color each node in the enemy graph as $\{0, 1\}$ with equal probability. If there is any edge in the enemy graph whose both end-points (components) are of color 1, then we re-color both of them as 0. We repeat this process until no such pair exists (which can be done in linear time by scanning through the edges). After this step, it is clear that all nodes of color 1 form an independent set; we can easily determine if a subset of nodes whose weight is k exists, using a knapsack-like algorithm.

Provided that an IR invitation of size k exists, this algorithm's probability of success is at least $(1/2^k) \cdot (1/2^{k\beta}) \geq 1/2^{k(1+f(k))}$. For any fixed IR invitation I^* of size k, we color all agents in I^* as color 1 with probability $1/2^k$, and with probability at least $1/2^{k\beta}$ we color the union of enemies of all agents in I^* as color 0. Regardless of coloring of all other agents, this coloring will ensure that all agents in I^* remain to be of color 1 in the enemy graph, and thus our algorithm can find I^* (or some other solution).

The overall runtime of our algorithm is $O(f(k)n)$ as all sub-routines can be implemented in linear time in size of each graph and each graph contains at most $O(n)$ nodes and $O(f(k)n)$ edges. We can repeat this randomized algorithm $2^{k(1+f(k))} \ln n$ times to increase the probability of success to $1 - 1/n$ (with overall runtime $2^{k(1+f(k))}(f(k)n \ln n)$). This algorithm can also be de-randomized using a family of k-

perfect hash functions (Alon, Yuster, and Zwick 1995). □

Lastly we show that k-IR-Invitations remains to be W[1]-complete, even under symmetric social relationships, when β is not bounded. Proof of W[1]-hardness is similar to that of Theorem 4, and completeness follows from Theorem 2. We omit proof of Theorem 13.

Theorem 13. *When agents have symmetric social relationships, k-IR-Invitations is W[1]-complete if β is not bounded above by any function of k.*

4.2 Symmetric k-Stable-Invitations

Interestingly, complexity of k-Stable-Invitations and that of k-IR-Invitations are identical, except when β is unbounded, if we assume symmetric social relationships. This implies that the combinatoric complexity due to social relationships plays an important role in SIP, and restrictions on social relationships (such as symmetry) can substantially reduce the complexity. Yet we emphasize that both polytime and FPT algorithms for k-Stable-Invitations are much more complicated than those for k-IR-Invitations, and much of its complexity is due to the additional requirement that uninvited agents must not be willing to attend.

We first present classical complexity results for k-Stable-Invitations under symmetric social relationships, followed by parameterized complexity results.

Theorem 14. *When agents have symmetric social relationships, k-Stable-Invitations can be solved in polynomial time if (i) $\beta = 0$, (ii) $\beta = 1$ and $\alpha \leq 1$, or (iii) $\beta = 2$ and $\alpha = 0$. Otherwise, the problem is NP-hard.*

Proof sketch. Let us first consider the case (i) when $\beta = 0$. We construct a friend graph as before, and find connected components in this graph. Any stable invitation must contain all or none of nodes in each connected component. For each connected component, we check two things: Whether a stable invitation can contain all of nodes in the component and whether it can contain none of nodes in it. To check the first, we simply check if the component is of size k or less and if everyone in the component approves size k. To check the second, we check if the component contains two or more nodes (then we can leave them out) or if it is a singleton component but the only agent in it does not approve size k (then we can leave the agent out). All of these checks can be done in linear time. Now we can use a dynamic programming algorithm to determine whether a subset of connected components that contains k nodes over all such that every component (whether selected or not) does not violate the stability conditions (which can be easily checked by the two conditions we processed earlier).

Our reductions for k-IR-Invitations in proof of Theorem 11 show NP-hardness for k-Stable-Invitations as well, because agents only approve invitations of size k in our reduction; it ensures that any uninvited agent would be unwilling to attend due to the size of an invitation. □

Theorem 15. *When agents have symmetric social relationships, k-Stable-Invitations is in FPT if $\beta \leq f(k)$ where $f(k)$ can be an arbitrary function of k.*

Proof sketch. The main idea is similar to that of our proof of Theorem 12, but we need an original idea to deal with stability conditions. As before, we proceed by creating a *friend graph*, removing certain connected components, creating an *enemy graph*, coloring components using two colors $\{0, 1\}$, and re-coloring any adjacent nodes of color 1 to color 0, as before. After the re-coloring step, any subset of nodes of color 1 forms an independent set in the enemy graph (and yields an IR invitation).

Hence we only need to worry about stability conditions while choosing a subset of nodes of color 1 to find a stable invitation. Doing so will exclude all nodes of color 0, but only the singleton nodes (i.e., agents with no friends) may violate stability condition. To avoid this, we are going to choose at least one enemy (of color 1) of each singleton node of color 0 who approves size $k + 1$ in brute-force manner; the search space is bounded because we can only choose up to k agents and each agent has at most $f(k)$ enemies (i.e., the search space is $O((f(k))^k)$). This pre-selection process is the crucial step in our algorithm (and where we need the condition that β is bounded). The rest of the algorithm is straightforward, and is omitted. □

Lastly, k-Stable-Invitations remains to be W[2]-complete, even under symmetric social relationships, when β is not bounded. W[2]-hardness is proved by the reduction for Theorem 10, and completeness follows from Theorem 6.

Theorem 16. *When agents have symmetric social relationships, k-Stable-Invitations is W[2]-complete if β is not bounded above by any function of k.*

5 Discussion and Future Work

In this work we investigated the parameterized complexity of the Stable Invitations Problem (SIP) for two different solution concepts – individual rationality and (Nash) stability, when the size of a solution is parameterized. We considered restrictions on inputs by limiting the number of friends and enemies each agent can have, and also studied the special case in which all agents have symmetric social relationships. Despite the fact that the majority of the problems we consider in this work are NP-hard, we showed that many special cases of the problem admit efficient FPT algorithms. Our results indicate that the computational complexity of SIP varies when its input is restricted or the solution concept changes, which is not distinguishable under the classic complexity. Our work leaves a few interesting open problems for future work. Lee and Shoham (2015) considered another solution concept in which agents who are not invited are not envious of those who are invited (motivated by 'envy-freeness'). It would be interesting to analyze the parameterized complexity of finding an envy-free invitation of size k. In addition, analyzing the parameterized complexity of the Group Activity Selection Problem (Darmann et al. 2012) is another interesting direction for future work.

Proofs Missing details of proof sketches and omitted proofs can be found in the extended version.

Acknowledgements This work was funded in part by the National Science Foundation (grant IIS-1347214), AFOSR MURI, and the Kwanjeong Educational Foundation.

References

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

Aziz, H., and Brandl, F. 2012. Existence of stability in hedonic coalition formation games. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, 763–770.

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

Bogomolnaia, A., and Jackson, M. O. 2002. The stability of hedonic coalition structures. *Games and Economic Behavior* 38(2):201–230.

Creignou, N., and Vollmer, H. 2015. Parameterized complexity of weighted satisfiability problems: Decision, enumeration, counting. *Fundamenta Informaticae* 136(4):297–316.

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

Downey, R. G., and Fellows, M. R. 2012. *Parameterized complexity*. Springer Science & Business Media.

Dreze, J. H., and Greenberg, J. 1980. Hedonic coalitions: Optimality and stability. *Journal of the Econometric Society* 987–1003.

Garey, M. R.; Johnson, D. S.; and Stockmeyer, L. 1976. Some simplified NP-complete graph problems. *Theoretical Computer Science* 1(3):237–267.

Impagliazzo, R., and Paturi, R. 1999. Complexity of k-SAT. In *Computational Complexity, 1999. Proceedings. Fourteenth Annual IEEE Conference on*, 237–240. IEEE.

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