

## A Structural Approach to Activity Selection

Eduard Eiben<sup>1</sup>, Robert Ganian<sup>2</sup>, Sebastian Ordyniak<sup>3</sup>

<sup>1</sup> Department of Informatics, University of Bergen, Norway

<sup>2</sup> Algorithms and Complexity group, TU Wien, Austria

<sup>3</sup> Algorithms group, University of Sheffield, UK

eduard.eiben@uib.no, rganian@gmail.com, sordyniak@gmail.com

### Abstract

The general task of finding an assignment of agents to activities under certain stability and rationality constraints has led to the introduction of two prominent problems in the area of computational social choice: Group Activity Selection (GASP) and Stable Invitations (SIP). Here we introduce and study the Comprehensive Activity Selection Problem, which naturally generalizes both of these problems. In particular, we apply the parameterized complexity paradigm, which has already been successfully employed for SIP and GASP. While previous work has focused strongly on parameters such as solution size or number of activities, here we focus on parameters which capture the complexity of agent-to-agent interactions. Our results include a comprehensive complexity map for CAS under various restrictions on the number of activities in combination with restrictions on the complexity of agent interactions.

### 1 Introduction

Imagine a host who is in charge of organizing an event for a larger number of people. At the beginning, the host might want to select which people to invite for his event. Later on, he or she might need to assign the attendees to groups: for instance, some participants might want to play a game of football while others would like to play poker, or later on the host may need to make table assignments for dinner. In all of these settings, the problem we are dealing with is to find an assignment of a certain set of *agents* to one or several *activities* subject to certain rationality and/or stability conditions given by the agents. This setting has motivated the introduction of two distinct NP-complete problems (see [Darmann and Lang, 2017] for a survey on group activity selection):

- In the STABLE INVITATIONS PROBLEM (SIP), we are given a set of agents, each of which has a list of *friends*, a list of *enemies*, and a set of natural numbers called the *approval set*. The task is to select a set of *satisfied* agents for a single activity, where an agent is satisfied if all of its friends are in the activity, none of its enemies are in the activity, and the size of the activity is in its approval set. The stability condition most frequently used here is based on *Nash stability*:

for each agent that is not selected, adding him to the activity would violate at least one of this agent’s three criteria for being satisfied. SIP was introduced by Lee and Shoham (2015) and studied by Lee and Williams (2017a).

- In the GROUP ACTIVITY SELECTION PROBLEM (GASP), we are given a set of agents along with a set of activities, and for each agent we get a separate *approval set* for every activity. The task is to assign agents into activities so that each agent is satisfied, meaning that each agent is in an activity whose size is in the agent’s approval set (for that activity). Several stability conditions have been defined and used for this problem, including Nash stability. GASP was introduced by Darmann et al. (2012; 2017b) and has been studied in various settings [Lee and Williams, 2017b; Darmann et al., 2017a; Darmann, 2015; Igarashi et al., 2017a; 2017c; Gupta et al., 2017].

This paper studies the problem underlying both SIP and GASP in its full generality. In particular, we introduce the COMPREHENSIVE ACTIVITY SELECTION PROBLEM (CAS), where we are given a set of agents along with their friends and enemies (as in SIP), a set of activities (as in GASP), and approval sets which specify the size constraints for each agent and activity. The stability conditions for CAS can be translated directly from GASP and SIP; as in previous work, here we focus primarily on Nash stability and EF (envy-freeness). Hence our task in the COMPREHENSIVE ACTIVITY SELECTION PROBLEM is to assign agents to activities so that each agent is satisfied in that activity (its friends are there, its enemies are missing, and it approves of its size), and agents that are not assigned to any activity have a reason not to want to join any activity (meaning a friend is missing there, an enemy is there, or the size would not be satisfactory). The advantage of CAS is that it allows us to capture more complex real-life situations than either of SIP or GASP on their own; aside from the aforementioned example of event organization, numerous applications come to mind in the context of social networks where information about friends (and sometimes also enemies) are readily available.

While both SIP and GASP are NP-hard, it is easy to see that the problems become polynomial-time tractable when one is only interested in assigning at most a constant number  $c$  out of  $n$  agents ( $c$  is then called the *solution size*). However, the running time of a naive algorithm would be roughly  $\mathcal{O}(n^c)$ , which is far from practi-

cal. Recent works [Lee and Williams, 2017a; 2017b; Igarashi *et al.*, 2017a; Gupta *et al.*, 2017] employed the *parameterized complexity paradigm* [Downey and Fellows, 2013; Cygan *et al.*, 2015] to obtain a more refined complexity analysis of the problem in the case of selecting only few agents. In parameterized complexity, one considers inputs which are associated with a numerical parameter  $k$ , and the central question is whether the (parameterized) problem can be solved in time  $f(k) \cdot n^{\mathcal{O}(1)}$ ; algorithms with this running time are called *fixed-parameter algorithms*, and parameterized problems that admit such algorithms are called *fixed parameter tractable* (FPT). For instance, Lee and Williams (2017a; 2017b) showed that neither GASP nor SIP are FPT when parameterized by solution size (however, they do become FPT under additional assumptions on the input); for GASP, parameters such as number of agents and number of activities have also been considered [Igarashi *et al.*, 2017a; Gupta *et al.*, 2017].

**Contribution.** In line with previous work on GASP and SIP, we also study CAS through the lens of parameterized complexity, but with a twist. Instead of focusing on solution size, we turn our attention to the development of algorithms which exploit the structure of agent interactions captured in terms of an *association graph*, which provides a concise representation of agent interactions in terms of friend and enemy relations. This approach has two main advantages: it circumvents the strong lower bounds obtained by Lee and Williams (2017a; 2017b) when parameterizing by solution size, and leads to algorithms which can perform well even for instances with large solution sizes. Previously, structural parameters have only been considered for a variant of GASP with connectivity constraints [Igarashi *et al.*, 2017c; 2017a; Gupta *et al.*, 2017].

We set the stage by identifying the limits of how far graph structure can take us for a problem as complex as CAS. Since GASP is NP-hard, it is obvious that CAS will in general remain NP-hard even on edgeless association graphs; in other words, there is little hope of solving CAS when the number of activities is unbounded. Hence we restrict our attention to three restrictions of the problem:

- the (base) case where the number of activities is a fixed constant  $c$  (denoted  $c$ -CAS),
- the (more general) case where the number of activities is upper-bounded by the parameter (denoted CAS\*),
- the case where the number of activities is upper-bounded by the parameter but additional restrictions are placed on the approval sets (denoted SIMPLE-CAS\*).

In particular, the restrictions in SIMPLE-CAS\* require that approval sets must be *continuous* and that there are only few *types of agents* (see Section 3 for an exact definition); we note that both of these restrictions were introduced and motivated already in the original paper by Darmann *et al.* (2012). As a warmup, we show that  $c$ -CAS admits a simple polynomial-time algorithm on edgeless association graphs. Somewhat surprisingly, we then show that CAS\* is W[1]-hard even on edgeless association graphs, meaning that it is not FPT under well-established complexity assumptions. Note that this hardness result is not covered by [Igarashi *et al.*, 2017b, Theorem 2], since there hardness is shown for a more expressive

	$c$ -CAS	SIMPLE-CAS*	CAS*
edgeless	P (Fact 1)	FPT (Thm. 4)	W[1]-h. (Thm. 2)
vcn	FPT (Thm. 5)	FPT (Thm. 8)	W[1]-h. (Thm. 2)
treewidth	FPT (Thm. 5)	W[1]-h. (Thm. 7)	W[1]-h. (Thm. 2)

Table 1: CAS through the lens of structural parameters. All cases that are W[1]-hard can be solved in polynomial time when the parameters are fixed constants.

version of GASP and specifically the constructed instance uses this additional expressive power. On the other hand, the more restricted setting of SIMPLE-CAS\* allows us to obtain a fixed-parameter algorithm on edgeless association graphs.

Having dealt with the base case of edgeless graphs, we proceed to the graph parameter *treewidth* [Robertson and Seymour, 1984]. Treewidth is, by far, the most prominent structural parameter and has been extensively used in many diverse areas of computer science [Bodlaender, 2008; Samer and Szeider, 2010; Ganian *et al.*, 2017; Eiben *et al.*, 2016]. Here we obtain a fixed-parameter algorithm for  $c$ -CAS parameterized by the treewidth of the association graph. As for CAS\*, the aforementioned lower bound on edgeless graphs rules out the fixed-parameter tractability of the problem parameterized by treewidth. However, we show that (unlike the case of edgeless association graphs), SIMPLE-CAS\* becomes W[1]-hard parameterized by treewidth.

The results obtained so far beg the following question: can one lift the fixed-parameter tractability of SIMPLE-CAS\* from edgeless graphs to graphs exhibiting a more restrictive form of structure than treewidth? We answer this affirmatively, in particular by parameterizing by *vertex cover number* ( $vcn$ ). The vertex cover number (i.e., the size of a minimum vertex cover) has been successfully used to obtain a number of fixed-parameter algorithms for notoriously difficult problems [Lampis, 2012; Fellows *et al.*, 2008]. Our results are summarized in Table 1.

Our final result revisits the setting of Lee and Williams (2017a) and provides a spark of hope after the strong negative results obtained for SIP. In particular, it was shown by these authors that SIP is W[1]-hard parameterized by solution size even if each agent has at most 1 friend and 1 enemy. Interestingly, the class of association graphs of such instances can still be “dense” in the sense of being *somewhere dense* [Nešetřil and de Mendez, 2012]; for example, one can easily obtain an association graph that is a subdivision of a clique. On the other hand, while we do expect realistic instances to include agents with more than one friend and/or enemy, many real-world networks were shown to be nowhere dense and in particular exhibit bounded expansion [Reidl, 2016; Demaine *et al.*, 2014]. We prove that CAS\* (a generalization of SIP) is FPT parameterized by solution size when the association graphs are nowhere dense.

## 2 Preliminaries

For an integer  $i$ , we let  $[i] = \{1, 2, \dots, i\}$ . We refer to the handbook by Diestel (2012) for standard graph terminology. We also refer to the standard textbooks [Downey and Fellows, 2013; Cygan *et al.*, 2015] for basic notions in *parameterized*

complexity, such as: *fixed-parameter algorithms*, *XP algorithms*, *treewidth*, *nice tree-decompositions* and the complexity classes FPT, XP, W[1]. The *vertex cover number* of a graph  $G$  is the minimum size of a vertex cover in  $G$ .

**The Comprehensive Activity Selection Problem (CAS).** In CAS, we are given: a set of agents  $V = \{v_1, \dots, v_n\}$ , a set of activities  $A = \{a_1, \dots, a_p\}$ , and for each agent  $v_i \in V$ , a *portfolio*  $\Gamma_i = (F_i, E_i, S_i)$  where  $F_i \subseteq V$  is the *friend set*,  $E_i \subseteq V$  is the *enemy set* and  $S_i \subseteq A \times [n]$  is the *approval set*. We let  $\mathcal{P} = \{\Gamma_i \mid i \in [n]\}$  denote the set of all portfolios.

For  $i \in [n]$ ,  $j \in [p]$  we use  $S_i(a_j)$  to denote the set of all admissible sizes of activity  $a_j$  for agent  $v_i$ , i.e.,  $\ell \in S_i(a_j)$  if and only if  $(a_j, \ell) \in S_i$ . An *assignment*  $\tau$  (also called *outcome* in the literature) is a mapping from  $V$  to  $A^* = A \cup \{a_0\}$ , where  $a_0$  is the “void activity” (i.e., agents assigned to  $a_0$  are interpreted as not participating in any of the activities in  $A$ ). For brevity, for an assignment  $\tau$  we will denote the set of agents assigned by  $\tau$  to activity  $a_j$  simply as  $\tau^{-1}(a_j)$ .

An *assignment* is *individually rational* (IR) if each agent is willing to participate; formally,  $\tau$  is IR iff  $\forall i \in [n]$  with  $a = \tau(v_i)$  and  $a \in A$  it holds that  $F_i \subseteq \tau^{-1}(a)$ ,  $E_i \cap \tau^{-1}(a) = \emptyset$ , and  $|\tau^{-1}(a)| \in S_i(a)$ . We note that the empty assignment is always IR, because no friends and enemy conditions are required for agents assigned to  $a_0$ . An *assignment*  $\tau$  is (*Nash*) *stable* if it is IR and furthermore no agent assigned to  $a_0$  would be willing to join an activity; formally,  $\forall i$  such that  $\tau(v_i) = a_0$  and  $\forall j \in [p]$ , at least one of the following must be true:  $F_i \not\subseteq \tau^{-1}(a_j)$ ,  $E_i \cap \tau^{-1}(a_j) \neq \emptyset$ , or  $|\tau^{-1}(a_j)| + 1 \notin S_i(a_j)$ . An *assignment*  $\tau$  is *envy-free* (EF) if it is IR and furthermore no agent assigned to  $a_0$  would be willing to join an activity *instead of another agent*; the formal definition is nearly identical to that of stability, with a slight modification to the third condition.

Finally, an assignment is *stable-EF* if it is stable and envy-free. We note that our algorithms will be able to solve CAS regardless of which of the above stability conditions are desired, and so we will assume that each CAS instance also includes a specification of which stability conditions are required (i.e., a value  $\iota \subseteq \{\text{stable}, \text{EF}\}$ ). A *solution* to a CAS instance  $(V, A, \mathcal{P}, \iota)$  is an assignment which satisfies  $\iota$ .

Given a CAS instance  $I$ , we will use  $G_I$  to denote the *association graph* of  $I$ , which provides a concise representation of agent interactions in terms of friend and enemy relations. The vertex set of  $G_I$  is the set  $V$  of agents in  $I$ , and two agents  $v_i, v_j$  are connected by an edge if  $v_i \in E_j \cup F_j$  or  $v_j \in E_i \cup F_i$  (or both).

Let us now briefly compare CAS to the previously studied GROUP ACTIVITY SELECTION PROBLEM (GASP) [Lee and Williams, 2017b] and STABLE INVITATIONS PROBLEM (SIP) [Lee and Williams, 2017a]. In particular, SIP is the restriction of CAS to the case where  $|A| = 1$ , and GASP is the restriction of CAS to the case where  $\forall v_i \in V : F_i = E_i = \emptyset$ . This implies that, on their own, neither restrictions on the structure of the association graph nor restricting the number of activities can lead to efficient algorithms for CAS. Our focus lies on showing that a suitable combination of both types of restrictions does, however, allow us to obtain efficient solutions for the problem. We now formally define *c-CAS*.

*c-CAS*

Input:  $V = \{v_1, \dots, v_n\}$ ,  $A = \{a_1, \dots, a_c\}$ ,  
 $\mathcal{P} = \{\Gamma_1, \dots, \Gamma_n\}$ ,  $\iota \subseteq \{\text{stable}, \text{EF}\}$ .  
 Task: Find an assignment that satisfies  $\iota$ .

CAS\* is defined similarly as *c-CAS*; the only difference is that instead of the number of activities being a fixed constant  $c$ , instead it is a parameter  $p$ . The third variant of interest, SIMPLE-CAS\*, will be introduced in the next section.

### 3 The Base Case: no Friends, no Enemies

This section sets the stage for our later results by showing exactly how far graph structure can take us for *c-CAS* and CAS\*; in particular, we will focus on the case where the association graph is edgeless. We note that this problem precisely corresponds to GASP viewed through the lens of the number of activities, and each new result obtained in this section can be equivalently stated as a new result for GASP. We begin by mentioning an already known result.

**Fact 1** ([Darmann *et al.*, 2012]). *c-CAS can be solved in time  $(n+1)^c n^{\mathcal{O}(1)}$  if the association graph is edgeless.*

Next, we show that the above result cannot be improved to a fixed-parameter algorithm for CAS\*; we note that this was left open in the previous work of Lee and Williams (2017b).

**Theorem 2.** *CAS\* is W[1]-hard, even on instances with an edgeless association graph.*

*Sketch of Proof.* We give a parameterized reduction from the MULTICOLORED CLIQUE problem (MCC), which is well-known to be W[1]-complete: Given a  $k$ -partite graph  $G = (V, E)$  with  $V = \{V_1, \dots, V_k\}$ , determine whether it contains a  $k$ -clique. We denote by  $E_{i,j}$  the set of edges of  $G$  with one endpoint in  $V_i$  and one endpoint in  $V_j$ . W.l.o.g. we assume that each  $V_i$  has size  $n$  and each  $E_{i,j}$  has size  $m$ .

We will first give a formal construction and then provide the underlying intuition. To simplify notation in our setting, we assume our CAS\* instance is given by a set  $N$  of agents, a set  $A$  of activities, and for every  $n \in N$  an approval set  $S_n \subseteq A \times \mathbb{N}$ . Given an instance of MCC, we construct an instance  $I = (N, A, (S_n)_{n \in N})$  of GASP as follows. The set  $A$  contains the following activities: One activity  $a$ , for every  $i$  with  $1 \leq i \leq k$  three activities  $a_i, a'_i$ , and  $c_i$ , and for every  $i$  and  $j$  with  $1 \leq i < j \leq k$  the activity  $a_{i,j}$ .

For every  $i$  and  $j$  with  $1 \leq i < j \leq k$  let  $\alpha_i$  be a bijection from  $V_i$  to the set  $\{1, 3, \dots, 2n-1\}$  and similarly let  $\alpha_{i,j}$  be a bijection from  $E_{i,j}$  to the set  $\{1, 3, \dots, 2m-1\}$ . We are now ready to define the set  $N$  of agents. Apart from an agent  $v$  with approval set  $S_v = \{(a, 1), (a, 3)\}$ , we have the following agents for every  $i$  with  $1 \leq i \leq k$  and every  $v \in V_i$ :

- one agent  $f_i^v$  whose approval set is given as the union of the sets  $\{(a_i, \alpha_i(v')) \mid v' \in V_i \wedge \alpha_i(v') \geq \alpha_i(v)\}$ ,  $\{(a_{i,j}, \alpha_{i,j}(e) + 1) \mid j > i \wedge e \in E_{i,j} \wedge v \in e\}$ , and  $\{(a_{j,i}, \alpha_{j,i}(e) + 1) \mid j < i \wedge e \in E_{j,i} \wedge v \in e\}$ .
- one agent  $\bar{f}_i^v$  whose approval set is given as the union of the sets  $\{(a_i, \alpha_i(v')) \mid v' \in V_i \wedge \alpha_i(v') > \alpha_i(v)\}$ ,  $\{(c_i, n+1)\}$ , and  $\{(a, 2)\}$ .

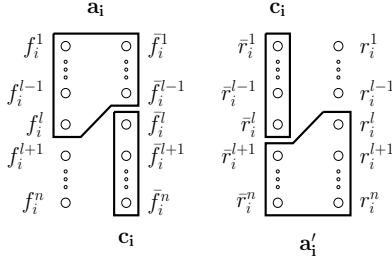


Figure 1: An illustration of a stable assignment for the agents in  $F(i) \cup R(i)$  for the case that vertex  $l$  is chosen to be in the  $k$ -clique—we assume that the vertices in  $V_i$  are numbered from 1 to  $n$ . The boxes indicate the distribution of the agents into activities, i.e., two agents contained in the same box are assigned to the same activity, which is given as the label of the box. Agents that are not contained in any box are assigned to the individual activity  $a_0$ . The picture also illustrates the sets  $F(i, l)$  and  $R(i, l)$ , which correspond to the set of all agents contained in the box titled  $a_i$  and  $a'_i$ , respectively.

- one agent  $r_i^v$  whose approval set is given as the union of the sets  $\{(a'_i, \alpha_i(v') \mid v' \in V_i \wedge \alpha_i(v') \geq 2n - 1 - \alpha_i(v) + 1\}$ ,  $\{(a_{i,j}, \alpha_{i,j}(e) + 1) \mid j > i \wedge e \in E_{i,j} \wedge v \in e\}$ , and  $\{(a_{j,i}, \alpha_{j,i}(e) + 1) \mid j < i \wedge e \in E_{j,i} \wedge v \in e\}$ .
- one agent  $\bar{r}_i^v$  whose approval set is given as the union of the sets  $\{(a'_i, \alpha_i(v') \mid v' \in V_i \wedge \alpha_i(v') > 2n - 1 - \alpha_i(v) + 1\}$ ,  $\{(c_i, n + 1)\}$ , and  $\{(a, 2)\}$ .
- for every  $i$  and  $j$  with  $1 \leq i < j \leq k$ , a set  $N_{i,j}$  of  $2m - 1$  agents with approval sets  $\{(a_{i,j}, 1), (a_{i,j}, 3), \dots, (a_{i,j}, \max\{2m - 1, 4n + 1\})\}$ .

For  $i$  and  $v$  with  $1 \leq i \leq k$  and  $v \in V_i$ , we denote by  $F_f(i)$ ,  $F_h(i)$ ,  $F(i)$ ,  $R_r(i)$ ,  $R_h(i)$ ,  $R(i)$ ,  $F(i, v)$ , and  $R(i, v)$  the sets of agents  $\{f_i^w \mid w \in V_i\}$ ,  $\{\bar{f}_i^w \mid w \in V_i\}$ ,  $F_f(i) \cup F_h(i)$ ,  $\{r_i^w \mid w \in V_i\}$ ,  $\{\bar{r}_i^w \mid w \in V_i\}$ ,  $R_r(i) \cup R_h(i)$ ,  $\{f_i^w, \bar{f}_i^w \mid w \in V_i \wedge \alpha_i(w) < \alpha_i(v)\} \cup \{f_i^v\}$ , and  $\{r_i^w, \bar{r}_i^w \mid w \in V_i \wedge \alpha_i(w) > \alpha_i(v)\} \cup \{r_i^v\}$ , respectively. See Figure 1 for an illustration of  $F(i, v)$  and  $R(i, v)$ .

This completes the construction of the instance  $I$ , and we conclude the proof sketch by outlining the main idea behind the reduction. First, using the approval set of the agents in  $N_{i,j}$  it is ensured that the size of the activity  $a_{i,j}$  in any stable assignment  $\pi$  for  $I$  indicates the edge from  $E_{i,j}$  that is chosen to be in the  $k$ -clique. More precisely, a size of  $s = |\pi^{-1}(a_{i,j})|$  for activity  $a_{i,j}$  corresponds to the edge  $\alpha^{-1}(s)$ . Moreover, for every  $i$  the approval sets for the agents in  $F(i) \cup R(i)$  ensure that there is an  $v \in V_i$  (with  $v$  corresponding to the vertex in  $V_i$  in the  $k$ -clique) such that all agents in  $F(i, v)$  are assigned to activity  $a_i$ , all agents in  $R(i, v)$  are assigned to activity  $a'_i$ , and all agents in  $F_h(i) \cup R_h(i)$  that have not yet been assigned are in activity  $c_i$  (see Figure 1 for an illustration of such an assignment).  $\square$

Observe that the above hardness result strongly relies on the fact that the approval set  $S_i$  of an agent  $v_i$  can be arbitrarily complex. However, we may often encounter situations where we only have limited information about our agents: for instance, their approval sets could be obtained through a questionnaire, and it is unlikely that people will fill in irregular sets of numbers with gaps for individual activities. Alternatively, in the context of social networks, approval sets could

be obtained automatically based on certain limited information we have about users. To model these more realistic settings with better fidelity, we will consider a version of CAS\* where there are additional restrictions on the approval sets of agents: agents can only choose from a limited number of approval sets, and the approval sets consist of a consecutive set of integers for each activity. Both of these restrictions have already been considered in the classical setting by Darmann et al. (2012). We formalize below.

**SIMPLE-CAS\*** Parameter:  $p + q$   
 Input:  $V = \{v_1, \dots, v_n\}$ ,  $A = \{a_1, \dots, a_p\}$ ,  $\mathcal{P} = \{\Gamma_1 = (F_1, E_1, S_1), \dots, \Gamma_n\}$ ,  $\iota \subseteq \{\text{stable}, \text{EF}\}$ .  
 Promise: There exist at most  $q$  approval sets which are pairwise distinct, and for each approval set  $S_i$  and each activity  $a_j$  the set  $S_i(a_j)$  must be consecutive.  
 Task: Find an assignment that satisfies  $\iota$ .

We note that GASP is known to be NP-hard even if all approval sets are consecutive [Darmann et al., 2012]. In any instance of SIMPLE-CAS\*, agents can be partitioned into at most  $q$  agent-types  $Q_1, \dots, Q_q$ , where agents  $v_i, v_j$  belong to the same agent-type iff  $S_i = S_j$ . Let  $\mathcal{Q}$  be the set of all agent-types and  $S_i^{\mathcal{Q}}$  be the approval set of the agents in  $Q_i$ .

Our last goal for this section will be to show that, unlike CAS\*, SIMPLE-CAS\* is in fact FPT on edgeless graphs. In order to do so, we will first introduce and solve an auxiliary problem, which we call RESTRICTED TOKEN ASSIGNMENT. Our algorithm for SIMPLE-CAS\* will later rely on making calls to a subroutine solving this problem.

#### RESTRICTED TOKEN ASSIGNMENT

Input: A set  $V = \{v_1, \dots, v_n\}$  of tokens, a set  $A = \{a_0, \dots, a_m\}$  of bins, and a specification  $\beta$  where  $\beta(a_j)$  is an interval and  $\beta(v_i)$  is a subset of  $A$ .  
 Task: Find an assignment  $\tau$  from  $V$  to  $A$  s.t. (T1) for each  $v_i \in V$ ,  $\tau(v_i) \in \beta(v_i)$ , and (T2) for each  $a_j \in A$ ,  $|\tau^{-1}(a_j)| \in \beta(a_j)$ .

**Lemma 3.** RESTRICTED TOKEN ASSIGNMENT can be solved in time  $\mathcal{O}((n + m)^4 \cdot \log(n + m))$ .

*Sketch of Proof.* The problem can be reduced to the CIRCULATION problem, a generalization of NETWORK FLOW; the run-time follows from the result of Orlin (1988).  $\square$

**Theorem 4.** SIMPLE-CAS\* can be solved in time  $2^{\mathcal{O}(q(p+1))} \cdot n^4 \log n$  if the association graph is edgeless.

*Sketch of Proof.* We begin by exhaustively branching over all possibilities of which agent-types are compatible with which activities based on their size. Formally, let the compatibility set  $C_i$  of activity  $a_i$  be an element of  $2^{\mathcal{Q}}$ , and let us branch over the at most  $(2^q)^{p+1}$  possible compatibility sets for each activity in  $A^*$ . For the following, let us fix one particular choice of compatibility sets  $C_0, C_1, \dots, C_p$ . We interpret our compatibility sets as the following assumptions:

- for  $i > 0$ , activity  $a_i$  has compatibility set  $C_i \neq \emptyset$  means that only agents from  $C_i$  can participate in activity  $a_i$  (i.e., the size of  $a_i$  lies in the approval set of  $C_i$ );

- for activity  $a_0$ ,  $C_0$  contains all agent-types which will contain *at least one* agent that is assigned to  $a_0$  (this is important due to the stability conditions).

Next, we will check that the chosen compatibility sets are valid: for each activity  $a_i \in A$  such that  $C_i \neq \emptyset$ , we compute a *pre-restriction*  $\alpha'_i = \bigcap_{Q_j \in C_i} S_j^Q(a_i)$ . Observe that  $\alpha'_i$  contains all sizes of  $a_i$  which are consistent with our assumption that all agents from  $C_i$  could participate in  $a_i$ . Our third task is to ensure that the stability conditions are not violated. Depending on  $\iota$ , we will prune each  $\alpha'_i$  so as to not invalidate our stability criteria in  $\iota$  (this has to be done for each agent-type in  $C_0$ ). If  $C_i \neq \emptyset$  but  $\alpha'_i = \emptyset$  then this means we have made an invalid assumption and we terminate our current branch. Observe that at this stage,  $\alpha'_i$  may consist of up to  $q + 1$  consecutive segments. So, we apply a second round of branching: for each  $\alpha'_i$ , we branch over its at most  $q + 1$  consecutive segments and denote the chosen segment  $\alpha_i$ . Let us consider a specific branch  $\eta$ , corresponding to a choice of compatibility sets and consecutive segments  $\alpha_1, \dots, \alpha_p$ .

Finally, for each branch  $\eta$ , we construct an instance  $\mathcal{I}_\eta$  of RESTRICTED TOKEN ASSIGNMENT as follows. The set of bins in  $\mathcal{I}_\eta$  is the set of activities including  $a_0$ . For each agent-type  $Q_j \in \mathcal{Q}$  such that  $Q_j \notin C_0$ , we proceed as follows: for each  $v \in Q_j$  we create a token  $v'$  and we set  $\beta(v') = \{a_i \mid Q_j \in C_i\}$ . On the other hand, for each agent-type  $Q_j \in C_0$  we arbitrarily choose one agent  $v_0 \in Q_j$ , for each agent  $v \in Q_j \setminus \{v_0\}$  we create a token  $v'$ , and we set  $\beta(v') = \{a_i \mid Q_j \in C_i\}$ . Finally, we set  $\beta(a_0) = [0, \infty]$  and for  $i > 0$  we set  $\beta(a_i) = \alpha_i$  (or, more precisely, the interval corresponding to consecutive segment  $\alpha_i$ ). This completes the construction of  $\mathcal{I}_\eta$ . The algorithm attempts to solve each instance  $\mathcal{I}_\eta$  obtained in our branches using Lemma 3 and outputs YES iff at least one  $\mathcal{I}_\eta$  is a YES-instance.  $\square$

## 4 Algorithms and Bounds using Treewidth

Given the above baseline results, we now turn our attention to the more general setting of association graphs with low treewidth. We begin with a fixed-parameter algorithm.

**Theorem 5.** *c-CAS can be solved in time  $n^{3c+1} \cdot 2^{\mathcal{O}(c \cdot \ell)}$  if the association graph has treewidth at most  $\ell$ .*

*Sketch of Proof.* We begin by computing a nice tree-decomposition  $(T, \mathcal{X})$  of the association graph  $G$  with width  $k \leq 5\ell + 4$  using the known approximation algorithm [Bodlaender *et al.*, 2016]. Next, we branch over all at most  $n^c$ -many target sizes of the activities in  $A$ . In each branch, we denote the size of activity  $a_i$  as  $r_i$ , and our task will be to find an assignment of agents to activities such that each activity  $a_i$  receives precisely  $r_i$  agents. We now have complete information about the allowed activities for each agent.

As our third step, we solve *c-CAS* using a leaf-to-root dynamic programming procedure along  $T$ . In this procedure, for each node  $t \in V(T)$  we will compute all the possible ways the activities in  $X_t$  can be assigned together with all the possible sizes of the activities based on agents assigned up to this point in the leaf-to-root run. In addition, we will also keep information for each agent in  $X_t$  assigned to  $a_0$

specifying which activities are currently violating the stability conditions of this agent.  $\square$

**Corollary 6.** *CAS\* admits an XP algorithm when additionally parameterized by the treewidth of the association graph.*

Naturally, the biggest question left for treewidth at this point is whether one can retain the fixed parameter tractability of SIMPLE-CAS\* while generalizing the graph structure to treewidth. Surprisingly, we show that this is not the case.

**Theorem 7.** *SIMPLE-CAS\* is W[1]-hard when additionally parameterized by the treewidth of the association graph.*

*Sketch of Proof.* We give a parameterized reduction from the EQUITABLE COLORING problem: given a graph  $G$  with  $n$  vertices and an integer  $q$  such that  $n/q = r$  is an integer, partition the vertices of  $G$  into  $q$  independent sets of size precisely  $r$ . EQUITABLE COLORING is W[1]-hard when parameterized by the treewidth of  $G$  and  $q$  [Fellows *et al.*, 2011].

The starting point of our reduction is an instance  $(G, q)$  of EQUITABLE COLORING. We construct an instance  $\mathcal{I}$  of SIMPLE-CAS\* as follows. The activity set of  $\mathcal{I}$  consists of  $q$  “main” activities  $a_1, \dots, a_q$  and  $q$  “auxiliary” activities  $a_{q+1}, \dots, a_{2q}$ . For each vertex  $v \in V(G)$  of  $G$ , we add an agent  $v$  to  $\mathcal{I}$  with no friends, enemies equal to its neighbors in  $G$ , and the approval set  $S_v = \{(a_i, m) \mid 1 \leq i \leq q, 1 \leq m \leq r + 1\}$ . We then add  $2q$  additional agents to  $\mathcal{I}$ , denoted  $x_{q+1}, \dots, x_{2q}$  and  $y_{q+1}, \dots, y_{2q}$ , with no friends and no enemies such that the approval set for each  $x_i$  is  $S_{x_i} = \{(a_i, 1), (a_{i-q}, r + 1)\}$ , and the approval set for each  $y_i$  is  $S_{y_i} = \{(a_i, 2)\}$ . To conclude the construction of  $\mathcal{I}$ , let  $\iota = \{\text{stable}\}$ . Observe that  $\text{tw}(G_{\mathcal{I}}) = \text{tw}(G)$ , and we note that  $\mathcal{I}$  contains no friends. In the end, it suffices to show that  $(G, q)$  is a yes-instance if and only if  $\mathcal{I}$  is a yes-instance.  $\square$

## 5 Solving SIMPLE-CAS\* using Vertex Covers

Here we give a fixed-parameter algorithm for SIMPLE-CAS\* par. by the vertex cover number of the association graph.

**Theorem 8.** *SIMPLE-CAS\* can be solved in time at most  $q^{\mathcal{O}(\ell^2 \cdot p \cdot \log p)} \cdot n^4 \log n$ , where  $\ell$  is the vertex cover number of the association graph.*

*Sketch of Proof.* Let  $\mathcal{I}$  be the given instance of SIMPLE-CAS\*. The general strategy is to apply two levels of branching and then reduce the resulting, simplified instance to RESTRICTED TOKEN ASSIGNMENT (cf. Lemma 3).

Our first step will aim at obtaining certain restrictions on the sizes of the activities, and in this sense its goal will be similar to that in Theorem 4. However, it will be more advantageous to use a slightly different approach here than in the aforementioned theorem: instead of beginning with the compatibility sets used in Theorem 4, here we will begin by directly branching on the intervals of sizes for individual activities, as follows. For each activity  $a_i$ , we let  $U_i'$  be the tuple containing the endpoints of the approval sets (intervals) for  $a_i$  over all approval sets  $S_1^Q, S_2^Q, \dots, S_q^Q$ , in ascending order and without multiplicities. We will use  $U_i'$  to define the set  $U_i$  of “relevant size intervals” for our activity  $i$ , as follows.  $U_i$  contains: **(A)** for each number  $z \in U_i'$ , the single-element

interval  $[z, z]$ ; **(B)** for each number  $z \in U'_i$  greater than 0, the single-element interval  $[z-1, z-1]$ ; **(C)** all consecutive intervals of natural numbers surrounding the single-unit intervals added up to this point between 0 and  $n$ .

To illustrate the above definitions, consider the example where  $q = 3$  and  $S_1^Q(a_i) = [2, 10]$ ,  $S_2^Q(a_i) = [6, 15]$  and  $S_3^Q(a_i) = [15, 50]$ . Then  $U'_i = (2, 6, 10, 15, 50)$ . Subsequently, the set  $U_i$  would then contain the following intervals:  $[0], [1], [2], [3, 4], [5], [6], [7, 8], [9], [10], [11, 13], [14], [15], [16, 48], [49], [50], [51, n]$ . On an intuitive level, one may observe that (as far as approval sets are concerned) whether  $|\tau(a_i)| = 30$  or  $|\tau(a_i)| = 40$  would make no difference as far as which agent is allowed to go to activity  $a_i$  or other activities. On the other hand, the case of  $|\tau(a_i)| = 9$  may differ from the case of  $|\tau(a_i)| = 10$  due to stability: in the former case an agent in  $Q_1$  assigned to  $a_0$  could violate stability due to preferring to be in  $a_i$ , while in the latter case no agent in  $Q_1$  would prefer to be in  $a_i$ .

To conclude the first step in our algorithm, we observe that for each  $i \in [p]$ ,  $|U_i| \leq 4q + 2$ . We will exhaustively branch by choosing an interval  $\beta_i$  from each of the sets  $U_i$ , resulting in a branching factor of at most  $(4q + 2)^p$ . Next, we will exploit the structure of the vertex cover through a second level of branching: we exhaustively branch over all  $(p + 1)^\ell$  possibilities of which agents in  $X$  are assigned to which activity. The algorithm will check whether the chosen assignment of  $X$  is compatible with the selected intervals  $\beta_i$  and also assign all agents whose assignments are already fixed by these choices (e.g., due to them having friends in  $X$ ). Modulo some technicalities, this allows us to deal with all friend and enemy relations in the instance at the cost of certain additional simple constraints; at that point, it is possible to reduce the problem to RESTRICTED TOKEN ASSIGNMENT using a reduction which extends the one used in the proof of Theorem 4.  $\square$

## 6 An Algorithm for CAS\* on Sparse Graphs

The goal in our last section is to establish the fixed-parameter tractability of CAS\* parameterized by solution size when restricted to instances whose association graphs belong to a fixed nowhere dense graph class. We note that the definition of nowhere dense graph classes [Nešetřil and de Mendez, 2012] is rather technical; however, it is worth mentioning that nowhere dense graph classes include many well-known graph classes such as graphs of bounded degree and planar graphs. The main tool we will use to establish our result is the fact that model checking for first order logic is FPT when parameterized by the size of the formula on nowhere dense labeled graph classes [Grohe *et al.*, 2017]. For clarity, we provide a full statement for the problem parameterized by solution size below, and note that SIP parameterized by solution size [Lee and Williams, 2017a] is the special case where  $|A| = 1$ .

S-CAS\* Parameter:  $p + k$   
 Input:  $V = \{v_1, \dots, v_n\}$ ,  $A = \{a_1, \dots, a_p\}$ ,  
 $\mathcal{P} = \{\Gamma_1, \dots, \Gamma_n\}$ ,  $\iota \subseteq \{\text{stable}, \text{EF}\}$ .  
 Task: Find an assignment that satisfies  $\iota$  which assigns precisely  $n - k$  agents to  $a_0$ .

**Theorem 9.** S-CAS\* is in FPT when restricted to instances whose association graphs belong to a fixed nowhere dense graph class  $\mathcal{C}$ .

*Sketch of Proof.* We begin by exhaustively branching over the at most  $k^p$  possible mappings  $\theta : A \rightarrow [k]$  such that  $\sum_{a \in A} \theta(a) = k$ . Clearly, our S-CAS\* instance  $I$  is a YES-instance if and only if there exists an assignment  $\tau$  which satisfies  $\iota$  and maps precisely  $\theta(a)$  agents to each  $a \in A$  for some choice of  $\theta$ . In the following we consider one fixed choice of  $\theta$  and show how to determine the existence of  $\tau$ .

Our next step is to perform a greedy preprocessing of all our agents in order to simplify our instance (w.r.t. our choice of  $\theta$ ). First of all, we loop over each agent  $v_i \in V$  and check whether  $v_i$  could violate the stability conditions given in  $\iota$  due to the final size of an activity  $a_j$  projected by  $\theta$ ; if this is the case, we mark  $v_i$  as *j-bad*. If an assignment  $\tau$  which respects  $\theta$  assigns a bad agent  $v_i$  to  $a_0$ , then  $\tau$  must either assign an enemy of  $v_i$  to  $a_j$  or not assign a friend of  $v_i$  to  $a_j$ . Second, for each  $v_i$ , we loop over each  $a_j \in A$  and we mark  $v_i$  as *j-compatible* if  $\theta(a_j) \in S_i(a_j)$ . Note that an IR assignment can only assign agents to activity  $a_j$  if they are *j-compatible*.

Our final task is to verify the existence of an assignment  $\tau$  which assigns precisely  $k$  agents to non-void activities while respecting the restrictions imposed by the friends and enemies and ensuring that all bad vertices do not violate  $\iota$  and that only *j-compatible* agents are assigned to activity  $a_j$ . All of these properties can be checked locally on the association graph (or, more precisely, a relational structure enriching the association graph to capture friend and enemy relations along with the properties of being bad and compatible), and in particular they can be expressed by a first-order formula. The proof then follows by the result of Grohe *et al.* (2017).  $\square$

## 7 Concluding Remarks

We initiated the study of the COMPREHENSIVE ACTIVITY SELECTION PROBLEM (CAS), a natural generalization of GASP and SIP. We investigated the complexity of CAS with respect to structural restrictions on the instance, and used this perspective to obtain a number of new algorithms (which can also be used for GASP and SIP).

Our algorithmic results are motivated and complemented by the lower bounds presented in Theorem 2 and 7. We note that Theorem 2 settles the complexity of GASP parameterized by the number of activities, a question left open in previous work [Darmann *et al.*, 2012]. In the base GASP setting, this still leaves open the complexity of the problem when parameterized by the number of agent types alone; we believe this is a natural and interesting open problem in the area.

## Acknowledgments

Eduard Eiben was supported by Pareto-Optimal Parameterized Algorithms (ERC Starting Grant 715744). Robert Ganian is also affiliated with FI MU, Brno, Czech Republic.

## References

[Bodlaender *et al.*, 2016] Hans L. Bodlaender, Pål Grønås Drange, Markus S. Dregi, Fedor V. Fomin, Daniel Lok-

- shtanov, and Michal Pilipczuk. A  $c^k n$  5-approximation algorithm for treewidth. *SIAM J. Comput.*, 45(2):317–378, 2016.
- [Bodlaender, 2008] Hans L. Bodlaender. Treewidth of graphs. In *Encyclopedia of Algorithms*. Springer, 2008.
- [Cygan *et al.*, 2015] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
- [Darmann and Lang, 2017] Andreas Darmann and Jérôme Lang. *Trends in Computational Social Choice*, chapter Group Activity Selection Problems. AI Access, 2017.
- [Darmann *et al.*, 2012] Andreas Darmann, Edith Elkind, Sascha Kurz, Jérôme Lang, Joachim Schauer, and Gerhard J. Woeginger. Group activity selection problem. In *WINE 2012*, volume 7695 of *Lecture Notes in Computer Science*, pages 156–169. Springer, 2012.
- [Darmann *et al.*, 2017a] Andreas Darmann, Janosch Döcker, Britta Dorn, Jérôme Lang, and Sebastian Schneckenburger. On simplified group activity selection. In *ADT 2017*, volume 10576 of *Lecture Notes in Computer Science*, pages 255–269. Springer, 2017.
- [Darmann *et al.*, 2017b] Andreas Darmann, Edith Elkind, Sascha Kurz, Jérôme Lang, Joachim Schauer, and Gerhard Woeginger. Group activity selection problem with approval preferences. *International Journal of Game Theory*, pages 1–30, 2017.
- [Darmann, 2015] Andreas Darmann. Group activity selection from ordinal preferences. In *ADT 2015*, volume 9346 of *Lecture Notes in Computer Science*, pages 35–51. Springer, 2015.
- [Demaine *et al.*, 2014] Erik D. Demaine, Felix Reidl, Peter Rossmanith, Fernando Sánchez Villaamil, Somnath Sikdar, and Blair D. Sullivan. Structural sparsity of complex networks: Random graph models and linear algorithms. *CoRR*, abs/1406.2587, 2014.
- [Diestel, 2012] Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.
- [Downey and Fellows, 2013] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer Verlag, 2013.
- [Eiben *et al.*, 2016] Eduard Eiben, Robert Ganian, and Sebastian Ordyniak. Using decomposition-parameters for QBF: mind the prefix! In *AAAI 2016*, pages 964–970, 2016.
- [Fellows *et al.*, 2008] Michael R. Fellows, Daniel Lokshtanov, Neeldhara Misra, Frances A. Rosamond, and Saket Saurabh. Graph layout problems parameterized by vertex cover. In *ISAAC 2008*, pages 294–305, 2008.
- [Fellows *et al.*, 2011] Michael R. Fellows, Fedor V. Fomin, Daniel Lokshtanov, Frances Rosamond, Saket Saurabh, Stefan Szeider, and Carsten Thomassen. On the complexity of some colorful problems parameterized by treewidth. *Information and Computation*, 209(2):143–153, 2011.
- [Ganian *et al.*, 2017] Robert Ganian, Sebastian Ordyniak, and M. S. Ramanujan. Going beyond primal treewidth for (M)ILP. In Satinder P. Singh and Shaul Markovitch, editors, *AAAI 2017*, pages 815–821. AAAI Press, 2017.
- [Grohe *et al.*, 2017] Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. *J. ACM*, 64(3):17:1–17:32, 2017.
- [Gupta *et al.*, 2017] Sushmita Gupta, Sanjukta Roy, Saket Saurabh, and Meirav Zehavi. Group activity selection on graphs: Parameterized analysis. In *SAGT 2017*, volume 10504 of *Lecture Notes in Computer Science*, pages 106–118. Springer, 2017.
- [Igarashi *et al.*, 2017a] Ayumi Igarashi, Robert Bredereck, and Edith Elkind. On parameterized complexity of group activity selection problems on social networks. In *AAMAS 2017*, pages 1575–1577, 2017.
- [Igarashi *et al.*, 2017b] Ayumi Igarashi, Robert Bredereck, and Edith Elkind. On parameterized complexity of group activity selection problems on social networks. *CoRR*, abs/1703.01121, 2017.
- [Igarashi *et al.*, 2017c] Ayumi Igarashi, Dominik Peters, and Edith Elkind. Group activity selection on social networks. In *AAAI 2017*, pages 565–571. AAAI Press, 2017.
- [Lampis, 2012] Michael Lampis. Algorithmic meta-theorems for restrictions of treewidth. *Algorithmica*, 64(1):190–37, 2012.
- [Lee and Shoham, 2015] Hooyeon Lee and Yoav Shoham. Stable invitations. In *AAAI 2015*, pages 965–971. AAAI Press, 2015.
- [Lee and Williams, 2017a] Hooyeon Lee and Vassilevska Williams. Complexity of the stable invitations problem. In *AAAI 2017*, pages 579–585. AAAI Press, 2017.
- [Lee and Williams, 2017b] Hooyeon Lee and Virginia Vassilevska Williams. Parameterized complexity of group activity selection. In *AAMAS 2017*, pages 353–361, 2017.
- [Nešetřil and de Mendez, 2012] Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity - Graphs, Structures, and Algorithms*. Springer, 2012.
- [Orlin, 1988] James B. Orlin. A faster strongly polynomial minimum cost flow algorithm. In *STOC 1988*, pages 377–387, 1988.
- [Reidl, 2016] F. Reidl. *Structural sparseness and complex networks*. PhD thesis, RWTH Aachen, 2016.
- [Robertson and Seymour, 1984] Neil Robertson and Paul D. Seymour. Graph minors. III. planar tree-width. *J. Comb. Theory, Ser. B*, 36(1):49–64, 1984.
- [Samer and Szeider, 2010] Marko Samer and Stefan Szeider. Constraint satisfaction with bounded treewidth revisited. *J. of Computer and System Sciences*, 76(2):103–114, 2010.