

## Committee Selection with Intraclass and Interclass Synergies

**Rani Izsak**

Weizmann Institute of Science  
Rehovot, Israel  
ran.izsak@weizmann.ac.il

**Nimrod Talmon**

Weizmann Institute of Science  
Rehovot, Israel  
nimrodtalmon77@gmail.com

**Gerhard J. Woeginger**

RWTH Aachen  
Aachen, Germany  
woeginger@algo.rwth-aachen.de

### Abstract

Voting is almost never done in void, as usually there are some relations between the alternatives on which the voters vote on. These relations shall be taken into consideration when selecting a winning committee of some given multiwinner election. As taking into account all possible relations between the alternatives is generally computationally intractable, in this paper we consider classes of alternatives; intuitively, the number of classes is significantly smaller than the number of alternatives, and thus there is some hope in reaching computational tractability. We model both intraclass relations and interclass relations by functions, which we refer to as *synergy functions*, and study the computational complexity of identifying the best committee, taking into account those synergy functions. Our model accommodates both positive and negative relations between alternatives; further, our efficient algorithms can also deal with a rich class of diversity wishes, which we show how to model using synergy functions.

### Introduction

The study of multiwinner elections is quite established by now, however it is usually assumed that the alternatives (over which the voters vote) are indistinguishable; the only differences between them are imposed by the preferences of the voters. Many times, however, this is not the case; for example, consider selecting a committee where it is desired to have: (1) the same proportion of men and women; and (2) as many experienced people as possible.

In a case like that, we would like to not only select people with great support from the electorate, but also to take diversity wishes into account. Notice how the alternatives are no longer indistinguishable, as there are in fact certain classes of alternatives (e.g., men and women). Indeed, in many cases the alternatives can be naturally partitioned into classes, and some intraclass or interclass synergies are known or are desired: as further examples, consider: (1) selecting different cutlery items, where the utility gained from selecting one fork and one knife might be greater than the utility gained from selecting two forks; (2) selecting cellphones with their power adapters, where the utility gained from selecting one cellphone with its matching power adapter might be greater than the utility gained from selecting two cellphones; or (3)

selecting tables with matching chairs, where selecting one table with 100 chairs might not increase the utility gained from selecting the same table with, e.g., 8 chairs.

In this paper we consider partitions of the set of alternatives into certain natural classes, and model (positive and negative) synergistic relations between alternatives of the same class (intraclass synergies), as well as synergistic relations and diversity wishes between alternatives of different classes (interclass synergies). Our main modelling tool is what we refer to as *synergy functions* (formal definitions are given in the Preliminaries section). We concentrate on the computational complexity of identifying a winning committee, while taking into account such intraclass and interclass synergies. Intuitively, we would like each committee member to have reasonable support from the electorate while maximizing (minimizing) the sum of the positive (negative) synergies.

We show that, while the corresponding optimization problem is generally intractable (specifically, for general synergy functions the problem is not even fixed-parameter tractable for the number of classes; roughly speaking, this means that even with very few classes no efficient algorithm for selecting an optimal committee exists), there are various cases for which efficient algorithms do exist. Specifically, we describe efficient algorithms that can identify committees satisfying certain diversity wishes, as well as committees that optimize certain non-trivial positive and negative synergies between the alternatives.

### Related Work

The most relevant work is the recent work of Izsak 2017 about committee selection with synergies between the candidates. Through *set functions*, Izsak allows voters to express their preferences of being represented by several *specific* candidates together; e.g., the score a voter gives to two candidates together can be higher than the sum of their individual scores (say, if they are known to work extremely well together). As set functions might be too complex, Izsak considers preference elicitation and algorithms based on the supermodular degree (Feige and Izsak 2013) and showed that existing algorithms (Feldman and Izsak 2014; 2017) can be used to obtain approximation guarantees that depend on the supermodular degree. The supermodular degree, however, might be too large when some candidates

have synergy with many other candidates. For example, assume that there are  $m/2$  chairs and  $m/2$  tables, where *every* chair is worth more together with *every* table. Then, the supermodular degree depends on  $m$ , and the guarantees of algorithms tailored for the supermodular degree might not be useful. Contrary, in the current paper we have an exact and efficient algorithm for this case. On the other hand, in an instance where one views the items as an ordered list, where every two consecutive items in the list have positive synergy, then, in the model of Izsak 2017, the supermodular degree is 2, while in our model, we need one class for every single item.

Another related model is the well-studied model of Marginal Contribution nets (MC-nets) (Ieong and Shoham 2005). MC-nets, which are vastly studied for cooperative game theory, are general and flexible; in particular, they can also capture synergies. We stress, however, that our model is different than that of MC-nets.

Other works within computational social choice consider relations between the alternatives. Among these, perhaps the most relevant are works related to voting in social networks, such as work studying the influence of social networks on election outcomes (Conitzer 2012; Doucette et al. 2015; Procaccia, Shah, and Sodomka 2015; Salehi-Abari and Boutilier 2014; Sina et al. 2015) and the effect of social networks on manipulative actions performed by external agents (Bredereck et al. 2015b; Bulteau et al. 2015; Chen et al. 2015).

## Preliminaries

Below we briefly discuss parameterized complexity and treewidth. We denote the set  $\{1, \dots, n\}$  by  $[n]$ .

### Parameterized Complexity

While the number  $m$  of alternatives in a given election might be huge, the number  $z$  of classes of alternatives might be significantly smaller than  $m$ . Therefore, considering  $z$  as a parameter, and studying our optimization problem from a parameterized complexity point of view is useful. Parameterized complexity is a very active research field; here we only briefly discuss it, and point the reader to the textbooks of Downey and Fellows 2013, Flum and Grohe 2006, or Niedermeier 2006.

An instance  $(I, k)$  of a parameterized problem consists of an instance  $I$  of a non-parameterized problem and an integer  $k$ , referred to as the *parameter*. A parameterized problem is called *fixed-parameter tractable* (or, is said to be in  $\mathcal{FPT}$ ) if there is an algorithm solving it in  $f(k) \cdot |I|^{O(1)}$  time, where  $f$  is some computable function. In contrast, an algorithm with running-time  $|I|^{f(k)}$  only shows membership in the class  $\mathcal{XP}$ . Trivially, it holds that  $\mathcal{FPT} \subseteq \mathcal{XP}$ .

One can show that a parameterized problem  $L$  is (presumably) not fixed-parameter tractable by devising an  $\mathcal{FPT}$  reduction from a  $W[1]$ -hard problem to  $L$ . An  $\mathcal{FPT}$  reduction from a parameterized problem  $L$  to another parameterized problem  $L'$  is a function that, given an instance  $(I, k)$ , computes in  $f(k) \cdot |I|^{O(1)}$  time an instance  $(I', k')$ , such

that  $k' \leq g(k)$  and  $(I, k) \in L \iff (I', k') \in L'$ , where  $g$  can be any computable function.

### Treewidth

Treewidth (see, e.g., (Downey and Fellows 2013)) is a measure of similarity of graphs to trees, where graphs which are similar to trees (in some specific sense) have small treewidth.

Let  $G$  be an undirected graph, let  $T$  be a tree and let  $B : V(T) \rightarrow 2^{V(G)}$ . The pair  $(T, B)$  is a *valid tree decomposition* of  $G$  if  $T$  is a tree in which every vertex  $x \in V(T)$  has an assigned set of vertices  $B_x \subseteq V(G)$  (called a bag) such that the following properties are satisfied:

- **(P1):**  $\bigcup_{x \in V(T)} B_x = V(G)$ .
- **(P2):** For each  $\{u, v\} \in E(G)$ , there exists an  $x \in V(T)$  such that  $u, v \in B_x$ .
- **(P3):** For each  $v \in V(G)$ , the set of vertices of  $T$  whose bags contain  $v$  induces a connected subtree of  $T$ .

The *width* of the tree decomposition  $(T, B)$  is  $\max_{x \in V(T)} |B_x| - 1$ . The treewidth of a graph  $G$ , usually denoted by  $\omega(G)$ , is the minimum width over all the valid tree decompositions of  $G$ .

### The CSC Problem and its Modeling Power

We are ready to describe our formal model and the specific combinatorial problem we aim at solving. First, we present our model. Then, we provide examples to demonstrate its power: (1) its applicability to situations where the winning committee shall adhere to some diversity requirements; (2) its ability to incorporate both positive and negative synergies; and (3) its ability to model both interclass and intra-class synergies.

### Formal Model and Problem Definition

We are given an integer  $k$ , a set  $A = \{a_1, \dots, a_m\}$  of *alternatives*, and a partition of  $A$  into  $z$  subsets:  $A = C_1 \cup \dots \cup C_z$ , and  $C_i \cap C_j = \emptyset$  for every  $i \neq j$ . For every  $i \in [z]$ , we refer to  $C_i$  as the  *$i$ th class* (of alternatives).

Our aim is to select a (winning) *committee*  $S \subseteq A$  that consists of exactly  $k$  alternatives. We would like our winning committee to consist of alternatives with reasonable support from the given electorate, which we model by individual weight functions, and, importantly, to maximize the additional weight gained by synergies between alternatives within the classes of alternatives or between different classes of alternatives, which we model by synergy functions. Formally speaking, we associate an individual weight/score  $w(a_i)$  for every alternative  $a_i \in A$ . We mention that these weights are sufficient to incorporate any separable committee scoring function, such as SNTV,  $k$ -Borda, and Bloc. (Intuitively, a separable committee scoring function is a multiwinner voting rule where the score given to a committee by a voter is the sum of the individual scores this voter gives to each committee member; see, e.g., (Faliszewski et al. 2016).)

Additionally, we associate a *synergy function* for every set of classes of alternatives. Formally, let  $\mathcal{C} \subseteq \{C_1, \dots, C_z\}$  be

a set of classes containing exactly  $d$  classes. Then, we associate with  $\mathcal{C}$  a synergy function  $w_{\mathcal{C}} : [m]^d \rightarrow \mathbb{N}$  of dimension  $d$ . Intuitively, the value of  $w_{\mathcal{C}}(x_1, \dots, x_d)$  is the amount of additional weight (i.e., score) gained by selecting the given numbers of alternatives of each corresponding classes contained in  $\mathcal{C}$  (when, without loss of generality, we assume some canonical ordering over the classes contained in  $\mathcal{C}$ ). Indeed, there are at most  $m$  items at each of the classes and not exactly  $m$ , but for clarity, we neglect that, and allow one to give a value of 0 for every non relevant combination. Further, sometimes we list the members of a set of classes without using set notation. For example, consider the 2-dimensional synergy function  $w_{C_1, C_4}$ : the value  $w_{C_1, C_4}(2, 4)$  measures the additional weight gained to a committee that includes exactly 2 alternatives of the class  $C_1$  and exactly 4 alternatives of the class  $C_4$ .

The main problem we study in this paper is the following.

#### COMMITTEE SELECTION WITH CLASSES (CSC)

**Input:** A set of alternatives  $A = \{a_1, \dots, a_m\}$  partitioned into  $z$  classes  $C_1, \dots, C_z$ , integer weights  $w(a_i)$  for the alternatives ( $i \in [m]$ ), synergy functions of dimension  $d$ , and committee size  $k \in [m]$ .

**Question:** Output a committee  $S \subseteq A$  of size  $k$  that maximizes the *committee score*:  $\sum_{a_i \in S} w(a_i) + \sum_{j_1, \dots, j_d \in [z]} w_{C_{j_1}, \dots, C_{j_d}}(|S \cap C_{j_1}|, \dots, |S \cap C_{j_d}|)$ .

Note that a synergy function of dimension  $d$  can easily model a synergy function of a lower dimension, by giving strictly positive values only for tuples containing zeros for a fixed subset of the coordinates.

**Example 1.** Consider the alternatives  $\{a_1, a_2, b_1, b_2\}$ , partitioned into  $C_1 = \{a_1, a_2\}$  and  $C_2 = \{b_1, b_2\}$ . Let  $w(a_1) = w(b_1) = 1$ ,  $w(a_2) = 2$  and  $w(b_2) = 3$ . Let the 2-dimensional synergy function  $w_{C_1, C_2}$  be  $w_{C_1, C_2}(x_1, x_2) = 2x_1$  and set  $k = 2$ . Then, two committees tie as winners, both with committee score 7:  $\{a_1, a_2\}$  and  $\{a_2, b_2\}$ .

## Modeling Power and Examples

Here we demonstrate some aspects of our model's modeling power. The following examples are meant to be toy examples to point to different modeling possibilities.

**Example 2** (Positive and negative synergies). Assume selecting a small committee from a given set of alternatives  $\{a, b, c, d, e, f\}$ . The committee members shall work together towards a certain goal. Therefore, it is desirable to satisfy both the following goals: (1) the committee members shall be well respected (say, very professional); and (2) the committee members should be able to work well together as a team. For the first goal, one may ask an assumed electorate for alternatives' approvals and count the number of approvals for each alternative separately; for the sake of the example, assume that  $a$ ,  $b$ , and  $c$  receive 3 approvals each, while  $d$ ,  $e$ , and  $f$  receive 5 approvals each. For the second goal, one may try to assess how well different alternatives work well together; for the sake of the example, assume that both  $a$  and  $b$  work well with  $c$ , while both  $d$  and  $e$  do not work well with  $f$ .

We can model the problem of choosing the best committee as a CSC instance as follows. The set of alternatives is  $A = \{a, b, c, d, f, e\}$  with the following weights:  $w(a) = w(b) = w(c) = 3$  and  $w(d) = w(e) = w(f) = 5$ . To model the positive and negative synergies, it is useful to first partition the set of alternatives into 4 classes, as follows:  $A = C_1 \cup C_2 \cup C_3 \cup C_4$ , where,  $C_1 = \{a, b\}$ ,  $C_2 = \{c\}$ ,  $C_3 = \{d, e\}$ , and  $C_4 = \{f\}$ . Then, the synergy functions might be as follows:  $w_{C_1, C_2}(1, 1) = 10$ ,  $w_{C_1, C_2}(2, 1) = 20$ ,  $w_{C_3, C_4}(1, 1) = -10$ ,  $w_{C_3, C_4}(2, 1) = -20$ . Notice how the positive synergies correspond to positive values while the negative synergies correspond to negative values.

**Example 3** ( $d$ -dimensional synergy functions). Consider selecting cutlery, such as forks, knives and spoons: assume the alternatives  $A = \{f_1, f_2, k_1, k_2, s_1, s_2\}$ , where,  $f_i$  are forks,  $k_i$  are knives, and  $s_i$  are spoons. Indeed, the set of alternatives can be naturally partitioned into  $A = \{C_1, C_2, C_3\}$ , where,  $C_1 = \{f_1, f_2\}$ ,  $C_2 = \{k_1, k_2\}$  and  $C_3 = \{s_1, s_2\}$ . Furthermore, one might prefer to select sets consisting of one fork, one knife, and one spoon (as compared to, say, two forks and one spoon).

As a CSC instance, one shall first assign weights to the individual cutlery items. Then, to account for the relations between the cutlery types, one may define the synergy functions to be, say, as follows:  $w_{C_1, C_2, C_3}(1, 1, 1) = 10$ ,  $w_{C_1, C_2, C_3}(2, 2, 2) = 20$ , while all other values are 0.

In the previous examples we demonstrated modeling interclass synergies, such as the utility added by selecting one fork together with one knife and one spoon. One can model intraclass synergies using CSC, as we show next.

**Example 4** (Intraclass synergies). Assume selecting a national soccer team, where the available athletes are  $A = \{a_1, a_2, b_1, c_1, c_2, c_3\}$ . Further,  $a_1$  and  $a_2$  currently play in the same soccer club, while  $c_1, c_2, c_3$  are in a different club. The set of alternatives can be naturally partitioned as follows:  $A = C_1 \cup C_2 \cup C_3$ , where  $C_1 = \{a_1, a_2\}$ ,  $C_2 = \{b_1\}$ , and  $C_3 = \{c_1, c_2, c_3\}$ . As athletes from the same club play well together, we do not only want to select the best athletes to the team, but take this aspect into account, as well. As a CSC instance, one might define the following synergy functions:  $w_{C_1}(2) = 20$ ,  $w_{C_3}(2) = 20$ ,  $w_{C_3}(3) = 40$ . Notice how, e.g., selecting all three athletes from  $C_3$  increases the committee score by 40. Further, notice how intraclass synergies are modeled using 1-dimensional synergy functions, while, in other examples, interclass synergies are modeled using  $d$ -dimensional ( $d > 1$ ) synergy functions.

In many cases diversity in the elected committee is required, or at least desired to some extent. Later we discuss general ways to model diversity goals using CSC; specifically, Theorem 2 and Theorem 3 describe an efficient algorithm for that task. Here we provide a simple example.

**Example 5** (Diversity requirements). Consider selecting a committee to include 4 committee members out of the set of alternatives  $A = \{m_1, m_2, m_3, w_1, w_2, w_3\}$ , where  $m_i$  are men,  $w_i$  are women, and it is required to have the same number of men and women in the committee. First we may assign weights to the alternatives based on their support from

the electorate. Then, after naturally partitioning the set of alternatives to  $A = \{C_1, C_2\}$ , where  $C_1 = \{m_1, m_2, m_3\}$  and  $C_2 = \{w_1, w_2, w_3\}$ , we can define the synergy functions to be, say:  $w_{C_1, C_2}(2, 2) = 100$ , while all other values of the synergy functions are 0, and the individual weights are much smaller than 100.

## CSC is Computationally Intractable

As exemplified in the previous section, many scenarios can be modeled as CSC instances. Therefore, a natural question to ask is whether solutions to CSC instances can be found efficiently. In this section, we start our computational journey by investigating the computational complexity of CSC. First, perhaps not surprisingly, we face intractability; interestingly, we even face parameterized hardness with respect to the number of classes.

**Theorem 1.** *CSC is  $\mathcal{NP}$ -hard and  $W[1]$ -hard with respect to the number  $z$  of classes.*

*Proof.* We reduce from the  $W[1]$ -hard problem MULTICOLORED CLIQUE (MCC) (Downey and Fellows 2013): given a graph  $G = (V, E)$  whose vertices are partitioned into  $h$  disjoint color classes  $V = \bigcup_{l \in [h]} V_l$ , where for every  $j \in [h]$ ,  $|V_j| = n$  and  $V_j$  is an independent set by  $G$ , the goal is to decide whether a set of  $h$  pairwise adjacent vertices  $v_1, \dots, v_h$  with  $v_i \in V_i$  for every  $i \in [h]$ , exists.

Let  $n$  and  $h$  be integers. Given an instance of MCC with  $h$  color classes  $V_1, \dots, V_h$ , where  $V_j = \{v_j^1, \dots, v_j^n\}$ , for every  $j \in [h]$ , we show how to reduce it to an instance of CSC.

We introduce  $z = h + 1$  classes of alternatives,  $C_1, \dots, C_h$  and  $D$ , where, for every  $j \in [h]$ ,  $C_j = \{a_j^1, \dots, a_j^n\}$  and  $D = \{d^1, \dots, d^{nh}\}$  (thus, we have  $m = 2nh$  alternatives in total). We set all the individual weights to zero:  $w(a_j^i) = 0$  ( $j \in [h], i \in [n]$ ) and  $w(d^i) = 0$  ( $i \in [nh]$ ). We set  $k$  to  $nh$ . For every pair of classes  $C_i$  and  $C_j$  ( $i \neq j \in [h]$ ), we define a synergy function of dimension 2, such that  $f_{C_i, C_j}(x, y) = 1$ , if the edge  $\{v_i^x, v_j^y\}$  is present in the graph  $G$ , and  $f_{C_i, C_j}(x, y) = 0$ , otherwise. This finishes the reduction.

We associate with each committee (of the reduced CSC instance), a set of at most  $h$  vertices (of the given MCC instance): with a committee  $S$  we associate the set of vertices  $V_S = \{v_j^{|S \cap C_j|} : j \in [h]\}$ ; that is, as  $S$  has  $|S \cap C_j|$  members of the  $j$ -th class,  $V_S$  has the  $|S \cap C_j|$ -th vertex of color  $j$ . Notice that we do not consider the class  $D$ . The crucial observation now is that the committee score of  $S$  equals the number of edges between the vertices of  $V_S$ ; this is so since all individual weights are set to zero, and every synergy function that corresponds to two classes  $C_i$  and  $C_j$  adds 1 to the committee score, if and only if there is an edge between the corresponding vertices of the  $i$ th and  $j$ th color. It follows that the committee score of  $S$  is maximum when  $V_S$  is a multicolored clique.  $\square$

The negative result above means that (as long as  $W[1] \not\subseteq FPT$ , which is widely believed), CSC is not fixed-parameter tractable with respect to the number  $z$  of classes.

## Breaking CSC's Computational Intractability

As in practice we might still want to find solutions to CSC instances, we continue our computational journey by considering possible ways of overcoming the intractability shown in the last section. First, we observe that if the number  $z$  of classes is a constant, then there is a polynomial-time algorithm for CSC.

**Observation 1.** *CSC is  $\mathcal{XP}$  wrt. the number  $z$  of classes.*

*Proof.* Consider the following brute-force algorithm: For every class  $C_j$  ( $j \in [z]$ ), guess (i.e., do exhaustive search for) a number  $k_j \in [k]$  which stands for the number of committee members to be selected from the  $j$ th class, and make sure that  $\sum_{j \in [z]} k_j = k$ . Then, for every class  $C_j$  ( $j \in [z]$ ), choose the  $k_j$  heaviest alternatives from that class. Since the highest number we should guess is  $k$ , the number of possibilities we check is upper bounded by  $k^z$ , which is polynomial in  $k$ , by our assumption that  $z$  is a constant.  $\square$

Next we consider certain kinds of synergy functions; specifically, we consider separable functions, monotone functions, piece-wise linear functions, and so-called constant-bounded functions (formally defined below). We show that CSC instances with synergy functions that are separable can be solved in polynomial time. We also show that CSC instances with synergy functions that are piece-wise linear concave or constant-bounded can be solved in FPT time with respect to the number of classes.

## Separable Synergies and Diversity Wishes

A function  $f : \mathbb{N}^d \rightarrow \mathbb{N}$  is separable if there are functions  $f_1, \dots, f_d$  such that  $f(x_1, \dots, x_d) = \sum_{i \in [d]} f_i(x_i)$ . Below we describe an efficient algorithm that solves CSC instances with only separable functions, and later we show how to model diversity wishes using such CSC instances. Note that if all the synergy functions are separable, then we can actually model the problem with only intraclass synergy functions (i.e. synergy functions of dimension 1). So, in fact we show how to compute exactly an optimal solution in the case of only intraclass synergies. Interestingly, such functions can model also diversity wishes *between different classes*.

**Theorem 2.** *Any instance of CSC containing only separable synergy functions is polynomial-time solvable.*

*Proof.* We describe an algorithm based on dynamic programming. We use the fact that, as all synergy functions are separable, it is sufficient to decide the number of committee members from each class individually. Informally speaking, our algorithm is as follows: we arbitrarily sort the classes and then apply dynamic programming according to that order, while remembering only the number of committee members that remain to be selected to the committee.

Formally, let  $C_1, \dots, C_z$  be the alternative classes, and consider them in this (arbitrary) order. We define  $\text{Rec}(j, k')$  to be a committee with the maximum score possible when selecting  $k'$  alternatives from the the first  $j$  classes,  $C_1, \dots, C_j$ . We show how to compute the values of the table

$\text{Rec}(j, k')$ , for every  $j \leq z, k' \leq k$ . At the end, we will return the committee stored at  $\text{Rec}(z, k)$ . Note that the scores of the committees in  $\text{Rec}$  can be stored as well.

We start by computing committees for  $\text{Rec}$  for  $j = 1$ , for every  $k' \leq k$ . This can be done as follows: one needs to choose  $k'$  alternatives from  $C_1$  of maximum individual weights. Given the values of  $\text{Rec}(j', k')$  for every  $k' \leq k$  and  $j' < j$ , in order to compute the committee for  $\text{Rec}(j, k')$ , we check all the possibilities of selecting a committee that contains  $k'' \leq k$  (heaviest) alternatives from  $C_j$ , and the rest from  $C_1, \dots, C_{j-1}$ , and choose the best. Observe that the number of possibilities is polynomial in  $k$  and the size of the table  $\text{Rec}$  is polynomial, thus the claimed complexity follows.  $\square$

To see how separable functions help to select committees with diversity desires, consider the following example.

**Example 6.** Consider selecting a committee of 6 members, given an election containing 6 men and 6 women. One natural diversity desire would be to have, if possible, the same number of men and women (i.e., 3) in the selected committee. We emphasize that this is not a “requirement”, as one shall be aware of the inherent tradeoff between the support each alternative has from the electorate, and the degree of diversity achieved.

One possible way to incorporate such diversity desire into an instance of CSC is to consider two classes, of men and of women, and to define one synergy function of dimension two  $f_{men,women}$  where  $f_{men,women}(x_1, x_2) = f_{men}(x_1) + f_{women}(x_2)$ ; then,

$$\begin{aligned} f_{men}(0) &= f_{women}(0) = 0, \\ f_{men}(1) &= f_{women}(1) = 1, \\ f_{men}(2) &= f_{women}(2) = 2, \\ f_{men}(3) &= f_{women}(3) = 3, \\ f_{men}(4) &= f_{women}(4) = 3, \\ f_{men}(5) &= f_{women}(5) = 3, \\ f_{men}(6) &= f_{women}(6) = 3. \end{aligned}$$

Notice how the maximum of  $f_{men,women}$  is achieved at  $f_{men,women}(3, 3) = 6$ , and how the value of  $f_{men,women}$  decreases for less diverse committees.

Interestingly, we can generalize the above example to incorporate various kinds of diversity desires from a winning committee, even for an arbitrary dimension.

**Theorem 3.** *Consider an election with  $z$  classes of alternatives and diversity preferences of the form “at least  $x_i$  elected members are from class  $i$ ”, where  $\sum_i x_i \leq k$ . Then, finding an optimal winning committee can be done in polynomial time.*

*Proof.* We formulate the problem as an instance of CSC with separable functions and generalize the idea shown in Example 6. Specifically, for every class  $i$ , we define the synergy function:

$$f_i(t) = \max\{t, x_i\}.$$

Then, the maximum possible value of  $\sum_i f_i(t_i)$  is  $\sum_i x_i$  and this value is achieved if and only if  $\forall_i t_i \geq x_i$ . Moreover, for every  $i$ , if  $t_i < x_i$ , then  $\sum_i f_i(t_i)$  is less than the optimum by at least  $x_i - t_i$ , which is the distance of  $t_i$  from the objective for the class  $i$ .  $\square$

## Monotone Synergies

Here we consider monotone functions: a function  $f : \mathbb{N}^d \rightarrow \mathbb{N}$  is monotone if, fixing any  $d - 1$  values, the residual one-dimensional function is monotone. This means that adding candidates to a subset can never decrease its value (intuitively, a candidate cannot interrupt to the rest of the committee). The next result, which follows by modifying the construction given in the proof of Theorem 1, shows that CSC remains intractable even if all the synergy functions are monotone.

**Theorem 4.** *CSC is  $\mathcal{NP}$ -hard and  $W[1]$ -hard with respect to the number  $z$  of classes even if all the synergy functions are monotone.*

*Proof.* The result follows by modifying the reduction constructed in the proof of Theorem 1. Specifically, recall that in the proof of Theorem 1, we defined the synergy functions  $f_{C_i, C_j}$  ( $i \neq j \in [h]$ ) to be  $f_{C_i, C_j}(x, y) = 1$  if the edge  $\{v_i^x, v_j^y\}$  is present in the graph  $G$ , and 0 otherwise; here, we define the synergy functions to be  $f_{C_i, C_j}(x, y) = 2x + 2y + 1$  if the edge  $\{v_i^x, v_j^y\}$  is present in the graph  $G$ , and  $2x + 2y$  otherwise. Further, instead of defining  $w(d^i) = 0$  ( $i \in [nz]$ ); as done in the proof of Theorem 1), here we define  $w(d^i) = 2(h - 1)i$  ( $i \in [nz]$ ).

The proof of correctness follows similar lines as the proof of theorem 1 when observing that all synergy functions are monotone, and the committee score of a committee  $S$  with  $k$  candidates is equal to  $2(h - 1)k$  plus the number of edges between the vertices of  $V_S$ .  $\square$

## Piece-wise Linear Synergies

While the last section shows that CSC remains intractable even if all the synergy functions are monotone, in this section we consider concave functions over linear combinations (defined within the next proof), and achieve (fixed-parameter) tractability.

We mention that the following theorem implies that CSC is fixed-parameter tractable if all the synergy functions are concave. Intuitively, concave synergy functions correspond to situations with decreasing marginal utilities; as motivating examples, consider (1) purchasing product clones (e.g. a second phone) and (2) having several committee members with the same expertise.

**Theorem 5.** *CSC is fixed-parameter tractable with respect to the number  $z$  of classes, if all the synergy functions are concave functions over linear combinations.*

*Proof.* We show how to reduce the problem into maximizing an integer linear program (ILP) with piece-wise linear concave functions; the claim then follows by applying the method developed by Bredebeck et al. 2015a.

We define  $z$  integer variables,  $z_l$  ( $l \in [z]$ ), where  $z_l$  stands for the number of committee members selected from the  $l$ th class,  $C_l$ . For each  $l \in [z]$  we add a constraint  $0 \leq z_l \leq |C_l|$ , as we cannot select more than  $|C_l|$  committee members from the class  $C_l$ . We add an additional constraint, namely  $\sum_{l \in [z]} z_l = k$ , to respect the required committee size.

It is convenient to define the following function for each  $l \in [z]$ : for  $x \in [|C_l|]$ , let  $g_l(x)$  be the total weight of the  $x$  heaviest alternatives in the class  $C_l$ . Notice that (1) we can precompute the values of  $g_l(x)$  ( $l \in [z]$ ,  $x \in [|C_l|]$ ); and (2) all the  $g_l$  functions are concave: roughly speaking, this is so since we take the heaviest items. To see the last claim more formally, order the alternatives of the  $l$ th class in decreasing weights, and denote them, w.l.o.g, by  $a_1, \dots, a_{|C_l|}$ , such that  $a_i \geq a_{i+1}$  ( $i \in [|C_l| - 1]$ ). Then, concavity follows since  $g_l(x+1) - g_l(x) = a_{x+1} \geq a_x = g_l(x) - g_l(x-1)$ . Logically, to maximize the committee score we shall first maximize the sum  $\sum_{l \in [z]} g_l(z_l)$ ; as it is not linear, we utilize its concavity, and transform it using extra real-valued variables, following the technique of Brederick et al. 2015a.

Further, we shall consider the addition to the committee score gained by the synergy functions. Here we assume that each  $d$ -dimensional synergy function  $w_{j_1, \dots, j_d}$  is such that it is a (piece-wise linear) concave function over linear combinations, that is,  $w_{j_1, \dots, j_d}(x_1, \dots, x_d) = g_{j_1, \dots, j_d}(f(x_1, \dots, x_d))$ , where  $g_{j_1, \dots, j_d}$  is a piece-wise linear concave function and  $f(x_1, \dots, x_d)$  is a linear combination of  $x_1, \dots, x_d$ . Thus, for each  $d$ -dimensional synergy function,  $w_{j_1, \dots, j_d}$  we define an integer variable  $z_{j_1, \dots, j_d}$  with the intent that  $z_{j_1, \dots, j_d}$  is the value of  $w_{j_1, \dots, j_d}(z_{j_1}, \dots, z_{j_d})$ . Logically, we shall add the constraint  $z_{j_1, \dots, j_d} \leq w_{j_1, \dots, j_d}(z_{j_1}, \dots, z_{j_d})$ ; again, following concavity, we can optimize it by transforming it using extra real-valued variables (Brederick et al. 2015a).

Finally, we maximize the following objective function:

$$\sum_{l \in [z]} z_l + \sum_{j_1 \neq \dots \neq j_d} z_{j_1, \dots, j_d},$$

and notice that we maximize piece-wise linear concave functions, thus our result follows by applying the method developed by Brederick et al. (Brederick et al. 2015a).  $\square$

We end this section with an open question: Is CSC  $W[1]$ -hard with respect to the number of synergy functions even if all synergy functions are convex functions over linear combinations (i.e., does CSC remains fixed-parameter tractable for increasing marginal utilities)?

### Constant-bounded Synergies

Consider a synergy function  $f_{C_{j_1}, \dots, C_{j_d}}$  of dimension  $d$ . We say that  $f_{C_{j_1}, \dots, C_{j_d}}$  is *constant-bounded* if it is defined only for constant values. That is, there are constants  $c_{j_1}, \dots, c_{j_d}$  such that, for each  $i$  and every possible values  $x_1, \dots, x_d$ , if  $x_i \geq c_i$ , then  $f_{x_{i_1}, \dots, x_{i_d}}(x_1, \dots, x_d) = f_{C_{i_1}, \dots, C_{i_d}}(x_1, \dots, x_{i-1}, c_i, x_{i+1}, \dots, x_d)$ . We refer to  $c_i$  as the *synergistic upper bound* of class  $i$ . The next result describes an efficient algorithm for instances of CSC consisting only of constant-bounded synergy functions.

**Proposition 1.** CSC with constant-bounded synergy functions is in FPT with respect to the number of classes  $z$ .

*Proof.* The idea is to guess a number  $b_i$  between 0 and  $c_i$ , for every class  $C_i$  with synergistic upper bound of  $c_i$  ( $i \in [z]$ ). This  $b_i$  will be the number of representatives selected from this class, unless the number selected is  $c_i$ , and then we will select *at least*  $c_i$  representatives from the corresponding class. After this guessing phase, we select the guessed number of representatives from each class according to the guess, as described above, in a greedy way; specifically, we always select the heaviest alternatives. Finally, if we have not yet selected  $k$  alternatives, then we complete the committee with alternatives selected greedily over all those classes  $C_i$  for which our guessed number is  $c_i$ . Note that our “guesses” can be implemented by an exhaustive search. So, for  $c_{max} = \max_i c_i$ , our running time is polynomial in  $(c_{max})^z$  and the number of alternatives, which proves Proposition 1, since  $c_{max}$  is a constant by the definition of constant-bounded synergy functions.  $\square$

### Structural Restrictions

Consider instances of CSC with dimension at most 2, and let the *classes-graph* be a graph containing one vertex for each class of alternatives and an edge between two vertices if there is a non-trivial (i.e. not the zero function) synergy function between the corresponding classes. In this section we show that instances of CSC with classes-graph of bounded treewidth can be solved in polynomial time. We mention that such classes-graphs contain as a special case graphs with (only) small connected components, which seem of high relevance to the motivating examples given in the Introduction.

**Theorem 6.** Any instance of CSC with classes-graph of constant treewidth is solvable in polynomial time.

*Proof.* The main idea of our polynomial-time algorithm is as follows. Let  $\omega$  be the treewidth of the classes-graph corresponding to the given instance of CSC. Since there exist small separators of size  $O(\omega)$  which separates the graph into two disconnected parts, it is possible, in polynomial time, to brute-force over all possibilities of selecting alternatives from classes within these separators. Intuitively, the latter possibilities suffice: in the classes-graph, there are no edges crossing these separators; thus, there are no synergies between classes from opposite sides of those separators.

For technical convenience, we will use a special type of tree-decompositions called *nice* tree-decompositions.

**Definition 1.** A tree-decomposition  $(T, B)$  of  $G$  is said to be nice if  $T$  is a rooted binary tree such that each vertex  $t \in T$  is one of the following four types:

- **Leaf Node:**  $t$  is a leaf in  $T$  and  $B_t = \{v\}$  for some  $v \in G$ .
- **Introduce Node:**  $t$  has exactly one child  $t'$  and  $B_t = B_{t'} \cup \{v\}$  for some  $v \notin B_{t'}$ .
- **Forget Node:**  $t$  has exactly one child  $t'$  and  $B_t = B_{t'} \setminus \{v\}$  for some vertex  $v \in B_{t'}$ .
- **Join Node:**  $t$  has exactly two children  $t', t''$  such that  $B_{t'} = B_t = B_{t''}$ .

The advantage of using nice tree-decompositions is that, when writing a dynamic program, as we do below, we only need to handle four types of nodes (intuitively, these node types distinguish between certain types of separators). It is known (Bodlaender and Koster 2008; Kloks 1994) that a general tree decomposition  $(T, B)$  (of treewidth  $\omega$ ) can be converted, in linear time, into a *nice tree decomposition*  $(T', B')$  of the same width such that  $|T'| = O(\omega \cdot n)$ .

As mentioned, we solve CSC using dynamic programming. In fact, we solve a slightly generalized version of CSC (it is more general, since we are given specific requirements for different classes); specifically, we evaluate the following values recursively: For a node  $t \in T$  whose bag vertices are (without loss of generality)  $B_t = \{v_1, \dots, v_{|B_t|}\}$ , an integer  $k' \in [k]$ , and a vector  $[k'_1, \dots, k'_{|B_t|}]$ , we define  $\text{Rec}(t, k', [k'_1, \dots, k'_{|B_t|}])$  to be the maximum committee score possible when selecting  $k'$  alternatives from the alternatives contained in the classes at the subtree rooted at  $t$  (including  $t$ ), when we select exactly  $k'_j$  alternatives from the class corresponding to the  $j$ th bag vertex  $v_j$  of  $t$ .

Below we describe how to evaluate these  $\text{Rec}(t, k', [k'_1, \dots, k'_{|B_t|}])$  values. First we mention that, as our output, we return the committee corresponding to the value of  $\max_{[k'_1, \dots, k'_{|B_t|}] \in [n]^{|B_t|}} \text{Rec}(r, k, [k'_1, \dots, k'_{|B_t|}])$ , where  $r$  is the root of the tree-decomposition of the classes-graph of the given CSC instance (i.e., we consider all vectors  $[k'_1, \dots, k'_{|B_t|}]$  and maximize over them).

Next we describe how to compute the values  $\text{Rec}(t, k', [k'_1, \dots, k'_{|B_t|}])$  for each node type in the nice tree decomposition. We let  $W_j(k'_j)$  to denote the total weight of  $k'_j$  heaviest alternatives of the class corresponding to  $v_j$ .

**LEAF NODE.** Let  $t$  be a leaf node whose bag vertices are  $B_t = \{v_1, \dots, v_{|B_t|}\}$ . First, make sure that  $k' = \sum_{j \in [|B_t|]} k'_j$ . If so, then select  $k'_j$  heaviest alternatives from each  $v_j \in B_t$ . Then, compute the committee score, taking into account synergies between the vertices of  $B_t$ . More formally, we return  $\text{Rec}(t, k', [k'_1, \dots, k'_{|B_t|}]) = \sum_{j \in [|B_t|]} W_j(k'_j) + \sum_{j_1 \neq j_2 \in [|B_t|]} w_{v_{j_1}, v_{j_2}}(k'_{j_1}, k'_{j_2})$ .

**INTRODUCE NODE.** Let  $t$  be an introduce node whose bag vertices are  $B_t = \{v_1, \dots, v_{|B_t|}\}$ , and where  $v_1$  is the introduced node; i.e.,  $t$  has a single child,  $t'$ , and  $B_{t'} = \{v_2, \dots, v_{|B_t|}\}$ . Select  $k'_1$  heaviest alternatives from  $v_1$  and compute the sum of synergies corresponding to pairs of vertices  $v_1, v_j$  ( $2 \leq v_j \leq |B_t|$ ); then, recursively call  $t'$ . More formally,  $\text{Rec}(t, k', [k'_1, \dots, k'_{|B_t|}]) = W_1(k'_1) + \sum_{2 \leq j \leq |B_t|} w_{v_1, v_j}(k'_1, k'_j) + \text{Rec}(t', k' - k'_1, [k'_2, \dots, k'_{|B_t|}])$ .

**FORGET NODE.** Let  $t$  be a forget node whose single child is  $t'$  with  $B_{t'} = \{v_1, \dots, v_{|B_{t'}|}\}$ , where  $v_1$  is the forgotten node; i.e.,  $t = \{v_2, \dots, v_{|B_{t'}|}\}$ . Guess how many alternatives to select from the class corresponding to the forgotten node, and recursively call  $t'$ . More formally, let  $k'' = k' - \sum_{2 \leq j \leq |B_{t'}|} k'_j$  (i.e.,  $k''$  is the number of remaining alternatives to select from classes in the subtree

of  $t$ , not including  $t$ ); then,  $\text{Rec}(t, k', [k'_2, \dots, k'_{|B_{t'}|}]) = \max_{k'_1 \in k''} \text{Rec}(t', k' - k'_1, [k'_1, \dots, k'_{|B_{t'}|}])$ .

**JOIN NODE.** Let  $t$  be a join node whose two children are  $t'$  and  $t''$ , such that  $B_t = B_{t'} = B_{t''} = \{v_1, \dots, v_{|B_t|}\}$ . Recursively call the two children, but, since the synergies within the bag vertices are computed in each of the children, decrease the total value accordingly. More formally,  $\text{Rec}(t, k', [k'_1, \dots, k'_{|B_t|}]) = \text{Rec}(t', k', [k'_1, \dots, k'_{|B_t|}]) + \text{Rec}(t'', k', [k'_1, \dots, k'_{|B_t|}]) - \sum_{j_1 \neq j_2 \in [|B_t|]} w_{v_{j_1}, v_{j_2}}(k'_{j_1}, k'_{j_2})$ .

This finishes the description of our dynamic program whose correctness follows by its exhaustive nature. The number of recursive values computed is  $z \cdot k \cdot k^\omega$  where  $z$  is the number of classes ( $k^\omega$  is the number of different vectors of the form  $[k'_1, \dots, k'_{|B_t|}]$  and is bounded by  $k^\omega$  since  $\omega$  is the treewidth of the classes graph); as  $\omega$  is assumed to be constant, the number of recursive values computed is polynomial in the input size. Each of these values is computed by issuing a polynomial number of recursive calls, thus complexity follows.  $\square$

## Discussion

We introduced the CSC problem, which is concerned with identifying winning committees in multiwinner elections, when some (positive and negative) relations between certain alternatives are present; specifically, we utilized the fact that in many cases (but not all; see below) there is a natural partition of the set of alternatives, such that the relations between them can be expressed as synergy functions involving these classes. We have demonstrated that CSC can model various of real-world scenarios, including scenarios involving both positive and negative relations, scenarios with both interclass and intraclass relations, and scenarios where some diversity requirements or desires are to be satisfied by the winning committee.

Even though, in its most general form, CSC is computationally intractable, We have developed algorithms which efficiently solve many CSC instances.

We end the paper by discussing generalizations of CSC and other avenues for future research, as well as some other subtle points concerning our modeling.

- In our formal model, we used individual weights for the alternatives as well as synergy functions to model the synergies between the alternatives. The individual weights for the alternatives can naturally correspond to scores corresponding to any weakly separable multiwinner voting rule operating on an implicit multiwinner election (see Example 2 and the discussion concerning weakly separable multiwinner voting rules in the section describing the Formal Model). One might generalize CSC and study a similar problem, but one where the assumed underlying multiwinner voting rule is not weakly separable, but, say, Chamberlin–Courant or PAV.
- As another problem variant, notice that here we assumed that the values of the synergy functions are given explicitly. It might be interesting to see whether different

access models to their values (e.g., as oracle functions) correspond better to some real-world scenarios, and if so, whether efficient (exact, parameterized, or approximation) algorithms exist for such variants of CSC.

- It is natural to try to further extend the modeling power of CSC; as some example, consider (1) A generalization of CSC where not only one partition of the set of alternatives is given, but several partitions: this might correspond to diversity wishes across multiple dimensions, such as both gender diversity and ethnic diversity; and (2) A generalization of CSC with individual synergy functions: it seems plausible to assume that, for different voters, different synergies play different roles and importance.

### Acknowledgements

We are very grateful to Irit Dinur for useful discussions and for her help with showing hardness. We are also very grateful to Uri Feige, Inbal Livni-Navon and Maayan Meir for useful discussions.

### References

- Bodlaender, H. L., and Koster, A. 2008. Combinatorial optimization on graphs of bounded treewidth. *The Computer Journal* 51(3):255–269.
- Bredereck, R.; Faliszewski, P.; Niedermeier, R.; Skowron, P.; and Talmon, N. 2015a. Elections with few candidates: Prices, weights, and covering problems. In *Proceedings of the 4th International Conference on Algorithmic Decision Theory (ADT '15)*, 414–431.
- Bredereck, R.; Faliszewski, P.; Niedermeier, R.; and Talmon, N. 2015b. Large-scale election campaigns: Combinatorial shift bribery. In *Proceedings of AAMAS'15*, 67–75.
- Bulteau, L.; Chen, J.; Faliszewski, P.; Niedermeier, R.; and Talmon, N. 2015. Combinatorial voter control in elections. *Theoretical Computer Science* 589:99–120.
- Chen, J.; Faliszewski, P.; Niedermeier, R.; and Talmon, N. 2015. Elections with few voters: Candidate control can be easy. In *Proceedings of AAI'15*, 2045–2051.
- Conitzer, V. 2012. Should social network structure be taken into account in elections? *Mathematical Social Sciences* 64(1):100–102.
- Doucette, J. A.; Hosseini, H.; Tsang, A.; Larson, K.; and Cohen, R. 2015. Voting with social networks: Truth springs from argument amongst friends. *Presented at the 2nd Workshop on Exploring Beyond the Worst Case in Computational Social Choice; Held as part of AAMAS'15*.
- Downey, R. G., and Fellows, M. R. 2013. *Fundamentals of Parameterized Complexity*. Springer.
- Faliszewski, P.; Skowron, P.; Slinko, A.; and Talmon, N. 2016. Committee scoring rules: Axiomatic classification and hierarchy. In *Proceedings of IJCAI'16*, 250–256.
- Feige, U., and Izsak, R. 2013. Welfare maximization and the supermodular degree. In *Innovations in Theoretical Computer Science, ITCS*, 247–256.
- Feldman, M., and Izsak, R. 2014. Constrained monotone function maximization and the supermodular degree. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, 160–175.
- Feldman, M., and Izsak, R. 2017. Building a good team: Secretary problems and the supermodular degree. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, 1651–1670.
- Flum, J., and Grohe, M. 2006. *Parameterized Complexity Theory*. Springer.
- Ieong, S., and Shoham, Y. 2005. Marginal contribution nets: a compact representation scheme for coalitional games. In *Proceedings 6th ACM Conference on Electronic Commerce (EC-2005), Vancouver, BC, Canada, June 5-8, 2005*, 193–202.
- Izsak, R. 2017. Working together: Committee selection and the supermodular degree. In *Proceedings of AAMAS'17*, 1578–1580.
- Kloks, T. 1994. *Treewidth: computations and approximations*, volume 842. Springer Science & Business Media.
- Niedermeier, R. 2006. *Invitation to Fixed-Parameter Algorithms*. Oxford.
- Procaccia, A. D.; Shah, N.; and Sodomka, E. 2015. Ranked voting on social networks. In *Proceedings of IJCAI'15*, 2040–2046.
- Salehi-Abari, A., and Boutilier, C. 2014. Empathetic social choice on social networks. In *Proceedings of AAMAS'14*, 693–700.
- Sina, S.; Hazon, N.; Hassidim, A.; and Kraus, S. 2015. Adapting the social network to affect elections. In *Proceedings of AAMAS'15*, 705–713.