Growth of Dimension in Complete Simple Games

Liam O'Dwyer and Arkadii Slinko

Department of Mathematics, University of Auckland, New Zealand

Abstract. Simple games are a mathematical concept used to model a wide range of situations from decision making in committees [18] to reliability of real life systems made from unreliable components [19] and McCulloch-Pitts units in threshold logic [17]. Weighted voting games are a natural and practically important class of simple games, in which each agent is assigned a numeric weight, and a coalition is winning if the sum of weights of agents in that coalition is above a certain threshold. The concept of dimension in simple games was introduced by Taylor and Zwicker in [22] as a measure of the remoteness of a given game from being weighted. In [23] they demonstrated that the dimension of a simple game can grow exponentially in the number of players. However, the problem of worst-case growth of the dimension in complete games was left open. Freixas and Puente [12] showed that complete games of arbitrary dimension exist and, in particular, their examples demonstrate that the worst-case growth of dimension in complete games is at least linear. In this paper, using a novel technique of Kurz and Napel [16], we demonstrate that the growth of dimension in complete games can be polynomial in the number of players. Whether or not it can be exponential remains an open question.

1 Introduction

The past two decades have witnessed an explosion of interest in computational and representational issues related to coalitional games (see, e.g., [5, 14]) and simple games, in particular (see, e.g., [4, 7, 8]). Simple games were introduced in their present form by von Neumann and Morgenstern [24] for applications in economics but found a wide range of applications across several disciplines, including multi-agent systems. A simple game consists of a finite set of players and a set of winning coalitions that satisfies the monotone property asserting that all supersets of a winning coalition are also winning.

One of the most important classes of simple games is the class of *weighted simple games*. In a weighted simple game every player is assigned a non-negative real weight so that a coalition is winning if the total weight of its players is at least some predetermined threshold. From the computational perspective weighted games are especially important since they can be succinctly represented by a finite sequence of integers [9]. If we allow the weights and the threshold to be vector-valued, then every simple game becomes weighted [23] and the smallest dimension of vectors which makes this representation possible is called the *dimension* of the game.

Another important class of simple games is the class of complete games [2]. In a complete game it is always possible to say which player among any two players is more desirable (influential) as a coalition partner and this desirability relation is a total order. This is a much broader class of games than weighted games which includes, for example, conjunctive and disjunctive hierarchical games which appear as the access structures of popular secret sharing schemes [20, 21]. Both disjunctive and conjunctive hierarchical games are seldom weighted [13].

Freixas and Puente [12] studied conjunctive hierarchical games (under the name of games with a minimum) and found that their dimension grows linearly in the number of players and asked a question as to whether or not in the class of complete games the dimension can grow polynomially or even exponentially. In the present paper we show that the growth of dimension of disjunctive hierarchical games is at least polynomial, partly answering their question.

2 Preliminaries

Simple Games. Let P be a set consisting of n players. For convenience P can be taken to be $[n] = \{1, 2, \ldots, n\}$.

Definition 1. *A simple game is a pair $G = (P, W)$, where W is a subset of the power set 2^P which satisfies the monotonicity condition:*

if $X \in W$ and $X \subset Y \subseteq P$, then $Y \in W$.

Elements of the set W are called *winning coalitions*. We also define the set $L = 2^P \setminus W$ and call elements of this set *losing coalitions*. A winning coalition is said to be *minimal* if every proper subset of it is a losing coalition. Due to monotonicity, every simple game is fully determined by the set of its minimal winning coalitions $W_{\min}$ or the set of maximal losing coalitions $L_{\max}$.

Weighted Simple Games and Criteria of Weightedness.

Definition 2. *A simple game G is called a* weighted (majority) game *if there exist non-negative reals $w_1, \ldots, w_n$, and a positive real number q, called the quota, such that $X \in W$ iff $\sum_{i \in X} w_i \geq q$. Such a game is denoted $[q; w_1, \ldots, w_n]$. We also call $[q; w_1, \ldots, w_n]$ a* voting representation *for G.*

Example 3. Let now $n = 2k - 1$ be odd and W be all subsets of P of cardinality k or greater. There are exactly 2^{n-1} elements in W. This game is called the simple majority voting game. It is weighted and $[k; 1, 1, \ldots, 1]$ is its voting representation.

A more interesting example is

Example 4. The UN Security Council consists of five permanent and 10 non-permanent members (which are sov-ereign states). A passage requires approval of at least nine countries, subject to a veto by any one of the permanent members. This is a weighted simple game with a voting representation

$$[39; 7, 7, 7, 7, 7, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1].$$

A sequence of coalitions

$$\mathcal{T} = (X_1, \ldots, X_j; Y_1, \ldots, Y_j)$$

of simple game G is a *trading transform* if the coalitions $X_1, \ldots, X_j$ can be converted into the coalitions $Y_1, \ldots, Y_j$ by rearranging players. It can also be expressed as

$$|\{i : a \in X_i\}| = |\{i : a \in Y_i\}| \qquad \text{for all } a \in P.$$

A trading transform $\mathcal{T}$ is called a *certificate of non-weightedness* for G if $X_1, \ldots, X_j$ are winning in G and $Y_1, \ldots, Y_j$ are losing. The absence of certificates of non-weightedness of any length is a necessary and sufficient condition of weightedness [6, 23] of the game G.

Definition 5. *Let $G = (P, W)$ be a simple game, $A \subsetneq P$ and A^c is the complement of A in P. Let us define subsets*

$$W_{sg} = \{X \subseteq A^c \mid X \in W\}, \quad W_{rg} = \{X \subseteq A^c \mid X \cup A \in W\}.$$

Then the game $G_A = (A^c, W_{sg})$ is called a subgame *of G and $G^A = (A^c, W_{rg})$ is called a* reduced game *of G.*

It is easy to show that every subgame and every reduced game of a weighted game is also a weighted game.

Let $G = (P, W)$ be a simple game and L be the set of its losing coalitions. We define the game $G^* = (P, W^*)$ dual to G by setting

$$W^* = \{P \setminus X \mid X \in L\},$$

i.e., the winning coalitions of G^* are complements to the losing coalitions of G.

Complete and Hierarchical Simple Games. Given a simple game $G = (P, W)$, after Isbell [15], we define a relation $\succeq_G$ on P by setting $i \succeq_G j$ if for every set $X \subseteq P$ not containing i and j

$$X \cup \{j\} \in W \Longrightarrow X \cup \{i\} \in W. \tag{1}$$

In such a case we will say that i is at least as *desirable* (as a coalition partner) as j. This relation is reflexive and transitive but not always complete (total) (e.g., see [2]). The corresponding equivalence relation on $[n]$ will be denoted $\sim_G$ and the strict desirability relation as $\succ_G$. If this can cause no confusion we will omit the subscript G.

Definition 6. *Any simple game whose desirability relation is complete is called* complete*.*

Example 7. Any weighted game is complete.

Later we will have more examples. Complete simple games are a very natural generalisation of weighted games. This class is much larger however so measures of non-weightedness, e.g., the dimension, for such games are important and interesting.

In a complete game $G = (P, W)$ the set of players P is partitioned into equivalence classes $P = P_1 \cup \ldots \cup P_m$ with respect to $\sim_G$. Any coalition $X \subseteq P$ defines a multiset $\{1^{\ell_1}, \ldots, m^{\ell_m}\}$, where ℓ_i is the number of elements from P_i in X. Due to completeness, the status of X, i.e.,whether it is winning or losing, can be deduced from this multiset. The multisets corresponding to winning coalitions will be called *models of winning coalitions*. We also define *models of losing coalitions*, respectively.

In a complete game $G = (P, W)$ a winning coalition X is *shift-minimal* if any coalition $(X \setminus \{i\}) \cup \{j\}$ is losing for any $i \in X$ and $j \notin X$ such that $i \succ_G j$, i.e., it ceases to be winning after any replacement of its player with a less desirable one. A losing coalition Y is called *shift-maximal* if it becomes winning after a replacement of any player with a more desirable one. A complete simple game is fully defined by the set of its shift-minimal winning coalitions or the set of its shift-maximal winning coalitions.

Suppose now that the set of players P is partitioned into m disjoint subsets $P = \cup_{i=1}^{m} P_i$ and let $\mathbf{k} = (k_1, \ldots, k_m)$ be a sequence of positive integers with $k_1 < k_2 < \ldots < k_m$. Then we define the game $H = H_\exists(P, \mathbf{k})$ by setting the set of winning coalitions to be

$$W_\exists = \left\{ X \in 2^P \mid \exists i \left(\left| X \cap \left(\cup_{j=1}^{i} P_i \right) \right| \geq k_i \right) \right\}.$$

Such a game is called a *disjunctive hierarchical game*. It has m thresholds and at least one of them must be reached for the coalition to be winning.

Suppose now that the set of players P is partitioned into m disjoint subsets $P = \cup_{i=1}^{m} P_i$, and let $k_1 < \ldots < k_{m-1} \leq k_m$ be a sequence of positive integers. Then we define the game $H_\forall(P, \mathbf{k})$ by setting the set of its winning coalitions to be

$$W_\forall = \left\{ X \in 2^P \mid \forall i \left(\left| X \cap \left(\cup_{j=1}^{i} P_i \right) \right| \geq k_i \right) \right\}.$$

Such a game is called a *conjunctive hierarchical game*. It has m thresholds and all of them must be reached for the coalition to be winning.

Both classes of hierarchical games are complete. Gvozdeva et al [13] give a sufficient and necessary conditions for having all m classes of players of different desirability, i.e.,

$$P_1 \succ_G P_2 \succ_G \ldots \succ_G P_m \tag{2}$$

is satisfied in which case the game G is called *m-partite*. In this case $P_1, \ldots, P_m$ are exactly the equivalence classes with respect to the equivalence relation $\sim_G$. We denote $|P_i| = n_i$. For a conjunctive hierarchical game [13] give the following necessary and sufficient conditions:

(a) $k_1 \le n_1$, and
(b) $k_i < k_{i-1} + n_i$ for every $i \in \{2, \ldots, m\}$.

Moreover, G has veto players iff $k_1 = n_1$, in which case P_1 is the set of veto players, and G has dummies iff $k_{m-1} = k_m$, in which case P_m is the set of dummies.

We note that these conditions imply

$$k_i \le n_1 + \ldots + n_i - (i-1). \tag{3}$$

In particular, the equation $k_i = n_1 + \ldots + n_i$ can be satisfied only when $i = 1$.

Lemma 8. *Let $G = (P, W)$ be an m-partite simple game with $P = P_1 \cup P_2 \cup \ldots \cup P_m$, where P_1 consists of veto players and P_m of dummy players. Let $A = P_1 \cup P_m$. Then the reduced game G^A is defined on the set of players $P_2 \cup \ldots \cup P_{m-1}$ and does not have veto or dummy players.*

Proof. Due to Proposition 5 of [13] G_A is an $(m-2)$-partite conjunctive hierarchical game with $\mathbf{n}' = (n_2, \ldots, n_{m-1})$ and with the vector of thresholds

$$\mathbf{k}' = (k_2 - k_1, \ldots, k_{m-1} - k_1).$$

Since $n_2 > k_2 - k_1$ this game does not have veto players and since $k_{m-2} - k_1 < k_{m-1} - k_1$ it does not have dummies.

2.1 Definition of the Dimension. The Criterion of Kurz and Napel

The *dimension* of a simple game G is the minimum dimension of the vectors required to express it as a vector-weighted game [22]. That is, a simple game has dimension k if it can be represented as a vector-weighted game with weights from $\mathbb{R}^k$, but not with weights from $\mathbb{R}^\ell$ for $\ell < k$. In practice it is more convenient to work with the following equivalent definition.

Let $G_i = (P, W_i)$, $i = 1, \ldots, n$, be simple games with the same set of players P. Then the *intersection* of these games is the simple game $G = (P, W)$, where $W = W_1 \cap W_2 \cap \cdots \cap W_n$. In other words, a coalition is winning in G if and only if it is winning in (P, W_j) for each $j = 1, \ldots, n$. We write $G = G_1 \wedge \ldots \wedge G_n$ for reasons that will be revealed later.

Definition 9. *A simple game has dimension d if it can be represented as the intersection of d weighted games but cannot be represented as the intersection of ℓ weighted games for $\ell < d$. We will denote the dimension of G by $\dim(G)$.*

Examples of games of dimension 2 include the United States Federal System and the procedure to amend the Canadian Constitution [23]. Freixas [10] proved that two European voting systems under the Nice rules (two complete simple games) had dimension 3, which became the first real-world example of dimension 3. Cheung and Ng [3] proved that the voting system in Hong Kong (which is not a complete simple game) has also dimension 3. In a recent result Kurz and Napel [16] have found that the revised voting rules of the Council of the European Union (EU Council) mean that a simple game representation of that voting body must have dimension at least 7. This is significantly larger than that of any other known simple game that occurs in the real world.

To calculate the dimension of a game exactly is not an easy task. Deĭneko and Woeginger [4] proved that the following problem is NP-hard: given k weighted majority games, decide whether the dimension of their intersection is exactly k.

To bound the dimension of a game from above the following observation can be used [23].

Proposition 10. *The dimension of a simple game $G = (P, W)$ is at most the cardinality of the set $|L_{max}|$ of its maximal losing coalitions.*

To bound the dimension of a game from below we will use the following useful criterion, which is Observation 1 in Kurz and Napel [16].

Theorem 11. *Let $G = (P, W)$ be a simple game, and let $S = \{Y_1, \ldots, Y_k\}$ be a set of losing coalitions such that for each pair $\{Y_i, Y_j\}$ with $i \neq j$, there is no possible weighted simple game for which every coalition in W is winning but Y_i and Y_j are both losing. Then the dimension of G is at least $|S| = k$.*

Kurz and Napel refer to these elements of S as *pairwise incompatible.* One way to use this theorem to prove that a simple game G has dimension at least k is to find, for every pair $\{Y_i, Y_j\} \subseteq S$, a certificate of non-weightedness $(X^1_{i,j}, X^2_{i,j}; Y_i, Y_j)$, where $X^1_{i,j}$, and $X^2_{i,j}$ are both winning in G.

3 The Main Results

Dimension. Freixas and Puente [12] studied a class of games, that they called games with minimum, which, as was proved in [13], is nothing other than the class of conjunctive hierarchical games. They formulated the following theorem. However their proof of the lower bound was inaccurate (which is explained in the next section where we clarify the concept of dimension in complete games). We will give a simpler proof of the upper bound as well.

Theorem 12. *The dimension d of an m-partite conjunctive hierarchical game without veto or dummy players satisfies the inequalities $\lceil \frac{m}{2} \rceil \leq d \leq m$.*

Proof. Let $H = H_\forall(P, \mathbf{k})$, where $P = P_1 \cup \ldots \cup P_m$, $\mathbf{k} = (k_1, \ldots, k_m)$, and the game is m-partite. Let us denote $n = |P|$. Let us define m weight functions on P by

$$w_s(p) = \begin{cases} 1, \text{if } p \in P_1 \cup \ldots \cup P_s \\ 0, \text{if } p \in P_{s+1} \cup \ldots \cup P_m \end{cases}$$

and m thresholds $q_s = k_s$, $s \in [m]$. We define game G_s on P by the voting representation $[q_s; w_s(1), \ldots, w_s(n)]$.

It is clear that a coalition X wins in G_s if and only if $\left|X \cap \left(\cup_{j=1}^{s} P_i\right)\right| \geq k_s$, hence $H = G_1 \wedge \ldots \wedge G_m$ and $d \leq m$.

To prove the lower bound we assume that there are $\ell < \lceil \frac{m}{2} \rceil$ weighted games $H_1, \ldots, H_\ell$ such that $H = H_1 \wedge \ldots \wedge H_\ell$. Then, in each game H_i, all winning coalitions of H are winning.

Let $|P_i| = n_i$, $i = 1, \ldots, m$. By Lemma 2 and Theorem 6 of [13] we have $n_i \geq 2$. Any shift-maximal losing coalition of H corresponds to one of the following m models:

$$\mathcal{M}_i = \{1^{a_1}, \ldots, i^{a_i}, (i+1)^{n_{i+1}}, \ldots, m^{n_m}\} \quad (i = 1, \ldots, m),$$

where $a_1, \ldots, a_i$ satisfy the following conditions:

(i) $a_j \leq n_j$ for all $j \leq i$ and $a_i \leq n_i - i$;
(ii) If $0 < a_t$, then $a_s = n_s$ for all $s < t$;
(iii) $a_1 + \ldots + a_i = k_i - 1$.

Here (i) follows from (3) and the rest are clear.

We remind the reader that there are $\ell < m/2$ games $H_1, \ldots, H_\ell$ and m shift-maximal losing coalitions. Hence, due to the pigeonhole principle there exist an index i such that in H_i at least three shift-maximal losing coalitions of H, belonging to different models, are losing in H_i. Then there are two coalitions among these, say L_1 and L_2, whose models, say $\mathcal{M}_i$ and $\mathcal{M}_j$, satisfy $i + 2 \leq j$. Then

$$L_1 = C_1 \cup \ldots \cup C_i \cup P_{i+1} \cup \ldots \cup P_m,$$
$$L_2 = D_1 \cup \ldots \cup D_j \cup P_{j+1} \cup \ldots \cup P_m,$$

where $C_i \subseteq P_i$, $D_i \subseteq P_i$, moreover, $|C_1|+\ldots+|C_i| = k_i-1$ and $|D_1|+\ldots+|D_j| = k_j - 1$ with $|C_i| < n_i$ and $|D_j| < n_j$. We note that $|C_s| \leq |D_s|$ for $s \in [i]$. If only $|C_s| > |D_s|$ for some $s \in [i]$, and we can take s the smallest with this property, then $|D_s| < n_s$ and $|D_{s+1}| = \ldots = |D_j| = 0$. Since $|C_s| > 0$ by (ii) we have $|C_1| + \ldots + |C_{s-1}| = n_1 + \ldots + n_{s-1}$. This implies $|C_1| + \ldots + |C_s| > |D_1| + \ldots + |D_s| = k_j - 1 \geq k_i$, a contradiction.

Let $s \in [j]$ be the largest positive integer with $D_s \neq \emptyset$. Then $|D_1|+\ldots+|D_s| = |D_1| + \ldots + |D_j| = k_j - 1$. Suppose $s \leq i$. As $k_j - 1 \geq k_{j-1} > k_i$, we have $|C_s| < |D_s|$. If $s > i$, then $D_i = P_i$ and $|C_i| < |D_i|$ is true. Thus, a transfer of a player x, from D_s to C_s in the first case and from D_i to C_i in the second, is possible.

After the transfer the conditions $|D_1|+\ldots+|D_t| \geq k_t$ for $t \in [j-2]$ would not be violated. And if $k_{j-1} > k_j - 1$, then the condition $|D_1| + \ldots + |D_{j-1}| \geq k_{j-1}$ would not be violated either. If, however, $k_{j-1} = k_j - 1$, then after the transfer we will have $|D_1|+\ldots+|D_{j-1}| = k_{j-1} - 1$. And, of course, $|D_1|+\ldots+|D_j| = k_j - 2$. We claim that in such a case $D_{j-1} \neq P_{j-1}$. Indeed, if $D_{j-1} = P_{j-1}$, then $D_\ell = P_\ell$ for all $\ell \in [j-1]$. But then by (3) $|D_1| + \ldots + |D_{j-1}| = n_1 + \ldots + n_{j-1} > k_{j-1}$ which gives us a contradiction.

Now since $i < j-1$ we have $L_1 \supset P_{j-1} \cup P_j$. Since $D_{j-1} \neq P_{j-1}$ and $D_j \neq P_j$ and $\min(n_{j-1}, n_j) \geq 2$, we can transfer $y \in P_{j-1}$ and $z \in P_j$ from L_1 to L_2. This will make $(L_2 \setminus \{x\}) \cup \{y, z\}$ winning. Let us also notice that for L_1 before the transfer we had

$$|C_1| + \ldots + |C_i| + |P_{i+1}| + \ldots + |P_s| \geq k_s + (s - i)$$

for any $s \geq i + 1$. Indeed,

$$\begin{aligned} |C_1| + \ldots + |C_i| + |P_{i+1}| + \ldots + |P_s| &= k_i - 1 + \sum_{t=i+1}^{s} n_t \\ \geq k_i - 1 + \sum_{t=i+1}^{s} (k_t - k_{t-1} - 1) &= k_s + (s - i). \end{aligned}$$

This means that $(L_1 \cup \{x\}) \setminus \{y, z\}$ is winning as well. We obtained a certificate of non-weightedness

$$((L_1 \cup \{x\}) \setminus \{y, z\}, (L_2 \setminus \{x\}) \cup \{y, z\}; L_1, L_2)$$

which gives us a contradiction.

To enhance this theorem we prove the following lemma.

Lemma 13. *Let $G = (P, W)$ be a weighted voting game. Let P_1 and P_2 be disjoint sets which are also disjoint with P. We extend the set of players to $P' = P_1 \cup P \cup P_2$ and modify the set of winning coalitions so that*

$$W'_{min} = \{X \cup P_1 \mid X \in W_{min}\}.$$

Then the new game $G' = (P', W')$ is again weighted, players of P_1 are veto players and players of P_2 are dummy players in G'.

Proof. Let $w\colon P \to \mathbb{R}$ be the weight function for G and let q be the corresponding threshold. Let u be a sufficiently large number ($u > w(P)$ would be sufficient). We leave weights for elements of P unchanged, we give weight u to all players from P_1 and weight 0 to all players from P_2. The new threshold for G' will be $q' = u|P_1| + q$. It is straightforward to check that the new system of weights defines G'.

We can now prove our first main result.

Theorem 14. *The dimension d of an m-partite conjunctive hierarchical game satisfies $\lceil \frac{m}{2} \rceil - 1 \leq d \leq m$.*

Proof. Let G be an m-partite conjunctive hierarchical game. Due to Lemmata 1 and 2 the dimension of G is equal to the dimension of m'-partite game where $m' \geq m - 2$. The result now follows from Theorem 12.

We have to acknowledge though that the original version of this theorem in [12] gives an algorithm of computing the dimension of H from parameters $\mathbf{k} = (k_1, \ldots, k_m)$ and $\mathbf{n} = (n_1, \ldots, n_m)$, where $n_i = |P_i|$.

The only way we can grow the dimension in hierarchical conjunctive games is to increase the number m of equivalence classes. This is not the case where disjunctive hierarchical games are concerned. Let us consider a non-weighted example of a disjunctive hierarchical game with the 'smallest' possible vector $\mathbf{k}$. This would be $\mathbf{k} = (2, 4)$ (as in non-trivial cases we have $k_1 \geq 2$ and the games with $\mathbf{k} = (2, 3)$ are weighted [13]).

Proposition 15. *Let $d \geq 2$ be a positive integer. Let $P = P_0 \cup P_1$ with $|P_0| = d$, $|P_1| = 2d$, and $\mathbf{k} = (2, 4)$. Then the disjunctive hierarchical game $H = H_\exists(P, \mathbf{k})$ has dimension at least d.*

Proof. Let $P_0 = \{a_0, \ldots, a_{d-1}\}$ and $P_1 = \{b_0, \ldots, b_{2d-1}\}$, define the sets $Y_i = \{a_i, b_{2i}, b_{2i+1}\}$, $i = 0, \ldots, d-1$, and let $S = \{Y_0, \ldots, Y_{d-1}\}$. All coalitions from S lose in H. Then S satisfies the conditions of the Kurz-Napel criterion since if $i \neq j$

$$(\{a_i, a_j\}, \{b_{2i}, b_{2i+1}, b_{2j}, b_{2j+1}\}; Y_i, Y_j)$$

is a certificate of non-weightedness for H as the coalitions $X^1_{i,j} = \{a_i, a_j\}$ and $X^2_{i,j} = \{b_{2i}, b_{2i+1}, b_{2j}, b_{2j+1}\}$ are both winning (the first achieves the first threshold and the second achieves the second). By the criterion, the dimension of H is at least d.

We have shown here that we have dimension at least $|P_0|$ when $|P_1| = 2|P_0|$. If $|P_1| > 2|P_0|$, then we have the same result, as we can use exactly the same set of pairwise incompatible losing coalitions. In this case our coalitions will not include all the members of P_1, however. Finally, if $|P_1| < 2|P_0|$, then we can again use the same set of pairwise incompatible coalitions, but in this case we will not use every member of P_0. We will end up with $\left\lfloor \frac{|P_1|}{2} \right\rfloor$ pairwise incompatible losing coalitions in this case, since if $|P_1|$ is odd we will have only $\frac{|P_1|-1}{2}$ pairs of members from $|P_1|$. We must have that $|P_1| \geqslant 2$ so that we can form the pair $\{b_{2i}, b_{2i+1}\}$. Thus we have the following result.

Corollary 16. *The $(2, 4)$–disjunctive hierarchical game with $P = P_0 \cup P_1$ has a dimension of at least $\min\left\{|P_0|, \left\lfloor \frac{|P_1|}{2} \right\rfloor\right\}$ when $|P_1| \geqslant 2$.*

This result is interesting from several perspectives. Firstly, due to [13], we know that the game H in the proposition is roughly weighted. Hence we see that a roughly weighted game can have an arbitrary dimension. Secondly, we showed that we can get linear growth in the number of players without increasing the number of classes. If we start increasing both we will get a growth faster than linear.

Theorem 17. *Let $P = P_0 \cup P_1 \cup \ldots \cup P_{m-1}$ with $|P_0| = k$, $|P_1| = |P_2| = \cdots = |P_{m-1}| = 2k$ and $\mathbf{k} = (2, 4, 6, \ldots, 2m)$. Then the disjunctive hierarchical simple game $H = H_\exists(P, \mathbf{k})$ has dimension d satisfying*

$$k^{m-1} \le d \le k^m (2k-1)^{m-1}.$$

Proof. We will denote by $p_i^{(j)}$ the jth player from part P_i, and as above, we will let $P_i^{(j)} = \{p_i^{(2j)}, p_i^{(2j+1)}\}$, where here $j \in \{0, \ldots, k-1\}$ and $i \in [m]$. Then all coalitions of the form

$$\{a_{i_0}\} \cup P_1^{(i_1)} \cup P_2^{(i_2)} \cup \cdots \cup P_{m-2}^{(i_{m-2})} \cup P_{m-1}^{(i')}, \tag{4}$$

where $i' = i_0 + i_1 + \ldots + i_{m-2} \pmod{k}$ will form the set S to be used in the Kurz-Napel criterion. Firstly, we note that all the coalitions in S are losing as no threshold is achieved. Let us show that any two of them are incompatible. Let

$$Y_1 = \{a_{i_0}\} \cup P_1^{(i_1)} \cup P_2^{(i_2)} \cup \cdots \cup P_{m-2}^{(i_{m-2})} \cup P_{m-1}^{(i')},$$
$$Y_2 = \{a_{j_0}\} \cup P_1^{(j_1)} \cup P_2^{(j_2)} \cup \cdots \cup P_{m-2}^{(j_{m-2})} \cup P_{m-1}^{(j')},$$

be two coalitions from S. There are two cases.

- If $i_0 = j_0$, then there is at least one $\ell \in \{1, \ldots, m-2\}$ for which $i_\ell \ne j_\ell$ (otherwise the coalitions would be identical), so $P_\ell^{(i_\ell)}$ is disjoint from $P_\ell^{(j_\ell)}$. If $i_r = j_r$ for all $r \in \{1, \ldots, \ell-1, \ell+1, \ldots, m-2\}$, then $i' \ne j'$, thus we may assume that there exists also $r \in [m-1]$ such that $r \ne \ell$ and $i_r \ne j_r$. Without loss of generality assume that $\ell < r$. Then we get a certificate of non-weightedness

$$(Y_1 \cup \{p_\ell^{(2j_\ell)}\} \setminus P_r^{(i_r)}, (Y_2 \setminus \{p_\ell^{(2j_\ell)}\} \cup P_r^{(i_r)}; Y_1, Y_2).$$

  by swapping one element $\{p_\ell^{(2j_\ell)}\}$ of Y_2 for two elements $P_r^{(i_r)}$ of Y_1. After the swap the first coalition will be winning since the ℓ-th threshold is achieved and the second will be also winning since the r-th threshold is achieved.
- If $i_0 \ne j_0$ but $i_r = j_r$ for all $r \in \{1, \ldots, m-2\}$, then $i' \ne j'$, hence we may assume that there exists $r \in [m-1]$ such that $i_r \ne j_r$. In this case we get a certificate of non-weightedness

$$(Y_1 \cup \{a_{j_0}\} \setminus P_r^{(i_r)}, (Y_2 \setminus \{a_{j_0}\} \cup P_r^{(i_r)}; Y_1, Y_2)$$

  by swapping $\{a_{j_0}\}$ and $P_r^{(i_r)}$.

Since $|S| = k^{m-1}$, this means that the dimension of such a game is at least k^{m-1}.

The upper bound is easily calculated with the help of Proposition 10 taking in consideration that each maximal losing coalition consists of one member from P_0 and two members from each of the $P_1, \ldots, P_n$.

This result answers directly the question from [12] about possibility of a polynomial growth in complete games.

Theorems 12 and 14 demonstrate inter alia that the dimension is not preserved under duality. Indeed, it is known [13] that the duality takes us from disjunctive hierarchical games to conjunctive ones and vice versa.

Codimension. A concept closely related to dimension is that of the codimension.

Definition 18. *The* codimension *of a simple game is the minimum number of weighted simple games whose union forms the given game. That is, the simple game $G = (P, W)$ has codimension n if $W = W_1 \cup \cdots \cup W_n$, where each of $(P, W_1), \ldots, (P, W_n)$ is weighted, and W cannot be represented as the union of fewer than n weighted games. We will denote the codimension of G by* $\mathrm{codim}(G)$*.*

This concept emerges in relation to duality of games.

Theorem 19 (Freixas-Marciniak, 2009). *If G is a simple game, then*

$$\mathrm{codim}(G^*) = \dim(G).$$

Proof. This is a simplification of Theorem 3.2(ii) in [11], in which we take $\mathcal{C}$ to be the class of weighted simple games, along with their observation that this class is closed under duality.

Due to Freixas-Marciniak theorem we can extract some consequences from our results with respect to codimension.

Corollary 20. *The codimension d of an m-partite disjunctive hierarchical game satisfies $\lceil \frac{m}{2} \rceil - 1 \leq d \leq m$.*

Proof. Follows from Theorem 14 and the fact that the dual game to a disjunctive hierarchical game is conjunctive hierarchical [13].

Corollary 21. *There is a sequence of disjunctive hierarchical games whose codimensions grow polynomially in the number of players.*

4 Problems with the concept of dimension

There are several disturbing properties of the concept of dimension. One of those can be observed from the results of this paper, namely, that the dimension is not preserved under duality. Indeed, in [13] it was proved that the dual of a disjunctive hierarchical games is a conjunctive hierarchical games and vice versa. However, Theorems 14 and 17 show that the dimensions of games in these classes are very different. Another problem with this concept is illustrated in the following example.

Motivating example.[1] It shows that when one represents a complete game G as an intersection of weighted games, it may be impossible to choose the weightings in a way that faithfully represents the desirability order $\succeq_G$.

Let $P = P_1 \cup P_2$ with $|P_1| = 2$ and $|P_2| = 5$. Consider a disjunctive hierarchical game $H = H_\exists(P, \mathbf{k})$ with $\mathbf{k} = (2, 5)$. Frstly, let us consider P as a multiset $\{1^2, 2^5\}$, i.e., consisting of two identical players of type 1 and five of type 2. We ask if it is possible to find games $H_1, \ldots, H_k$, where the ith game has voting representation $[q_i; w_i(1), w_i(2)]$, i.e., assigning weight $w_i(1)$ for players of the first type, weight $w_i(2)]$ to players of the second type and having threshold q_i.

Considering all coalitions as submultisets we see that the minimal winning ones are $\{1^2\}$, $\{1, 2^4\}$, $\{2^5\}$. We can graphically depict them on the following diagram where the area of winning coalitions is presented in grey.

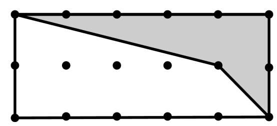


Figure 1: Vertical axes shows the number of players of the first type and horizontal of the second.

This grey area is not convex; its convex hull also contains the point corresponding to the only maximal losing coalition $\{1, 2^3\}$. This means that the games $H_1, \ldots, H_k$ cannot be found. What can be done?

Let $P_1 = \{b_1, b_2\}$ and $P_2 = \{c_1, c_2, c_3, c_4, c_5\}$. Consider the two following representations:

First representation. We define two weighted games $G_1 = (P, W_1)$ and $G_2 = (P, W_2)$ as follows:

$$\begin{aligned} w_1(b_1) = 4,\ w_1(b_2) = 1.1, \\ \forall_{j\in[5]}\ w_1(c_j) = 1 \text{ and quota} = 5. \end{aligned}$$

and

$$\begin{aligned} w_1(b_1) = 1.1,\ w_1(b_2) = 4, \\ \forall_{j\in[5]}\ w_1(c_j) = 1 \text{ and quota} = 5. \end{aligned}$$

Obviously, $H = G_1 \cap G_2$ and, since H is not weighted, we have $\dim H = 2$.

Second representation. The above representation failed to represent the 'equivalence' part of $\sim_H$ for the players of P_1, but did succeed with the P_2 players (in that they did get equal weight for each of the two weightings).

Next, we look at a representation that switches roles: it similarly represents the strict part of $\sim_H$ for all players, and it gives the P_1 players equal weight

[1] The example we present in this section is the result of an email discussion with Bill Zwicker.

for each of the weightings (but the five players of P_2 get different weights). For every subset $X \subset [5]$ such that $|X| = 3$ we define a game $G_X = (P, W_X)$ by

$$
\begin{aligned}
&w_X(b_1) = w_X(b_2) = 3,\\
&\forall_{i\in X}\ w_X(c_i) = 2,\ \forall_{i\notin X}\ w_X(c_i) = 0,\\
&quota = 6
\end{aligned}
$$

There are 10 such sets X of cardinality 3, and thus 10 weighted games, and it is easy to show that their intersection is the hierarchical game H. The nice property of this particular vector of weightings is that it respects the 'strict' part $\sim_H$ of the individual desirability order of H in the following sense: $x \prec_H y$ iff $w_X(x) < w_X(y)$ holds for each three-element subset X of [5].

This example explains why the lower bound in the theorem of Freixas and Puente (Theorem 2) had to be reproved. In their proof of the lower bound they allowed only weighted games that assign equal weights to players who are equivalent in the original game. We see that it is not sufficient to claim that the lower bound in their theorem holds.

Boolean dimension of simple games. Boolean dimension of a simple game was introduced in [8]. Let $\Phi = \{p, q, \ldots\}$ be a set of propositional variables and let $\mathcal{L}$ denote the set of (well-formed) formulas of the first-order propositional logic over Φ containing only logical connectives $\wedge$ and $\vee$[2]. For a formula $\phi \in \mathcal{L}$ let $|\phi|$ be the number of variables used to express ϕ. Suppose also $\top$ is a tautology and $\perp$ is a contradiction. Let $G_i = (P, W_i)$, $i = 1, \ldots, q$, be simple games with the same set of players P. We will define the game $G = (P, W)$ by setting for a coalition $C \subseteq P$

$$C \in W := \phi(C \in W_1, \ldots, C \in W_q) = \top.$$

We will denote this game $\phi(G_1, \ldots, G_q)$. We illustrate this definition with the following simple games which plays an important role in the theory of secret sharing [1]. They are called there tripartite.

Example 22. Let $\mathbf{n} = (n_1, n_2, n_3)$ and $\mathbf{k} = (k_1, k_2, k_3)$, where n_1, n_2, n_3 and k_1, k_2, k_3 are positive integers. The game $\Delta_1(\mathbf{n}, \mathbf{k})$ is defined on the set $P = P_1 \cup P_2 \cup P_3$ which is a union of disjoint sets P_1, P_2, P_3 of cardinalities n_1, n_2, n_3, respectively,. A coalition $C = C_1 \cup C_2 \cup C_3$, where $C_i \subseteq P_i$, $i = 1, 2, 3$, is winning iff

$$(|C_1| \geq k_1) \vee [(|C_1| + |C_2| \geq k_2) \wedge (|C_1| + |C_2| + |C_3| \geq k_3)] = \top,$$

where $k_1 < k_3$, $\quad k_2 < k_3$, $\quad n_1 \geq k_1$, $\quad n_2 > k_2 - k_1$ and $n_3 > k_3 - k_2$. Obviously, it is organised as $G_1 \vee (G_2 \wedge G_3)$, where G_1, G_2, G_3 are weighted games.

Definition 23. *Let G be a simple game. The smallest positive integer d such that G can be represented as $G = \phi(G_1, \ldots, G_n)$, where $G_1, \ldots, G_n$ are weighted simple games, and $|\phi| = d$ is called the Boolean dimension of G.*

[2] In paper [8] negations were also allowed, however, the authors of that paper considered also non-monotonic simple games which we do not consider.

The Boolean dimension of the game $\Delta_1(\mathbf{n}, \mathbf{k})$ from Example 22 is obviously 3 while its classical dimension may (and certainly will) depend on the parameters $\mathbf{n}$ and $\mathbf{k}$..

Apart from a better ability to reflect the descriptive complexity of games, the Boolean dimension has some nice properties absent in the classical dimension.

Proposition 24. *The Boolean dimension of a simple game is equal to the Boolean dimension of its dual.*

Proof. Let $G_i = (P, W_i)$, $i = 1, 2$, be a simple game and $G_i^* = (P, W_i^*)$, $i = 1, 2$, be their dual games. Taylor and Zwicker in [23] (see, e.g., Proposition 1.4.3) showed that the following two de Morgan laws are satisfied

$$(W_1 \cup W_2)^* = W_1^* \cap W_2^*,$$
$$(W_1 \cap W_2)^* = W_1^* \cup W_2^*.$$

From here it immediately follows that

$$(G_1 \vee G_2)^* = G_1^* \wedge G_2^*,$$
$$(G_1 \wedge G_2)^* = G_1^* \vee G_2^*,$$

which imply the statement.

However, some features of Boolean dimension are the same as the corresponding features of dimension. In particular, as is the case for dimension, in general the calculation of the exact Boolean dimension of a simple game is NP-hard [8].

5 Conclusion and open questions

We have answered a question of Freixas and Puente [12] by showing that the dimension of a complete simple game can grow polynomially in the number of players. The games used to demonstrate this are the so-called disjunctive hierarchical games. We have discussed the pitfalls of the concept of dimension and found some disturbing features of it which led us to reproving and strengthening the theorem of Freixas and Puente about the range in which the dimension of a conjunctive hierarchical game may lie. This research prompts the following questions:

- Can the dimension of a complete simple game grow exponentially in the number of players?
- Can anything be said about the Boolean dimension of a complete game?
- Let $G = (P, W)$ be a complete game with equivalence classes for the desirability relation $P_1, \ldots, P_m$ so that $P = P_1 \cup \ldots, \cup P_m$. Then for each $i \in [m]$ does there exist a representation of G as an intersection of weighted games such that each game in this representation assigns equal weights to players from P_i and respects the strict part of the desirability order $\succeq_G$?

References

1. A. Beimel, T. Tassa, and E. Weinreb. Characterizing ideal weighted threshold secret sharing. *SIAM J. Discrete Math.*, 22(1):360–397, 2008.
2. F. Carreras and J. Freixas. Complete simple games. *Mathematical Social Sciences*, 32(2):139–155, 1996.
3. W.-S. Cheung and T.-W. Ng. A three-dimensional voting system in hong kong. *European Journal of Operational Research*, 236:292–297, 2014.
4. V. G. Deĭneko and G. J. Woeginger. On the dimension of simple monotonic games. *European Journal of Operational Research*, 170(1):315–318, 2006.
5. X. Deng and C. H. Papadimitriou. On the complexity of cooperative solution concepts. *Math. Oper. Res.*, 19(2):257–266, 1994.
6. C. Elgot. Truth functions realizable by single threshold organs. In *Proceedings of the Second Annual Symposium on Switching Circuit Theory and Logical Design*, pages 225–245. AIEE, 1961.
7. E. Elkind, L. A. Goldberg, P. W. Goldberg, and M. Wooldridge. On the dimensionality of voting games. In *Proceedings of the 23rd Conference on Artificial Intelligence*, pages 69–74, 2008.
8. P. Faliszewski, E. Elkind, and M. Wooldridge. Boolean combinations of weighted voting games. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 185–192, 2009.
9. J. Freixas. Different ways to represent weighted majority games. *Top*, 5(2):201–211, 1997.
10. J. Freixas. The dimension for the european union council under the nice rules. *European Journal of Operational Research*, 156:415–419, 2004.
11. J. Freixas and D. Marciniak. A minimum dimensional class of simple games. *Top*, 17(2):407–414, 2009.
12. J. Freixas and M. A. Puente. Dimension of complete simple games with minimum. *European Journal of Operational Research*, 188(2):555–568, 2008.
13. T. Gvozdeva, A. Hameed, and A. Slinko. Weightedness and structural characterization of hierarchical simple games. *Mathematical Social Sciences*, 65(3):181–189, 2013.
14. S. Ieong and Y. Shoham. Marginal contribution nets: A compact representation scheme for coalitional games. In *Proceedings of the 6th ACM Conference on Electronic Commerce*, EC '05, pages 193–202, New York, NY, USA, 2005. ACM.
15. J. R. Isbell. A class of simple games. *Duke Mathematical Journal*, 25(3):423–439, 1958.
16. S. Kurz and S. Napel. Dimension of the Lisbon voting rules in the EU Council: A challenge and new world record, 2015. arXiv:1503.02859.
17. S. Muroga. *Threshold Logic and its Applications*. Wiley, New York, 1971.
18. B. Peleg. Game-theoretic analysis of voting in committees. *Handbook of social choice and welfare*, 1:395–423, 2002.
19. K. Ramamurthy. *Coherent structures and simple games*, volume 6. Springer Science & Business Media, 2012.
20. G. J. Simmons. How to (really) share a secret. In *Proceedings of the 8th Annual International Cryptology Conference on Advances in Cryptology*, pages 390–448. Springer-Verlag, 1990.
21. T. Tassa. Hierarchical threshold secret sharing. *Journal of Cryptology*, 20(2):237–264, 2007.

22. A. Taylor and W. Zwicker. Weighted voting, multicameral representation, and power. *Games and Economic Behavior*, 5(1):170–181, 1993.
23. A. D. Taylor and W. S. Zwicker. *Simple Games: Desirability Relations, Trading, Pseudoweightings*. Princeton University Press, 1999.
24. J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

Distributing Coalition Value Calculations to Coalition Members

Luke Riley, Katie Atkinson, Paul E. Dunne and Terry R. Payne

Department of Computer Science, University of Liverpool,
Liverpool L69 3BX, United Kingdom
{L.J.Riley, K.M.Atkinson, P.E.Dunne, T.R.Payne}@liverpool.ac.uk

Abstract. Within characteristic function games, agents have the option of joining one of many different coalitions, based on the utility value of each candidate coalition. However, determining this utility value can be computationally complex since the number of coalitions increases exponentially with the number of agents available. Various approaches have been proposed that mediate this problem by distributing the computational load so that each agent calculates only a subset of coalition values. However, current approaches are either highly inefficient due to redundant calculations, or make the benevolence assumption (i.e. are not suitable for adversarial environments). We introduce *DCG*, a novel algorithm that distributes the calculations of coalition utility values across a community of agents, such that: (i) no inter-agent communication is required; (ii) the coalition value calculations are (approximately) equally partitioned into shares, one for each agent; (iii) the utility value is calculated only once for each coalition, thus redundant calculations are eliminated; (iv) there is an equal number of operations for agents with equal sized shares; and (v) an agent is only allocated those coalitions in which it is a potential member. The *DCG* algorithm is presented and illustrated by means of an example. We formally prove that our approach allocates all of the coalitions to the agents, and that each coalition is assigned once and only once.

Note: This paper originally appeared in AAAI-15 [7].

1 Introduction

Coalition formation is the process by which a number of agents partition themselves into temporary teams (i.e. coalitions), where each coalition collaborates to achieve mutually beneficial results. Coalition formation is a well studied research area in multi-agent systems and has a wide range of potential applications, including: electronic auctions/market places; communication networks; the smart grid; grid computing; distributed vehicle routing; sensor networks; multi-agent planning; and computational trust [3].

Coalition formation can be divided into a three stage process [8]: (1) calculating the utility value of each possible coalition; (2) finding a satisfiable set of coalitions; and (3) dividing the utility values of the coalitions into a stable distribution (where no agent can object to its assigned payoff). The first stage of coalition formation requires the agents to calculate the utility value of each coalition, which can be computationally complex as the number of coalitions that can form given a population of n agents is $2^n - 1$. Furthermore, the complexity of calculating an individual coalition's value can vary, and potentially be exponential [9]. Thus, even if each agent only calculates the value of those coalitions that it can potentially form (i.e. $\frac{2^n}{2}$), then this can still result in a significant overlap of calculations, such that this redundancy can converge to 100%, as $\lim_{n\to\infty} \frac{2^n-1}{n2^{(n-1)}} = 0$.

Various studies have explored how to distribute coalition value calculations amongst agents, to reduce the computation cost for each agent, and possibly reduce the overall computation time [10–12, 6, 13]. Furthermore, as the agents themselves can potentially determine the value of the coalitions in which they participate, this eliminates the need for a trusted central authority/independent third-party responsible for determining the value calculations. However, existing approaches are not necessarily efficient: some only guarantee that every coalition value will be calculated *at least once*, potentially resulting in redundant calculations [10–12]; while others distribute the calculations unequally amongst the agents [13]. The approach given in [6] does not suffer from these deficiencies but allocates some coalitions to agents that do not appear in them, which can be undesirable in adversarial environments where deception may occur.

We present a novel algorithm for distributing coalition value calculations, named the *Distributed Coalition Generation* (DCG) algorithm, that addresses these limitations by allocating all coalitions to agents that appear in them and balancing the computational load approximately evenly (wrt. share size and number of operations) across the agents. This process is completed without inter-agent communication in a way that eliminates redundant coalition value calculations.

Furthermore, the DCG algorithm can be combined with other approaches to complete the coalition formation process. For example, if agents want to find a core/ϵ-core stable solution [5], then DCG can be used as input to algorithms, such as [2, 14], that determine such solutions.

In this paper, the distributed coalition value calculation problem is presented and related work critiqued in Section 2. Section 3 describes the underlying coalition ordering mechanism and provides an example, while the DCG algorithm is presented in Section 4. Proofs for the properties of this work are presented in Section 5 which focus on how: (a) all coalitions are assigned; and (b) all coalitions are assigned once and only once. After the DCG algorithm and related theory has been fully introduced, comparisons with related work are made in Section 6. Possible lines of future work are detailed in Section 7 and the paper concludes in Section 8.

2 Related Work

The *characteristic function game* model of coalition formation [5] is denoted: $\mathcal{G} = \langle N, v \rangle$ where $N = \{1, 2, ..., n\}$ is the set of agents, and v is the *characteristic function* ($v(2^N) \to \mathbb{R}$) which assigns every possible coalition a real numeric payoff. To find an *outcome* of a characteristic function game, the value of each coalition needs to be calculated and the coalitions compared. As it assumes that each coalition's value is static, deterministic and independent of the other coalitions that could form, a *characteristic* *function game* has the property that each coalition's value needs to be calculated only once.

This property was originally exploited by Shehory and Kraus [10–12] to reduce the number of redundant calculations. Instead of each agent calculating each coalition value in which it is a member, they introduced an algorithm (referred to here as SK) where each agent negotiated over which coalitions (that also comprised that agent) should be allocated to its value calculation share. However, this algorithm suffers from several limitations [6]: (i) many messages need to be sent between the agents to facilitate the negotiation; (ii) there is *no* guarantee that the number of coalition value calculations performed by each agent is approximately equal; and (iii) there is no guarantee that every coalition value is calculated *once and only once* (SK only guarantees that every coalition value will be calculated *at least once*).

Rahwan and Jennings argued that the SK algorithm utilised the resources of the system *inefficiently*, and addressed this by proposing the *Distributed Coalition Value Calculation* algorithm (*DCVC*) [6]. Their algorithm grouped coalitions into lists, and divided the lists into *shares*, one for each agent. They showed that: (i) no inter-agent communication was necessary; (ii) the agents' coalition value calculation shares were approximately equal; and (iii) each coalition value was calculated *once and only once*. However, a weakness of DCVC is that a coalition assigned to an agent i's share may not include i.

More recently, Vinyals *et al.* [13] proposed an algorithm (referred to as VBFR) that distributes the coalition value calculations to agents when they are connected in a network, with the property that each coalition must include member agents that are connected together in a graph. However their algorithm failed to evenly distribute the coalition value calculations (for a fully connected graph), because the number of calculations that an agent had to perform was correlated with an ID associated to each agent.

3 Preliminaries and Introductory Example

The DCG algorithm (in Section 4) exploits a novel method for representing and ordering coalitions, such that different coalitions can be allocated to each agent, in such a way as to facilitate the construction of *shares* (one per agent) that eliminate redundant coalition value calculations.

3.1 Ordering and Integer Partitions

In this paper, a coalition $C \subseteq \{1, 2, ..., n\}$ is represented as an ordered sequence of identifiers (IDs) that form a *coalition array*, where no agent appears more than once in any coalition, and where the coalition size $s = |C|$. An *integer increment value* between two contiguous agents i and j in a coalition array corresponds to the difference in the agents' IDs[1]. For example, if we have a coalition array $[3, 6, 1]$, then there is

[1] As agent IDs are in the range $[1, n]$, IDs modified using an integer increment will result in an ID modulo n. The agent ID n will be returned when the ID 0 is found because $0 \equiv n \pmod{n}$.

	L_3			
	$L_{3,\langle 3,0,0\rangle}$	$L_{3,\langle 2,1,0\rangle}$	$L_{3,\langle 2,0,1\rangle}$	$L_{3,\langle 1,1,1\rangle}$
CV_1	1,5,6	1,4,6	1,4,5	
CV_2	2,6,1	2,5,1	2,5,6	
CV_3	3,1,2	3,6,2	3,6,1	
CV_4	4,2,3	4,1,3	4,1,2	4,6,2
CV_5	5,3,4	5,2,4	5,2,3	5,1,3
CV_6	6,4,5	6,3,5	6,3,4	

Table 1. Coalition value calculation shares (CV) for all coalitions of size $s = 3$, when there are $n = 6$ agents.

one integer increment value between the ID pair $3, 6$ and one between $6, 1$. There is an additional increment between the last and the first agent IDs in the array; i.e. the ID pair $1, 3$. The *integer increment value* between two agents i and j can be decomposed into a *baseline increment* (which is assumed to be 1, since agent IDs are unique) and an *offset increment*, denoted $t_i = (j - i) - 1 \bmod n$ (i.e. integers modulo n). Thus, if $t_i = 0$, the difference between the IDs for agents i and j corresponds only to the baseline increment; whereas if $t_i \neq 0$, then t_i represents an additional offset increment. An *increment array* (IA) denoted $\underline{t} = \langle t_0, t_1, \ldots, t_{s-1}\rangle$ therefore represents the offset increments between the identifiers of the coalition array. For example, given the coalition array $[3, 6, 1]$, the corresponding IA will be $\langle 2, 0, 1\rangle$.

An *integer partition* of x is a combination of positive integers that add up to exactly x. The DCG algorithm uses integer partitions to identify the offset increments between consecutive pairs of IDs in the coalition array. The full set of integer partitions is denoted $\mathcal{I}(n-s)$; for example, given $n = 6$ and $s = 3$, $\mathcal{I}(n-s) = \{\{3\}, \{2, 1\}, \{1, 1, 1\}\}$. Increment arrays can be formed from an integer partition I for coalitions of size s, *only* when $I \in \mathcal{I}(n - s)$ and $|I| \leq s$, by including additional zero values to satisfy the property:

$$\sum_{i=0}^{s-1} t_i = (n - s)$$

For example, when $n = 6$ and $s = 3$, the integer partition $\{2, 1\}$ could be used to form various possible increment arrays: $\langle 2, 1, 0\rangle$, $\langle 2, 0, 1\rangle$, etc. The integer increment values corresponding to the increment array $\langle 2, 1, 0\rangle$ result from the two following coalition arrays $[1, 4, 6]$ and $[2, 5, 1]$, as the ID pairs $1, 4$ and $2, 5$ share $(2 + 1)$, whereas the ID pairs $4, 6$ and $5, 1$ share $(1 + 1)$. As IAs are shared between coalition arrays, the new ordering method introduced in this paper divides the coalitions into 2-dimensional lists $L_{s,\underline{t}}$.

Each *increment array* $\underline{t}$ represents the necessary offset increments from one agent ID of the coalition array to the next. For agent i to generate a coalition C assigned to itself using $\underline{t}$, the first element of the coalition array will be i to *motivate* i to compute the coalition's value. The second agent ID j in the coalition array will be $= (i + t_0 + 1) \bmod n$; and the third agent ID k will be $= ((i + t_0 + 1) + t_1 + 1) \bmod n$. This continues until the coalition's size s limit has been reached.

Table 1 presents a subset of the coalition arrays, grouped by IAs of size $s = 3$ for $n = 6$ agents. Each column represents a single list $L_{s,\underline{t}}$ for some IA $\underline{t}$, whereas the rows present the *coalition value calculation shares* (CVs) comprising the different coalition arrays with a common first element (where CV_i is agent i's share). The table represents all coalition arrays necessary for coalitions of size $s = 3$. To assign all of the coalitions, multiple IAs are needed; however, every coalition is assigned once and only once. Note that an integer partition may form more than one increment array; for example the two increment arrays $\langle 2, 1, 0\rangle$ and $\langle 2, 0, 1\rangle$ are formed from the $\{2,1\}$ integer partition.

Four different IAs are required for all the coalitions to be allocated in the above example. The IA $\underline{t}^x = \langle 2, 0, 1\rangle$ is valid as $\{2, 1\}$ is a candidate integer partition of $\mathcal{I}(6-3)$ that satisfies $|\{2, 1\}| \leq s = 3$. Yet as $|\{2, 1\}| \neq 3$, additional zeros are needed to fill up the IA to make the IA the required size s. If agent 2 used $\underline{t}^x$, the coalition array would comprise:
$= \{i, (i + t^x_0 + 1) \bmod n, ((i + t^x_0 + 1) + t^x_1 + 1) \bmod n\}$
$= \{2, (2 + 2 + 1) \bmod 6, ((2 + 2 + 1) + 0 + 1) \bmod 6\}$
$= \{2, 5, 0\} \equiv \{2, 5, 6\}$
In the above example, the ID 0 was generated. As $0 \equiv n \pmod n$, this is replaced with ID $= n = 6$ in this coalition.

Each IA should be used n times (once for each agent) unless the IA includes a sequence that is repeated throughout the IA. In Table 1, $\langle 1, 1, 1\rangle$ is the only IA with a repeated sequence, with $\{1\}$ being repeated $m = 3$ times. The number of times an IA with a repeating sequence should be used relates to the size of the repeating sequence and is given by r (introduced in the next subsection). The choice of agents that should use this type of IA will depend on the allocation of other coalitions, and is described in Section 4.

If any other IA was used other than the ones listed in Table 1, it would result in a coalition's value being calculated more than once. For example, if agent 6 used $t^y = \langle 1, 2, 0\rangle$, the coalition array $[6, 2, 5]$ would be generated despite this coalition being generated by agent 2 using $t^x = \langle 2, 0, 1\rangle$.

3.2 A Distributed Method for Coalition Generation

The *period of* $\underline{t}$, denoted by $\pi(\underline{t})$ is defined as:

$$\min_{1\leq p\leq s} \underline{t} = \langle t_0, t_1, ..., t_{p-1}, t_0, t_1, ..., t_{p-1}, ..., t_0, t_1, ..., t_{p-1}\rangle$$

Hence, $\underline{t}$ is formed by m identical copies of a sequence of length $\pi(\underline{t})$. Given $C \subseteq \{1, 2, \ldots, n\}$, an agent ID ag generates C from ag if $C = \{ag_1, ag_2, \ldots, ag_s\}$ and:

$$ag_i = \begin{cases} ag & \text{if } i = 1 \\ (ag + \phi_i) \mod n & \text{if } (ag + \phi_i) \mod n \neq 0 \\ n & \text{if } (ag + \phi_i) \mod n = 0 \end{cases}$$

where:

$$\phi_i = \sum_{k=0}^{i-2} t_k + (i-1)$$

Additionally, $C(ag, \underline{t})$ denotes the subset of $\{1, 2, \ldots, n\}$ generated by the IA $\underline{t}$ from agent ag. It is possible to demonstrate that each $\underline{t}$ only needs to be used by $r =$

$(n \times \pi(\underline{t}))/s$ different agents. If more than r agents use $\underline{t}$ to generate a coalition, then repeated coalitions will be generated. For example, if the chosen IA from Table 1 is $\underline{t}^q = \langle 1,1,1\rangle$ then $r = (n \times \pi(\underline{t}^q))/s = (6 \times 1)/3 = 6/3 = 2$ agents should use $\underline{t}^q$, which is true as any other agent using $\underline{t}^q$ would repeat either coalition $\{1,3,5\}$ or $\{2,4,6\}$. Finally, if $\underline{t}^x$ and $\underline{t}^y$ generate the same coalition C for two different agents $i, j \in C$ (i.e. $C(i,\underline{t}^x) = C(j,\underline{t}^y)$), then $\underline{t}^x$ and $\underline{t}^y$ are classified as belonging to the same *equivalence class*, denoted $\underline{t}^x \approx \underline{t}^y$. For example, the IAs in the following equation belong to the same equivalence class: $C(2,\langle 2,0,1\rangle) = C(6,\langle 1,2,0\rangle) = \{2,5,6\}$. We write $\underline{t}^x \approx \underline{t}^y$ when $t^x = \langle t^y_k, ..., t^y_{s-1}, t^y_0, ..., t^y_{k-1}\rangle$ for some $0 \le k \le s-1$. Section 5 proves that rather than considering *every* possible IA, it suffices only to consider a *single* representative from each equivalence class $\approx$.

4 The Distributed Coalition Generation Algorithm

The DCG algorithm used by each agent i to generate all of its coalitions in its coalition value calculation share is presented in Algorithm 1. The $balance$ pointer is initialised, and references the next agent to calculate a coalition's value. It is similar to the α pointer in [6], and its use by SingleSize ensures that all the agents' shares are either equal in size or have a difference in size of $+/-$ 1. Line 21 allows only the next r agents to calculate a coalition, starting from the agent referred to by the $balance$ pointer, and continuing in an ascending order. If agent n (i.e. the agent with the highest ID) is assigned to calculate a coalition's value, and z more coalition value calculations are required, then line 21 also allows agents 1 to z to calculate a coalition's value.

The DCG algorithm calls SingleSize for every possible size of the coalitions. This function returns all coalitions of size s in agent i's share. The while loop (lines 18-30) uses a black box function, named build (lines 17 and 29), to determine the next IA of a new equivalence class. This new IA is used to generate another coalition for agent i, if i is one of the next r agents to be assigned a coalition (lines 19 to 23). Regardless of who uses the new IA, the $balance$ pointer is updated (lines 24-27). When the build function returns $null$, this indicates that all the coalitions of size s for agent i's calculation share have been found.

A possible implementation of the black box build function is presented in Algorithm 2. It relies on an indexing scheme that defines the function $place : \underline{t} \to \mathbb{N}_0$ to map an IA $\underline{t} = \langle t_0, \ldots, t_{s-1}\rangle$ to a non-negative integer; i.e.

$$place(\langle t_0, t_1, \ldots, t_{s-1}\rangle) \;=\; \sum_{i=0}^{s-1} (t_i \times (n-s+1)^i)$$

Informally, $place(\underline{t})$ treats $\underline{t}$ as an integer expressed in base $n-s+1$. With this convention, the number of distinct IAs is bounded by $(n-s+1)^s$. For example, using the $place$ function for the IA $t^x = \langle 2,1,0\rangle$, would give the value $val = 2+4+0 = 6$ because: $(t^x_0 = 2) \times (6-3+1)^0 = 2$; $(t^x_1 = 1) \times (6-3+1)^1 = 4$; and $(t^x_2 = 0) \times (6-3+1)^2 = 0$.

This build function finds the next index value k ($> y$) of a representative of an equivalence class not used so far, while decode returns the s-tuple corresponding to

Algorithm 1: The Distributed Coalition Generation (DCG) Algorithm

1: **global int** $balance\ :=\ 1$;
2:
3: **function** `DCG` (**int** `n, i`) **returns** $\langle \mathcal{C}^s \rangle_{s=1}^{n}$;
4: **Input:** $\langle n, i \rangle$ $(1 \leq i \leq n)$; where n is the number of agents and i the agent ID.
5: **Output:** $\langle \mathcal{C}^s \rangle_{s=1}^{n}$; $\mathcal{C}^s = \{C_1^s, C_2^s, \ldots, C_k^s\}$, $C_j^s \subseteq \{1, 2, \ldots, n\}$, $|C_j^s| = s$; where $\langle \mathcal{C}^s \rangle_{s=1}^{n}$ is the collection of coalitions, of all sizes $1 \leq s \leq n$, assigned to agent i's share.
6: **begin**
7: **for** (**int** $s = 1;\ s \leq n;\ s++$) **do**
8:     $\mathcal{C}^s\ :=$ `SingleSize(n,s,i);`
9: **end for**
10: **return** $\langle \mathcal{C}^1, \ldots, \mathcal{C}^n \rangle$;
11: **end**;
12:
13: **function** `SingleSize` (**int** `n, s, i`) **returns** $\mathcal{C}^i$;
14: **Input:** $\langle n, s, i \rangle$ $(1 \leq s \leq n)$; where n is the number of agents, s the size of the coalitions and i the agent ID.
15: **Output:** $\mathcal{C}^i$; $\mathcal{C}^i = \{C_1^i, C_2^i, \ldots, C_k^i\}$, $C_j^i \subseteq \{1, 2, \ldots, n\}$, $|C_j^i| = s$; where $\mathcal{C}^i$ is the collection of coalitions (of size s) assigned to agent i's share.
16: **begin**
17: $t^y\ :=$ `build(n,s,0);`
18: **while** ($t^y \neq null$) **do**
19:     $p\ :=\ \pi(\underline{t})$;
20:     $r\ :=\ (n \times p)/s$;
21:     **if** ($balance \leq i < balance + r$) **or** ($balance + r > n$ **and** $1 \leq i < balance + r - n$) **then**
22:         $C\ :=\ C(i, \underline{t})$;
23:         $\mathcal{C}^i\ := \mathcal{C}^i \cup C$;
24:         $balance := balance + r$;
25:         **if** $balance > n$ **then**
26:             $balance\ :=\ balance - n$;
27:         **end if**
28:     **end if**
29:     $t^y\ :=$ `build(n,s,y+1);`
30: **end while**
31: **return** $\langle \mathcal{C}_1^i, \ldots, \mathcal{C}_k^i \rangle$;
32: **end**

the index value $val = k$. Each element of the s-tuple returned by `decode` is at least 0 and at most $n - s$; however, the s-tuple is not necessarily an IA as defined earlier (as it may not sum to $n - s$). This `build` function filters out any s-tuples returned by the `decode` function (lines 19-20) that either: (i) do not sum to $n - s$; or (ii) are a member of an equivalence class previously used. This `build` function knows which equivalence classes have been used so far since all the indexes of IAs of the same equivalence class are marked as used in the Boolean $Used$ array (lines 21-25). When this `build` function finds an s-tuple that is not filtered out by (i) and (ii), then this

Algorithm 2: One possible method to find representative IAs from equivalence classes of $\approx$.

1: **function** `decode` (**int** `n, s, k`) **returns** s-tuple;
2: **Input:** $\langle n, s, k \rangle$; where n is the number of agents, s is the size of the coalition ($1 \leq s \leq n$) and k is the index position to convert to an s-tuple.
3: **Output:** $\langle t_0, t_1, \ldots, t_{s-1} \rangle$ with $\sum_{i=0}^{s-1} t_i \times (n-s+1)^i \ = \ k$
4: **begin**
5: $val \ := \ k$;
6: **for** (**int** $index = 0; \ index < s; \ index++$) **do**
7: $\quad t_{index} \ := \ \mathbf{remainder}(val, n-s+1)$;
8: $\quad val \ := \ (val - t_{index})/(n-s+1)$;
9: **end for**
10: **return** $\langle t_0, t_1, \ldots, t_{s-1} \rangle$;
11: **end**;
12:
13: **function** `build` (**int** `n, s, y`) **returns** s-tuple;
14: **Input:** $\langle n, s, y \rangle$; where n is the number of agents, s is the size of the coalition ($1 \leq s \leq n$) and y is the index position to start searching from ($0 \leq y < (n-s+1)^{s+1}$).
15: **Output:** $\langle t_0^k, t_1^k, \ldots, t_{s-1}^k \rangle$; the next IA of a class $[\approx]_i$ not used so far.
16: **begin**
17: **for** (**int** $k = y; \ k < (n-s+1)^{s+1}; \ k++$) **do**
18: $\quad t^k = \langle t_0^k, \ldots, t_{s-1}^k \rangle \ := $ `decode(n,s,k);`
19: $\quad$ **int** $tot \ := \ \sum_{i=0}^{s-1} t_i^k$;
20: $\quad$ **if** $tot = n - s$ and $\neg Used[k]$ **then**
21: $\quad\quad Used[k] \ := $ **true**
22: $\quad\quad p \ := \ \pi(\underline{t}^k)$;
23: $\quad\quad$ **for** $(z = p-1; \ z > 0; \ z--)$ **do**
24: $\quad\quad\quad Used[place(\langle t_z^k, \ldots, t_{s-1}^k, t_0^k, \ldots, t_{z-1}^k \rangle)]$:= **true**;
25: $\quad\quad$ **end for**
26: $\quad\quad$ **return** t^k;
27: $\quad$ **end if**
28: **end for**
29: **return** $null$;
30: **end**;

s-tuple is an IA of an equivalence class not previously used, and is returned (line 26). If no IA of a new equivalence class can be found, then $null$ is returned (line 29).

To illustrate how the `decode` function works: decoding $val = 6$ gives as expected $t^x = \langle 2, 1, 0 \rangle$, as: $t_0^x = \mathbf{remainder}(6, 6-3+1) = \mathbf{remainder}(6, 4) = 2$ while val becomes $val := (6-2)/(6-3+1) = 4/4 = 1$; $t_1^x = \mathbf{remainder}(1, 6-3+1) = \mathbf{remainder}(1, 4) = 1$ while val becomes $val := (1-1)/(6-3+1) = 0/4 = 0$; and $t_2^x = \mathbf{remainder}(0, 6-3+1) = \mathbf{remainder}(0, 4) = 0$.

Finally, if all the agents are required to only store one coalition in memory at a time (i.e. perform `DCG` with minimal memory requirements), then the following modifications to Algorithm 1 need to be made: (a) do not use the $\langle \mathcal{C}^s \rangle_{s=1}^n$ data object of line 3 and 10; (b) do not use the $\mathcal{C}^i$ data object of line 15 and 31; and (c) replace line 23 with *perform* $v(C)$.

5 Theoretical Evaluation

To evaluate the DCG algorithm, we prove that the algorithm will generate all of the coalitions possible for a community of n agents; and that each coalition is assigned once and only once, thus eliminating redundancy. We start with the following Lemma that shows that all coalitions will be generated:

Lemma 1. *Let $C \subseteq \{1, 2, \ldots, n\}$ with $|C| = s$. There is an IA $\underline{t}$ and $i \in C$ such that $C = C(i, \underline{t})$.*

Proof. Let $C = \{x, x_0, x_1, \ldots, x_{s-2}\}$. Without loss of generality we may assume:

$$x \;<\; x_0 \;<\; x_1 \;<\; \cdots \;<\; x_i \;<\; x_{i+1} \;<\; \cdots \;<\; x_{s-2}$$

It suffices to find $\underline{t} = \langle t_0, \ldots, t_{s-1} \rangle$ with $C(x, \underline{t}) = C$ and $\sum_{i=0}^{s-1} t_i = n - s$, i.e.:

$$\begin{aligned} t_0 &= x_0 - (x+1) \\ \ldots & \\ t_k &= x_k - \left(\sum_{i=0}^{k-1} t_i + x + k + 1 \right) \\ \ldots & \\ t_{s-1} &= n - \sum_{i=0}^{s-2} t_i \end{aligned}$$

The next Lemma shows that IAs of an equivalent class generate the same coalitions:

Lemma 2. *If $\underline{t} \approx \underline{u}$ then*

$$\bigcup_{i=1}^{n} \{\ C(i, \underline{t})\ \} \;\;=\;\; \bigcup_{i=1}^{n} \{\ C(i, \underline{u})\ \}$$

Proof. Without loss of generality we may assume that $\underline{t} \approx \underline{u}$ is witnessed by the choice $k = s - 1$, i.e.

$$\langle u_0, u_1, \ldots, u_{s-1} \rangle \;\;=\;\; \langle t_{s-1},\ t_0,\ t_1, \ldots, t_{s-3},\ t_{s-2} \rangle$$

Define φ_r for $1 \leq r \leq s+1$

$$\varphi_r \;=\; \begin{cases} 0 & \text{if } \ r = 1 \\ \sum_{k=0}^{r-2} \ t_k \ + \ (r-1) & \text{if } \ 2 \leq r \leq s+1 \end{cases}$$

Note that $\varphi_{s+1} = n$. Define ψ_r for $1 \leq r \leq s+1$ via

$$\psi_r \;=\; \begin{cases} 0 & \text{if } \ r = 1 \\ t_{s-1} + \sum_{k=0}^{r-3} \ t_k + r - 1 & \text{if } \ 2 \leq r \leq s+1 \end{cases}$$

Comparing respective terms we see that for all $2 \leq k \leq s$:

$$\psi_k \;=\; \varphi_k \;+\; (t_{s-1} \;-\; t_{k-2})$$

This leads to the following:

$$C(i,\underline{t}) = \begin{cases} C(n - t_{s-1} + i{-}1, \underline{u}) & \text{if } 1 \leq i \leq t_{s-1}{+}1 \\ C(i - t_{s-1}{-}1, \underline{u}) & \text{if } t_{s-1}{+}2 \leq i \leq n \end{cases}$$

To see this, consider when $1 \leq i \leq t_{s-1} + 1$. We have,

$$C(i,\underline{t}) \;=\; \bigcup_{k=1}^{s} \{i + \varphi_k\}$$

When $\lambda = n - t_{s-1} + i - 1$ the above is claimed to be:

$$\begin{aligned} C(\lambda,\underline{u}) &= \bigcup_{k=1}^{s} \{\lambda + \psi_k\} \\ &= \{\lambda\} \cup \bigcup_{k=2}^{s} \{\lambda + \varphi_k + t_{s-1} - t_{k-2}\} \end{aligned}$$

Consider the terms:

$$n - t_{s-1} + i - 1 + \varphi_k + t_{s-1} - t_{k-2}$$

For $2 \leq k \leq s$, from the fact that $\varphi_k = \sum_{j=0}^{k-2} t_j + k - 1$, these are equal to:

$$n - t_{s-1} + i - 1 + \sum_{j=0}^{k-3} t_j \;+\; k - 1 \;+\; t_{s-1} \;\;=\;\; n + i + \varphi_{k-1}$$

In total we have, for $1 \leq i \leq t_{s-1} + 1$:

$$\begin{aligned} C(i,\underline{t}) &= \bigcup_{k=1}^{s} \{i + \varphi_k\} \\ C(\lambda,\underline{u}) &= \{n - t_{s-1} + i - 1\} \;\cup \\ & \quad\; \bigcup_{k=2}^{s} \{n + i + \varphi_{k-1}\} \end{aligned}$$

Given that $n + i + \varphi_{k-1}$ and $i + \varphi_{k-1}$ are congruent modulo n, which are elements of $C(i,\underline{t})$, the only terms unaccounted for are $\{n - t_{s-1} + i - 1\} \in C(\lambda,\underline{u})$ and $\{i + \varphi_s\} \in C(i,\underline{t})$. For these, however,

$$\begin{aligned} i + \varphi_s &= i + \sum_{j=0}^{s-2} t_j \;+\; s - 1 \\ &= i + (n - s - t_{s-1}) + s - 1 \\ &= i + n - t_{s-1} - 1 \end{aligned}$$

When $t_{s-1} + 2 \leq i \leq n$ and $\omega = i - t_{s-1} - 1$, it is claimed that

$$C(i, \underline{t}) \;=\; \bigcup_{k=1}^{s} \{i + \varphi_k\}$$

corresponds to:

$$\begin{aligned} C(\omega, \underline{u}) \;&=\; \bigcup_{k=1}^{s} \{\omega + \psi_k\} \\ &=\; \{\omega\} \cup \bigcup_{k=2}^{s} \{\omega + \varphi_k + t_{s-1} - t_{k-2}\} \end{aligned}$$

Now inspecting the terms for $2 \leq k \leq s$:

$$i - t_{s-1} - 1 + \varphi_k + t_{s-1} - t_{k-2}$$

These, again, simplify to $i + \varphi_{k-1}$, so that

$$C(i, \underline{t}) = \bigcup_{k=1}^{s} \{i + \varphi_k\}$$

$$C(\omega, \underline{u}) = \{\omega\} \cup \bigcup_{k=2}^{s} \{i + \varphi_{k-1}\}$$

When $1 \leq k \leq s-1$, the term $i+\varphi_k$ appears in both $C(i, \underline{t})$ and $C(\omega, \underline{u})$. For the terms $i + \varphi_s \in C(i, \underline{t})$ and $\omega \in C(\omega, \underline{u})$ we have already seen that $i + \varphi_s = i + n - t_{s-1} - 1$ is congruent modulo n with $i - t_{s-1} - 1 = \omega$. This establishes the claim of the Lemma.

Lemma 2 shows that IAs belonging to the same equivalence class of $\approx$ generate exactly the same set of coalitions. Our next two results establish that this is the *only* way in which two distinct IAs can produce the same coalition.

Lemma 3. *Let $C = \{x_1,\ x_2, \ldots, x_i, \ldots, x_s\}$ and $C = C(x_i, \underline{t})$ with $x_i < x_{i+1}$ for all $1 \leq i < s$. There is an IA $\underline{u}$, for which $\underline{t} \approx \underline{u}$ and $C(x_1, \underline{u})$ generates C in strictly increasing ordering of x_i, i.e. $\forall\ 2 \leq i \leq s$:*

$$x_i \in \left\{ x_1 + \sum_{k=0}^{i-2} u_i + i - 1, x_1 + \sum_{k=0}^{i-2} u_i + i - 1 - n \right\}$$

Proof. Given $\underline{t}$, suppose:

$$C(x_i, \underline{t}) \;=\; \{x_1, x_2, \ldots, x_i, \ldots, x_s\}$$

First observe that the terms

$$x_i \;+\; \sum_{k=0}^{r-2} t_k \;+\; r - 1 \;=\; x_i \;+\; \varphi_r$$

are strictly increasing. It follows that if $x_i \neq x_1$ there must be a *unique* index, p, for which:

$$x_i \; + \; \varphi_r \quad \text{is} \quad \begin{cases} \leq \; n & \text{if } r < p \\ > n & \text{if } r \geq p \end{cases}$$

In consequence, $x_1 \; = \; x_i + \varphi_p - n$ otherwise we cannot have $x_1 \in C(x_i, \underline{t})$. More generally, it must hold that:

$$\begin{array}{l} x_k = x_i + \varphi_{p+k-1} - n \;\; \forall \, 1 \; \leq \; k \; \leq \; s - p + 1 \\ x_k = x_i + \varphi_{p-(s-k)-1} \;\; \forall \, s - p + 2 \; \leq k \; \leq s \end{array}$$

This, however, corresponds to the behaviour of the IA $\underline{u}$, whose definition is:

$$\underline{u} \;\; = \;\; \langle t_{p+1}, \; t_{p+2}, \; \ldots, \; t_{p+k}, \; \ldots, \; t_s, \; t_0, \; \ldots, \; t_p \rangle$$

Clearly $\underline{u} \approx \underline{t}$ and $C(x_1, \underline{u}) = C(x_i, \underline{t})$ as claimed.

As a consequence of Lemma 3 we obtain:

Lemma 4. *Let $\underline{t}$ and $\underline{u}$ be IAs for which $\underline{t} \not\approx \underline{u}$, then:*

$$\bigcup_{i=1}^{n} \; \{C(i, \underline{t})\} \; \bigcap \; \bigcup_{i=1}^{n} \; \{C(i, \underline{u})\} \;\; = \;\; \emptyset$$

Proof. Suppose the contrary and that $C = \{x_1, \ldots, x_s\}$ can be generated by $C(x_i, \underline{t})$ and $C(x_j, \underline{u})$ for choices of $\underline{t}$ and $\underline{u}$ of different equivalence classes of $\approx$. As a consequence of Lemma 3 we know that there are IAs, $\underline{t}'$ and $\underline{u}'$ for which:

$$\begin{array}{l} \underline{t} \approx \underline{t}', \; \underline{u} \approx \underline{u}' \text{ and} \\ C(x_1, \underline{t}') \; = \; C(x_i, \underline{t}) \; = \; C(x_j, \underline{u}) \; = \; C(x_1, \underline{u}') \end{array}$$

Furthermore, $C(x_1, \underline{t}')$ and $C(x_1, \underline{u}')$ produce the elements of C in increasing ordering of $x_i \in C$. This, however, is only possible if

$$x_i \;\; = \;\; x_1 \; + \; \sum_{k=0}^{i-2} t'_k \; + \; i - 1 \;\; = \;\; x_1 \; + \; \sum_{k=0}^{i-2} u'_k \; + \; i - 1$$

that is, $t'_i = u'_i$ for each $0 \leq i \leq s - 1$. This, however, implies that $\underline{t} \approx \underline{u}$, in contradiction to our starting premise.

Finally, the following Theorem states that each IA chosen by the DCG algorithm is required to be used r times.

Theorem 1. *For any IA $\underline{t}$, and for all $1 \leq i \leq j \leq n$,*

$$C(i, \underline{t}) = C(j, \underline{t}) \Leftrightarrow$$

$$\exists \, 0 \leq r \leq \frac{(n-i)s}{n\pi(\underline{t})} \; : j = i + r \left(\frac{n\pi(\underline{t})}{s} \right)$$

Proof. We show that if and only if $j = i + (rn\pi(\underline{t})/s)$ for r in the range stated, then $C(i, \underline{t})$ and $C(j, \underline{t})$ are identical. This is completed using the previously introduced Lemmas. The proof is omitted for brevity.

Property	SK	DCVC	VBFR	DCG
Eliminates Comms.	No	Yes	Yes	*Yes*
Approx. Equal Shares	No	Yes	No	*Yes*
Eliminates Redundancy	No	Yes	Yes	*Yes*
Equal Operations	No	No	No	*Yes*
Coalition Self Assessed	Yes	No	Yes	*Yes*

Table 2. A comparison of the properties of the DCG algorithm and the three related algorithms (Section 2).

6 Comparison to Related Work

Table 2 shows that unlike related approaches (Section 2), the DCG algorithm satisfies *all* of the following properties:

i *Communication is eliminated* as the algorithm determines the allocation of coalitions to each agent. Thus, no further coordination between the agents is necessary.
ii The inclusion of the *balance* pointer ensures that agents calculate *approximately equal* shares of coalitions, as it minimises the maximum difference between the size of any two agents' shares (to one).
iii *Redundancy is eliminated* as: (a) only one IA of each equivalence class is used; (b) each IA of a new equivalence class is used only r times; and (c) IAs of different equivalence classes cannot generate the same coalition.
iv An *equal number of operations* will be performed by agents that are allocated equal sized shares, as each IA requires the exact same number of operations of additions to find the corresponding coalition.
v *Every coalition is assessed by one of its member agents* as the first agent generated for each coalition in agent i's allocated share is agent i itself.

The DCG algorithm is better suited for adversarial environments compared to the related approaches, as one of the agents in each coalition will be allocated that coalition's value calculation, while DCG additionally balances the value calculations efficiently across the community of agents in a manner that incurs no communication overheads.

7 Future Work

There are many intriguing avenues for future work. The DCG algorithm could be exploited by distributed coalition structure generation (CSG) algorithms to find the optimal coalition structure. The first such algorithm for solving the CSG problem optimally was D-IP [4], which used the DCVC algorithm [6] as input. Using DCG as input to a distributed CSG algorithm could result in new and interesting properties compared to D-IP.

In competitive or adversarial environments, there is a possibility that self-interested agents may attempt to deceive others to gain an advantage. Within our DCG algorithm, an agent *may* benefit by artificially increasing the value of one (or more) of its coalition value calculations; however, we considered deception out of scope for this paper, and focussed instead on the properties of the DCG algorithm itself. An interesting line of research would be to investigate how to make DCG *incentive compatible*[2] or *near-incentive compatible*[3] [1].

Finally, an investigation is underway to determine an optimal approach (in terms of efficiency) for the black box build function that is used within DCG. Preliminary results suggest that the optimal approach will run in time according to the number of equivalence classes of IAs and with storage requirements according to the number of agents.

8 Conclusions

In this paper, the *Distributed Coalition Generation Algorithm* was presented that distributes the coalition value calculations across a community of agents. This algorithm is based on a mechanism that exploits integer partitions to generate increment arrays that represent the difference between agent IDs when a coalition is represented in an ordered sequence. The algorithm has been evaluated theoretically, resulting in the proofs that: (a) all coalitions are assigned; and (b) each coalition is assigned once and only once.

9 Acknowledgements

We thank the AAAI-15 anonymous reviewers for their useful comments.

References

1. Blankenburg, B., Dash, R.K., Ramchurn, S.D., Klusch, M., Jennings, N.R.: Trusted kernel-based coalition formation. In: Proceedings of the 4th International Conference on Autonomous Agents and Multiagent Systems (AAMAS). pp. 989–996 (2005)
2. Cesco, J.C.: A convergent transfer scheme to the core of a TU-game. Revista de Matemticas Aplicadas 19, 23–35 (1998)
3. Chalkiadakis, G., Elkind, E., Wooldridge, M.: Computational Aspects of Cooperative Game Theory. Morgan & Claypool Publishers (2011)
4. Michalak, T., Sroka, J., Rahwan, T., Wooldridge, M., McBurney, P., Jennings, N.R.: A distributed algorithm for anytime coalition structure generation. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent System (AAMAS). pp. 1007–1014 (2010)
5. Osbourne, M.J., Rubinstein, A.: A Course in Game Theory. MIT Press (1994)
6. Rahwan, T., Jennings, N.R.: An algorithm for distributing coalition value calculations among cooperating agents. Artificial Intelligence 171, 535–567 (2007)
7. Riley, L., Atkinson, K., Dunne, P., Payne, T.R.: Distributing coalition value calculations to coalition members. In: Proceedings of the 29th Conference on Artificial Intelligence (AAAI). pp. 2117–2123 (2015)
8. Sandholm, T.W., Larson, K.S., Andersson, M., Shehory, O., Tohme, F.: Coalition structure generation with worst case guarantees. Artificial Intelligence 111, 209–238 (1999)

[2] Where the agents fare best when they are truthful.

[3] Where the agents cannot determine how to lie profitably.

9. Sandholm, T.W., Lesser, V.R.: Coalitions among computationally bounded agents. Artificial Intelligence 94, 99–137 (1997)
10. Shehory, O., Kraus, S.: Task allocation via coalition formation among autonomous-agents. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI). pp. 655–661 (1995)
11. Shehory, O., Kraus, S.: Formation of overlapping coalitions for precedence-order task-execution among autonomous agents. In: Proceedings of the 2nd International Conference on Multiagent Systems (ICMAS). pp. 330–337 (1996)
12. Shehory, O., Kraus, S.: Methods for task allocation via agent coalition formation. Artificial Intelligence 101, 165–200 (1998)
13. Vinyals, M., Bistaffa, F., Farinelli, A., Rogers, A.: Coalitional energy purchasing in the smart grid. In: Proceedings of the IEEE International Energy Conference & Exhibition (ENERGYCON). pp. 848–853 (2012)
14. Wu, L.S.Y.: A dynamic theory for the class of games with nonempty cores. SIAM Journal on Applied Mathematics 32, 328–338 (1977)

Graphical Hedonic Games of Bounded Treewidth*

Dominik Peters
dominik.peters@cs.ox.ac.uk

Department of Computer Science, University of Oxford, UK

Abstract. Hedonic games are a well-studied model of coalition formation, in which selfish agents are partitioned into disjoint sets and agents care about the make-up of the coalition they end up in. The computational problems of finding stable, optimal, or fair outcomes tend to be computationally intractable in even severely restricted instances of hedonic games. We introduce the notion of a graphical hedonic game and show that, in contrast, on classes of graphical hedonic games whose underlying graphs are of bounded treewidth and degree, such problems become easy. In particular, problems that can be specified through quantification over agents, coalitions, and (connected) partitions can be decided in linear time. The proof is by reduction to monadic second order logic. We also provide faster algorithms in special cases, and show that the extra condition of the degree bound cannot be dropped. Finally, we note that the problem of allocating indivisible goods can be modelled as a hedonic game, so that our results imply tractability of finding fair and efficient allocations on appropriately restricted instances.

1 Introduction

Hedonic games, first studied by Banerjee et al. [2001] and Bogomolnaia and Jackson [2002], provide a general framework for the study of *coalition formation*. Hedonic games subsume the well-studied *matching problems* (stable marriage, stable roommates, hospital-residents), but are able to express more general preference structures. They have been applied to problems in public good provision, voting, and clustering, and, as we show below, they also encapsulate a variety of allocation problems.

A *hedonic game* consists of a set N of agents, each of whom has a preference ordering over all coalitions $S \subseteq N$ containing her. The outcome of such a game is a *partition* of the agent set into disjoint coalitions, with agents preferring those partitions in which they are in a preferred coalition.

Unfortunately, it has turned out that many key questions about hedonic games are computationally hard to answer. For example, it is typically NP-complete to decide whether a hedonic game admits a *Nash stable* outcome; it is typically

NP-hard to maximise social welfare; and it is often even Σ_2^p-complete to identify hedonic games with non-empty *core*. See Ballester [2004], Sung and Dimitrov [2010], Woeginger [2013], and Peters and Elkind [2015] for a selection of such results.

A standard criticism of hardness results such as these is that they apply only in the *worst case*. Instances arising in practice can be expected to show much more structure than the highly contrived instances produced in 3SAT-reductions. In non-cooperative game theory, *graphical games* [Kearns et al., 2001] are an influential way to allow formalisation of the notion of a 'structured' game. In a graphical game, agents form the vertices of an undirected graph, and each agent's utility function only depends on the actions taken by her neighbours. The underlying graph can guide algorithms in finding a stable outcome, and imposing restrictions on the graph topology can yield tractability [Gottlob et al., 2005].

A particularly successful restriction on graph topology is *bounded treewidth*. The treewidth of a graph [Robertson and Seymour, 1986] measures how 'tree-like' a given graph is. Many NP-hard problems become fixed-parameter tractable with respect to the treewidth of some graph naturally associated with the problem instances. Indeed, dynamic programming over a given *tree decomposition* often yields algorithms that are exponential in the treewidth, but *linear* in the problem size. See Bodlaender's (1994) 'tourist guide through treewidth' for an introduction to this area.

The treewidth approach was first applied to the domain of hedonic games by Elkind and Wooldridge [2009], who propose a representation formalism for hedonic games called *hedonic coalition nets* that expresses agents' preferences by weighted boolean formulas. They also introduce a notion of treewidth for hedonic games—very close to ours—and show that (when numbers in the input are polynomially bounded) it is fixed-parameter tractable with respect to treewidth to decide whether a given partition is core-stable.

In this paper, we study *graphical hedonic games*, which we define in analogy to graphical games. Here, agents are again arranged in an underlying graph, and need to be partitioned into coalitions. Every agent only cares about which of her neighbours are in the same coalition as her. Every hedonic game can be made graphical by introducing edges whenever one agent's utility depends on the other's presence. We will then consider graphical hedonic games whose agent graphs have bounded treewidth and bounded degrees.

In the context of hedonic games, restricting treewidth and also degrees in the underlying social network seems particularly natural. Consider for example *Dunbar's number* [Dunbar, 1992], a suggested limit on the number of individuals that a single human being can maintain stable social relationships with. This number has been suggested to lie between 100 and 250, which gives us a natural bound on the degree of any social network. Intuitively, it also seems sensible to suppose that social networks have relatively small treewidth, though see Adcock et al. [2013] who find mixed empirical support for this proposition.

We show that when restricted to a class of graphical hedonic games whose agent graphs have bounded treewidth and bounded degrees, many standard problems related to these hedonic games become linear-time solvable. More precisely, by a somewhat involved translation to monadic second-order logic and by appealing to Courcelle's theorem, it follows that we can decide in linear time whether a given such hedonic game satisfies any logical sentence of what we call *HG-logic*, which allows quantification over partitions, coalitions, and agents. Using this approach and on this restricted domain, we can efficiently find stable or fair outcomes of a hedonic game for all notions of stability that are commonly discussed in the literature.

We also show that HG-logic is expressive enough to capture problems that would at first appear to lie outside the domain of hedonic games, such as the problem of fair and efficient allocation of indivisible goods. This implies that questions regarding those problems can also be answered efficiently when we restrict treewidth and degree.

Our appeal to Courcelle's meta-algorithmic result, while powerful, comes at the cost of the hidden 'constant factor' growing dramatically as the treewidth and degree of the hedonic game increase (indeed, this growth cannot be bounded by an elementary function unless P = NP). To show that despite this the restriction to bounded treewidth is useful in practice, we present a variety of more specific problems that can be solved in the more manageable runtime $\tilde{O}(2^{kd^2}n)$, where k is the bound on treewidth, d is the bound on the degrees, and n is the number of agents. Such results suggest that finding good outcomes of hedonic games should be tractable on instances arising in practice.

Many standard NP-complete problems defined on graphs become easy when bounding treewidth, without requiring a further restriction on the degrees of the graph. In the last section we show that this is not the case for graphical hedonic games: we give reductions showing that finding a core-stable partition and similar problems remain NP-hard even on games of treewidth 1 or 2 (but when degrees are unbounded).

2 Preliminaries

A *hedonic game* $\langle N, (\succcurlyeq_i)_{i \in N}\rangle$ is given by a finite set N of agents, and for each agent $i \in N$ a complete and transitive preference relation over $\mathcal{N}_i = \{S \subseteq N : i \in S\}$. We let $\succ_i$ and $\sim_i$ denote the strict and indifference parts of $\succcurlyeq_i$. The outcome of a hedonic game is a *partition* π of the agent set into disjoint coalitions. We write $\pi(i)$ for the coalition $S \in \pi$ that contains $i \in N$. We are interested in finding partitions that are *stable*, *optimal*, and/or *fair*. There are multiple ways of formalising these goals. For example, a partition π is *Nash stable* if no agent wants to join another (possibly empty) coalition of π, that is $\pi(i) \succcurlyeq_i S \cup \{i\}$ for all $S \in \pi \cup \{\varnothing\}$ and all $i \in N$. It is *individually stable* if there is no agent i and coalition $S \in \pi \cup \{\varnothing\}$ such that $S \cup \{i\} \succ_i \pi(i)$ and $S \cup \{i\} \succcurlyeq_j S$ for all $j \in S$. We say π is *core-stable* if there is no non-empty *blocking* coalition $S \subseteq N$ such that $S \succ_i \pi(i)$ for each $i \in S$. We say π is *strict*-core-stable if there is no non-empty coalition $S \subseteq N$ such that $S \succcurlyeq_i \pi(i)$ for each $i \in S$, with at least one preference strict. For fairness, we say that π is *envy-free* if no agent prefers taking another agent's place: $\pi(i) \succcurlyeq_i \pi(j) \setminus \{j\} \cup \{i\}$ for all $i, j \in N$ with $\pi(i) \neq \pi(j)$. We will only consider the concept of envy-freeness for hedonic games where every coalition is acceptably to every player, i.e., if $S \succcurlyeq_i \{i\}$ for all $S \in \mathcal{N}_i$, so that a player never envies a player being alone.

Note that the preference relations $\succcurlyeq_i$ have an exponentially sized domain of 2^{n-1} coalitions. For computational purposes, we need to use a language that represents such preferences *succinctly*, so that the representation preferably uses only $\text{poly}(n)$ symbols, where n is the number of agents. An attractive such representation is given by *additively separable hedonic games*, in which each agent specifies a *valuation function* $v_i : N \to \mathbb{R}$ assigning each agent a numeric value. We then say that $S \succcurlyeq_i T$ if and only if $\sum_{j \in S} v_i(j) \geqslant \sum_{j \in T} v_i(j)$. An additively separable game is thus given by n^2 numbers.

A more expressive representation is proposed by Elkind and Wooldridge [2009]. They define *hedonic coalition nets* (or *HC-nets*) in which each agent specifies a set of weighted propositional formulas, called *rules*, with propositional atoms given by the agents. For example, the rule $i_2 \wedge i_3 \wedge \neg i_4 \mapsto_{i_1} 5$ means that agent i_1 derives utility 5 when i_1 is together with i_2 and i_3 but not together with i_4. If an agent specifies multiple rules, the agent obtains the sum of the weights of those formulas that are satisfied in the given coalition. By taking rules of form $j \mapsto_i v_i(j)$, we see that HC-nets can encode additively separable games. Elkind and Wooldridge [2009] show that many other standard representations can also be encoded in HC-nets.

In hedonic games with cardinal (numeric) utilities (such as those given by HC-nets), we can define the *social welfare* of a given partition π. Its *utilitarian* welfare is $\sum_{i \in N} u_i(\pi(i))$, its *egalitarian* welfare is $\min_{i \in N} u_i(\pi(i))$, and its *Nash product* is $\prod_{i \in N} u_i(\pi(i))$. A partition that maximises a chosen welfare notion is seen as *optimal*. Another optimality notion is that of a *perfect* (sometimes called *wonderfully stable*) partition in which every agent belongs to a most preferred coalition. Finally, a partition π is *Pareto optimal* if for no partition π' we have $\pi'(i) \succcurlyeq_i \pi(i)$ for all $i \in N$, with at least one preference strict.

Let us now define the treewidth of a graph. Given a graph G, we write $V(G)$ for its set of vertices, and $E(G)$ for its set of edges. A *tree decomposition* of an undirected graph G is given by a tree T and a map $\beta : V(T) \to 2^{V(G)}$ which assigns to each vertex $w \in V(T)$ of the tree T a *bag* $\beta(w) \subseteq V(G)$ of vertices of G, satisfying the following two conditions:

1. For each $v \in V(G)$, the set $\beta^{-1}(v)$ of bags containing v is non-empty and connected (in T).
2. For each edge $\{u, v\} \in E(G)$ in G, there is a bag containing both u and v, i.e., there is $w \in V(T)$ with $\{u, v\} \subseteq \beta(w)$.

The *width* of a tree decomposition is $\max_{w \in V(T)} |\beta(w)| - 1$, that is, one less than the maximum size of the bags. Then, the *treewidth* of G is the minimum width of a tree decomposition of G. Bodlaender [1994] gives more intuition and examples.

3 Graphical Hedonic Games

We now introduce the main notion of this paper:

Definition 1. *A* ***graphical hedonic game*** *is a pair of a hedonic game $\langle N, (\succcurlyeq_i)_{i\in N}\rangle$ and an undirected graph $G = (N, E)$ that jointly satisfy the following condition: for each agent $i \in N$ and all coalitions $S, T \in \mathcal{N}_i$, we have*

$$S \succcurlyeq_i T \text{ if and only if } S \cap \Gamma(i) \succcurlyeq_i T \cap \Gamma(i),$$

where $\Gamma(i) \subseteq N$ is the set of neighbours of i in G.

We will call any graph $G' = (N, E)$ satisfying this condition an (agent) dependency graph *for the hedonic game $\langle N, (\succcurlyeq_i)_{i\in N}\rangle$.*

Thus, in a graphical hedonic game, how much an agent i likes a coalition depends only on the presence of the neighbours of i in the dependency graph.

Example 1. Pairing any hedonic game with the complete graph over the agent set yields a graphical hedonic game. The complete graph over the agent set is the unique agent dependency graph for *anonymous* hedonic games, where agents only care about the cardinality of their coalition. The same is true for *fractional hedonic games* [Aziz et al., 2014].

In additively separable hedonic games, the graph with

$$\{i, j\} \in E \iff v_i(j) \neq 0 \text{ or } v_j(i) \neq 0$$

is the edge-minimal dependency graph.

For a hedonic game given by an HC-net, we could take

$$\{i, j\} \in E \iff i \text{ appears in a rule of } j \text{ or vice versa},$$

though this might not be edge-minimal. □

We are usually interested in dependency graphs with as few edges as possible, so that the vertex-neighbourhoods and the treewidth of the graph are small.

Lemma 1. *For each hedonic game, there exists a unique edge-minimal agent dependency graph.*

Proof. We show that if edge sets $E_1, E_2 \subseteq N^{(2)}$ induce a dependency graph for a hedonic game $\langle N, (\succcurlyeq_i)_{i\in N}\rangle$, then so does the edge set $E_1 \cap E_2$. Let $i \in N$ be an agent, and let $S, T \ni i$ be coalitions. Then using the definition of a dependency graph once for E_1 and once for E_2, we see that

$$\begin{aligned} S \succcurlyeq_i T &\iff S \cap \Gamma_1(i) \succcurlyeq_i T \cap \Gamma_1(i) \\ &\iff S \cap \Gamma_1(i) \cap \Gamma_2(i) \succcurlyeq_i T \cap \Gamma_1(i) \cap \Gamma_2(i). \end{aligned}$$

The neighbourhoods $\Gamma_1(i) \cap \Gamma_2(i)$ are induced by $E_1 \cap E_2$. Hence $E_1 \cap E_2$ is the edge set of a dependency graph.

It is often easy to find the minimal dependency graph, as in the case of additively separable hedonic games. On the other hand, finding the *minimal* graph of an HC-net is coNP-hard, since it is hard to decide whether some variable is redundant in a boolean formula.

Definition 2. *Let $(\langle N, (\succcurlyeq_i)_{i\in N}\rangle, G)$ be a graphical hedonic game. Its* treewidth *is the treewidth of G, and its* degree *is the maximum degree of G.*

We will show that many computational problems concerning hedonic games become easy when restricting attention to graphical hedonic games of small treewidth and small degree.

Example 2. The class of hedonic games in which agents can be placed in a cycle and only care about the presence of their immediate neighbours is of bounded treewidth and degree.

A crucial observation about graphical hedonic games is that, as far as the agents' preferences are concerned, we can often restrict our attention to *connected* coalitions.

Definition 3. *In a graphical hedonic game with dependency graph G, a coalition $S \subseteq N$ is* connected *if $G[S]$ is connected, that is if S induces a connected subgraph in G. A partition π of N is* connected *if each $S \in \pi$ is connected.*

Notice that given a non-connected partition π', we can split each coalition $S \in \pi'$ into its connected components, obtaining a connected partition π. Then, from the definition of dependency graphs, every agent is indifferent between π and π': so $\pi(i) \sim_i \pi'(i)$ for all $i \in N$.

Representation. In our computational study of graphical hedonic games, we will require games to be represented in some reasonably concise fashion. As we mostly deal with classes of graphical hedonic games that have bounded degree d, where we regard d to be small, it can be sensible to explicitly list every agent's preferences over all subsets of her neighbourhood, taking $O(2^{2d} \cdot n)$ space. If the preference relations $\succcurlyeq_i$ can be evaluated in time only depending on d, but not on n, this will be just as well for our FPT results. In other cases, we assume that we have oracle access to the $\succcurlyeq_i$.

4 A Logic for Hedonic Games

In this section, we define a logic that captures standard properties of hedonic games, for example the existence of stable partitions. The logic uses variables $i, j, k, \dots$ ranging over agents, variables $S, T, \dots$ ranging over coalitions, and variables $\pi, \pi', \dots$ ranging over partitions of the agents.

Definition 4. *The formulas of* ***hedonic game logic (HG-logic)*** *are defined recursively as*

1. *atomic formulas:* $i = j$, $i \in S$, $S = \pi(i)$, $S \succcurlyeq_i T$.
2. *boolean combinations of formulas:* $\neg\phi$, $(\phi \vee \psi)$, $(\phi \wedge \psi)$.
3. *quantification over agents:* $\forall i\, \phi$, $\exists i\, \phi$.
4. *quantification over coalitions:* $\forall S\, \phi$, $\exists S\, \phi$.
5. *quantification over partitions:* $\forall \pi\, \phi$, $\exists \pi\, \phi$.

We will use standard abbreviations in writing formulas of HG-logic. For example,

- $S \succcurlyeq_i \pi(i)$ means $\exists T\, (T = \pi(i) \wedge S \succcurlyeq_i T)$,
- $S \subseteq T$ means $\forall i\, (i \in S \to i \in T)$,
- $\exists i \in S\, \phi$ means $\exists i\, (i \in S \wedge \phi)$,
- $S \succ_i T$ means $(S \succcurlyeq_i T \wedge \neg T \succcurlyeq_i S)$.

A *sentence* of HG-logic is a formula in which no variable occurs free. In general, every hedonic game can form a model of a sentence in HG-logic in the natural way. In our approach, however, the models of HG-logic are *graphical* hedonic games. A given sentence ϕ of HG-logic is *true* in a given graphical hedonic game $(\langle N, (\succcurlyeq_i)_{i\in N}\rangle, G)$ if it is true when the formula is evaluated according to the obvious semantics using the universe N and relations $\succcurlyeq_i$ as specified by the hedonic game model, but where quantifications over partitions range only over *connected* partitions (according to the dependency graph G). Thus, according to our graphical semantics, the sentence $\forall i\, \forall j\, \exists \pi\, \pi(i) = \pi(j)$ is not valid, since i and j might not be together in any connected partition.

Let us give some examples of properties of hedonic games and partitions expressible in HG-logic.

- *a core-stable partition exists*: $\exists \pi\, \forall S\, \exists i \in S\, \pi(i) \succcurlyeq_i S$
- π *is Pareto-optimal*: $\neg\exists \pi'\, (\forall i\, \pi'(i) \succcurlyeq_i \pi(i) \wedge \exists j\, \pi'(j) \succ_j \pi(j))$
- π *is Nash stable:* $\forall i\, (\pi(i) \succcurlyeq_i \{i\} \wedge \forall S\, (\exists j\, S = \pi(j) \to \pi(i) \succcurlyeq_i S \cup \{i\}))$.
- π *is perfect*: $\forall i\, \forall S \ni i\, S \not\succ_i \pi(i)$.
- π *is envy-free:* $\forall i\, \forall j\, \pi(i) \succcurlyeq_i \pi(j) \setminus \{j\} \cup \{i\}$.
- π' *is reachable from* π *by actions of* S*:* $\forall i\, \forall j\, (i \notin S \wedge j \notin S \to (\pi(i) = \pi(j) \leftrightarrow \pi'(i) = \pi'(j)))$.

5 Main Result

An important computational problem we wish to solve is the model-checking problem of HG-logic.

ϕ-HEDONIC GAMES
Instance: a graphical hedonic game $(\langle N, (\succcurlyeq_i)_{i\in N}\rangle, G)$ and a formula ϕ of HG-logic
Question: does $(\langle N, (\succcurlyeq_i)_{i\in N}\rangle, G) \models \phi$, i.e. is the graphical hedonic game a model of the formula ϕ?

The perhaps most important special case of this problem is deciding the existence of stable partitions in a hedonic game.

Theorem 1. *The problem ϕ-HEDONIC GAMES is fixed-parameter tractable with respect to the length $|\phi|$ of the formula ϕ, and the treewidth k and degree d of the graph G. That is, the problem can be solved in time $O(f(|\phi|, k, d) \cdot n)$ where f is a computable function, and n is the number of agents. Here we assume that the relation "$S \succcurlyeq_i T$" can be decided in time only depending on d, but not on n.*

This means that for any formula ϕ and any class $\mathcal{C}$ of graphical hedonic games of bounded treewidth and bounded degree, we can decide in linear time whether ϕ is true in a given game $(\langle N, (\succcurlyeq_i)_{i \in N} \rangle, G) \in \mathcal{C}$. In case computing "$S \succcurlyeq_i T$" takes time depending on n, we will need $2^{2d} \cdot n$ calls to an oracle deciding this relation during a pre-processing step, after which the linear-time bound applies again.

Let us make explicit some special cases of Theorem 1.

Corollary 1. *For every class of graphical hedonic games of bounded treewidth and degree, there exist linear-time algorithms that can decide whether a given such game admits a partition that is (i) core-stable, (ii) strict-core-stable, (iii) Pareto-optimal, (iv) perfect (v) Nash-stable, (vi) individually stable, (vii) envy-free, or that satisfies any combination of these properties.*

Proof. This would follow immediately from Theorem 1 and the formulas in Section 4, except that we need to check that the fact that our semantics only quantify over connected partitions makes no difference. For (i)-(iv), this is immediate, since a partition π satisfies the relevant criterion if and only if the connected partition π' obtained from π by splitting its coalitions into their connected components satisfies the same property.

For (v)-(vii), we can achieve a similar behaviour by adding extra edges to the dependency graph G of the input game to obtain a new dependency graph G'. Precisely, whenever there are edges $\{u, v\} \in E(G)$ and $\{v, w\} \in E(G)$, then we add the edge $\{u, w\}$ to G' (note that this is different from taking a transitive closure since we do not apply this step repeatedly). It can then be seen that splitting a Nash- or individually stable partition into its G'-components preserves stability, and similarly for envy-freeness under the additional assumption that all coalitions are individually rational. Further, note that G' has treewidth at most kd and degree at most d^2, so that both parameters are still bounded.

By calling such an algorithm repeatedly, we can adaptively build up a connected partition satisfying any of these properties (if it exists).

Theorem 1 is proved by reducing ϕ-HEDONIC GAMES to the model checking problem of monadic second-order logic, which by Courcelle's theorem is fixed-parameter tractable with parameter the treewidth of the underlying logical structure and the length of the input formula. We will explain this reduction in the following two sections.

6 MSO and Courcelle's Theorem

Here we give a statement of Courcelle's theorem (1990) and introduce monadic second-order logic (MSO).

First, we need some definitions. A *signature* σ is a finite collection of relation symbols $(R_1, \ldots, R_k)$, with each symbol $R_i \in \sigma$ being endowed with an *arity* $\mathrm{ar}(R_i) \geqslant 1$. A *$\sigma$-structure* $\mathcal{A} := \langle A, (R_1^{\mathcal{A}}, \ldots, R_k^{\mathcal{A}})\rangle$ is given by a finite set A, the *universe* of $\mathcal{A}$, as well as a realisation $R_i^{\mathcal{A}} \subseteq A^{\mathrm{ar}(R_i)}$ for each relation symbol R_i. The *size* of $\mathcal{A}$ is given by $\|\mathcal{A}\| = |\sigma| + |A| + \sum_{R_i\in\sigma} |R_i^{\mathcal{A}}| \cdot \mathrm{ar}(R_i)$.

Given a signature σ, the *language* $\mathrm{MSO}[\sigma]$ *of monadic second order logic* is given by the grammar

$$\begin{aligned} \phi ::= {} & x{=}y \mid R_i x_1 \ldots x_{\mathrm{ar}(R_i)} \mid Xx \mid (\phi \vee \phi) \mid (\phi \wedge \phi) \mid \neg\phi \\ & \mid \exists x\, \phi \mid \forall x\, \phi \mid \exists X\, \phi \mid \forall X\, \phi, \end{aligned}$$

where $x, y, x_1, x_2, \ldots$ are first-order variables, and X denotes set variables. Notice that MSO allows quantification only over *unary* relations, i.e. over subsets of the universe A. For a formula ϕ of $\mathrm{MSO}[\sigma]$ and a σ-structure $\mathcal{A}$, we define the semantic notion of $\mathcal{A} \models \phi$ in the obvious way.

Next, let us define the notion of treewidth for a σ-structure. A *tree decomposition* of a σ-structure $\mathcal{A}$ is given by a tree T, each vertex v of T being associated with a subset $\beta(v) \subseteq A$ of the universe, called a *bag*, satisfying the following two conditions: (1) each $a \in A$ is contained in some bag, and the set $\beta^{-1}(a)$ of bags containing a forms a connected subtree of T, and (2) for each $R_i \in \sigma$ and all $a_1, \ldots, a_{\mathrm{ar}(R_i)} \in A$ such that $(a_1, \ldots, a_{\mathrm{ar}(R_i)}) \in R_i^{\mathcal{A}}$, we have that $\{a_1, \ldots, a_{\mathrm{ar}(R_i)}\} \subseteq \beta(v)$ for some vertex v of T. The *width* of such a tree decomposition is the maximum cardinality of the bags minus 1, and the *treewidth* $\mathrm{tw}(\mathcal{A})$ of $\mathcal{A}$ is the minimum width among tree decompositions of $\mathcal{A}$. The (usual) treewidth of a graph is a special case of this definition: just take σ to consist of a single binary relation, namely the adjacency relation of the graph.

We are now ready to state Courcelle's (1990) theorem.

Theorem 2 (Courcelle). *Given a formula ϕ of $\mathrm{MSO}[\sigma]$ and a σ-structure $\mathcal{A}$, we can in time $g(|\phi|, \mathrm{tw}(\mathcal{A})) \cdot |A| + O(\|\mathcal{A}\|)$ decide whether $\mathcal{A} \models \phi$, where g is a computable function.*

Courcelle's theorem is often stated for fixed formulas ϕ, but the more general statement as we are stating it here is true: the model checking problem for MSO is fixed-parameter tractable with respect to the joint parameter consisting of the formula length $|\phi|$ and the treewidth of $\mathcal{A}$. We will use this stronger result in what follows.

7 Encoding of HG-logic into MSO

In this section we prove Theorem 1. Suppose we are given a formula ϕ of HG-logic and a graphical hedonic game $(\langle N, (\succcurlyeq_i)_{i\in N}\rangle, G)$, where G has treewidth k and max-degree d. In the following, we will generate a relatively large (in d, not n) σ-structure containing all information about the game, and then rewrite the formula ϕ into a formula of MSO.

Step 1: $\boldsymbol{\sigma}$-structure. Our signature σ will have four relation symbols: unary symbols VERT and EDGE, a binary symbol INCI, and a $(2d+1)$-ary symbol PREF.

We build a σ-structure $\mathcal{G}$ with universe $N \cup E \cup \{*\}$, and relations VERT $= N$, EDGE $= E$, the vertex-edge incidence relation INCI $= \{(i, e) : i \in N, e \in E, i \in e\}$, and for each $i \in N$, let $(i, i_1, \ldots, i_d, i_{d+1}, \ldots, i_{2d}) \in$ PREF if and only if $i_s \in N \cup \{*\}$ for all $1 \leqslant s \leqslant 2d$, and each $i_s \in N$ is a neighbour of i in G, and

$$\{i_1, \ldots, i_d\} \setminus \{*\} \succcurlyeq_i \{i_{d+1}, \ldots, i_{2d}\} \setminus \{*\}.$$

That is, we use the defining property of dependency graphs to encode every agent's essential preferences in the relation PREF. The structure $\mathcal{G}$ can be computed in $O(2^{2d} \cdot n)$ calls to an oracle deciding $\succcurlyeq_i$. For convenience, let us define within MSO$[\sigma]$ the adjacency relation between vertices as

$$\mathrm{adj}(u, v) \;\equiv\; u \neq v \wedge \exists e\,(\mathrm{INCI}\, ue \wedge \mathrm{INCI}\, ve).$$

Step 2: bounding treewidth. By assumption, the agent dependency graph G has treewidth at most k. We show that the σ-structure $\mathcal{G}$ constructed above has treewidth at most $k \cdot (d+1) + 1$, which is still bounded for bounded d and k. First find in time linear-in-n a tree-decomposition of G of width at most k using Bodlaender's (1993) algorithm. For each of the $O(n)$ edges $e = \{u, v\}$ of G, find a bag $\beta(w)$ that contains both u and v, and introduce a new bag $\{e, u, v\}$ that gets attached as a leaf to w in the tree underlying the tree decomposition, not increasing its width. For each vertex v and every bag $\beta(w)$ of the original tree decomposition containing v, add to $\beta(w)$ the set $\mathit{\Gamma}_G(v)$ of the at most d neighbours of v in G. This operation increases the width by at most $k \cdot d$. Finally, add $*$ to every bag, increasing the width by 1, for a total width of at most $k + kd + 1$. As is easy to see, this is a tree decomposition of $\mathcal{G}$.

Step 3: encoding partitions. In our encoding, a partition π of the agent set will be a 'transitive' subset $E' \subseteq E$ of the edge set of the dependency graph. (The set E' represents the equivalence relation associated with the partition π, in the sense that the two endpoints of an edge $e \in E'$ are in the same coalition of π.) Formally, a set variable X represents a partition if $X \subseteq E$ and whenever $e_1 = \{x, y\}, e_2 = \{y, z\}, e_3 = \{x, z\}$ are edges in E with $e_1, e_2 \in X$, then also $e_3 \in X$. This condition can clearly be expressed in MSO$[\sigma]$. We can also express the relation "two agents have an edge between them, and this edge is part of X" by a formula. It is well-known that MSO can express the transitive closure of every binary relation it can express [Courcelle and Engelfriet, 2012]. Hence we can express the transitive closure of the preceding relation, which is "two agents are connected by a path of edges that are in X", which is equivalent in our understanding to "two agents are in the same coalition in partition X". This we can use to express "$S = \pi(i)$".

Step 4: encoding preference. We now encode the relation $S \succcurlyeq_i T$. This depends crucially on the definition of the agent dependency graph, so that we actually encode the equivalent relation $S \cap \Gamma(i) \succcurlyeq_i T \cap \Gamma(i)$. To do this, we use d variables for S and d variables for T which will represent the agents from $\Gamma(i)$ that are present in S and T respectively. If there are fewer than d such agents, we assign $*$ as a placeholder. Note that the relation $x \in N \cup \{*\}$ is expressible in $\mathrm{MSO}[\sigma]$ as $\neg \mathrm{EDGE}\, x$. With this, we can express $S \succcurlyeq_i T$ as

$$\begin{aligned}
&\exists x_1, \dots, x_d, y_1, \dots, y_d \in N \cup \{*\} \\
&\qquad \forall x \in S \,(\mathrm{adj}(i, x) \to x = x_1 \vee \cdots \vee x = x_d) \\
&\quad\ \wedge \forall y \in T \,(\mathrm{adj}(i, y) \to y = y_1 \vee \cdots \vee y = y_d) \\
&\quad\ \wedge \mathrm{PREF}\, i\, x_1 \dots x_d\, y_1 \dots y_d.
\end{aligned}$$

Step 5: encoding HG-syntax. Using steps 3 and 4, we can translate ϕ (a formula of HG-logic) into a formula ϕ' of $\mathrm{MSO}[\sigma]$. Here, we replace quantifications over partitions by quantifications over edge sets, as indicated in step 3.

This finishes our translation of HG-logic into $\mathrm{MSO}[\sigma]$. Using Courcelle's algorithm, we can now check whether $\mathcal{G} \models \phi'$. We achieve the claimed time bound by noting that any blow-ups in formula size and treewidth are still bounded whenever k and d are bounded.

8 Faster Algorithms in Special Cases

In the algorithms arising through our use of Courcelle's theorem, the dependence on k and d in their runtime is quite bad; indeed, we cannot bound the function $f(|\phi|, k, d)$ by an elementary function unless P = NP [Frick and Grohe, 2004]. This phenomenon is especially bad in our case as we are using multiple quantifier alternations in our MSO-encoding. Clearly, we cannot just ignore this as a merely 'constant factor'. In this sense, Theorem 1 should be seen as an *existence result*, but not as providing an actually usable algorithm.

In this section, we use an alternative approach due to Bodlaender [1988] that produces algorithms with more manageable dependence on k and d for some important stability and optimality problems. Bodlaender [1988] defines the very general class of *local condition composition problems* (short LCC or 1-LCC) and shows that they are linear-time solvable on classes of graphs of bounded treewidth and degree. In the following, we will give a definition of an LCC-problem, suitably specialised for our purposes.

Let $G = (V, E)$ be a graph with max-degree d for which a tree decomposition of width k is given. Let $E_2(v)$ be the set of edges from E whose endpoints are both within distance 2 of v. For maps $f : E \to \{0, 1\}$, let $P(v, f|_{E_2(v)})$ be a 0/1-property, and let $W(v, f|_{E_2(v)})$ be an integer-valued function, both computable in time polynomial in their input length. Finally, let $\oplus$ denote the binary operation of a totally ordered commutative monoid over the integers. Examples include taking $\oplus$ to be the sum, product, or minimum of the values. Now consider the following computational problem:

Instance: Graph $G = (V, E)$, additional data about G, target value K.
Question: Does there exist a map $f : E \to \{0, 1\}$ such that for each $v \in V$ the property $P(v, f|_{E_2(v)})$ is true, and $\bigoplus_{v \in V} W(v, f|_{E_2(v)}) \geqslant K$?

Bodlaender [1988] shows that any problem of this form can be solved in $\tilde{O}(2^{kd^2} n)$ time. Here, the soft-$\tilde{O}$ hides factors polynomial in k and d which will depend on the runtime of evaluating P and W.

Using this apparatus, we can encode hedonic games problems in a similar fashion as before. Again, a connected partition π of N will be represented by a set $E' \subseteq E$ of edges, where $E' = f^{-1}(1) = \{e \in E : f(e) = 1\}$. Using the property P, we can enforce transitivity of E'. We can also calculate in P the utility of a given agent in the partition described by E' (since $f|_{E_2(v)}$ tells us which relevant players are in the same coalition as v), and we know who is in the coalition of every agent w adjacent to v. For example, we can thus let $P(v, f|_{E_2(v)})$ express that (i) E' is transitive at v and (ii) v does not want to Nash deviate under the partition specified by E'. Hence deciding the existence of a connected Nash-stable partition is an LCC problem. Since we can calculate players' utilities in E', maximising utilitarian or egalitarian or Nash social welfare is also an LCC problem. Using this general technique, we find the following.

Theorem 3. *There is an $\tilde{O}(2^{kd^2} n)$ algorithm that, given a graphical hedonic game and a tree decomposition, decides whether there exists a connected partition π of the agent set that satisfies (a combination of) (i) individual rationality, (ii) Nash stability, (iii) individual stability, (iv) envy-freeness. Subject to any combination (or none) of the preceding conditions, we can also maximise utilitarian, egalitarian, or Nash social welfare under π.*

A perfect partition can be found in slightly worse time. We can always modify the dynamic programming implementation to actually return a partition π (if it exists) in the same time bound. By using the technique of Corollary 1, we can drop the condition that π is connected in exchange for a worse time bound of $\tilde{O}(2^{kd^5} n)$.

We can also use the LCC approach to get the following result about verifying whether a *given* partition satisfies a stability or optimality criterion.

Theorem 4. *There is an $\tilde{O}(2^{kd^2} n)$ algorithm that given a hedonic game, an associated dependency graph, a tree decomposition, and a partition π of N, decides whether π is (i) Pareto optimal, (ii) core-stable, (iii) strict-core-stable.*

While the method of reduction to LCC problems is evidently quite powerful, it does not seem to capture Σ_2^p-questions like whether a core-stable outcome exists.

9 Allocation of Indivisible Goods

In the problem of allocating indivisible goods, we are given a set $\mathcal{O} = \{o_1, \dots, o_m\}$ of objects that need to be allocated to agents N who have preferences over bundles $B \subseteq \mathcal{O}$ of objects (see Bouveret et al. [2016] for a survey). Throughout, we will assume that no bundle is unacceptable to any agent (a weak free-disposal assumption). This setting can quite naturally be captured as a hedonic game with agent set $N \cup \mathcal{O}$, where no coalition containing 2 different agents from N is allowed, and $i \in N$ likes a coalition $S \in \mathcal{N}_i$ just as much as i likes the bundle $S \setminus \{i\}$. The objects, on the other hand, are indifferent between all outcomes. With this implementation, the hedonic-game- and allocation-notions of envy-freeness, Pareto-optimality, and of maximising social welfare coincide perfectly.

This hedonic game is also a graphical hedonic game whose dependency graph is bipartite with N on one side and $\mathcal{O}$ on the other, with an edge from i to o whenever i cares about whether o is part of i's bundle. Note that this graphical hedonic game does not capture the requirement that coalitions may only contain a single agent from N, but we will enforce this condition later.

Let us now take a class of allocation problems whose associated bipartite graphs have bounded treewidth and bounded degree. The latter condition implies that every agent desires a bounded number of objects, and every object is desired by a bounded number of agents. The results developed over the preceding sections will imply that on such a restricted class, we can efficiently find allocations that are fair and/or efficient, in contrast to many hardness results in the unrestricted case.

Using HG-logic, we can identify agents that belong to $\mathcal{O}$ as those agents that are indifferent between all coalitions containing them. Hence HG-logic is expressive enough to require that a given coalition contains at most 1 non-object agent. Hence, using HG-logic, we have an algorithm that decides the existence of a Pareto-optimal and envy-free allocation, which is Σ_2^p-complete for general preferences represented in a logic representation [Bouveret and Lang, 2008] and even for additive utilities [de Keijzer et al., 2009]. Similarly, there is an algorithm that decides the existence of an envy-free and *complete* allocation (where every object must be allocated); this problem is NP-hard for general additive utilities [Lipton et al., 2004]. We can also use an LCC-based algorithm to find an allocation that maximises social welfare among envy-free ones, or an algorithm that finds a complete allocation of minimum envy. Looking at the allocation problem from a hedonic game perspective, we can also readily define intriguing notions of *stable* allocations in which agents don't want to swap items (possibly even in larger swap cycles). Many such properties can be described in HG-logic.

In the context of combinatorial auctions, Conitzer et al. [2004] provide a different way of exploiting a tree decomposition to efficiently allocate objects. Here, the objects are arranged in an *item graph* of bounded treewidth, and agents are assumed to only demand bundles inducing a connected subgraph of the item graph. With this restriction, winner determination becomes feasible in polynomial time. Note, however, that constructing a suitable item graph of small treewidth is computationally hard [Gottlob and Greco, 2007].

10 Necessity of the Degree Bound

In this section, we show that a variety of problems of type ϕ-HEDONIC GAMES are NP-hard for games of bounded treewidth but unbounded degree. This establishes that unless P = NP it is necessary for our fixed-parameter tractability result that we bound the degree of the hedonic games. The bound on the treewidth is also necessary; this follows from slight modifications of standard hardness reductions (e.g., those of Sung and Dimitrov [2010]).

Theorem 5. CORE-EXISTENCE *is NP-hard even for graphical hedonic games of treewidth 2 that are given by an HC-net.*

Proof. By reduction from 3SAT. Given a formula ϕ which we may assume not to be satisfied by setting all variables false, introduce one agent $x_1, \dots, x_n$ for each variable occurring in ϕ, and add 3 other agents a, b, c. The variable agents are indifferent between all outcomes (no associated rules). The preferences of agents a, b, c are cyclic and given by the net

$$\begin{aligned} \phi \mapsto_a 3, \quad b \mapsto_a 2, \quad c \mapsto_a 1, \quad b \wedge c \mapsto_a -10; \\ c \mapsto_b 2, \quad a \mapsto_b 1, \quad c \wedge a \mapsto_b -10; \\ a \mapsto_c 2, \quad b \mapsto_c 1, \quad a \wedge b \mapsto_c -10. \end{aligned}$$

An agent dependency graph of this game is given by a triangle on $\{a, b, c\}$ plus n leaves attached to a; this is easily seen to have treewidth 2 (and actually even pathwidth 2).

We show the game admits a core-stable outcome if and only if ϕ is satisfiable. Suppose ϕ is satisfied by some assignment. Then take the partition π where a is together with all true variable agents, with $\{b, c\} \in \pi$, and with all false variable agents together. Then π is core-stable, because the variable agents (being indifferent) are not part of any blocking coalition, so that a does not obtain utility larger than 3 in any blocking coalition.

Conversely, suppose that the game admits a core-stable partition π. We show that a obtains utility 3 in π, which implies that a is together with variables that satisfy ϕ so that ϕ is satisfiable, as required. Suppose not. By individual rationality, a, b, c are not all together, so one of them is not together with either of them, say b. But then $\{a, b\}$ blocks. Such a blocking coalition exists for any choice of lonely agent as a obtains utility $\leqslant 2$, so π is not stable, contradiction.

This result cannot be improved to apply to HC-nets of treewidth 1, which follows from results of Igarashi and Elkind [2016]. By adapting the reduction of Peters [2015], this problem should be Σ_2^p-complete, at least for treewidth 4.

Theorem 6. NASH-STABLE-EXISTENCE *is NP-hard even for graphical hedonic games of treewidth 1 that are given by an HC-net.*

Proof. By reduction from X3C. Given an instance with elements $x_1, \dots, x_n$ and sets $s_1, \dots, s_m$, we construct an HC-net where the agents are given by the elements x_i and an extra stalker agent. Every element hates the stalker, but the stalker loves every coalition of elements—except coalitions of form s_j. Note that the dependency graph of this game is a star with the stalker in the center. If the X3C-instance has a solution, then partitioning the x_i as in the solution and putting the stalker in a singleton is Nash stable. Conversely, in every Nash stable partition, the stalker needs to be alone by individual rationality for the elements. By Nash stability, the stalker does not want to join any coalition, and so every coalition must be of form s_j; thus the partition of the elements gives an X3C-solution.

A similar construction works for individual stability for treewidth 2, see also Peters and Elkind (2015, Thm. 2). A reduction from 3SAT gives hardness for PERFECT-EXISTENCE on trees (agents for literals, complementary ones hate each other, a formula agent is satisfied iff the formula is satisfied).

11 Conclusions and Future Work

We have shown that restricting treewidth and degree of hedonic games is a potent avenue to obtaining tractability results for a broad array of important computational problems concerning hedonic games. Our application to the problem of allocating indivisible goods shows how useful tractability results for hedonic games can be: because hedonic games are a very general model encompassing far more than just 'how to find friends', a possibility result for hedonic games translates to easiness for any problem that involves partitions with some elements having preferences over who gets what. It will be interesting to see whether this idea can be further applied elsewhere.

The notion of a graphical hedonic game suggests a wide variety of interesting questions for future work: Are there alternative conditions on graph topology that yield tractability? Examples could be bipartiteness, planarity, or H-minor freeness. Can we say anything about the structure of stable outcomes in dependence on the structure of the graphical hedonic game?

An open problem more closely related to this paper is the problem of finding faster algorithms than those provided through HG-logic for Σ_2^p-type questions like the existence of a core-stable partition or of finding a Pareto-optimal partition. We should also note that the hardness results in the preceding section are not easily generalised to, say, additively separable hedonic games. It would be interesting to know if we can dispense with the degree bound on this more restricted class. Some results for welfare-maximisation can be found in Bachrach et al. [2013].

Acknowledgements Thanks to four anonymous AAAI reviewers whose feedback improved this paper considerably, and to Edith Elkind for advice and guidance. I am supported by EPSRC.

Bibliography

Aaron B Adcock, Blair D Sullivan, and Michael W Mahoney. Tree-like structure in large social and information networks. In *International Conference on Data Mining (ICDM) 2013*, pages 1–10. IEEE, 2013.

Haris Aziz, Felix Brandt, and Paul Harrenstein. Fractional hedonic games. AAMAS '14, pages 5–12, 2014.

Yoram Bachrach, Pushmeet Kohli, Vladimir Kolmogorov, and Morteza Zadimoghaddam. Optimal coalition structure generation in cooperative graph games. AAAI '13, 2013.

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

Suryapratim Banerjee, Hideo Konishi, and Tayfun Sönmez. Core in a simple coalition formation game. *Social Choice and Welfare*, 18(1):135–153, January 2001.

Hans Bodlaender. Dynamic programming on graphs with bounded treewidth. *Automata, Languages and Programming*, pages 105–118, 1988.

Hans L Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. STOC 1993, pages 226–234. ACM, 1993.

Hans L Bodlaender. A tourist guide through treewidth. *Acta cybernetica*, 11 (1-2):1, 1994.

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

Sylvain Bouveret and Jérôme Lang. Efficiency and envy-freeness in fair division of indivisible goods: Logical representation and complexity. *Journal of Artificial Intelligence Research*, pages 525–564, 2008.

Sylvain Bouveret, Yann Chevaleyre, and Nicolas Maudet. Fair allocation of indivisible goods. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 15. Cambridge University Press, 2016.

Vincent Conitzer, Jonathan Derryberry, and Tuomas Sandholm. Combinatorial auctions with structured item graphs. AAAI '04, pages 212–218, 2004.

Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and computation*, 85(1):12–75, 1990.

Bruno Courcelle and Joost Engelfriet. *Graph structure and monadic second-order logic: a language-theoretic approach*, volume 138. Cambridge University Press, 2012.

Bart de Keijzer, Sylvain Bouveret, Tomas Klos, and Yingqian Zhang. On the complexity of efficiency and envy-freeness in fair division of indivisible goods with additive preferences. In *Algorithmic Decision Theory*, pages 98–110. Springer, 2009.

Robin IM Dunbar. Neocortex size as a constraint on group size in primates. *Journal of Human Evolution*, 22(6):469–493, 1992.

Edith Elkind and Michael Wooldridge. Hedonic coalition nets. AAMAS '09, pages 417–424, 2009.

Markus Frick and Martin Grohe. The complexity of first-order and monadic second-order logic revisited. *Annals of Pure and Applied Logic*, 130(1):3–31, 2004.

Georg Gottlob and Gianluigi Greco. On the complexity of combinatorial auctions: structured item graphs and hypertree decomposition. EC '07, pages 152–161. ACM, 2007.

Georg Gottlob, Gianluigi Greco, and Francesco Scarcello. Pure nash equilibria: Hard and easy games. *Journal of Artificial Intelligence Research*, pages 357–406, 2005.

Ayumi Igarashi and Edith Elkind. Hedonic games with graph-restricted communication. page (in preparation), 2016.

Michael Kearns, Michael L Littman, and Satinder Singh. Graphical models for game theory. In *Proceedings of the seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 253–260, 2001.

Richard J Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. EC '04, pages 125–131. ACM, 2004.

Dominik Peters. Σ_2^p-complete problems on hedonic games. page arXiv:1509.02333 [cs.GT], 2015.

Dominik Peters and Edith Elkind. Simple causes of complexity in hedonic games. IJCAI '15, pages 617–623, arXiv:1507.03474 [cs.GT], 2015.

Neil Robertson and Paul D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *Journal of algorithms*, 7(3):309–322, 1986.

Shao-Chin Sung and Dinko Dimitrov. Computational complexity in additive hedonic games. *European Journal of Operational Research*, 203(3):635–639, June 2010.

Gerhard J Woeginger. Core stability in hedonic coalition formation. In *SOFSEM 2013: Theory and Practice of Computer Science*, pages 33–50. Springer, 2013.

Optimally Protecting Elections

Yue Yin[1], Yevgeniy Vorobeychik[2], Bo An[3], and Noam Hazon[4]

[1]University of Chinese Academy of Sciences
[2] Electrical Engineering & Computer Science, Vanderbilt University
[3] School of Computer Engineering, Nanyang Technological University
[4] Dept of Computer Science and Mathematics, Ariel University

Abstract. The problem of voting outcome manipulation is both a fundamental theoretical problem in social choice, as well as a major practical concern to democratic institutions. Consequently, this issue has received considerable attention, particularly as it pertains to different voting rules. In contrast, the problem of how election control can be prevented or deterred has been largely ignored. We introduce the problem of optimal protection against election control, where manipulation is allowed at the granularity of groups of voters (e.g., voting locations), through a denial-of-service attack, and the defender allocates limited protection resources to prevent control. We show that for plurality voting, election control through group deletion to prevent a candidate from winning is in P, while it is NP-Hard to prevent such control. We then present a double-oracle framework for computing an optimal prevention strategy, developing exact mixed-integer linear programming formulations for both the defender and attacker oracles (both of these subproblems we show to be NP-Hard), as well as heuristic oracles. Experiments conducted on both synthetic and real data demonstrate that the proposed computational framework can scale to realistic problem instances.

Keywords: Election Control, Stackelberg Game, Double Oracle

1 Introduction

Democratic institutions rely on the integrity of the voting process. A major threat to this integrity is the possibility that the process can be subverted by malicious parties to their own goals. Indeed, actual incidents of vote manipulation and control, sometimes through violence, bear out this concern. For example, the 2013 election in Pakistan was marred by a series of election-day bombings, resulting in over 30 dead and over 200 injured, in an overt attempt to subvert the voting process [5], and the 2010 Sri Lanka election exhibited 84 major and 202 minor incidents of poll-related violence [4]. Moreover, with the dawn of electronic and internet voting, the additional threat of election control and manipulation through cyber means has emerged, with a number of documented demonstration attacks [8, 11, 9, 12, 10].

The problem of election control and manipulation has received considerable attention in prior literature [13, 6, 16, 17, 14, 15, 18, 7, 19, 20]. For example, it is known that while most voting rules are NP-Hard to control, plurality—which is widely used and is of particular relevance to our work—can be controlled through voter deletion in polynomial time [6]. In the context of election control, which is most relevant to our work, the results typically focus on studying hardness of control through means such as adding or deleting voters. In this literature, a voting rule is viewed as resistant if control is NP-Hard, and vulnerable otherwise. Moreover, control is almost unversally at the granularity of individual voters, and protection, when considered, is at the form of designing voting rules which are resistant in this sense [15]. While these considerations are crucial if one is to understand vulnerability of elections, they are also limited in several respects. First, as the incidents of control described above attest, control can be exercised for groups of voters through a single attack, such as a denial-of-service attack on a voting station or a polling center (of which bombing is an extreme example); we are the first (to our knowledge) to consider this generalization. Second, NP-Hardness of control is insufficient evidence for resistance: it is often possible to solve large instances of NP-Hard problems in practice (see, e.g., [21] in the case of SAT). Resistance to election control in the broader sense, such as through allocation of limited protection resources to prevent attacks on specific voter groups, has, to our knowledge, neither been modeled nor investigated to date.

To address these limitations, we consider the problem of optimally protecting elections against control. We model control as a denial-of-service (deletion) attack on a subset of voter *groups*, which may represent polling places or electronic voting stations, with the goal of preventing a specific candidate from winning. We then show that, for plurality voting, optimal election control under this model can be computed in polynomial time. Next, we consider the problem of protection against election control, modeling it as a Stackelberg game in which an outside party deploys limited protection resources to protect a collection of voter groups, allowing for randomization, and the adversary responds by attempting to subvert (control) the election. Protection resources may represent actual physical security for polling centers or voting stations, or resources devoted to frequent auditing of specific electronic voting systems. In this model, we assume that the defender's goal is to ensure that the same candidate wins with or without an election control attack. We show that the problem of choosing the minimal set of resources that guarantee that an election cannot be controlled is NP-Hard. For the more general problem, we propose a double-oracle framework to compute an optimal protection, given limited security resources. We prove that both the defender, and attacker oracles are NP-Hard when randomized strategies are allowed. On the positive side, we develop novel mixed-integer linear programming formulations for both oracles that enable us to compute a provably optimal solution for protecting elections. Moreover, we develop heuristic defender and attacker oracles which significantly speed up the framework. Our experiments demonstrate the effectiveness and scalability of our algorithmic approach.

In summary, we make the following contributions:

- A new model of election control whereby groups, rather than individual voters, can be deleted in a single action,
- A new model of protecting elections from group-level election control attacks,
- A polynomial-time algorithm for group-level election control,
- Complexity analysis of guaranteeing that an election cannot be controlled,
- A scalable double-oracle framework for choosing optimal allocation of protection resources.

2 Election Control by Deleting Voter Groups

A common question in election control is whether it is possible to prevent a specific candidate from winning by deleting a subset of voters. We begin by generalizing this control problem to allow attackers to delete (or deploy a denial-of-service attack against) groups of voters, which may represent polling locations. Formally, suppose that there is a set I of n non-overlapping groups of voters and a set of candidates C over which voters have preferences. Throughout, we focus on *plurality voting*, in which only a single candidate is selected by each voter, and the candidate with the most votes wins (we assume that the tie-breaking rule is adversarial to the defender). For each group $i \in I$, let v_{ic} be the number of votes for candidate c, and let $v_c = \sum_i v_{ic}$ be the total vote tally for $c \in C$. Let $\omega \in C$ be the candidate who would have won with the original set of voters: $\omega = \arg\max_c v_c$. We now consider the problem of election control in which the attacker may choose to delete a subset of at most $k \leq n$ groups, with the goal of preventing ω from winning.[1]

It is well known that optimal control of plurality by deleting individual voters can be computed in polynomial time [6]. Allowing the attacker to select specific groups may appear to significantly complicate the problem. Surprisingly, we show that optimal control can still be computed in polynomial time, significantly generalizing the previous result. Intuitively, control succeeds as long as there exists a candidate $c \in C$ who has at least as many votes as ω after k groups are removed. Consequently, the attacker can consider one candidate c at a time, checking if k groups can be deleted so that c has a higher vote count than ω. Moreover, if we fix $c \in C$, it is easy to check whether it is possible to get more votes for c than ω: we would just delete the k groups in which ω is most favored over c.

Formally, let $d^c = \langle d_i^c : i \in I \rangle$ be a vector with $d_i^c = v_{ic} - v_{i\omega}$, that is, the vote advantage of c over ω in group $i \in I$. For a vector d^c, define $sum(d^c) = \sum_i d_i^c$. Then, $sum(d^c)$ is the total difference of votes between c and ω. For example, suppose that d^c is $\langle -3, -2, 1 \rangle$. This means that ω has more votes than c in the first two groups, but fewer (by 1) in the third. If the attacker can attack 2 groups, he will succeed by attacking the first two, leading c to have 1 more vote left than ω. The following proposition shows that it is, in fact, sufficient to delete k groups with smallest d_i^c to verify whether it is possible to make c have a larger vote count than ω. For convenience, define d^{c-k} to be the portion of the vector d^c remaining after the k groups with smallest d_i^c have been deleted.

Proposition 1 *For a given candidate $c \in C$, it is possible to delete k groups to ensure that $v_c > v_\omega$ iff $sum(d^{c-k}) > 0$.*

Proof (Proof Sketch). The $\Leftarrow$ direction is straightforward: if $sum(d^{c-k}) > 0$, then by definition of d^{c-k} we have accomplished our goal and $v_c > v_\omega$. For the $\Rightarrow$ direction, if deleting the smallest k elements in d^c still leaves $sum(d^{c-k}) < 0$, then it is impossible to find any other subset of groups $G \subseteq I$ to delete and have $v_c > v_\omega$, since we chose the k groups with the largest $v_{i,\omega} - v_{i,c}$, and, consequently, added the largest possible

[1] Note that "traditional" election control by deleting votes is a special case of our setting, where each group contains a single voter.

$\sum_i v_{i,\omega} - v_{i,c}$ to $sum(d^c)$. Since the remaining tally difference is still negative, it is not possible to make c have more votes than ω.

The process of computing a group-level election control approach is shown in Algorithm 1. For each candidate $c \in C \setminus \{\omega\}$, denoted by $C^{-\omega}$, Lines 1 - 4 check whether

Algorithm 1: Optimal Election Control by Group Deletion

1 **for** $c \in C^{-\omega}$ **do**
2 $\quad d^{c-k} \leftarrow$ Sort d^c in ascending order, delete the first k elements in d^c;
3 $\quad$ **if** $sum(d^{c-k}) > 0$ **then**
4 $\quad\quad$ **return** Attack voter groups corresponding to deleted elements;
5 **return** No control approach;

there exists an attack where c beats ω (based on Proposition 1). If no such attack exists for all candidates in $C^{-\omega}$, election control is not possible. It is not difficult to see that the complexity of Algorithm 1 is O($|C|n \log n$), which yields the following:

Theorem 1 *Election control preventing a candidate ω from winning by deleting k voter groups can be accomplished in O($|C|n \log n$) time.*

3 Protecting Elections

Given that plurality is extremely vulnerable to control by deleting voter groups, we now pose the dual question: is it possible for a party interested in maintaining election integrity (henceforth, *defender*) to ensure that plurality is resilient to control? To address this question, we consider the following model of protection. The defender can deploy $m \leq n$ protection resources (e.g., physical protection, electronic auditing, etc) to protect individual voter groups from attacks. If a group i is protected, we assume that it cannot be deleted by the adversary. We now ask: how hard is it for the defender to guarantee that a given set of resources m is sufficient to protect the election, that is, to ensure that it is impossible for an attacker to make ω lose by deleting unprotected voter groups?

Definition 1 (Hitting Set Problem) *A set G, a set U consisting of subsets $\hat{G}$ of G.* **Question:** *does there exist a 'hitting set' $G' \subseteq G$ with $|G'| = m$, so that $\forall \hat{G} \in U, G' \cap \hat{G} \neq \emptyset$.*

Theorem 2 *Checking whether m protection resources is sufficient to prevent control is NP-Complete.*

Proof. It is easy to see that this decision problem is in NP. To show that it is NP-Hard, we reduce from the hitting set problem. Specifically, we show that for any hitting set problem, we can construct an election with n voter groups, so that there exists a hitting set G' iff it is possible to prevent any control with m resources.

Given a hitting set problem, we construct a corresponding election as follows. There are $n = |G|$ voter groups and $|U| + 1$ candidates. Each $i \in G$ corresponds to a voter group. Each $\hat{G} \in U$ corresponds to a candidate other than ω. For candidate c corresponding to $\hat{G}$, for any voter group i, if $i \in \hat{G}$, then $d_i^c = -1$, i.e., c has 1 less vote than ω in group i; otherwise $d_i^c = 0$, i.e., c and ω ties in group i. Assume that the attacker can attack up to $k = n - m$ groups.

The $\Leftarrow$ direction: If there exists a defender strategy which protects m voter groups, i.e., $G' \subset G$ with $|G'| = m$, so that the attacker has no way to control the election, it indicates that for each candidate c, i.e., an element $\hat{G} \in U$, at least one voter group i in which $d_i^c = -1$ is protected, i.e., $G' \cap \hat{G} \neq \emptyset$. This is because if there exists a candidate c, all voter groups with $d_i^c = -1$ are unprotected, then the attacker can successfully attack all such groups and c will tie with ω in the left votes. Thus the protected voter groups satisfy that $\forall \hat{G} \in U, G' \cap \hat{G} \neq \emptyset$, which is a required hitting set.

The $\Rightarrow$ direction: Given a hitting set $G' \subset G$, the defender can protect all voter groups $i \in G'$. Thus, even if the attacker attacks all the unprotected voter groups, each candidate $c \in C^{-\omega}$ still has at least 1 vote less than ω. Therefore, no attacker strategy can control the election.

Theorem 2 leaves us with two questions: 1) does this mean that we cannot protect elections in practice, and 2) is all hope lost if m is insufficient to protect an election? In answering question 2, clearly we cannot protect the election if protection resources are allocated deterministically. However, when resources are limited, randomized allocation can offer tremendous value, increasing uncertainty and raising the stakes for attackers [22]. We propose to address both of these questions through a single framework: a Stackelberg game model in which the defender (of the election) first chooses a randomized allocation of m protection resources, and the attacker follows by choosing k groups to attack. Formally, let s denote a pure strategy of the defender, where $s_i \in \{0, 1\}$ indicates whether a voter group i is protected. Similarly, the attacker's pure strategy is a vector a where a_i indicates whether group i is attacked. We use $\mathcal{S}$ and $\mathcal{A}$ to represent the strategy space of the defender and the attacker respectively. Let $P(s, a) \in \{0, 1\}$ be an indicator denoting whether an attack a succeeds when a pure protection strategy s is played. Implicitly, we have assumed that both the attacker and defender know the net voting tallies for each location i. We relax this assumption in Section 5. Utilities of the attacker and defender are then defined by $u_A(s, a) = P(s, a)$ and $u_D(s, a) = -P(s, a)$, respectively, so that the game is zero-sum. Since we allow randomization for the defender, let $\mathbf{x}$ denote its randomized (mixed) strategy, with x_s the probability that a pure strategy $s \in \mathcal{S}$ is used.

Since the game is zero-sum, the Stackelberg equilibrium strategy for the defender is equivalent to its Nash equilibria [23]. Consequently, one can use a well-known linear programming formulation, shown as a Linear Program 1 (henceforth, Core-LP) below, for solving zero-sum normal-form games [24].

$$\text{Core-LP}(\mathcal{S}, \mathcal{A}): \qquad \min_{\mathbf{x}} p \tag{1a}$$

$$p \geq \sum\nolimits_{s \in \mathcal{S}} x_s P(s, a), \quad \forall a \in \mathcal{A}. \tag{1b}$$

The central challenge with this approach is that it requires one to explicitly enumerate all pure strategies for both the defender and attacker. Since in our cases the strategy space for both players is combinatorial, this is a non-starter. We therefore develop a *Double Oracle* approach for addressing this scalability issue.

4 Double Oracle Approach

The double oracle framework is a common approach for solving zero-sum games with exponential strategy spaces of both players [3, 2]. The idea is to start with a small set of strategies for both players, compute equilibrium in this restricted game using Core-LP, and check whether either player has a best response in the full strategy space that improves their payoff. If such a strategy exists for either player, it is added to the Core-LP, which is re-solved. Otherwise, we have proven that the resulting restricted equilibrium is a Stakcelberg / Nash equilibrium of the full game.

The Double-Oracle approach is not itself an algorithm, as it does not specify how to compute a best response for each player in the full strategy space. Indeed, in general this would require full enumeration of player strategies. The key is to develop effective approaches to compute such best responses—that is, effective oracles for both players—which is problem dependent. For example, none of the prior approaches (e.g., [2]) are applicable in our case, because of modeling differences. Our central contributions in this section are therefore: 1) novel mixed-integer linear programming (MILP) formulations for both oracles, and 2) heuristic algorithms to speed up the computation of the oracles.

Our full double-oracle method is shown in Algorithm 2. Line 3 computes the mixed strategy equilibrium of the restricted game, $(\mathbf{x}, \mathbf{y})$, where $\mathbf{y}$ is the dual solution of Core-LP representing attacker's mixed strategy. We then make use of two types of oracles: heuristic oracles, which allow us to quickly check the existence of *better* responses (AO-Better and DO-Better, for attacker and defender, respectively), and exact oracles (AO-MILP and DO-MILP), which are optimal.

Algorithm 2: Double Oracle Approach

1 Input: $\mathcal{S}' \subset \mathcal{S}$; $\mathcal{A}' \subset \mathcal{A}$;
2 **while do**
3     $(\mathbf{x}, \mathbf{y}) \leftarrow$ Core-LP$(\mathcal{S}', \mathcal{A}')$;
4     $a \leftarrow$ AO-Better$(\mathbf{x})$;
5   **if** $a = \emptyset$ **then** $a \leftarrow$ AO-MILP$(\mathbf{x})$;
6     $s \leftarrow$ DO-Better$(\mathbf{y})$;
7   **if** $s = \emptyset$ **then** $s \leftarrow$ DO-MILP$(\mathbf{y})$;
8   **if** $a \in A$ *and* $s \in S$ **then**
9   return $\mathbf{x}$;
10   **else**
11         $\mathcal{A}' \leftarrow \mathcal{A}' \cup \{a\}, \mathcal{S}' \leftarrow \mathcal{S}' \cup \{s\}$;

Next, we describe both the exact and heuristic oracles for the defender and attacker, observing in the process that both best response problems are NP-Hard.

4.1 Attacker Oracle

Complexity: It would seem that in Theorem 1 we had already shown that controling election in our model is in P. However, this result assumed that no protection is deployed (equivalently, that protection is deterministic). Surprisingly, when protection is randomized, election control, which we also refer to as the attacker's *best response* or *oracle*, is NP-Hard, as the following result attests (in this result, $\mathcal{S}'$ represents the support of the defender's mixed strategy).

Theorem 3 *Let $\mathcal{S}'$ be a set of defender strategies. Checking whether there exist k groups an attack on which would control an election no matter which $s \in \mathcal{S}'$ is played by the defender is NP-Complete, even with only two candidates.*

Proof. It is easy to see that this decision problem is in NP. To show that it is NP-Hard, we reduce from the hitting set problem shown in Definition 1. Specifically, we show that for any hitting set problem, we can construct an election with n voter groups, 2 candidates, and a set $\mathcal{S}'$ of defender strategies, so that there exists a hitting set G' iff there exists an attacker strategy which can control the election no matter which defender strategy $s \in \mathcal{S}'$ is played.

Given a hitting set problem as is shown in Definition 1, we construct an election with $|G| + 1$ voter groups, two candidates, ω and another candidate c. Each $i \in G$ corresponds to a voter group, in which $d_i^c = -1$, i.e., c has one less vote than ω in voter group i. In the extra voter group j which does not correspond to any element in G, $d_j^c = |G| - 1$. Thus c has 1 less vote than ω in total. Each $\hat{G} \in U$ corresponds to a defender pure strategy, in which voter group j and voter groups $i \in G \setminus \hat{G}$ are protected. For example, still assume that $G = \{1, 2, 3\}$ and $U = \{\{1, 2\}, \{2, 3\}, \{2\}, \{3\}\}$. Then the vote states of the candidate c under all defender strategies are shown in Table 1, e.g., P in position $(3, s^{\{1,2\}})$ means that in defender strategy corresponding to $\{1, 2\} \in U$, voter group $3 \in G$ is protected. Assume that the attacker can attack up to m groups.

	$s^{\{1,2\}}$	$s^{\{2,3\}}$	$s^{\{2\}}$	$s^{\{3\}}$
1	-1	P	P	P
2	-1	-1	-1	P
3	P	-1	P	-1
4	P	P	P	P

Table 1. Votes under defender strategies

The $\Leftarrow$ direction: If there exists an attacker strategy which attacks m voter groups i.e., $G' \subset G$ with $|G'| = m$, so that he can control the election no matter which defender strategy $s \in \mathcal{S}'$ is played, it indicates that for each defender strategy s, i.e., an element

$\hat{G} \in U$, at least one voter group $i \in G$ with $d_i^c = -1$ is attacked, i.e., $G' \cap \hat{G} \neq \emptyset$. Otherwise the attacker cannot control the election if s is played. Therefore, G' is a required hitting set.

The $\Rightarrow$ direction: Given a hitting set $G' \subset G$, the attacker can attack all voter groups $i \in G'$. Thus no matter which $s \in \mathcal{S}'$ is played by the defender, at least one unprotected voter group with $d_i^c = -1$ is attacked. Since ω only has 1 more vote than c in the original voting, the attacker can prevent ω from winning no matter which $s \in \mathcal{S}'$ is played.

Exact Solution: Although computing attacker's best response (oracle) is NP-Hard, we now develop an exact compact mixed-integer linear program (MILP) for it, which we term **AO-MILP**. Formally, the attacker's best response involves solving

$$\max_{a \in \mathcal{A}} \sum_{s \in \mathcal{S}'} P(s,a) x_s \tag{2}$$

for a given mixed strategy $\mathbf{x}$. Our first step is to formulate the attacker oracle as a mathematical (non-linear) program. The main technical challenge involved is representing $P(s,a)$, which is a non-trivial function of s and a. We do this implicitly in AO-MP by using an auxiliary binary variable z_s.

$$\text{AO-MP}: \qquad \max_a \quad \sum\nolimits_{s \in \mathcal{S}'} \quad z_s \cdot x_s \tag{3a}$$

$$\sum\nolimits_i a_i \leq k \tag{3b}$$

$$\sum\nolimits_{c \in C^{-\omega}} e_s^c = 1, \quad \forall s \in \mathcal{S}' \tag{3c}$$

$$z_s \sum\nolimits_{c \in C^{-\omega}} e_s^c \left(\sum\nolimits_i d_i^c (1 - (1 - s_i) a_i) \right) \geq 0, \forall s \in \mathcal{S}' \tag{3d}$$

$$a_i, z_s, e_s^c \in \{0, 1\}. \tag{3e}$$

Constraint (3b) enforces feasibility of the attacker's strategy vector a. Next we explain Constraints (3c)-(3d). Given a strategy pair (s,a), votes in group i are deleted only if $s_i = 0$ and $a_i = 1$. Thus for each candidate $c \in C^{-\omega}$, the vote difference between c and ω is $d^{c\prime} = \sum_i d_i^c - \sum_i (1 - s_i) a_i d_i^c$. Note that $z_s = 1$, i.e., attacker succeeds given strategy pair (s,a), as long as there exists one candidate who has no fewer votes left than ω given (s,a), i.e., $d^{c\prime} \geq 0$. Variables e_s^c are thus introduced to check whether there exists such a candidate. Constraints (3c), (3d), and the objective together ensure that if there exists such a candidate c^* for some s, the corresponding $e_s^{c^*}$ will be set as 1 and e_s^c for all other candidates will be set as 0. Thus, $\sum_{c \in C^{-\omega}} e_s^c \left(\sum_i d_i^c (1 - (1 - s_i) a_i) \right) \geq 0$, and the associated $z_s = 1$, yielding, in combination with Constraint (3b) a pure strategy for the attacker that maximizes its success probability given $\mathbf{x}$.

While AO-MP includes non-linear constraint (3d), because all variables involved are binary, this constraint can be linearized in a standard way using McCormick inequalities [25], yielding an MILP for computing the attacker's best response.

Heuristic "Better" Response: The main issue with AO-MP is its poor scalability. However, note that we need only compute a *better* response for the attacker in each iteration of the Double-Oracle method to make progress; by doing so quickly, even if heuristically, we can considerably speed up equilibrium computation. As long as we ultimately fall back on the MILP to check optimality, we lose no solution guarantee in the process.

We take two steps to find a better response for the attacker. First, we look for a subset $\mathcal{S}'' \subset \mathcal{S}'$ with $\sum_{s \in \mathcal{S}''} x_s > p$, where p is the objective value of Core-LP restricted to a small subset of attacker strategies $\mathcal{A}'$ in the previous iteration. Second, we look for an attacker pure strategy a which can successfully affect voting result no matter which pure strategy $s \in \mathcal{S}''$ is played by the defender, i.e., $P(s,a) = 1 \ \forall s \in \mathcal{S}''$. If we can successfully find such a set $\mathcal{S}''$ and a pure strategy a, the attacker will succeed with a probability of at least $\sum_{s \in \mathcal{S}''} x_s$ if he plays pure strategy a. Since $\sum_{s \in \mathcal{S}''} x_s > p$, a is a better strategy than any $a' \in \mathcal{A}'$.

The full heuristic approach, **AO-Better**, is shown in Algorithm 3. We first sort the defender strategies in $\mathcal{S}'$ in decreasing order of x_s, obtaining a sorted vector $\bar{S}$ with s^ρ the ρth largest element (Line 3). We then look for set $\mathcal{S}''$ consisting of adjacent strategies in $\bar{S}$ (Lines 5 - 6). For each $\mathcal{S}''$, we check that if there exists a candidate c, such that if the attacker attacks k areas which are not protected by any strategy $s \in \mathcal{S}''$, c will have more votes remaining than ω. If there exists such a candidate, then the corresponding attacker strategy leads to success no matter which $s \in \mathcal{S}''$ is played by the defender, and is better than any in $\mathcal{A}'$ (Lines 8 - 11). If no better strategy is found, then **AO-Better** returns an empty set.

Algorithm 3: Attacker's Better Response (**AO-Better**).

1 input: $\mathcal{S}', \mathbf{x}, p$;
2 $\bar{S} = \langle s^\rho, \rho \in 1,2,3,\cdots \rangle \leftarrow$ sort $s \in \mathcal{S}'$ by decreasing x_s;
3 **for** ρ *in* $1..|\bar{S}|$ **do**
4     $p' \leftarrow x_{s^\rho}, \mathcal{S}'' \leftarrow \{s^\rho\}, \rho' \leftarrow \rho + 1$;
5   **while** $p' \leq p$ *and* $\rho' \leq |\bar{S}|$ **do**
6         $p' \leftarrow p' + x_{s^{\rho'}}, \mathcal{S}'' \leftarrow \mathcal{S}'' \cup \{s^{\rho'}\}, \rho' \leftarrow \rho' + 1$;
7   **if** $p' > p$ **then**
8   **for** $c \in C^{-\omega}$ **do**
9             $d^{c\prime} \leftarrow \langle d_i^c : i$ with $s_i = 0, \forall s \in \mathcal{S}'' \rangle$;
10             $d^{(c-k)\prime} \leftarrow$ delete the smallest k elements in $d^{c\prime}$;
11   **if** *sum*($d^{(c-k)\prime}$) ≥ 0 **then**
12   **return** attack the k groups corresponding to deleted elements;
13 **return** $\emptyset$;

4.2 Defender Oracle

We now proceed to analyze the NP-Hard defender oracle (Theorem 2).

Exact Solution: The defender's oracle, or best response, can be defined as

$$\max_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}'} (1 - P(s,a)) y_a. \tag{4}$$

Just as in the attacker oracle formulation, we proceed to develop the (non-linear) mathematical integer program to compute the defender's best response.

$$\text{DO-MP:} \qquad \max_s \sum_{a \in \mathcal{A}'} z_a \cdot y_a \tag{5a}$$

$$\sum_i s_i \le m \tag{5b}$$

$$z_a \sum_i (d_i^c (1-(1-s_i)a_i)+1) \le 0, \forall c \in C^{-\omega}, a \in \mathcal{A}' \tag{5c}$$

$$s_i, z_a \in \{0,1\}. \tag{5d}$$

There is an important difference from the attacker oracle: in particular, $z_a = 1$ (that is, the defender successfully blocks an attack strategy $a \in \mathcal{A}'$, where $\mathcal{A}'$ is the attacker strategy from the previous iteration of Double-Oracle) only if all candidates $c \in C^{-\omega}$ have fewer votes remaining than ω. Constraint (5c) ensures that $z_a = 1$ only when $\forall c \in C^{-\omega}$, $\sum_i d_i^c - \sum_i (1 - s_i) a_i d_i^c < 0$, while Constraint (5b) enforces feasibility of the defender's strategy. The resulting DO-MP thereby chooses the defender strategy which minimizes the probability of a successful attack for a fixed attacker mixed strategy $\mathbf{y}$. We can then linearize the nonlinear constraint (5c) by using McCormick inequalities [25], obtaining an MILP formulation of the defender oracle.
Heuristic "Better" Response (Algorithm 4): We first look for a subset $\mathcal{A}'' \subset \mathcal{A}'$ with $\sum_{a \in \mathcal{A}''} y_a > 1 - p$. Then we look for a defender pure strategy s which can "block" all attacker strategies $a \in \mathcal{A}''$, ensuring that the attacker will succeed with probability less than p. If such a strategy is found, then it is a better response for the defender. Algorithm 4 presents the full heuristic procedure.

Algorithm 4: Defender Oracle with Better Response

$s = \langle s_i = 0 : \forall i \in \{1, \cdots, n\} \rangle$, res = 0;
for *each* $\mathcal{A}''$ *with* $\sum_{a \in \mathcal{A}''} y_a > 1 - p$ **do**
  for $c \in C^{-\omega}$ **do**
    $d^{c\prime} \leftarrow \langle d_i^c : i$ with $a_i = 0, \forall a \in \mathcal{A}''$ or $s_i = 1 \rangle$;
    while *sum*$(d^{c\prime}) \ge 0$ *and res* $< m$ **do**
      $d^{c\prime\prime} \leftarrow d^c \setminus d^{c\prime}, i^* \leftarrow \text{argmin}_i \{d_i^{c\prime\prime}\}$;
      $d^{c\prime} \leftarrow d^{c\prime} \cup \{d_{i^*}^c\}$, $s_{i^*} \leftarrow 1$, res $\leftarrow$ res + 1;
  if $\forall c \in C^{-\omega}, d^{c\prime} < 0$ **then**
    return s;
return $\emptyset$;

5 Uncertainty about Voter Preferences

Our entire treatment of the problem so far assumed complete information about voter preferences for both the attacker and defender. We now show that this assumption is relatively straightforward to relax (from a technical perspective). Specifically, we retain the assumption that attacker has complete information, but assume that the defender is uncertain about voter preferences. Formally, let V denote a particular voting preference outcome, with R_V the defender's prior distribution over V. The defender's goal in this setting is to minimize the expected probability that the attacker can successfully control the election. Since the attacker knows V, this gives rise to a Bayesian Stackelberg game with V the attacker's type. Let $p_V(s, a)$ be a binary indicator representing whether the attacker can successfully control the voting given voting preferences V and a strategy pair (s, a). The optimal mixed strategy for the defender can then be computed by solving the following LP, which is a Bayesian extension of the Core-LP above:

$$\text{Baysian-LP}(\mathcal{S}, \mathcal{A}): \qquad \min_{\mathbf{x}} \sum_{V} R_V P_V \tag{6a}$$

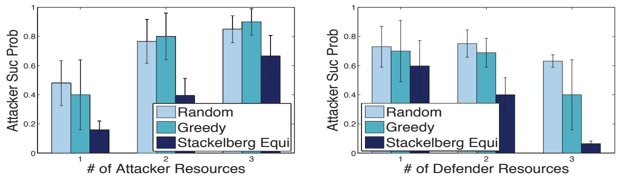
$$P_V \geq \sum_{s \in \mathcal{S}} x_s p_V(s, a), \quad \forall a \in \mathcal{A}, \forall V \tag{6b}$$

Note that this formulation is amenable to the same double oracle framework that was used to solve the complete information game. The primary difference is that now the attacker oracle must be run for each V, whereas the defender oracle requires a modified objective involving expected probability of election being controlled with respect to R_V. In practice, since the space of relevant voting preferences V is extremely large, we can take a collection of samples from this distribution and solve the linear program (6) solely using these samples to obtain an approximately optimal defense.

6 Evaluation

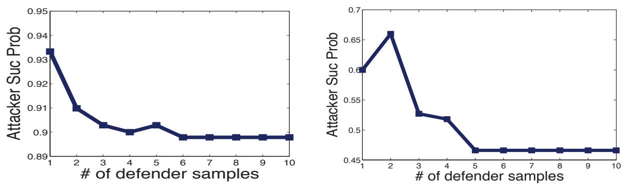
We evaluate the proposed algorithms on both synthetic and real data with respect to solution quality as well as scalability. Solution quality of our approach is compared to two baselines. The first, termed *Random*, is a uniformly random defense strategy. The second, terms *Greedy* is a "greedy" defender strategy in which security resources are assigned to protect voter groups in order of relative advantage ω has over the second ranked candidate in that group. Linear and mixed integer programs were solved using CPLEX 12.6.1.

We generated synthetic data by creating 30 voter groups and 5 candidates, and then randomly generated a tally for each candidate within each group uniformly in $[0, 100]$. Each data point is an average over 30 such samples. As we can observe from Figure 1(a) and 1(b), the Stackelberg equilibrium solution always outperforms both baselines above, in most cases quite dramatically. In addition, we compared solution quality of our approach extended to account for defender's uncertainty about voter preferences with the two baselines. The results were qualitatively the same: the Bayesian Stackelberg game approach significantly outperformed the alternatives. In addition, we consider the effect of the number of samples from the entire voter preference outcome space used in the Bayesian Stackelberg game to compute an approximate defense under uncertainty. We model uncertainty by taking baseline voting tallies (generated as described above), and adding zero-mean Gaussian noise. We study two cases: low uncertainty, where the variance of Gaussian noise is 10, and high uncertainty, where tallies of candidates are drawn uniformly in $[1, 400]$ and variance is 20. In both cases, we take 60 attacker types (drawn from this distribution) to be the ground truth. In Figures 2(a) and 2(b), the x-axis is the number of samples taken by the defender to solve Baysian-LP, while the y-axis indicates the optimal expected success probability of attackers. We observe that in both treatments very few samples (≤ 6) suffice to achieve a near-optimal solution. Additionally, we performed several robustness experiments, considering the impact of errors in problem parameters (e.g., voter preferences, in the complete information case and probability distribution over types in Bayesian games) on solution quality. We found that solutions are robust to such errors.



(a) Changing attack resources. (b) Changing defense resources.

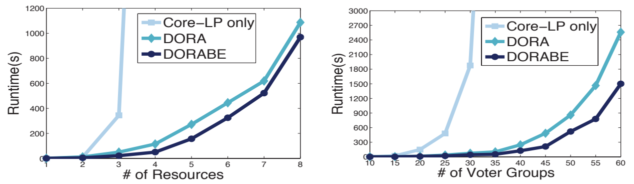
Fig. 1. Comparison of solution quality on synthetic data. "Stackelberg Equi" is the Stackelberg equilibrium solution.



(a) Low uncertainty. (b) High uncertainty.

Fig. 2. Bayesian-LP: Impact of the number of samples on solution quality.

Next we compare the scalability of the Core-LP algorithm ("Core-LP only") with the two proposed double oracle approaches: the first using only MILP oracles (DORA), and the second leveraging the heuristic methods as well (DORABE). The results in Figures 3(a) and 3(b) show that as we increase problem size, either in terms of the number of voter groups or defender resources, the double oracle approaches significantly out-

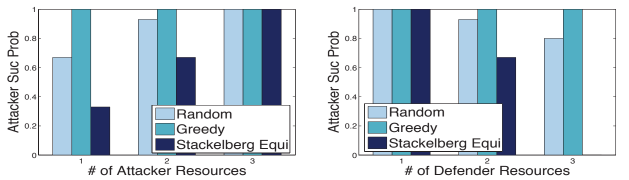


(a) Changing defense resources (20 voter groups).

(b) Changing the # of voter groups (3 defense resources).

Fig. 3. Scalability on synthetic data.

perform Core-LP. Moreover, DORABE offers significant computational savings compared to DORA.



(a) Changing attack resources.

(b) Changing defense resources.

Fig. 4. Solution quality on real data

Finally, we evaluate our algorithms on the 2002 French president election dataset [1], consisting of 2597 votes for 16 candidates by voters in 6 districts (voter groups). Figures 4(a) and 4(b) again compares the baselines to our algorithmic approach in terms of solution quality. As in the experiments with synthetic data, our approach demonstrates substantial improvement in defender's performance compared to baselines: in an extreme case, the attack success probability drops from 1 to nearly 0. (Thus Figure 4(b) only shows 2 lines with 3 resources.)

7 Conclusion

We study the problem of optimally protecting an election against group-deletion-control. We show that although plurality voting is vulnerable to control, it is NP-Hard to protect an election against it. We propose a double-oracle framework for computing an optimal protection strategy and develop compact mixed integer linear programs for both oracles, even though these are NP-Hard. We also propose heuristic oracles to further speed the double oracle framework up. Experimental results show that our algorithms outperform baseline alternatives, and scale to realistic problem instances.

References

1. J.-F. Laslier and K. Van der Straeten. A live experiment on approval voting. Experimental Economics, 11(1):97–105, 2008.
2. M. Jain, V. Conitzer, and M. Tambe. Security scheduling for real-world networks. In Proceedings of the 12th international conference on Autonomous agents and multi-agent systems(AAMAS), pages 215–222, 2013.
3. H. B. Mcmahan, G. J. Gordon, and A. Blum. Planning in the presence of cost functions controlled by an adversary. In Proceedings of the Twentieth International Conference on Machine Learning(ICML), pages 536–543, 2003.
4. S. Bhattacharjya. Low turnout and invalid votes mark first post war general polls. `http://www.sundaytimes.lk/100411/News/nws_16.html`, 2010.
5. RT. Election day bombings sweep pakistan: Over 30 killed, more than 200 injured. `https://www.rt.com/news/pakistan-election-day-bombing-136/`, 2013.
6. J. J. Bartholdi, C. A. Tovey, and M. A. Trick. How hard is it to control an election? Mathematical and Computer Modelling, 16(8):27–40, 1992.
7. K. Wojtas and P. Faliszewski. Possible winners in noisy elections. In AAAI, 2012.
8. J. Bannet, D. W. Price, A. Rudys, J. Singer, and D. S. Wallach. Hack-a-Vote: Security issues with electronic voting systems. IEEE Security and Privacy, 2(1):32–37, 2004.
9. S. Wolchok, E. Wustrow, D. Isabel, and J. A. Halderman. Attacking the Washington, DC internet voting system. In Conference on Financial Cryptography and Data Security, pages 114–128, 2012.
10. J. A. Halderman and V. Teague. The new south wales ivote system: Security failures and verification flaws in a live online election. In International Conference on E-voting and Identit, 2015.
11. S.Wolchok, E.Wustrow, J. A. Halderman, H. K. Prasad, A. Kankipati, S. K. Sakhamuri, V. Yagati, and R. Gonggrijp. Security analysis of indias electronic voting machines. In ACM Conference on Computer and Communications Security, 2010.
12. D. Springall, T. Finkenauer, Z. Durumeric, J. Kitcat, H. Hursti, M. MacAlpine, and J. A. Halderman. Security analysis of the estonian internet voting system. In ACM Conference on Computer and Communications Security, 2014.
13. J. J. Bartholdi, C. A. Tovey, and M. A. Trick. The computational difficulty of manipulating an election. Social Choice and Welfare, 6:227–241, 1989.
14. E. Elkind and H. Lipmaa. Small coalitions cannot manipulate voting. In Financial Cryptography and Data Security, pages 285–297, 2005.
15. E. Elkind and H. Lipmaa. Hybrid voting protocols and hardness of manipulation. In International Symposium on Algorithms and Computation, pages 206–215, 2005.
16. V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? Journal of the ACM, 54(3), 2007.
17. T. Coleman and V. Teague. On the complexity of manipulating elections. In Proceedings of the Thirteenth Australasian Symposium on Theory of Computing - Volume 65, pages 25–33, 2007.
18. A. Procaccia and J. S. Rosenschein. Junta distributions and the average-case complexity of manipulating elections. Journal of Artificial Intelligence Research, 28:157–181, 2007.
19. P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. Weighted electoral control. Journal of Artificial Intelligence Research, 52:507–542, 2015.
20. J. Chen, P. Faliszewski, R. Niedermeier, and N. Talmon. Elections with few voters: Candidate control can be easy. In AAAI Conference on Artificial Intelligence, pages 2045–2051, 2015.
21. L. Xu, F. Hutter, H. H. Hoos, and K. Leyton-Brown. Satzilla: Portfolio-based algorithm selection for sat. Journal of Artificial Intelligence Research, 32(1):565– 606, 2008.

22. P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordonez, and S. Kraus. Playing games with security: An efficient exact algorithm for Bayesian Stackelberg games. In Proc. 7th International Conference on Autonomous Agents and Multiagent Systems, pages 895–902, 2008.
23. D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, and M. Tambe. Stackelberg vs. Nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. Journal of Artificial Intelligence Research, 41:297–327, 2011.
24. V. Conitzer and T. Sandholm. Computing the optimal strategy to commit to. In ACM Conference on Electronic Commerce, pages 82–90, 2006.
25. G. McCormick. Computability of global solutions to factorable nonconvex programs: Part I - convex underestimating problems. Mathematical Programming, 10:147–175, 1976.

Strategyproof Matching with Minimum Quotas and Initial Endowments

Naoto Hamada, Ryoji Kurata, Suguru Ueda, Takamasa Suzuki, and Makoto Yokoo

Kyushu University,
744, Motooka, Nishi-ku,
Fukuoka, 819-0395, Japan,
{nhamada@agent.,kurata@agent.,ueda@,tsuzuki@,yokoo@}inf.kyushu-u.ac.jp

Abstract. We develop a strategyproof matching mechanism called Priority-List based Deferred Acceptance mechanism with Minimum Quotas (PLDA-MQ). Although minimum quotas are important in many real-world markets, existing strategyproof mechanisms require an unrealistic assumption that all students consider all schools acceptable (and vice-versa). PLDA-MQ can work under more realistic assumptions: (i) a student considers (at least) one particular school, which we call her initial endowment school, acceptable, and vice-versa, and (ii) the initial endowments satisfy all the minimum quotas. We require a matching to respect initial endowments; each student must be assigned to a school that is at least as good as her initial endowment. These assumptions are reasonable in public school choice programs and student reallocation problems. In general, standard stability is incompatible with minimum quotas. Thus, we introduce an alternative condition called Priority-List based (PL)-stability. Since the initial endowments satisfy all required properties (respecting minimum quotas/initial endowments and PL-stability), our research goal is to develop a strategyproof mechanism that improves students' welfare while satisfying these properties. We show that PLDA-MQ is theoretically optimal in terms of students' welfare; it obtains the student-optimal matching within all matchings that satisfy these properties.

1 Introduction

The theory of matching has been extensively developed for markets in which schools[1] have maximum quotas that cannot be exceeded [18]. However, in many real-world markets, *minimum* quotas are also present. For example, school districts may need at least a certain number of students in each school for the school to operate [2]. In a study on the market for Japanese medical residents [12], the Japanese government wants to assign more doctors to rural hospitals, and imposing minimum quotas is one possible approach. The United States Military Academy solicits cadet preferences over assignments to various Army branches,

[1] For concreteness, we use the terms of school choice throughout the paper, but our results can be applied to many other two-sided matching problems.

each of which has manning requirements [21, 22], which can be implemented as minimum quotas.

Although minimum quotas are relevant in many real-world settings and strategyproofness (i.e., no student has an incentive to misreport her preference) is important to many policymakers, there is a lack of strategyproof mechanisms that consider them. A notable exception is [6], in which they develop strategyproof mechanisms that can handle minimum quotas.

However, to guarantee that these mechanisms obtain feasible matchings (which respect minimum quotas), it is required that all students consider all schools *acceptable* and vice-versa. This requirement seems unrealistic in many applications. For example, in a public school choice program, it is not likely that a student is willing to attend a public school located very far away from her residence, unless the school offers some very appealing characteristics.

In this paper, we develop a mechanism that works under much more realistic assumptions. We require that a student consider at least one particular school acceptable, which we call her initial endowment school. In a public school choice program, this initial endowment school corresponds to the school that is closest to her residence, and by default, she is going to attend it unless she has special interest in other schools. We assume each school can consider some students unacceptable. However, we require that each school consider its initial endowment students acceptable. We also assume that the matching corresponds to the initial endowments is feasible, i.e., it satisfies all minimum/maximum quotas.

These assumptions are reasonable for a public school choice program. Before the school choice program was introduced, the only option for a student was to attend her nearby default school. The school choice program is introduced in order to provide her more options. Thus, it is reasonable to consider she will accept her default school, which was the only option without the program. Here, the introduction of the school choice program never hurts any student, i.e., a student is guaranteed to have a seat in her default school. On the other hand, to apply the mechanisms in [6], a student is not allowed to declare some schools unacceptable. Thus, the introduction of the school choice program can hurt some student, i.e., she might be assigned to a school that is worse than her default school.

Another application domain where these assumptions are appropriate is a student reallocation problem. Assume a feasible initial allocation is determined, and students experience a trial period. After it, each student has a chance to apply to another school if her interest changes. In this reallocation problem, it is natural to require that no student is reallocated to a school that is worse than her current assignment. Also, it is natural to assume the current allocation is feasible. We require a matching to respect initial endowments, i.e., each student must be assigned to a school that is at least as good as her initial endowment.

Our newly developed mechanism, which we call Priority-List based Deferred Acceptance mechanism with Minimum Quotas (PLDA-MQ), is an instance of the generalized Deferred Acceptance (DA) mechanism [10]; its skeleton resembles the well-known DA mechanism. Thus, our mechanism would be easy to adopt for policymakers and schools/students. This is a major advantage compared to applying a completely new/unfamiliar mechanism.

In general, fairness and standard nonwastefulness, both of which compose stability, are incompatible when minimum quotas are imposed [4]. Thus, in this paper, we introduce alternative definitions called Priority-List based (PL-) fairness and PL-nonwastefulness. They compose PL-stability.

There are two simple extensions of existing mechanisms that can work in our setting: (i) Artificial Cap Deffered Acceptance mechanism (ACDA) is a mechanism that artificially reduces the maximum quota of each school so that minimum quotas are automatically satisfied, and (ii) Top Trading Cycles mechanism (TTC), which is modified from the standard TTC [20] to respect initial endowments. ACDA is fair but it is neither PL-fair nor PL-nonwasteful. TTC does not satisfy any of these properties. Also, a trivial mechanism that simply returns the initial endowments is strategyproof, PL-stable, and respects minimum quotas.

Thus, we set our research goal to develop a strategyproof mechanism that improves students' welfare while satisfying the required properties. We show that PLDA-MQ is theoretically optimal in terms of students' welfare, i.e., it obtains the *student-optimal* matching within all matchings that satisfy these properties. Thus, as long as a mechanism satisfies these properties, it cannot obtain a better matching than that of PLDA-MQ.

We also conduct quantitative evaluation via computer simulation, which show that PLDA-MQ significantly improves students' welfare compared to other mechanisms.

The rest of this paper is organized as follows. We first discuss related works (Section 2). Next, we define the model of two-sided matching problem with minimum/maximum quotas and initial endowments (Section 3). Then, we describe two simple mechanisms; ACDA and TTC (Section 4). Next, we show our newly developed mechanism PLDA-MQ and its properties (Section 5). Finally, we conduct quantitative evaluation via computer simulation (Section 6).

2 Related Literature

In the context of school choice, minimum quotas are often imposed on different *types* of students (e.g., gender, socioeconomic status) [3, 4, 9, 14, 16, 23]. The crucial difference between our setting and these works is that minimum quotas are hard constraints that must be satisfied by any matching, while these works treat minimum quotas as "soft" constraints that may or may not actually be satisfied.

Ehlers et al. [4] show that if the constraints are interpreted as hard constraints, no mechanism that is fair and satisfies a definition they call constrained nonwastefulness can simultaneously be strategyproof. Due to this impossibility result, Fragiadakis et al. [6] develop two strategyproof mechanisms that renounce fairness or nonwastefulness. One is called Extended Seat Deferred Acceptance mechanism (ESDA), which is fair but wasteful, and the other is called Multi-Stage Deferred Acceptance mechanism (MSDA), which is nonwasteful but not

Table 1. Properties of mechanisms

	resp. initial endow.	PL-fairness	fairness	PL-NW	NW
PLDA-MQ	yes	yes	yes	yes	no
ESDA	no	no	yes	no	no
MSDA	no	no	no	yes	yes
TTC	yes	no	no	no	no
ACDA	yes	no	yes	no	no

fair. Based on their work, Goto et al. [7] develop a strategyproof mechanism that can handle hierarchical minimum quotas. We cannot use these mechanisms in our setting since they do not respect initial endowments.

The problem of matching with minimum quotas has also been addressed in the computer science community [2, 5, 11]. These works examine the complexity of verifying the existence of some types of *stable* matchings, but they do not provide explicit mechanisms or consider incentive issues.

A standard way to improve students' welfare with initial endowments is using Top Trading Cycles mechanism (TTC) [20]. However, we need to handle the indifference in students' preferences. This is not trivial in general [19]. However, in our setting, all students consider all of the seats in the same school as indifferent. Thus, we can use a mechanism based on Algorithm III in [13], which uses a simple common tie-breaking order. With it, the number of students assigned to each school does not change. Thus, the reallocation obtained by it is feasible.

However, in our setting, a school may have empty seats, i.e., it can accept more students than the initial endowment students. By allocating empty seats, the welfare of students can be improved. Although there exist mechanisms that can simultaneously handle initial endowments and empty seats [1], to the best of our knowledge, there exists no TTC-based mechanism that can handle both initial endowments and empty seats, while respecting minimum quotas.

If we modify the maximum quota of each school to the number of its initial endowment students, and also modify the priority in each school over students such that initial endowment students have higher priorities over other students, the matching obtained by the standard DA is feasible. We call this mechanism Artificial Cap Deferred Acceptance mechanism (ACDA). However, it is not PL-stable.

Table 1 summarizes the properties of mechanisms that respect minimum quotas ("NW" stands for nonwastefulness). Here, for ESDA and MSDA, we assume all students consider all schools acceptable and vice versa. Without this assumption, they cannot satisfy minimum quotas.

We utilize a general framework for developing strategyproof matching mechanisms under various distributional constraints [15]. Although they provide a useful toolkit for this purpose, developing a concrete mechanism that works for new types of constraints remains challenging; we need to appropriately design a choice function of schools so that their framework is applicable while the required design goals can be achieved.

3 Model

A matching market is given by $(S, C, X, \omega, \succ_S, \succ_C, \succ_{PL}, q_C, p_C)$. The meaning of each element is as follows.

- $S = \{s_1, s_2, \dots, s_n\}$ is a finite set of students.
- $C = \{c_1, c_2, \dots, c_m\}$ is a finite set of schools.
- $X \subseteq S \times C$ is a finite set of contracts. Contract $x = (s, c) \in X$ represents that student s is assigned to school c. For any $X' \subseteq X$, let X'_s denote $\{(s, c) \in X' \mid c \in C\}$, i.e., the sets of contracts related to student s. Also, let X'_c denote $\{(s, c) \in X' \mid s \in S\}$, i.e., the sets of contracts related to school c.
- $\omega : S \to C$ is an initial endowment function. $\omega(s)$ returns $c \in C$, which is s's initial endowment. When $\omega(s) = c$, we say student s is school c's initial endowment student. Let X^* denote $\bigcup_{s \in S}\{(s, \omega(s))\}$, i.e., X^* is the set of contracts, where each element is a contract between a student and her initial endowment school.
- $\succ_S = (\succ_{s_1}, \succ_{s_2}, \dots, \succ_{s_n})$ is a profile of the students' preferences. For each student s, $\succ_s$ represents the preference of s over X_s. We assume $\succ_s$ is strict for each s. We say $(s, c) \in X_s$ is acceptable for s if $(s, c) \succ_s (s, \omega(s))$ or $c = \omega(s)$ holds. We sometimes use such notations as $c \succ_s c'$ instead of $(s, c) \succ_s (s, c')$.
- $\succ_C = (\succ_{c_1}, \succ_{c_2}, \dots, \succ_{c_m})$ is a profile of the schools' priorities. For each school c, $\succ_c$ represents the priority of c over $X_c \cup \{(\phi, c)\}$, where (ϕ, c) represents an outcome such that c is unmatched. We assume $\succ_c$ is strict for each c. Contract (s, c) is acceptable for c if $(s, c) \succ_c (\phi, c)$ or $\omega(s) = c$ holds. We assume each contract x in X_c is acceptable for c. This is without loss of generality because if some contract is unacceptable to a school, we assume it is not included in X. We sometimes write $s \succ_c s'$ instead of $(s, c) \succ_c (s', c)$.
- $\succ_{PL}$ is a serial order over X called *priority list (PL)*, which represents a tie-breaking order among contracts.[2] We assume $\succ_{PL}$ respects the initial endowments, i.e., for each $x \in X^*$ and $x' \in X \setminus X^*$, $x \succ_{PL} x'$ holds. Also, we assume $\succ_{PL}$ respects $\succ_C$, i.e., for any $(s, c), (s', c) \in X \setminus X^*$, $(s, c) \succ_{PL} (s', c)$ holds iff $(s, c) \succ_c (s', c)$ holds.
- $q_C = (q_{c_1}, q_{c_2}, \dots, q_{c_m})$ is a vector of the schools' maximum quotas. Also, $p_C = (p_{c_1}, p_{c_2}, \dots, p_{c_m})$ is a vector of the schools' minimum quotas. We assume $\sum_{c \in C} p_c \le n \le \sum_{c \in C} q_c$ holds.

With a slight abuse of notation, for two sets of contracts, X' and X'', we denote $X'_s \succ_s X''_s$ if either (i) $X'_s = \{x'\}$, $X''_s = \{x''\}$, and $x' \succ_s x''$ for some $x', x'' \in X_s$ that are acceptable for s, or (ii) $X'_s = \{x'\}$ for some $x' \in X_s$ that is acceptable for s and $X''_s = \emptyset$. Furthermore, we denote $X'_s \succeq_s X''_s$ if either $X'_s \succ_s X''_s$ or $X'_s = X''_s$. Also, for $X'_s \subseteq X_s$, we say X'_s is acceptable for s if $X'_s = \{x\}$ and x is acceptable for s.

[2] A similar technique is used for handling regional maximum quotas in [8].

Definition 1 (feasibility). *$X' \subseteq X$ is* student-feasible *if for all $s \in S$, X'_s is acceptable for s. X' is* school-feasible *if for all $c \in C$, $p_c \leq |X'_c| \leq q_c$ holds. X' is* feasible *if it is student- and school-feasible. We call a feasible set of contracts a* matching*.*

We assume X^* is feasible, i.e., for each $c \in C$, $p_c \leq |X^*_c| \leq q_c$ holds.

For set of matchings $\mathcal{X}$, $X' \in \mathcal{X}$ is *student-optimal* within $\mathcal{X}$ if $X'_s \succeq_s X''_s$ holds $\forall X'' \in \mathcal{X}$, $\forall s \in S$. There is a chance that no student-optimal matching exists in $\mathcal{X}$. If a student-optimal matching does exist in $\mathcal{X}$, it must be unique.

A *mechanism* is a function that takes a profile of students' preferences as input and returns a matching. We say a mechanism is *strategyproof* if no student ever has any incentive to misreport her preference, regardless of what the other students report.

Definition 2 (fairness). *We say student s has justified envy toward $s' \neq s$ in matching X', where $(s, c) \in X'$, $(s', c') \in X' \setminus X^*$, and $(s, c') \in X \setminus X'$, if $(s, c') \succ_s (s, c)$ and $(s, c') \succ_{c'} (s', c')$ hold. Matching X' is* fair *if no student has justified envy. A mechanism is fair if it always gives a fair matching.*

In words, student s, who is allocated to c, has justified envy toward another student s' who is allocated to c', if s prefers c' over c, s has a higher priority than s' in c', and c' is not the initial endowment of s'.

Definition 3 (nonwastefulness). *We say student s claims an empty seat of c' in matching X', where $(s, c) \in X'$ and $(s, c') \in X \setminus X'$, if $(s, c') \succ_s (s, c)$, $|X'_{c'}| < q_{c'}$, and $|X'_c| > p_c$ hold. Matching X' is* nonwasteful *if no student claims an empty seat. A mechanism is nonwasteful if it always gives a nonwasteful matching.*

In words, student s, who is allocated to c, claims an empty seat of c', if s prefers c' over c, and the set of contracts obtained by moving s from c to c' is school-feasible.

There exists a case where no matching is simultaneously fair and nonwasteful, as shown in the following example.[3]

Example 1. Assume $S = \{s_1, s_2\}, C = \{c_1, c_2, c_3\}$, where $\omega(s_1) = \omega(s_2) = c_1$. $q_c = 2$ for all $c \in C$. $p_{c_1} = 1$, $p_{c_2} = p_{c_3} = 0$. The preferences of students are given as follows: $c_2 \succ_{s_1} c_3 \succ_{s_1} c_1$, $c_3 \succ_{s_2} c_2 \succ_{s_2} c_1$. The priorities of schools and $\succ_{PL}$ are given as: $s_1 \succ_{c_1} s_2$, $s_2 \succ_{c_2} s_1$, $s_1 \succ_{c_3} s_2$, $(s_1, c_1) \succ_{PL} (s_2, c_1) \succ_{PL} (s_2, c_2) \succ_{PL} (s_1, c_3) \succ_{PL} (s_1, c_2) \succ_{PL} (s_2, c_3)$.

Here, c_1 is the least popular school for both s_1 and s_2, but at least one student must be assigned to c_1 since $p_{c_1} = 1$. Assume s_1 is allocated to c_1. Then, s_2 must be allocated to her most preferred school c_3, or otherwise, s_2 claims an empty seat of c_3. However, then s_1 has justified envy toward s_2 since $s_1 \succ_{c_3} s_2$. Similarly, assume s_2 is allocated to c_1. Then s_1 must be allocated to her most preferred school c_2, or otherwise, s_1 claims an empty seat of c_2. However, then s_2 has justified envy toward s_1 since $s_2 \succ_{c_2} s_1$.

[3] This example is based on the example used in the proof of Theorem 1 in [4].

Given this impossibility result, we first introduce a weaker definition of nonwastefulness.

Definition 4 (PL-nonwastefulness). *Student s claims an empty seat of c' in matching X' based on PL, where $(s,c) \in X'$ and $(s,c') \in X \setminus X'$, if $(s,c') \succ_s (s,c)$, $|X'_{c'}| < q_{c'}$, $|X'_c| > p_c$ and $(s,c') \succ_{PL} (s,c)$ hold. Matching X' is* PL-nonwasteful *if no student claims an empty seat based on PL. A mechanism is PL-nonwasteful if it always gives a PL-nonwasteful matching.*

This definition weakens standard nonwastefulness, i.e., the claim of student s who is assigned to c to obtain an empty seat of c' is considered legitimate only if the tie-breaking rule supports this, i.e., $(s,c') \succ_{PL} (s,c)$.

Next, we introduce a *stronger* condition than standard fairness.

Definition 5 (PL-fairness). *We say student s has justified envy toward $s' \neq s$ in matching X' based on PL, where $(s,c),(s',c') \in X'$ and $(s,c'') \in X \setminus X'$, if $(s,c'') \succ_s (s,c)$, $|X'_{c''}| < q_{c''}$, $|X'_{c'}| > p_{c'}$, and $(s,c'') \succ_{PL} (s',c')$ hold. Matching X' is* PL-fair *if no student has justified envy or justified envy based on PL. A mechanism is PL-fair if it always gives a PL-fair matching.*

In words, if student s is assigned to c even though she hopes to be assigned to c'', and c'' can accept one more student, while another student s' is assigned to c' even though c' already satisfies its minimum quotas, then s can have justified envy toward s' if the tie-breaking rule supports this, i.e., $(s,c'') \succ_{PL} (s',c')$ holds.

Intuitively, PL-fairness requires that if we need to reject a contract to satisfy the minimum quota of some school, and there exist several candidate contracts to reject, the mechanism should reject the one in a fair way based on PL, i.e., the contract with the lowest priority according to PL should be rejected.

We say a matching is *PL-stable* if it is PL-fair and PL-nonwasteful. We say a mechanism is PL-stable if it always gives a PL-stable matching.

Let us consider the situation of Example 1, but all the schools prefer s_2 over s_1, and $\succ_{PL}$ is given as follows (which respects schools' priorities): $(s_1,c_1) \succ_{PL} (s_2,c_1) \succ_{PL} (s_2,c_2) \succ_{PL} (s_2,c_3) \succ_{PL} (s_1,c_2) \succ_{PL} (s_1,c_3)$. Since $p_{c_1} = 1$, at least one student must be assigned to c_1 even though c_1 is the least popular school for both students. A mechanism needs to decide which contract should be rejected, e.g., among $\{(s_1,c_3),(s_2,c_2)\}$. Here, since both schools c_2 and c_3 unanimously prefer s_2 over s_1, it is natural to assume (s_1,c_3) is rejected. Indeed, PL-fairness requires the mechanism to reject (s_1,c_3) that has the lowest priority according to PL (which respects schools' priorities).

Note that from these definitions, X^* is PL-stable. Since a student cannot have justified envy toward another student allocated to her initial endowment, X^* is fair. Also, for any $x \in X^*$ and $x' \in X \setminus X^*$, $x \succ_{PL} x'$ holds. Then, a student cannot have justified envy based on PL, or can claim an empty seat based on PL. Thus, X^* is PL-stable. Therefore, there always exists at least one PL-stable matching.

4 Simple extensions

In this section, we introduce two simple extensions of existing mechanisms that can work in our setting. First, we introduce ACDA. The idea of ACDA is used in the Japan Residency Matching Program, i.e., reducing maximum quotas of hospitals in urban areas such as Tokyo, so that more doctors will apply to hospitals in rural areas [12].

Definition 6 (ACDA).
For each school c, the original maximum quota q_c is decreased to $|X_c^*|$, i.e., the number of its initial endowment students. Also, the original school priority $\succ_c$ is modified so that each of its initial endowment contracts has a higher priority than any contract that is in $X_c \setminus X_c^*$. Then, the mechanism performs the standard DA procedure described as follows, which repeats the following steps.

Step t (≥ 1) *A student applies to her most preferred school from which she is not rejected so far. Then, each school deferred accepts students applying to it up to its maximum quotas. The rest of students are rejected. If no student is rejected, then return the current assignment as a final matching. Otherwise, go to Step $t + 1$.*

Next, we show TTC-based mechanism, which is one instance of Algorithm III in [13]. We assume there exists a common strict ordering $\succ_{ML}$ over students called the master list. Note that the master list is completely different from the priority list, which is a serial order over X. Without loss of generality, we assume $\succ_{ML}$ is defined as: $s_1 \succ_{ML} s_2 \succ_{ML} \ldots \succ_{ML} s_n$.

Definition 7 (TTC).

Step t (≥ 1) *For each school c, choose one remaining initial endowment student s as its representative, who has the highest priority according to $\succ_{ML}$. If there is no remaining initial endowment student, c has no representative. If there is no representative students from any school, the mechanism terminates. Otherwise, among these representative students, run the standard TTC procedure, i.e., each student points to a student who is endowed with her most favorite school (a student can point to herself). Then, there exists at least one cycle. Students in a cycle exchange seats of their schools and they leave the mechanism. Then, go to Step $t + 1$.*

5 New mechanism

In this section, we develop the Priority-List based Deferred Acceptance mechanism with Minimum Quotas (PLDA-MQ) and clarify its properties.

5.1 Definition of PLDA-MQ

We utilize the general framework for developing a strategyproof mechanism with various distributional constraints recently developed by Kojima et al. [15]. Their framework exploits *choice functions* for students (Ch_S) and schools (Ch_C), which are defined as follows:

Definition 8 (students' choice function). *For each student s, her choice function Ch_s is defined as follows. Given $X' \subseteq X$, let Y_s denote $\{x \in X'_s \mid x \text{ is acceptable for } s\}$. $Ch_s(X')$ returns $\{x\}$, s.t. $x \in Y_s$ and x is the most preferred contract in Y_s for s. If $Y_s = \emptyset$, then $Ch_s(X')$ returns $\emptyset$. Then the choice function of all students Ch_S is defined as $Ch_S(X') := \bigcup_{s \in S} Ch_s(X')$.*

Definition 9 (schools' choice function). *Given $X' \subseteq X$, the choice function of schools returns set of contracts $Ch_C(X')$, which is defined as follows:*

$$Ch_C(X') := \arg\max_{X'' \subseteq X'} f(X'').$$

Here, f is an evaluation function that aggregates the schools' priorities and distributional constraints. We assume f is unique-selecting, i.e., for all $X' \subseteq X$, there exists a unique $X'' \subseteq X'$ that maximizes $f(X'')$.

Based on these choice functions, the generalized Deferred Acceptance (DA) mechanism is defined as follows:

Definition 10 (generalized DA).

Step 1 $Re \leftarrow \emptyset$.
Step 2 $X' \leftarrow Ch_S(X \setminus Re), X'' \leftarrow Ch_C(X')$.
Step 3 *If $X' = X''$, then return X', otherwise, $Re \leftarrow Re \cup (X' \setminus X'')$, go to Step 2.*

Here, Re represents a set of contracts that are proposed by students and rejected by schools. Students are not allowed to propose a contract in Re, which is initially empty. Thus, they can choose their most preferred contracts and propose them to the schools. This set is X'. Then schools choose the most preferred subset X'' from X'. If no contract is rejected, the mechanism terminates. Otherwise, the rejected contracts are added to Re, and the mechanism repeats the same procedure.

Theorem 1, which is identical to Theorem 1 in [15], clarifies conditions so that the generalized DA satisfies several desirable properties.

Theorem 1 (Theorem 1 in [15]). *Suppose that the preference of the schools can be represented by an $M^\natural$-concave function f. Then,*

1. *the generalized DA mechanism is strategyproof and the obtained matching is student optimal in all* Hatfield-Milgrom (HM)-stable *matchings, and*
2. *the time complexity of the generalized DA mechanism is $O(T(f) \cdot |X|^2)$, assuming f can be calculated in $T(f)$ time.*

Here, matching X' is HM-stable if $X' = Ch_S(X') = Ch_C(X')$ holds and there exists no $x \in X \setminus X'$ such that $x \in Ch_S(X' \cup \{x\})$ and $x \in Ch_C(X' \cup \{x\})$ hold.

Furthermore, they show a sufficient condition where f becomes M$^\natural$-concave. Assume $f(X')$ is given by $\widehat{f}(X') + \widetilde{f}(X')$. $\widehat{f}$ represents a hard distributional constraint; it returns $-\infty$ if X' violates the hard constraint, and otherwise, it returns 0. $\widetilde{f}(X')$ represents the soft preference/priorities of schools. The effective domain of $\widehat{f}$ (denoted as $\mathrm{dom}\,\widehat{f}$) is defined as $\{X' \subseteq X \mid \widehat{f}(X') \neq -\infty\}$.

Theorem 2, which is identical to Theorem 3 in [15], shows a sufficient condition so that f becomes M$^\natural$-concave.

Theorem 2 (Theorem 3 in [15]). *If (i) $(X, \mathrm{dom}\,\widehat{f})$ is a mathematical structure called a matroid [17] and (ii) $\widetilde{f}(X')$ is given as $\sum_{x \in X'} v(x)$, then $f = \widehat{f} + \widetilde{f}$ is M$^\natural$-concave.*

Here, $v : X \to (0, \infty)$ is a function such that $x \neq x'$ implies $v(x) \neq v(x')$. We can assume $v(x)$ represents the value of contract x.

A matroid is defined as follows:

Definition 11 (matroid). *Let X be a finite set and $\mathcal{F}$ be a family of subsets of X. Pair $(X, \mathcal{F})$ is a* matroid *if it satisfies the following conditions.*

1. *$\emptyset \in \mathcal{F}$.*
2. *If $X' \in \mathcal{F}$ and $X'' \subset X'$, then $X'' \in \mathcal{F}$ holds.*
3. *If $X', X'' \in \mathcal{F}$ and $|X'| > |X''|$, then there exists $x \in X' \setminus X''$ such that $X'' \cup \{x\} \in \mathcal{F}$.*

Given these results, we appropriately define $\widehat{f}$ and $\widetilde{f}$ so that Theorem 2 can be applied. Defining $\widetilde{f}(X')$ is straightforward. Based on $\succ_{PL}$, we can define $v(\cdot)$ such that $x \succ_{PL} x'$ implies $v(x) > v(x')$.

Next we show how to appropriately define $\widehat{f}$.

Definition 12 (hard constraint). *$\widehat{f}(X')$ is defined as follows:*

$$\widehat{f}(X') := \begin{cases} 0 \text{ if } |X'_c| \leq q_c \text{ for all } c \in C, \\ \quad\quad \text{and } \sum_{c \in C} \max(|X'_c|, p_c) \leq n, \\ -\infty \text{ otherwise.} \end{cases}$$

Intuitively, if $\widehat{f}(X') = -\infty$ holds, there exists no $X'' \supseteq X'$ such that X'' is school-feasible.

With this definition of $\widehat{f}$, we call the generalized DA, where $f(X')$ is defined by $\widehat{f}(X') + \sum_{x \in X'} v(x)$, Priority-List based Deferred Acceptance mechanism with Minimum Quotas (PLDA-MQ).

Let us show the execution of PLDA-MQ.

Example 2. Assume $S = \{s_1, s_2, s_3, s_4\}, C = \{c_1, c_2, c_3\}$, where $\omega(s_1) = c_1, \omega(s_2) = c_2, \omega(s_3) = \omega(s_4) = c_3$. $q_c = 4$ for all $c \in C$. $p_{c_1} = p_{c_2} = 0, p_{c_3} = 1$. The preferences of students are given as follows: s_1 considers only c_1 as acceptable, $c_1 \succ_{s_2} c_2$, $c_1 \succ_{s_3} c_2 \succ_{s_3} c_3$, and $c_2 \succ_{s_4} c_3$. The priorities of schools and $\succ_{PL}$ are given as follows: $s_1 \succ_{c_1} s_2 \succ_{c_1} s_3$, $s_2 \succ_{c_2} s_3 \succ_{c_2} s_4$, $s_3 \succ_{c_3} s_4$, $(s_1, c_1) \succ_{PL} (s_2, c_2) \succ_{PL} (s_3, c_3) \succ_{PL} (s_4, c_3) \succ_{PL} (s_3, c_2) \succ_{PL} (s_2, c_1) \succ_{PL} (s_4, c_2) \succ_{PL} (s_3, c_1)$.

First, each student chooses her most preferred acceptable contract. Thus, $X' = \{(s_1, c_1), (s_2, c_1), (s_3, c_1), (s_4, c_2)\}$. Here $\widehat{f}(X') = -\infty$ since $\sum_{c \in C} \max(|X'_c|, p_c) = 3 + 1 + 1 = 5 > 4$. (s_3, c_1) is rejected since it has the lowest priority according to $\succ_{PL}$ within X'.

Next, s_3 chooses her second preferred contract (s_3, c_2), while other students choose the same schools as before. Thus, $X' = \{(s_1, c_1), (s_2, c_1), (s_3, c_2), (s_4, c_2)\}$. $\widehat{f}(X') = -\infty$ since $\sum_{c \in C} \max(|X'_c|, p_c) = 2 + 2 + 1 = 5 > 4$. Then (s_4, c_2) is rejected since it has the lowest priority according to $\succ_{PL}$ within X'.

Finally, s_4 chooses her second preferred contract (s_4, c_3). Then $X' = \{(s_1, c_1), (s_2, c_1), (s_3, c_2), (s_4, c_3)\}$. $\widehat{f}(X') = 0$ since $\sum_{c \in C} \max(|X'_c|, p_c) = 2 + 1 + 1 = 4$. Since no contract is rejected, the mechanism terminates.

In the situation described in Example 1, the matching obtained by PLDA-MQ is $\{(s_1, c_1), (s_2, c_2)\}$. This is PL-stable; s_2 cannot claim an empty seat of c_3 based on PL since $(s_2, c_2) \succ_{PL} (s_2, c_3)$ holds.

5.2 Properties of PLDA-MQ

The following lemmas hold. We omit proofs for space reasons.

Lemma 1. *$(X, \mathrm{dom}\ \widehat{f})$ is a matroid.*

Lemma 2. *$Ch_C(X')$ is equivalent to the following procedure:*

1. *$Y \leftarrow \emptyset$.*
2. *Remove (s, c) from X' such that $v((s, c))$ is largest in X'. If there exists no such contract, terminate the procedure and return Y.*
3. *If $\widehat{f}(Y \cup \{(s, c)\}) = 0$ then $Y \leftarrow Y \cup \{(s, c)\}$. Go to 2.*

Lemma 3. *Student s is never rejected by her initial endowment school $\omega(s)$, i.e., whenever $(s, \omega(s)) \in X'$, $(s, \omega(s)) \in Ch_C(X')$ holds.*

Lemma 4. *Matching X' is HM-stable iff it is PL-stable.*

From these lemmas, we obtain the following theorems.

Theorem 3. *PLDA-MQ obtains a feasible set of contracts.*

Proof. Assume PLDA-MQ obtains set of contracts X'. If $X'_s = \emptyset$, then $(s, \omega(s)) \in Ch_S(X' \cup \{(s, \omega(s))\})$ holds. Also, from Lemma 3, $(s, \omega(s)) \in Ch_C(X' \cup \{(s, \omega(s))\})$ holds. This violates the fact that the generalized DA obtains an HM-stable matching. If $X'_s = \{(s, c)\}$ holds, it is clear that for each student s, (s, c) is acceptable for s since $(s, c) \in Ch_S(X')$. Thus, X' is student-feasible. Since each student is accepted to some school, $|X'| = n$ holds.

Next, we show that X' is school-feasible. It is clear that all contracts in X'_c are acceptable for c. Also, for all $c \in C$, $|X'_c| \leq q_c$ holds, since otherwise, $\widehat{f}(X') = -\infty$ and $Ch_C(X')$ cannot be X'. Assume for contradiction, there exists $c \in C$ such that $|X'_c| < p_c$. Then $\sum_{c \in C} \max(|X'_c|, p_c) > |X'| = n$. Thus, $\widehat{f}(X') = -\infty$ and $Ch_C(X')$ cannot be X'.

Theorem 4. *PLDA-MQ is strategyproof, PL-stable, and obtains the student-optimal matching within all the PL-stable matchings.*

Proof. From Lemma 1 and Theorem 2, the f used in PLDA-MQ is M$^\natural$-concave. Thus, from Condition 1 of Theorem 1, PLDA-MQ is strategyproof and obtains the student-optimal matching within all the HM-stable matchings. From Lemma 4, HM-stability is equivalent to PL-stability. Thus, PLDA-MQ satisfies these properties.

Theorem 5. *The time complexity of PLDA-MQ is $O(|X|^3)$.*

Proof. It is clear that f can be calculated in $O(|X|)$, since both $\widehat{f}$ and $\widetilde{f}$ can be calculated in $O(|X|)$. From the fact that PLDA-MQ is M$^\natural$-concave and Condition 2 of Theorem 1, the time complexity of PLDA-MQ is $O(|X|^3)$.

6 Evaluation

In this section, we conduct quantitative evaluation via computer simulation. We set $n = 720, m = 36$. The minimum and maximum quotas of each school are 5 and 60, respectively. We assume schools are located in a 6×6 grid space (Fig. 1), where the right-end school is connected to the left-end school in each row, and the lower-end school is connected to the upper-end school in each column (i.e., schools constitute a torus). We employ this structure to avoid boundary effects. We assume the residence of each student is also located in this grid space. We allocate to each location 20 students who are initially endowed to the school in the location.

Also, we assume a student considers only nearby schools acceptable, i.e., a student considers school c acceptable only when the Manhattan distance between her residence and c is at most 2, i.e., each student considers at most 13 schools acceptable. For example, in Fig. 1, a student located in the black circle considers 13 schools located at the black and gray circles acceptable. We generate the students' preferences as follows. We draw one common vector v of the cardinal utilities from set $[0,1]^m$ uniformly at random. We then draw private vector u_s of the cardinal utilities of each student s from the same set, again uniformly at random. Next we construct cardinal utilities over all m schools for student s as $\alpha v + (1-\alpha)u_s$ for some $\alpha \in [0,1]$. Here the i-th element of this vector represents the cardinal utility for school c_i. We then convert these cardinal utilities into an ordinal preference relation for each student over the acceptable schools. The higher the value of α is, the more correlated the students' preferences are. In this experiment, we vary α from 0.0 to 1.0 in increments of 0.1.

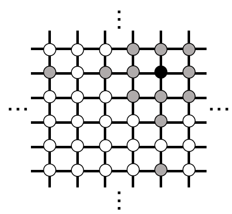


Fig. 1. School locations

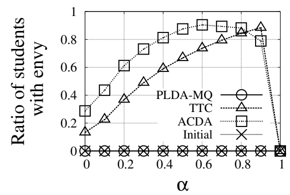


Fig. 2. Ratio of students with envy

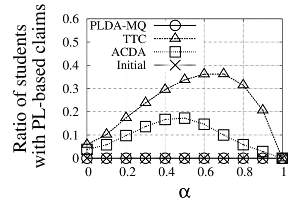


Fig. 3. Ratio of students with PL-based claims

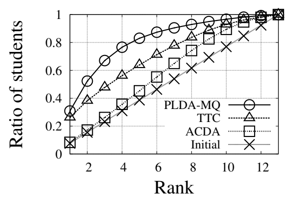


Fig. 4. CDFs of students' welfare ($\alpha = 0.6$)

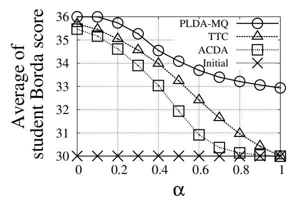


Fig. 5. Borda score

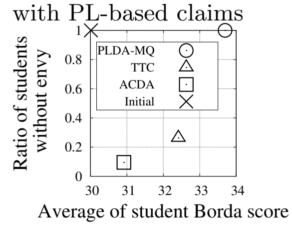


Fig. 6. Comparison of efficiency/fairness ($\alpha = 0.6$)

The priority of school c for the students is determined as follows. We draw one common vector u from set $[0, 1]^n$ uniformly at random. Here u_i represents the utility for student s_i. School c prefers student s_i over s_j if s_i lives closer to c than s_j. If the distance between s_i and c is equal to the distance between s_j and c, then c prefers student s_i over s_j if $u_i > u_j$ holds. We create 100 problem instances for each parameter setting. In ACDA, we set the maximum quota of each school to 20. In TTC, we use $\succ_{ML}$ that is based on the common vector, i.e., for two students s_i and s_j, $s_i \succ_{ML} s_j$ holds if $u_i > u_j$.

Figure 2 shows the ratio of students with justified envy or justified envy based on PL. Since PLDA-MQ and the initial endowments are PL-fair, no student has such envy. In TTC and ACDA, the number of students with envy quickly increases as α increases.

Figure 3 shows the ratio of students who claim empty seats based on PL. PLDA-MQ and the initial endowments are PL-nonwasteful. Thus, no student claims an empty seat based on PL. In TTC and ACDA, on the other hand, there exist students with PL-based claims. Note that when $\alpha = 1$, all students basically have the same preference. Then all the students are allocated to their initial endowment schools in TTC and ACDA; the ratio of students with envy (Fig. 2) and the ratio of students with PL-based claims (Fig. 3) becomes 0.

Figure 4 shows the students' welfare by plotting the cumulative distribution functions (CDFs) of the average number of students matched with their k-th or higher ranked school under each mechanism when $\alpha = 0.6$. The students' welfare under PLDA-MQ is higher than their welfare under TTC and ACDA. Also, PLDA-MQ significantly improves students' welfare compared to the initial endowments.

We also compare the average Borda score of students in by varying α (Fig. 5). If a student is assigned to her k-th-choice school, her Borda score is $m - k + 1$.

Thus, a higher score is more desirable. We can see the Borda score of PLDA-MQ is higher than other mechanisms regardless of α.

To illustrate the efficiency/fairness obtained by these mechanisms, we plot the average point of obtained matchings in a two-dimensional space in Fig. 6 (when $\alpha = 0.6$), where the x-axis shows the average Borda score of the students, and the y-axis shows the ratio of students without envy. Thus, the point located in the north-east is preferable. PLDA-MQ is PL-fair and much more efficient than TTC and ACDA.

7 Conclusions

In this paper, we developed a new strategyproof mechanism called PLDA-MQ that can handle minimum quotas. This mechanism can work under more realistic assumptions than existing mechanisms, i.e., a student consider her initial endowment school acceptable, and the initial endowments is feasible. We proved that PLDA-MQ satisfies required properties (respecting minimum quotas/initial endowments, PL-stability), and obtains the student-optimal matching within all matchings that satisfy the required properties. We also conducted quantitative evaluation via computer simulation, which show that PLDA-MQ significantly improves students' welfare compared to other mechanisms.

Our future works include developing a TTC-based mechanism that can allocate empty seats while respecting minimum quotas. We expect this mechanism to be more efficient than PLDA-MQ, although it is not fair.

References

1. Abdulkadiroğlu, A., Sönmez, T.: House allocation with existing tenants. Journal of Economic Theory 88, 233–260 (1999)
2. Biró, P., Fleiner, T., Irving, R.W., Manlove, D.F.: The college admissions problem with lower and common quotas. Theoretical Computer Science 411(34-36), 3136–3153 (2010)
3. Braun, S., Dwenger, N., Kübler, D., Westkamp, A.: Implementing quotas in university admissions: An experimental analysis. Games and Economic Behavior 85, 232–251 (2014)
4. Ehlers, L., Hafalir, I.E., Yenmez, M.B., Yildirim, M.A.: School choice with controlled choice constraints: Hard bounds versus soft bounds. Journal of Economic Theory 153, 648–683 (2014)
5. Fleiner, T., Kamiyama, N.: A matroid approach to stable matchings with lower quotas. In: Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-12). pp. 135–142 (2012)
6. Fragiadakis, D., Iwasaki, A., Troyan, P., Ueda, S., Yokoo, M.: Strategyproof matching with minimum quotas. ACM Transactions on Economics and Computation (2015), forthcoming (an extended abstract in AAMAS, pages 1327–1328, 2012)
7. Goto, M., Hashimoto, N., Iwasaki, A., Kawasaki, Y., Ueda, S., Yasuda, Y., Yokoo, M.: Strategy-proof matching with regional minimum quotas. In: Proceedings of Thirteenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2014). pp. 1225–1232 (2014)

8. Goto, M., Iwasaki, A., Kawasaki, Y., Yasuda, Y., Yokoo, M.: Improving fairness and efficiency in matching markets with regional caps: Priority-list based deferred acceptance mechanism (2014), mimeo (the latest version is available at http://mpra.ub.uni-muenchen.de/53409/)
9. Hafalir, I.E., Yenmez, M.B., Yildirim, M.A.: Effective affirmative action in school choice. Theoretical Economics 8(2), 325–363 (2013)
10. Hatfield, J.W., Milgrom, P.R.: Matching with contracts. American Economic Review 95(4), 913–935 (2005)
11. Huang, C.C.: Classified stable matching. In: Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-10). pp. 1235–1253 (2010)
12. Kamada, Y., Kojima, F.: Efficient matching under distributional constraints: Theory and applications. American Economic Review 105(1), 67–99 (2015)
13. Kesten, O.: On two competing mechanisms for priority-based allocation problems. Journal of Economic Theory 127, 155 – 171 (2006)
14. Kojima, F.: School choice: Impossibilities for affirmative action. Games and Economic Behavior 75(2), 685–693 (2012)
15. Kojima, F., Tamura, A., Yokoo, M.: Designing matching mechanisms under constraints: An approach from discrete convex analysis. In: Proceedings of the Seventh International Symposium on Algorithmic Game Theory (SAGT-2014) (2014), the full version is available at http://mpra.ub.uni-muenchen.de/62226)
16. Kurata, R., Goto, M., Iwasaki, A., Yokoo, M.: Controlled school choice with soft bounds and overlapping types. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-2015). pp. 951–957 (2015)
17. Oxley, J.: Matroid Theory. Oxford University Press (2011)
18. Roth, A.E., Sotomayor, M.A.O.: Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis (Econometric Society Monographs). Cambridge University Press. (1990)
19. Sabán, D., Sethuraman, J.: House allocation with indifferences: a generalization and a unified view. In: Proceedings of the fourteenth ACM conference on Economics and Computation (EC-2014). pp. 803–820 (2013)
20. Shapley, L., Scarf, H.: On cores and indivisibility. Journal of Mathematical Economics 1(1), 23 – 37 (1974)
21. Sönmez, T.: Bidding for army career specialties: Improving the ROTC branching mechanism. Journal of Political Economy 121(1), 186–219 (2013)
22. Sönmez, T., Switzer, T.B.: Matching with (branch-of-choice) contracts at the united states military academy. Econometrica 81(2), 451–488 (2013)
23. Westkamp, A.: An analysis of the German university admissions system. Economic Theory 53, 561–589 (2013)

Optimal Interdiction on Cooperative Links to Prevent Attackers from Forming Coalitions*

Qingyu Guo[1], Bo An[1], Yevgeniy Vorobeychik[2], Long Tran-Thanh[3], Jiarui Gan[1], Chunyan Miao[1]

[1]Nanyang Technological University, Singapore
[2]Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN
[3]Electronics and Computer Science, University of Southampton, UK

Abstract. Game theoretic models of security, and associated computational methods, have emerged as critical components of security posture across a broad array of domains, including airport security and coast guard. These approaches consider terrorists as motivated but independent entities. There is, however, increasing evidence that attackers, be it terrorists or cyber attackers, communicate extensively and form coalitions that can dramatically increase their ability to achieve malicious goals. To date, such cooperative decision making among attackers has been ignored in the security games literature. To address the issue of cooperation among attackers, we introduce a novel *coalitional security game (CSG)* model. A CSG consists of a set of attackers connected by a (communication or trust) network who can form coalitions as connected subgraphs of this network so as to attack a collection of targets. A defender in a CSG can delete a set of edges, incurring a cost for deleting each edge, with the goal of optimally limiting the attackers' ability to form effective coalitions (in terms of successfully attacking high value targets). We first show that a CSG is, in general, hard to approximate. Nevertheless, we develop a novel branch and price algorithm, leveraging a combination of column generation, relaxation, greedy approximation, and stabilization methods to enable scalable high-quality approximations of CSG solutions on realistic problem instances.

1 Introduction

Recent decades have seen a number of major terrorist attacks, such as WTC 9/11 attack, Jemaah Islamiyahs Bali bombing, and 7/7 London bombing, that have killed thousands of lives and caused significant economic losses. An important reason for the increasing threat of terrorism is cooperation between terrorist groups [21]. For example, three terrorist groups in Africa have been reported to share funds, training, and explosive materials with each other [25], and Chechen terrorists were reported to obtain weapons from terrorist organizations in the Middle East [18]. Such sharing of skills and resources among terrorist groups is

* This paper will appear at the 15th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2016).

common because it significantly increases their capability of achieving malicious goals, such as attacking high-value targets.

An important way to prevent individual terrorist groups from forming powerful coalitions is to cut off connections between them. This can be done by blocking their bank accounts, increasing surveillance of strategic exchange points, setting sentries in arterial roads, etc. However, since it is impractical to block all possible ways attackers can communicate, a central decision problem is to choose a subset of such connections to block so as to minimize expected efficacy of formed coalitions and resulting attacks. Addressing this problem entails several challenges: i) the set of possible combinations of edges to cut is exponential in the number of connections among attackers; ii) the number of possible coalitions to account for is exponential in the number of attackers; iii) coalition stability is an important consideration in assessing which coalitions will form, and it significantly increases problem difficulty; iv) the decision space of attackers includes the choice of targets to attack, which must also be accounted for by the defender.

While there is existing research on terrorist networks, it has been limited in scope to either social network analysis of terrorist groups [12, 27, 20, 11], or using cooperative game theory as a means for identifying key members of terrorist networks [16, 17]. The related research in security games, on the other hand, tends to model attackers as independent actors (indeed, only a single attack by a single attacker or group is typically considered) [28, 10, 14]. The ability and proclivity of attackers to form coalitions is thus largely ignored within the security games research.

To address the problem of optimally inhibiting formation of attack coalitions, we introduce a formal Coalitional Security Game (CSG) model. We show that the associated problem is MAX SNP-hard for the defender, indicating no polynomial time approximation scheme unless P=NP, and the decision problem is NP-hard for the attackers. To overcome the computational challenges, we develop a sophisticated branch and price algorithm, involving a novel combination of column generation and linear programming relaxation. Since the slave problem of column generation is formulated as a bilevel mixed-integer linear program (BMILP), we further improve the performance of the algorithm by using a novel linear relaxation approximation to reformulate the slave problem as an easily solvable single level MILP with formal constant factor approximation guarantee. Furthermore, we provide an interior-point stabilization to improve convergence properties of column generation by generating an interior dual optimal solution of the master problem, and novel heuristics for generating multiple columns in each iteration to support the linear relaxation approximation and further speed up the column generation. Extensive experimental results show that our algorithms scale up to realistic-sized problems with high quality solutions.

2 Motivating Domain

Cooperation among terrorist groups is common and necessary for their survival. During the past few decades, almost half of terrorist groups have had an ally [21].

Various kinds of cooperation are found among them, including transferring funds, weapons support, training, and other sharing of critical skills and resources. For example, the Al-Taqwa banking system financed the activities of multiple terrorist organizations, including Hamas [15]. Levitt [15] pointed out that there is significant overlap and cooperation between Palestinian terrorist groups like Hamas and other groups, such as al-Qaeda in the area of terrorist financing and logistical support. In 2000, Al-Qaeda held training camps which served as open universities, educating terrorists from a wide array of local and international terrorist groups [23]. Terrorist groups also cooperate with each other to coordinate attacks. For example, the PRC is a conglomeration of members of Islamic Jihad, Hamas and the various terrorist groups, and has conducted several infamous terrorist attacks, including the roadside bombing attack in February 14, 2002 [4]. Since cooperation among terrorist groups is achieved through commu-

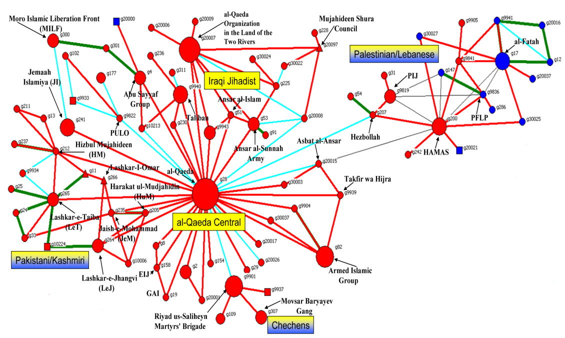


Fig. 1. Mideast Terrorist Network [1].

nication channels, such as front companies, charities for transferring funds [15], and transportation hubs including roadways, ports, and rivers for weapon supply, training, and coordinated attacks, an important measure for preventing terrorist groups from forming coalitions is to cut off connections among them. For example, FBI Assistant Director John Pistole testified to Congress that investigations into the financial activities of terrorist supporters in the United States helped prevent four different terrorist attacks abroad [22]. Moreover, the US strategy for combating terrorism asserts that *"The interconnected nature of terrorist organizations necessitates that we pursue them across the geographic spectrum to ensure that all linkages between the strong and the weak organizations are broken, leaving each of them isolated, exposed, and vulnerable to defeat"* [6].

In order to cut connections among terrorist groups, security agencies can shut down the front companies and charities, which are providing and transferring money for terrorist groups, to stem the flow of funds between terrorist groups [15]. Besides, blocking critical transportation hubs can significantly reduce efficiency of coordination attempts such as weapon support and terrorist training. Nevertheless, activities aimed at inhibiting communication among attackers, such as blocking roadways by setting sentries are costly, and it is in-feasible to cut all the connections among terrorist groups. This motivates our investigation of optimal allocation of security resources to block attacker linkages, taking into account the associated costs, as well as the nature of coalitions that would form as a result.

3 Coalitional Security Games

A *coalitional security game* (CSG) is modeled based on the *coalitional skill game* [2], and consists of T types of targets, each with a large number of copies, which is reasonable since there is a large pool of potential targets of terrorism around the world. There is a set N of terrorists (individuals or groups) who want to attack these targets. Each terrorist i has a set S_i of skills, and attacking a target of type $t \in T$ requires a set $S(t)$ of skills. Let $S = \bigcup_{i \in N} S_i$. We assume $S(t) \subseteq S$ for all $t \in T$ without loss of generality. Each type $t \in T$ has a value $p_t > 0$ for the terrorists. Typically, each terrorist i has a capacity $m_{is} \in \mathbb{N}$ for each of his skills $s \in S_i$, so that he can use s in at most m_{is} attacks. For example, a terrorist can provide weapons for a number of terrorist attacks.

The terrorists form coalitions to share skills to launch more attacks. Similar to existing work [17], we use a graph $G = (N, E)$ to represent the cooperation network of terrorists, where N is the set of terrorists and E is the set of edges representing connections between pairs of terrorists. We denote by (i, j) an edge connecting terrorists i and j. A set $C \subseteq N$ of terrorists can form into a coalition only when the *induced subgraph* $G_C = (C, E(C))$ is connected, where $E(C) = \{(i, j) \in E | i \in C, j \in C\}$. We use the set C to represent the coalition formed.

A terrorist coalition can choose multiple targets to attack simultaneously, so long as it possesses sufficient required skills and resources. For example, in 2008, a group of terrorists carried out a series of twelve coordinated shooting and bombing attacks in Mumbai [13], and in the September 11 attack, 4 airplanes were hijacked to attack several targets in 3 different cities. We model these types of threats through the definition of the value of a coalition. Formally, let $\mathbf{a} = \langle a_t \rangle$ be an attacking plan of a coalition C, where a_t is the number of targets of type t that coalition C plans to attack.

An attacking plan $\mathbf{a}$ is feasible if

$$\sum\nolimits_{t \in T,\, s \in S(t)} a_t \leq \sum\nolimits_{i \in C,\, s \in S_i} m_{is}, \ \forall s \in S, \tag{1}$$

The payoff $u(\mathbf{a})$ for an attacking plan $\mathbf{a}$ is the sum of values of all targets attacked, i.e., $u(\mathbf{a}) = \sum_t a_t p_t$. We assume that terrorists are utility-maximizers. Thus, once a coalition is formed, they choose an attack plan which maximizes payoff over all feasible plans.

$$v(C) = \max\nolimits_{\mathbf{a}\text{: satisfying Eq.(1)}} u(\mathbf{a}). \tag{2}$$

The value $v(C)$ of a coalition C is defined as the maximum achievable payoff in Eq.(2). We use a payoff vector $\mathbf{y} = \langle y_i \rangle \geq \mathbf{0}$ to denote the payoff for each terrorist i. A payoff vector represents how the value of every coalition is divided among their members, so that for a coalition structure CS (i.e., a partition of N into disjoint coalitions) we require $\sum_{i \in C} y_i \leq v(C)$ for any $C \in CS$. The pair $(CS, \mathbf{y})$ is called an *outcome* of a coalitional game.

3.1 Stable Coalition Structure

Since the terrorists are self-interested and profit driven, we assume that a coalition structure CS formed by attackers must be stable in the sense that any subset of attackers has no (or little) incentive to break off into another coalition for higher payoff. To enforce coalition structure stability, we adopt the widely-used solution concept of *ϵ-core* [7].

Definition 1 (ϵ-core) *The ϵ-core, for $\epsilon > 0$, is the set of all outcomes $(CS, \mathbf{y})$ such that for any coalition $C \subseteq N$, $\sum_{i \in C} y_i \geq v(C) - \epsilon$.*

Attackers always prefer the outcome in ϵ-core with as lower ϵ value as possible, and the minimal value ϵ^* for which the ϵ-core is non-empty is called the *least-core value* of the game, with the corresponding ϵ^*-core called the *least-core*. For a coalition structure CS, if there exists an outcome $(CS, \mathbf{y})$ in the least-core, we call CS a *stable coalition structure*. Although computing the least-core is extremely hard for general coalitional games [7], given the coalition value function v defined in Eq.(2), the coalitional game turns out to be *superadditive* (Lemma 1). In this case, the attackers are willing to form as large coalitions as they can, and the coalition structure CS^*, whose induced subgraphs are all connected components, is a stable coalition structure (Lemma 2).

Lemma 1 *Given the value function defined in Eq.(2), the terrorists' coalitional game $G = (N, v)$ is superadditive, i.e., $v(C \cup D) \geq v(C) + v(D)$ for every pair of disjoint coalitions $C, D \subseteq N$.*[1]

Lemma 2 *The coalition structure consisting of all coalitions whose induced subgraphs are connected components of graph $G(N, E)$ is a stable coalition structure, and it has the maximum total value among all coalition structures.*

3.2 Defender Strategy

The defender's goal is to optimally cut off connections within terrorists to minimize threat due to attacks by formed coalitions. We use a symmetric matrix $\mathbf{B} = \langle B_{ij} \rangle$ to represent the defender's strategy, such that $B_{ij} = 1$ if edge (i, j) is blocked and $B_{ij} = 0$ otherwise. We let $B_{ii} = 0$ for all $i \in N$. Blocking an edge (i, j) incurs a cost, λ_{ij}. When the defender adopts strategy $\mathbf{B}$, the blocked edges are removed from the network, resulting in a new network $G(\mathbf{B}) = (N, E(\mathbf{B}))$ where $E(\mathbf{B}) = E \backslash \{(i, j) \in E | B_{ij} = 1\}$. Given $\mathbf{B}$, the attackers play a coalitional skill game on the induced graph $G(\mathbf{B}) = (N, E(\mathbf{B}))$, and form a coalition structure $CS^*(\mathbf{B})$ consisting of all coalitions whose induced subgraphs are connected components of $G(\mathbf{B})$. The defender's utility is then

$$U_d(\mathbf{B}) = -\sum\nolimits_{C \in CS^*(\mathbf{B})} v(C) - \sum\nolimits_{(i,j) \in E} B_{ij} \lambda_{ij}$$

[1] Please see online Appendix A for all the proofs available at: `http://csgappendix.weebly.com/`

3.3 Complexity Analysis

We first investigate the computational complexity of finding the optimal strategies for both the defender and the attackers, and show that the defender's and the attackers' decision-making problems are MAX SNP-hard and NP-hard.

Defender's Decision-Making Problem We reduce the *k-TERMINAL CUT* problem [8], whose MAX SNP-hardness has been proved for any fixed $k \geq 3$, to the defender's decision-making problem by a *linear reduction* which preserves the approximation property (MAX SNP-hardness) [19]. The following definitions will be useful to introduce our key result (Theorem 1).

Theorem 1 *Finding an optimal defender strategy for a coalitional security game is MAX SNP-hard.*

Theorem 1 indicates that computing an optimal defender strategy does not admit a polynomial time approximation scheme unless P=NP. Indeed, what makes the defender's problem particularly challenging is the fact that the attacker's problem is hard as well, as we will demonstrate later.

Attackers' Decision-Making Problem For a given coalition C, the attackers' decision-making problem of choosing the optimal attacking plan $\mathbf{a}$ is NP-hard, as the following theorem asserts.

Theorem 2 *Computing the attackers' optimal attacking plan is NP-hard.*

Despite these rather negative results, we nevertheless undertake the task of devising a method for computing optimal CSG solutions, showing the challenges can be overcome for realistic problem instances, even if not in the worst case.

4 Solution Approach

We now turn to the computational issues in CSGs. We first propose an *integer program* (IP) to solve it exactly. Because the IP has an exponentially many variables, we propose a *branch and price* algorithm to tackle it.

4.1 Coalition Enumeration Approach

We start with the IP computing the optimal solution. Suppose $\mathcal{C}$ is the set of all coalitions for which the induced subgraphs on $G = (N, E)$ are connected, and $C_k \in \mathcal{C}$ is the k^{th} coalition in $\mathcal{C}$ with coalition value denoted by v_k. Let $\alpha_{ki} = 1$ if $i \in C_k$ and $\alpha_{ki} = 0$ otherwise. Let the binary variables $\mathbf{x}$ represent the attackers' strategy of forming a coalition structure CS, such that $x_k = 1$ iff $C_k \in CS$. If we suppose that all coalitions in $\mathcal{C}$ as well as their associated values have been precomputed, the defender's optimization problem can be formulated as the following integer program, with the objective of minimizing the defender's loss and Eqs.(3b)–(3c) restricting the stable coalition structure:

$$\min_{\mathbf{x},\mathbf{B}} \sum\nolimits_{C_k \in \mathcal{C}} v_k x_k + \sum\nolimits_{(i,j) \in E} B_{ij} \lambda_{ij} \tag{3a}$$

$$\text{s.t.} \quad \sum\nolimits_{C_k \in \mathcal{C}} \alpha_{ki} x_k = 1 \qquad i \in N \tag{3b}$$

$$\sum\nolimits_{C_k \in \mathcal{C}} \alpha_{ki} \alpha_{kj} x_k \geq 1 - B_{ij} \quad (i,j) \in E \tag{3c}$$

$$\mathbf{x} \in \{0,1\}^{|\mathcal{C}|}, \mathbf{B} \in \{0,1\}^{|N| \times |N|} \tag{3d}$$

Eq.(3b) ensures that each attacker is in exactly one coalition, so that CS denoted by $\mathbf{x}$ is a partition of N. Eq.(3c) means that once an edge $(i,j) \in E$ is not blocked by the defender, i.e., $B_{ij} = 0$, then i and j must be in the same coalition of CS. Let $\mathcal{C}_{\mathbf{B}}$ be the set of coalitions whose induced subgraphs are connected components of $G(\mathbf{B})$. According to Eqs.(3b)–(3c), for a feasible solution $\mathbf{x}$, $x_k = 1$ if and only if $C_k \in \mathcal{C}_{\mathbf{B}}$ or C_k is a superset of several coalitions in $\mathcal{C}_{\mathbf{B}}$. With the superadditive coalition's value and the minimizing objective, the solution $\mathbf{x}$ with $x_k = 1$ for $C_k \in \mathcal{C}_{\mathbf{B}}$ is always optimal for a fixed defender strategy $\mathbf{B}$, corresponding with Lemma 2.
Although IP (3) can obtain the optimal defender strategy $\mathbf{B}^*$, it has exponentially many ($|\mathcal{C}|+|E|$) variables, which makes it impossible to scale up to large game instances with standard *branch and cut* method adopted by popular commercial solvers, such as CPLEX. Therefore, we propose a *branch and price* framework, which combines *branch and bound* and *column generation*.

4.2 Branch and Price Framework

Before we introduce our branch and price framework, the following lemma shows that the exponentially large number of binary variables $\mathbf{x}$ in IP (3) can be relaxed without sacrificing optimality (note that $\mathbf{B}$ remains to be binary). This observation will be useful to significantly reduce the size of the branch and bound tree.

Lemma 3 *The formulation* (3) *is equivalent to its relaxed formulation where* $\mathbf{x}$ *is continuous.*

Figure 2 shows the flow of the branch and price framework. The left part of the figure shows the classic branch and bound tree of solving IP (3) (with $\mathbf{x}$ being relaxed), where the root node in the tree corresponds to the IP (3) and it keeps an *upper bound* (**UB**) on its optimal objective, which can be the objective value of any feasible solution of IP (3). For each created node (an integer program), a *lower bound* (**LB**) is obtained by further relaxing variable $\mathbf{B}$. We refer to the fully relaxed formulation LP relaxation, and denote by $\tilde{\mathbf{B}}^*$ its optimal solution. Two basic operators in branch and bound are *pruning* and *branching*. A node is pruned once its **LB** is not less than **UB** or $\tilde{\mathbf{B}}^*$ is integral. Otherwise, the branching operator will be conducted on a fractional variable B_{ij} in $\tilde{\mathbf{B}}^*$, and two child nodes will be created, one by fixing B_{ij} to 0 and the other with $B_{ij} = 1$. Once a node happens to obtain an **LB** with $\tilde{\mathbf{B}}^*$ being integral, the **UB** of root node will be updated as $\mathbf{UB} = \min\{\mathbf{LB}, \mathbf{UB}\}$. The method terminates when all leaf nodes are pruned, and the final **UB** of root node will be equal to the optimum of IP (3).

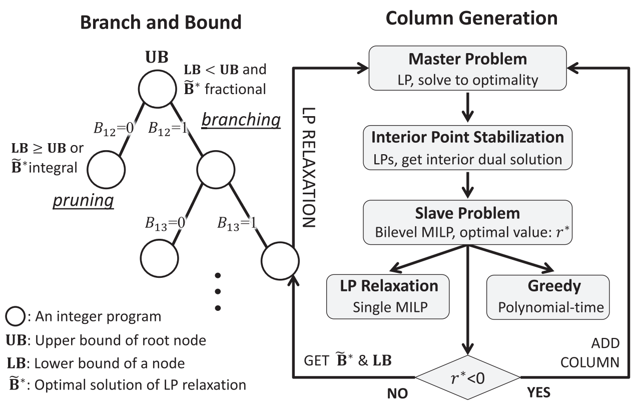


Fig. 2. Branch and Price Framework.

Since the LP relaxation of each node in the branch and bound tree has an exponential number of variables, we use column generation to iteratively compute $\mathbf{LB}$ and $\tilde{\mathbf{B}}^*$ as illustrated by the right part of the Figure 2. Column generation begins by a master LP with a small subset of variables, and solves the slave problem to add new column(s) or variable(s) with negative reduced cost(s) to the master LP, then resolves the master LP, repeating until no such column(s) exists. The column generation approach terminates with the optimal relaxed solution $\tilde{\mathbf{B}}^*$.

The central challenge and novelty of any column generation approach is how to efficiently compute which columns to add, and to guarantee that when this computation adds no new columns, the solution is optimal. The issues involved in adapting a column generation method to our setting are sufficiently non-trivial as to warrant a separate section.

5 Column Generation

The LP relaxation at each node in Figure 2 is decomposed into *master* and *slave* problems for column generation. The former solves for the relaxed defender strategy $\tilde{\mathbf{B}}$, given a restricted set of coalitions $\mathcal{C}' \subset \mathcal{C}$. The objective for the slave is updated based on the solution of the master, and the slave is solved to identify the best new coalition to be added to the $\mathcal{C}'$ of the master problem, as measured by *reduced cost* (explained later). If no new column can improve the solution the algorithm terminates with an optimal solution.

5.1 Master Problem

The master problem (4) starts with a small set of coalitions $\mathcal{C}'$ and solves for the optimal relaxed defender strategy $\tilde{\mathbf{B}}^*$. Note that Eqs.(4a)-(4c) have ensured $\mathbf{x} \leq \mathbf{1}$ and $\tilde{\mathbf{B}} \leq \mathbf{1}$.

$$\min_{\mathbf{x},\tilde{\mathbf{B}}} \ \sum\nolimits_{C_k \in \mathcal{C}'} v_k x_k + \sum\nolimits_{(i,j) \in E} \tilde{B}_{ij} \lambda_{ij} \tag{4a}$$

$$\text{s.t.} \quad \sum\nolimits_{C_k \in \mathcal{C}'} \alpha_{ki} x_k = 1 \qquad i \in N \tag{4b}$$

$$\sum\nolimits_{C_k \in \mathcal{C}'} \alpha_{ki} \alpha_{kj} x_k \geq 1 - \tilde{B}_{ij} \quad (i,j) \in E \tag{4c}$$

$$\mathbf{x} \geq \mathbf{0}, \tilde{\mathbf{B}} \geq \mathbf{0} \tag{4d}$$

Let $\mathbf{f}$ and $\mathbf{g}$ be dual variables of master constraints (4b) and (4c) respectively. After the master problem is solved, the optimal dual solution $(\mathbf{f}^*, \mathbf{g}^*)$ is computed and passed to the slave problem for finding the column (coalition) to add to the current $\mathcal{C}'$, as we will discuss later.

Interior Point Stabilization. Before the slave problem is introduced, it is necessary to understand that the optimal dual solution $(\mathbf{f}^*, \mathbf{g}^*)$ plays a critical role in generating a good column. Since the standard technique computes an optimal dual solution, which is an extreme point of the optimal dual polyhedron and can be characterized by very large values for some dual variables, an *Interior Point Stabilization* (IPS) method [24] is adopted to compute an optimal dual solution taking values in the center (or at least the interior) of the master problem's optimal dual polyhedron by averaging several extreme points of this polyhedron. We provide the details of IPS in online Appendix B[1].

5.2 Slave Problem

The slave problem finds the best column (i.e., coalition) to add to the current coalitions in $\mathcal{C}'$. This is done using *reduced cost*, which captures the total change in the defender's utility if a candidate coalition is added to $\mathcal{C}'$. The candidate coalition with minimum reduced cost improves the defender's utility most [5]. The reduced cost r_k of coalition C_k, associated with variable x_k, is given in Eq.(5), where the dual solution $(\mathbf{f}^*, \mathbf{g}^*)$ measures the influence of the associated constraint on the objective and can be calculated using standard techniques or the IPS method.

$$r_k = v_k - \sum\nolimits_{i \in N} \alpha_{ki} f_i^* - \sum\nolimits_{(i,j) \in E} \alpha_{ki} \alpha_{kj} g_{ij}^* \tag{5}$$

The slave problem then boils down to solving $\min_k r_k$ (i.e., minimizing reduced cost). Clearly, if we were to simply iterate through all coalitions k (of which there are an exponential number), nothing would be gained by column generation. We then exploit the structure of this problem by first formulating the reduced cost minimization problem as a *bilevel mixed-integer linear program* (B-MILP) (6), where $(\mathbf{f}^*, \mathbf{g}^*)$ is the optimal dual solution of the master problem (4). What makes this problem bilevel is the fact that, in minimizing reduced cost, we must also compute the value of an associated coalition, since we have avoided precomputing all coalitions' values with branch and price.

In the BMILP (6), the coalition C is represented by binary variable $\boldsymbol{\alpha}$ such that $\alpha_i = 1$ if $i \in C$ and $\alpha_i = 0$ otherwise. The upper level constraints ensure that the coalition C induced a connected subgraph, i.e., $C \in \mathcal{C}$, while the lower level program computes the coalition value $v(C)$.

Eq.(6b) ensures that $\xi_{ij} = \alpha_i \alpha_j$ for edge $(i, j) \in E$. Eqs.(6c)–(6f) enforce $\boldsymbol{\alpha}$ to induce a connected subgraph of G using flow balance techniques [26]: According to constraints (6c) and (6d), $w_i = 1$ if and only if i is the smallest index such that $\alpha_i = 1$, which will be set as the starting point of the flows $\mathbf{h}^+$ and $\mathbf{h}^-$; h^+_{lij} denotes the amount of flow from starting point to ending point l, such that passes through edge (i, j) in positive direction $i \to j$, while h^-_{lij} in negative direction $j \to i$; Eq.(6f) restricts the flow to pass through only edges that are in the subgraph induced by C, i.e., (i, j) with $\xi_{ij} = 1$; Eq.(6e) is the flow balance constraint with $\mathcal{N}(i)$ denoting the set of vertices connected with i, making sure there exists a path, in subgraph induced by C, from starting point $i : w_i = 1$ to any $l \in C$ $(\alpha_l = 1)$. Therefore the subgraph induced by C is connected.

The lower level program (6g) computes the coalition's value $v(C)$ where Eq.(6h) restricts C to have enough skill capacities to conduct the attacking plan $\mathbf{a}$ according to Eq.(1).

$$
\begin{aligned}
\min_{\boldsymbol{\alpha},\, \boldsymbol{\xi},\, \mathbf{w},\, \mathbf{h}^+,\, \mathbf{h}^-} \quad & \sum\nolimits_{t\in T} p_t a_t - \mathbf{f}^{*\mathbf{T}}\boldsymbol{\alpha} - \mathbf{g}^{*\mathbf{T}}\boldsymbol{\xi} && \text{(6a)}\\
\text{s.t.} \quad & \xi_{ij} \le \alpha_i, \xi_{ij} \le \alpha_j, && \\
& \xi_{ij} \ge \alpha_i + \alpha_j - 1 \quad \forall (i,j) \in E && \text{(6b)}\\
& w_i \le \alpha_i, && \\
& w_i \le 1 - \alpha_j \quad \forall i, j \in N : j < i && \text{(6c)}\\
& \sum\nolimits_{i\in N} w_i = 1 && \text{(6d)}\\
& \sum\nolimits_{j\in\mathcal{N}(i)} h^+_{lij} - \sum\nolimits_{j\in\mathcal{N}(i)} h^-_{lij} \ge && \\
& w_i - (1 - \alpha_l), \quad \forall i, l \in N : i \ne l && \text{(6e)}\\
& h^+_{lij} \le \xi_{ij}, && \\
& h^-_{lij} \le \xi_{ij}, \quad \forall (i,j) \in E, l \in N && \text{(6f)}\\
\max_{\mathbf{a}} \quad & \sum\nolimits_{t\in T} p_t a_t && \text{(6g)}\\
\text{s.t.} \quad & \sum\nolimits_{t\in T} \beta_{ts} a_t \le \sum\nolimits_{i\in N} \alpha_i \gamma_{is} m_{is} \quad s \in S && \text{(6h)}\\
& \boldsymbol{\alpha} \in \{0,1\}^{|N|}, \boldsymbol{\xi} \in [0,1]^{|E|}, \mathbf{a} \in \mathbb{N}^{|T|} && \\
& \mathbf{h}^+ \ge \mathbf{0}, \mathbf{h}^- \ge \mathbf{0}, \mathbf{w} \ge \mathbf{0}. && \text{(6i)}
\end{aligned}
$$

The key challenge of the BMILP is that the inner maximization program described in (6g) has integer variables. If we were to relax the integrality constraint, we could reformulate the bilevel program into a single mixed integer linear program, as we describe next.

Linear Relaxation Approximation. To reformulate the BMILP to an MILP, we first relax the integer program (6g) of computing the coalition's value in the lower level, such that the decision variable $\mathbf{a} \in \mathbb{N}^{|T|}$ is relaxed to $\tilde{\mathbf{a}} \ge \mathbf{0}$. For this relaxed linear program (RLP), let $\mathbf{u}$ be the dual variable associated with constraint (6h). A pair of feasible solutions $\tilde{\mathbf{a}}$ and $\mathbf{u}$ are optimal for the primal and dual RLPs if and only if the following *complementary slackness conditions* are satisfied [5]:

$$(\sum_{s\in S}\beta_{ts}u_s - p_t)\cdot \tilde{a}_t = 0 \quad \forall t\in T \tag{7a}$$

$$(\sum_{i\in N}\alpha_i\gamma_{is}m_{is} - \sum_{t\in T}\beta_{ts}\tilde{a}_t)\cdot u_s = 0 \quad \forall s\in S \tag{7b}$$

Therefore, the RLP is equivalent with a set of constraints consisting of the primal and dual constraints restricting the feasibility of $\tilde{\mathbf{a}}$ and $\mathbf{u}$ and the complementary slackness conditions ensuring optimality, and the relaxed BMILP is reformulated as an MILP (8), with Eq.(8c) corresponding to the dual constraint of the RLP, and Eqs.(8d)–(8e) equivalent with the complementary slackness conditions (7).

$$\min_{\substack{\boldsymbol{\alpha},\boldsymbol{\xi},\mathbf{w},\mathbf{u},\boldsymbol{\phi}\\ \mathbf{h}^+,\mathbf{h}^-,\tilde{\mathbf{a}},\boldsymbol{\varphi}}} \quad \sum_{t\in T}p_t\tilde{a}_t - \mathbf{f}^{*\mathbf{T}}\boldsymbol{\alpha} - \mathbf{g}^{*\mathbf{T}}\boldsymbol{\xi} \tag{8a}$$

$$\text{s.t.}\quad \text{Eqs.(6b)–(6f), (6h)–(6i)} \tag{8b}$$

$$\sum_{s\in S}\beta_{ts}u_s \geq p_t, \quad \forall t\in T \tag{8c}$$

$$\sum_{i\in N}\alpha_i\gamma_{is}m_{is} - \sum_{t\in T}\beta_{ts}\tilde{a}_t \leq M(1-\phi_s),$$
$$u_s \leq M\phi_s \quad \forall s\in S \tag{8d}$$

$$\sum_{s\in S}\beta_{ts}u_s - p_t \leq M(1-\varphi_t),$$
$$\tilde{a}_t \leq M\varphi_t \quad \forall t\in T \tag{8e}$$

$$\tilde{\mathbf{a}}\geq \mathbf{0}, \mathbf{u}\geq \mathbf{0}$$
$$\boldsymbol{\phi}\in\{0,1\}^{|S|}, \boldsymbol{\varphi}\in\{0,1\}^{|T|}. \tag{8f}$$

Our next results show that the solution quality of MILP (8) and the resulting branch and price framework achieves a competitive ratio only depending on $|S|$.

Lemma 4 *The coalition value $\tilde{v}(C)$ computed by LP relaxation of IP* (6g) *is bounded by $\tilde{v}(C)\leq(1+|S|)v(C)$.*

Theorem 3 *The branch and price approach, in which the slave problem of column generation is solved by the linear relaxation approximation, can obtain a defender strategy $\mathbf{B}'$ such that $U_d(\mathbf{B}')\geq(1+|S|)U_d(\mathbf{B}^*)$. Note that $U_d(\mathbf{B})\leq 0$.*

Greedy Approximation. Although the MILP of linear relaxation approximation can obtain an approximately-optimal coalition to add to $\mathcal{C}'$ of master problem, it is still relatively slow. Observe, however, that in each iteration of column generation we actually need not find a minimal reduced cost; rather, any reduced cost that improves solution quality would suffice. Consequently, a fast heuristic approach for generating columns would be satisfactory in most iterations, except when the heuristic is unable to find a solution improving column, at which point we can fall back on the MILP. To this end, we propose a Greedy with Multi-Start (GMS) heuristic (see Algorithm 1). This heuristic is fast, and has an important advantage of enabling generation of multiple coalitions in a single iteration of the column generation algorithm (hence, the name multi-start), significantly reducing the number of column generation iterations.

Algorithm 1: Greedy with Multi-Start (GMS)

Input: optimal dual solution $(\mathbf{f}^*, \mathbf{g}^*)$ of master problem, reduced cost function $r(C)$ defined in Eq.(5)

Output: a set of coalitions with negative reduced costs

$\mathcal{C}_{GMS} = \emptyset$;
for $i \in N$ **do**
    $C = \{i\}$, $continue = true$;
    while $continue$ **do**
        $\hat{i} = \arg\min_{i \in N \setminus C} r(C \cup \{i\}) - r(C)$;
        if $r(C \cup \{\hat{i}\}) - r(C) < 0$ **then** $C \leftarrow C \cup \{\hat{i}\}$;
        else $continue = false$;
    if $r(C) < 0$ **then** $\mathcal{C}_{GMS} \leftarrow \mathcal{C}_{GMS} \cup \{C\}$;
return $\mathcal{C}_{GMS}$;

6 Experimental Evaluation

We demonstrate the effectiveness of our algorithmic framework through extensive numerical evaluations. We use the CPLEX (version 12.6) to solve all linear programs. All computations were performed on a 64-bit PC with 16 GB RAM and a quad-core 3.4 GHz processor. All values are averaged over 40 instances unless otherwise specified. All scale-free graphs are generated by standard Barabási-Albert model [3] widely used for simulating networks with realistic topological properties, denoted by $BA(d)$, in which d represents the average node degree. All random graphs are generated by standard Erdős-Rényi model denoted by $ER(p)$ where p represents the probability of existence of an edge between any pair of nodes [9]. According to the different approaches adopted for the slave problem, the abbreviations of different branch and price algorithms are as follows: LR for *Linear Relaxation Approximation*, GLR for the combination of the GMS with LR such that LR is called only when GMS returns empty set; The optimal dual solution of the master problem is generated through *Interior Point Stabilization* (IPS) for IGMS, ILR, and IGLR. IP (3) is represented by EXACT. Our benchmark is a heuristic algorithm *Genetic Algorithm* (GA), and for the ease of reading, we put the details of GA in online Appendix C[1]. By default, the instances are parameterized as follows: $|T| = 10$ and $|S| = 20$, the necessary skills for each target type and the ones owned by each attacker drawn randomly from S, $m_{is} \sim \{1, 2, 3, 4, 5\}$; $p_t \sim [0, 1.0]$ and $\lambda_{ij} \sim [0, r]$ where $r \sim [0, 1]$.

6.1 Scalability and Optimality

Runtime. We test different algorithms on 5 types of networks, and the results are depicted in Figures 3(a)–3(e), from which we can observe the significant efficiency improvement provided by the branch and price framework. The IPS procedure and the GMS approach are shown to reduce the number of iterations of column generation drastically as they further improve the branch and price

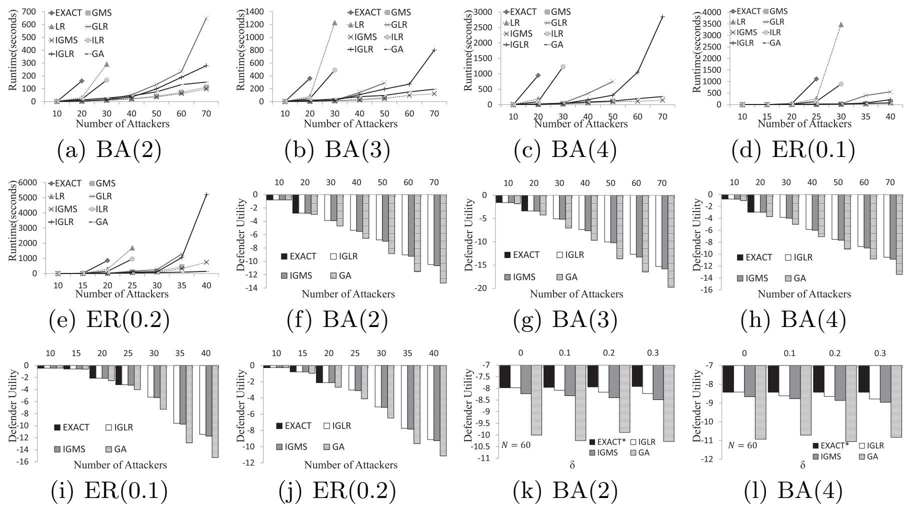


Fig. 3. Scalability: (a)–(e); Optimality: (f)–(j); Robustness: (k)–(l).

framework to scale up to realistic-sized problems. For example, Figure 1 contains 76 groups with mean degree 3.4, which is well within the scalability of IGLR.

Defender Utility. We compare the defender utility of IGLR and IGMS, with EXACT and GA as benchmarks. The results are illustrated in Figures 3(f)–3(j), from which we can see that IGLR can achieve an almost optimal solution which outperforms GA significantly, in accordance with the theoretical analysis of Theorem 3. The solutions obtained by IGMS, which also outperform GA, are near-optimal especially for networks with higher connectivity, such as large-scale networks of type ER(0.1) and ER(0.2). A tradeoff between runtime and solution quality is observed for IGMS and IGLR, as IGMS shows the better scalability while IGLR obtains the solution with higher quality, however, both of them outperform the alternatives significantly.

6.2 Robustness

In the real world, the defender's estimation of target value p_t may not be perfect from attackers' perspective. Therefore, we analyze the performance of our algorithmic framework under the existence of noise of p_t. Let $\bar{p}_t$ be the defender's estimation of p_t while p_t is drawn uniformly within $\bar{p}_t \cdot [1-\delta, 1+\delta]$. We compare the defender utility of IGLR and IGMS under different degrees of uncertainty, with the near-optimal solution of IGLR, denoted by EXACT* (IP (3) cannot scale to $N = 60$), and heuristic solution of GA as comparison. The results are shown in Figures 3(k)–3(l), from which a decreasing efficiency is observed for IGLR and IGMS when δ increases. However, IGLR and IGMS solutions still outperform GA significantly under the existence of uncertainty of p_t, and IGLR can obtain an almost optimal solution even when $\delta = 0.3$, which shows the strong robustness of our algorithmic framework.

7 Conclusion

For the first time, this paper studies the problem of blocking attacker coalition through efficient allocation of security resources. This paper provides the following key contributions: 1) We formally define and model coalitional security games. 2) We prove the MAX SNP-hardness of defender's decision-making problem and NP-hardness of attackers' decision-making problem. 3) To address the MAX SNP-hardness, we propose an exact integer program and provide a branch and price algorithm, a linear relaxation based column generation with a constant factor approximation bound, an interior point stabilization procedure, and a greedy method to further improve the scalability. 4) Experiments demonstrate that our methods can scale up to realistic-sized instances and achieve near-optimal performance.

Acknowledgments

This research is supported by the NSF (CNS-1238959), ONR (N00014-15-1-2621), AFRL (FA8750-14-2-0180) and the National Research Foundation, Prime Minister's Office, Singapore under its IDM Futures Funding Initiative and administered by the Interactive and Digital Media Programme Office.

References

1. V. Asal and R. K. Rethemeyer. The nature of the beast: Terrorist organizational characteristics and organizational lethality. *Journal of Politics*, 70(2):437–449, 2008.
2. Y. Bachrach and J. S. Rosenschein. Coalitional skill games. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1023–1030, 2008.
3. A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.
4. J. Bennett. Israeli killed as his commandos demolish west bank house. *The New York Times*, 2002.
5. D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.
6. G. W. Bush. *National strategy for combating terrorism*. Wordclay, 2009.
7. G. Chalkiadakis, E. Elkind, and M. Wooldridge. Computational aspects of cooperative game theory. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(6):1–168, 2011.
8. E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *The SIAM Journal on Computing*, 23(4):864–894, 1994.
9. P. Erdős and A. Rényi. On random graphs I. *Publ. Math. Debrecen*, 6:290–297, 1959.
10. M. Jain, B. An, and M. Tambe. An overview of recent application trends at the aamas conference: Security, sustainability and safety. *AI Magazine*, 33(3):14, 2012.

11. P. Klerks. The network paradigm applied to criminal organizations: Theoretical nitpicking or a relevant doctrine for investigators? Recent developments in the netherlands. *Connections*, 24(3):53–65, 2001.
12. V. Krebs. Uncloaking terrorist networks. *First Monday*, 7(4), 2002.
13. K. A. Kronstadt. Terrorist attacks in Mumbai, India, and implications for US interests. *Congressional Research Service*, 2008.
14. J. Letchford and V. Conitzer. Solving security games on graphs via marginal probabilities. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI)*, pages 591–597, 2013.
15. M. Levitt. Untangling the terror web: Identifying and counteracting the phenomenon of crossover between terrorist groups. *SAIS Review*, 24(1):33–48, 2004.
16. R. Lindelauf, H. Hamers, and B. Husslage. Cooperative game theoretic centrality analysis of terrorist networks: The cases of Jemaah Islamiyah and Al Qaeda. *European Journal of Operational Research*, 229(1):230–238, 2013.
17. T. P. Michalak, T. Rahwan, N. R. Jennings, P. L. Szczepański, O. Skibski, R. Narayanam, and M. J. Wooldridge. Computational analysis of connectivity games with applications to the investigation of terrorist networks. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 293–301, 2013.
18. A. Nekrassov. Chechen attack: 'terrorists get weapons from abroad, linked to mideast groups'. *Russia Today*, 2014.
19. C. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (ACM STOC)*, pages 229–234, 1988.
20. M. B. Peterson. *Applications in Criminal Analysis: A Sourcebook*. Greenwood Press Westport, 1994.
21. B. J. Phillips. *How Terrorist Organizations Survive: Cooperation and Competition in Terrorist Group Networks*. PhD thesis, University of Pittsburgh, 2012.
22. J. S. Pistole. Identifying, Tracking and Dismantling the Financial Structure of Terrorist Organizations. 2003.
23. C. L. Powell. Remarks to the United Nations Security Council, February 5, 2003.
24. L. Rousseau, M. Gendreau, and D. Feillet. Interior point stabilization for column generation. *Operations Research Letters*, 35(5):660–668, 2007.
25. T. Shanker and E. Schmitt. Three terrorist groups in Africa pose threat to US, American commander says. *The New York Times*, 2011.
26. S. Shen, J. C. Smith, and R. Goli. Exact interdiction models and algorithms for disconnecting networks via node deletions. *Discrete Optimization*, 9(3):172–188, 2012.
27. M. K. Sparrow. The application of network analysis to criminal intelligence: An assessment of the prospects. *Social networks*, 13(3):251–274, 1991.
28. M. Tambe. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, 2011.

On the complexity of solutions for cooperative games on cycle-complete graphs

Ayumi Igarashi

Department of Computer Science, University of Oxford, Oxford, U.K.
ayumi.igarashi@cs.ox.ac.uk

Abstract. We consider a cooperative transferable utility game where cooperation among the players is restricted by a graph structure: a subset of players can form a coalition if and only if they are connected in the given graph. The major issue in this line of research is the computational difficulty in proposed solutions. In this paper, we introduce the relaxed notion of supermodularity, called quasi-supermodularity, and show that various computational problems are tractable for the class of quasi-supermodular games if the given graph is cycle-complete. We also prove that without supermodularity, these problems become hard even if the underlying graph is a tree.

Keywords: Cooperative games, complexity, coalition structure, cycle-complete

1 Introduction

The standard framework of transferable utility games expects that any subsets of players can form a coalition. In many situations, however, we often encounter restrictions on cooperation: for instance, a group of political parties may not be able to cooperate without any help from intermediate parties. For the purpose of modelling such situations, Myerson [17] proposed a transferable utility game restricted by an undirected graph, or simply a graph game, where nodes correspond to players and edges represent communication links between them. Under Myerson's proposal, a subset of players can form a coalition if and only if they are connected in the underlying graph structure.

The complexity of such graph-restricted games may vary depending on what collaboration structure players have. Typically, tree-like structure appears to be an attractive restriction that can decrease complexity; indeed, in the seminal paper by Demange [4], an efficient procedure to obtain a specific core element was presented for supperadditive games on trees. Several authors [3, 7] have also observed that many solutions for graph games can be computed efficiently if the number of connected coalitions in the underlying graph is bounded by a polynomial in the number of players, as is the case for paths and cycles. Elkind [7] unified these approaches, by giving a characterization of graph families with polynomially bounded number of connected coalitions as well as by showing that various solutions can be computed in time polynomial in the number of connected coalitions.

In this paper, we consider computational complexity of games with tree-like restrictions; specifically, we focus on *cycle-complete*[1] graphs, which often represent networks having distinct communicty structure. We focus on several well-known solution concepts such as the core, the least core, the nucleolus, and the kernel, and investigate which restriction on the characteristic function is a necessary and sufficient condition for a game on a cycle-complete graph to have an efficient algorithm to compute these solutions.

In the standard settings, supermodularity of a characteristic function allows us to calculate various solutions in polynomial time. Notable results include the greedy algorithm to find an element of the core [6] and the polynomial-time algorithm for the nucleolus [8]. One of the issues that arise when considering such games under graph-restricted settings is that the standard definition of supermodular games is no longer applicable, since the set of connected coalitions is not necessarily closed under union and intersection. We hence introduce a relaxed notion of supermodularity, called *quasi-supermodularity*, by imposing supermodularity only on subsets of the family that are closed under union and intersection. In the existing literature on combinatorial optimization, it is known that if a characteristic function satisfies a relaxed form of supermodularity and the graph is cycle-complete, the so-called restricted game also has supermodularity [10]. Utilizing this property, we derive polynomial-time solvability of computational problems related to the core, the least core, the nucleolus and the kernel for quasi-supermodular games on cycle-complete graphs. Furthermore, we prove that the hereditary property of supermodularity is unlikely to hold unless the graph is cycle-complete.

It turns out that quasi-supermodularity is necessary for games on cycle-complete graphs; we prove that without quasi-supermodularity, many complexity questions become intractable even if the underlying graph is a tree. Indeed, a similar result concerning the core has been obtained by Chalkiadakis et al. [3], who showed that many core-related questions are hard for games on trees or graphs having bounded tree-width. Specifically, we show that the membership and search problems related to the least core, the nucleolus, or the kernel are co-NP-hard even if the characteristic function is cohesive, and the underlying graph is a tree. We also prove that these problems become Δ_2^p-hard if we allow arbitrary values for characteristic functions (Δ_2^p is the class of all decision problems solvable in polynomial time by using an **NP** oracle). A summary of our computational complexity results is presented in Table 1.

Related work The most closely related to ours is perhaps the work by Chalkiadakis et al [3]. However, there are a number of differences between this work and our work: first, in their work, the membership and search questions for other solution concepts such as the least core, the nucleolus, and the kernel are not addressed; second, they explore games constrained by graphs with bounded tree-width, whereas we focus on a different class of tree-like graphs; and finally, they do not explore the input of supermodularity.

[1] The term "cycle-complete" is used in [20]; such graphs are also called *block graphs* [15].

	Complete (unrestricted cases)		Cycle-complete	Trees	
	supermodular	general	q-supermodular	cohesive	general
In-Core	P ([19])	co-NP-c ([5])	P (Th. 6)	co-NP-c ([3])	co-NP-c ([3])
Core-Existence	$O(1)$ ([6])	co-NP-c ([5])	$O(1)$ ([2])	$O(1)$ (Lem. 5)	co-NP-c (Th. 10)
Core-Find	P ([6])	NP-h ([5])	P (Th. 5)	?	NP-h (Cor. 3)
In-LeastCore	P ([8]+[19])	Δ_2^p-c ([12])	P (Th. 9)	co-NP-h (Th. 11)	Δ_2^p-c (Th. 12)
LeastCore-Find	P ([8])	NP-h ([9])	P (Th. 7)	NP-h (Cor. 4)	Δ_2^p-h (Cor. 5)
In-Nucleolus	P ([8])	Δ_2^p-c ([13])	P (Th. 9)	co-NP-h (Th. 11)	Δ_2^p-c (Th. 12)
Nucleolus-Find	P ([8])	NP-h ([9])	P (Th. 7)	NP-h (Cor. 4)	Δ_2^p-h (Cor. 5)
In-Kernel	P ([19])	Δ_2^p-c ([12])	P (Th. 8)	co-NP-h (Th. 11)	Δ_2^p-c (Th. 12)
Kernel-Find	P ([8])	?	P (Th. 7)	NP-h (Cor. 4)	Δ_2^p-h (Cor. 5)

Table 1. Computational complexity for transferable utility games. The hardness results for search problems hold with respect to Turing reductions. Note that the existence questions for the least core, the nucleolus and the kernel are trivial, since such solutions are nonempty if and only if the imputation set is nonempty.

2 Preliminaries

We start by introducing basic notations and definitions of graph theory and set systems.

Graphs An *undirected graph*, or simply a *graph*, is a pair (N, E), where N is a finite set of *nodes* and $E \subseteq \{\{a,b\} \mid a,b \in N, a \neq b\}$ is a collection of *edges* between nodes. Given a set of nodes S, the *subgraph of* (N,E) *induced by* S is the graph (S, E_S), where $E_S = \{\{a,b\} \in E \mid a,b \in S\}$. For a graph (N,E), a sequence of distinct nodes $(a_1, a_2, \ldots, a_k)$, $k \geq 2$, is called a *path* in E if $\{a_h, a_{h+1}\} \in E$ for $h = 1, 2, \ldots, k-1$. A path $(a_1, a_2, \ldots, a_k)$, $k \geq 3$, is said to be a *cycle* in E if $\{a_k, a_1\} \in E$. A graph (N,E) is said to be a *forest* if it contains no cycles. A subset $S \subseteq N$ is said to be *connected in* (N,E) if for every pair of distinct nodes $a, b \in S$ there is a path between a and b in E_S. The collection of all connected subsets of N in (N,E) is denoted by $\mathcal{F}_E$. By convention, we assume that $\emptyset \in \mathcal{F}_E$. A forest (N,E) is said to be a *tree* if N is connected in (N,E). A tree (N,E) is called a *star* if there exists a central node $c \in N$ such that $E = \{\{c,a\} \mid a \in N \setminus \{c\}\}$. A subset $S \subseteq N$ of a graph (N,E) is said to be a *clique* if for every pair of distinct nodes $a,b \in S$ we have $\{a,b\} \in E$. An undirected graph (N,E) is said to be *cycle-complete* if every cycle forms a clique. Note that forests are clearly cycle-complete.

Subfamilies and set functions Let N be a finite set and $\mathcal{F}$ be a family of subsets of N. We write $\mathcal{F}(a) = \{S \in \mathcal{F} \mid a \in S\}$ for $a \in N$, and $\mathcal{F}(a \setminus b) = \mathcal{F}(a) \setminus \mathcal{F}(b)$ for $a, b \in N$. We say that $\mathcal{F}$ is an *intersecting family* if for all $S, T \in \mathcal{F}$ with $S \cap T \neq \emptyset$, $S \cup T \in \mathcal{F}$ and $S \cap T \in \mathcal{F}$. The class of intersecting families include the set of connected subsets of a cycle-complete graph.

Lemma 1 ([15]). *Given a graph $G = (N,E)$. The graph G is cycle-complete if and only if $\mathcal{F}_E$ is an intersecting family.*

A family $\mathcal{F}$ is called a *distributive lattice* if it is closed under union and intersection, i.e., for all $S, T \in \mathcal{F}$, $S \cup T \in \mathcal{F}$ and $S \cap T \in \mathcal{F}$. It is known that a distributive lattice $\mathcal{F}$ can be fully characterized by the smallest set M and the largest set L in $\mathcal{F}$, together with *a preorder*[2] $\preceq$ on N defined by

$$a \preceq b \Longleftrightarrow \mathcal{F}(b) \subseteq \mathcal{F}(a). \tag{2.1}$$

Given such representations, a subset S of N belongs to a distributive lattice $\mathcal{F}$ if and only if $M \subseteq S \subseteq L$ and S is a lower ideal in $\preceq$, i.e., S satisfies the condition that $a \in S$ whenever there exists $b \in S$ such that $a \preceq b$.

Throughout this paper, we only consider set functions $f : \mathcal{F} \to \mathbb{R}$ where $f(\emptyset) = 0$ whenever $\emptyset \in \mathcal{F}$. A set function f on a distributive lattice $\mathcal{F}$ is said to be *supermodular* if for all $S, T \in \mathcal{F}$,

$$f(S) + f(T) \leq f(S \cup T) + f(S \cap T). \tag{2.2}$$

A set function f on an intersecting family $\mathcal{F}$ is said to be *intersecting supermodular* if for all $S, T \in \mathcal{F}$ with $S \cap T \neq \emptyset$, the supermodular inequality (2.2) holds. We say that $f : \mathcal{F} \to \mathbb{R}$ is *superadditive* if for all $S, T \in \mathcal{F}$ such that $S \cap T = \emptyset$ and $S \cup T \in \mathcal{F}$, it holds that $f(S) + f(T) \leq f(S \cup T)$. For a nonempty subset $S \subseteq N$, a partition $\{X_i\}_{i \in I}$ of S is said to be an $\mathcal{F}$*-partition* if $X_i \in \mathcal{F}$ for all $i \in I$. We define $\hat{\mathcal{F}}$ as the collection of disjoint unions of sets in $\mathcal{F}$, namely,

$$\hat{\mathcal{F}} = \{\, S \subseteq N \mid S \neq \emptyset, \text{there exists an } \mathcal{F}\text{-partition of } S \,\} \cup \{\emptyset\},$$

Notice that $\hat{\mathcal{F}} = 2^N$ whenever $\{a\} \in \mathcal{F}$ for all $a \in N$. Also, it is not difficult to show that if $\mathcal{F}$ is an intersecting family, then $\hat{\mathcal{F}}$ is a distributive lattice. For a set function $f : \mathcal{F} \to \mathbb{R}$, the *Dilworth truncation*, or simply the *truncation* of f is the function $\hat{f} : \hat{\mathcal{F}} \to \mathbb{R}$ given by

$$\hat{f}(S) := \max\{ \sum_{i \in I} f(X_i) \mid \{X_i\}_{i \in I} \text{ is an } \mathcal{F}\text{-partition of } S \,\} \tag{2.3}$$

for any nonempty $S \in \hat{\mathcal{F}}$, and $\hat{f}(\emptyset) = 0$. For a rational-valued set function f on a family $\mathcal{F} \subseteq 2^N$, given by an oracle returning $f(S)$ for each $S \in \mathcal{F}$, we define $\langle f \rangle$ as an upper bound on the encoding lengths of outputs of f.

3 TU-games with a graph structure

Definition 1. *A cooperative transferable utility game with graph structure, or simply a graph game, is a triple (N, v, E) where N is a finite set of players, $v : \mathcal{F}_E \to \mathbb{R}$ is a characteristic function, and E is the set of communication edges between players.*

The subsets S of N are referred to as *coalitions*. A coalition $S \subseteq N$ is said to be *feasible* if $S \in \mathcal{F}_E$. Throughout the paper, we assume that the grand coalition N is connected, that is, $N \in \mathcal{F}_E$, and that $v(\emptyset) = 0$. For a vector $\boldsymbol{x} \in \mathbb{R}^n$, we use

[2] A *preorder* is a binary relation that is reflexive and transitive

notation $x(S) = \sum_{a \in S} x(a)$ for any $S \subseteq N$. Here, $x(\emptyset) = 0$ for convention. A characteristic function $v : \mathcal{F}_E \to \mathbb{R}$ is said to be *cohesive* if $v(N) = \hat{v}(N)$. We call a graph game (N, v, E) superadditive (respectively, cohesive) if the characteristic function is superadditive (respectively, cohesive).

3.1 Solution concepts

An *imputation* for a graph game (N, v, E) is a vector $\boldsymbol{x} \in \mathbb{R}^N$ satisfying *efficiency* : $x(N) = v(N)$, and *individual rationality* : $x(\{a\}) \geq v(\{a\})$, for all $a \in N$. For a graph game (N, v, E), let $\mathcal{I}(N, v, E)$ denote the set of imputations of (N, v, E).

Definition 2 (core). *The core $\mathcal{C}(N, v, E)$ of a graph game (N, v, E) is the set of all imputations $\boldsymbol{x}$ such that $x(S) \geq v(S)$, for all $S \in \mathcal{F}_E \setminus \{N, \emptyset\}$ }.*

Given an imputation $\boldsymbol{x} \in \mathcal{I}(N, v, E)$ and a feasible coalition $S \in \mathcal{F}_E \setminus \{N, \emptyset\}$, we define $e(\boldsymbol{x}, S) := v(S) - x(S)$. We call $e(\boldsymbol{x}, S)$ the *excess* of S at $\boldsymbol{x}$ for a graph game (N, v, E), which represents the loss (or gain, if it is negative) to the coalition S at $\boldsymbol{x}$. We denote $e_1(\boldsymbol{x})$ by the maximum excess with respect to $\boldsymbol{x}$, i.e., $e_1(\boldsymbol{x}) = \max_{S \in \mathcal{F}_E \setminus \{N, \emptyset\}} e(\boldsymbol{x}, S)$.

Definition 3 (least core). *The least core $\mathcal{LC}(N, v, E)$ of a graph game (N, v, E) is the set of all imputations $\boldsymbol{x}$ such that $e_1(\boldsymbol{x}) \leq e_1(\boldsymbol{y})$ for all $\boldsymbol{y} \in \mathcal{I}(N, v, E)$.*

For each imputation $\boldsymbol{x} \in \mathcal{I}(N, v, E)$, we denote by $\theta(\boldsymbol{x})$ the sequence of the components $e(\boldsymbol{x}, S)$ ($S \in \mathcal{F}_E \setminus \{N, \emptyset\}$) of $\boldsymbol{x}$ arranged in non-increasing order, i.e., $\theta(\boldsymbol{x}) = (e(\boldsymbol{x}, S_1), e(\boldsymbol{x}, S_2), \cdots, e(\boldsymbol{x}, S_k))$ with $e(\boldsymbol{x}, S_1) \geq e(\boldsymbol{x}, S_2) \geq \cdots \geq e(\boldsymbol{x}, S_k)$, where $\mathcal{F}_E \setminus \{N, \emptyset\} = \{S_1, S_2, \cdots, S_k\}$. For real k-sequences $\boldsymbol{u} = (u_1, u_2, \cdots, u_k)$ and $\boldsymbol{v} = (v_1, v_2, \cdots, v_k)$, $\boldsymbol{u}$ is *lexicographically smaller than or equal to* $\boldsymbol{v}$ (denoted by $\boldsymbol{u} \leq_L \boldsymbol{v}$) if and only if $\boldsymbol{u} = \boldsymbol{v}$, or $\boldsymbol{u} \neq \boldsymbol{v}$ and for the minimum index j such that $u_j \neq v_j$ we have $u_j < v_j$.

Definition 4 (nucleolus). *The nucleolus $\mathcal{N}(N, v, E)$ of a graph game (N, v, E) is the set of all imputations $\boldsymbol{x}$ such that $\theta(\boldsymbol{x}) \leq_L \theta(\boldsymbol{y})$, for all $\boldsymbol{y} \in \mathcal{I}(N, v, E)$.*

Given $\boldsymbol{x} \in \mathcal{I}(N, v, E)$ and $a, b \in N$ $(a \neq b)$, we define the *surplus* $s_{ab}(\boldsymbol{x})$ of a against b at $\boldsymbol{x}$ as $s_{ab}(\boldsymbol{x}) = \max\{\, e(\boldsymbol{x}, S) \mid S \in \mathcal{F}_E(a \setminus b) \,\}$. We say that player a has *stronger barganing power* than b at $\boldsymbol{x}$ if $s_{ab}(\boldsymbol{x}) > s_{ba}(\boldsymbol{x})$.

Definition 5 (kernel). *The kernel $\mathcal{K}(N, v, E)$ of a graph game (N, v, E) is the set of all imputations $\boldsymbol{x}$ such that for all players $b \in N$, if there exists another player $a \in N \setminus \{b\}$ who has stronger bargaining power than b, then $x(b) = v(\{b\})$.*

We remark the following containment relations among these classes of outcomes: $\mathcal{N}(N, v, E) \subseteq \mathcal{K}(N, v, E) \cap \mathcal{LC}(N, v, E)$ and $\mathcal{N}(N, v, E) \subseteq \mathcal{LC}(N, v, E) \subseteq \mathcal{C}(N, v, E)$ whenever $\mathcal{C}(N, v, E)$ is nonempty. The containment of the nucleolus in the kernel can be shown by a simple adaptation of the proof of Theorem 3 in [18], whereas the other relations immediately follow from the definitions.

In case where $E = \{\, \{a, b\} \mid a, b \in N, a \neq b \,\}$, the game (N, v, E) is said to have *full coalition structure* and is simply denoted by (N, v). The imputation set (respectively, the core, the least core, the nucleolus, and the kernel) of a graph game with full coalition structure is denoted by $\mathcal{I}(N, v)$ (respectively, $\mathcal{C}(N, v)$, $\mathcal{LC}(N, v)$, $\mathcal{N}(N, v)$, and $\mathcal{K}(N, v)$).

3.2 Computational setting

We address the following three natural complexity questions: for a solution concept $\mathcal{S}$,

- In-$\mathcal{S}$: given a graph game (N, v, E) and an imputation $\boldsymbol{x}$, decide whether $\boldsymbol{x} \in \mathcal{S}(N, v, E)$.
- $\mathcal{S}$-Existence : given a graph game (N, v, E), decide whether $\mathcal{S}(N, v, E) \neq \emptyset$.
- $\mathcal{S}$-Find : given a graph game (N, v, E), find an element $\boldsymbol{x} \in \mathcal{S}(N, v, E)$ if $\mathcal{S}(N, v, E) \neq \emptyset$ or output "$\mathcal{S}(N, v, E) = \emptyset$".

Throughout the paper, we only consider a graph game $\mathcal{G} = (N, v, E)$ whose characteristic function is an oracle computable in polynomial time in the size $||\mathcal{G}||$ of the game representation (e.g. see [3]). Formally, we assume that our game encoding $||\mathcal{G}||$ includes the graph $G = (N, E)$, and that the characteristic function v is given by an oracle such that the value $v(S)$ for any $S \in \mathcal{F}_E$ can be computed in time polynomial in $||\mathcal{G}||$.

4 Tractable cases: supermodular games

In this section, we introduce a relaxed notion of supermodularity, called *quasi-supermodularity*, and show that a variety of computational problems are tractable for the class of quasi-supermodular games on cycle-complete graphs.

Definition 6 (quasi-supermodularity). *Let $\mathcal{F}$ be a family of subsets of a finite set N. A set function $f : \mathcal{F} \to \mathbb{R}$ is called quasi-supermodular if for each $S, T \in \mathcal{F}$ with $S \cup T \in \mathcal{F}$ and $S \cap T \in \mathcal{F}$, the supermodular inequality* (2.2) *holds.*

A graph game (N, v, E) is said to be *quasi-supermodular* if the characteristic function v is quasi-supermodular. Notice that by definition, if a set function $f : \mathcal{F} \to \mathbb{R}$ such that $\emptyset \in \mathcal{F}$ and $f(\emptyset) = 0$ is quasi-supermodular, it is also superadditive. A similar notion of supermodularity can be found in Bilbao et al. [2], who defined quasi-convexity for games on convex geometries. In fact, these two concepts coincide when the family of feasible coalitions is a convex geometry.

We shall prove our tractability results in the following two steps: first, we reduce our problems to those on supermodular games with full coalition structure; and second, we derive polynomial solvability, by using results previously proven for the class of supermodular games in the unrestricted settings.

4.1 Restricted games

Given a graph (N, E), notice that $\hat{\mathcal{F}}_E = 2^N$ since each $S \subseteq N$ is guaranteed to have an $\mathcal{F}_E$-partition that consists of the singletons in S. For a graph game (N, v, E), we define its *restricted game* as a pair $(N, \hat{v})$, that is, the game with full coalition structure whose characteristic function is $\hat{v} : 2^N \to \mathbb{R}$. In what follows, we will observe that one can reduce some computational problems on quasi-supermodular graph games (N, v, E) on cycle-complete graphs, to those on supermodular games with full coalition structure $(N, \hat{v})$.

Theorem 1 ([1, 14]). *Given a graph game (N, v, E), the following statements hold.*

(i) *If v is cohesive, then $\mathcal{C}(N, v, E) = \mathcal{C}(N, \hat{v})$.*
(ii) *If v is superadditive and $\mathcal{C}(N, v, E) \neq \emptyset$, then $\mathcal{N}(N, v, E) = \mathcal{N}(N, \hat{v})$.*

Having observed that the core and the nucleolus of the restricted game coincide with those of the original game, we will see next whether the supermodularity of a characteristic function is also preserved to the restricted game. It is known that the relaxed supermodularity of a set function on an intersecting family leads to the supermodularity of the trunctation function.

Theorem 2 ([10]). *Let N be a finite set and $\mathcal{F}$ be a family of subsets of N with $N, \emptyset \in \mathcal{F}$. If $\mathcal{F}$ is an intersecting family and $f : \mathcal{F} \to \mathbb{R}$ is intersecting supermodular, then $\hat{f}$ is a supermodular function on a distributive lattice $\hat{\mathcal{F}}$.*

A quasi-supermodular function f on an intersecting family $\mathcal{F}$ is intersecting supermodular by definition; hence, its truncation is a supermodular function on a distributive lattice due to the previous theorem.

Corollary 1. *If (N, v, E) is a quasi-supermodular graph game whose underlying graph (N, E) is cycle-complete, then $\hat{v} : 2^N \to \mathbb{R}$ is supermodular and $\mathcal{C}(N, v, E)$ is nonempty.*

Proof. The restricted game $(N, \hat{v})$ is a supermodular game by Lemma 1 and Theorem 2, and thereby $\mathcal{C}(N, \hat{v})$ is nonempty [6]. This implies that $\mathcal{C}(N, v, E) = \mathcal{C}(N, \hat{v})$ is nonempty.

It is natural to ask whether the condition for $\mathcal{F} \subseteq 2^N$ to be an intersecting family is necessary to preserve supermodulartiy of f. The next results will show that it is indeed necessary.

Lemma 2. *Let N be a finite set and $\mathcal{F}$ be a family of subsets of N with $N, \emptyset \in \mathcal{F}$ such that $\hat{\mathcal{F}}$ is a distributive lattice. If the truncation $\hat{f} : \hat{\mathcal{F}} \to \mathbb{R}$ is supermodular for every quasi-supermodular function $f : \mathcal{F} \to \mathbb{R}$, then $\mathcal{F}$ is an intersecting family.*

Proof (Sketch). Let $\mathcal{F}$ be a subset of 2^N such that $N, \emptyset \in \mathcal{F}$ and $\hat{\mathcal{F}}$ is a distributive lattice. Suppose that $\mathcal{F}$ is not an intersecting family. Then, there exists a pair of nonempty subsets $S, T \in \mathcal{F}$ such that $S \cap T \neq \emptyset$, and $S \cup T \notin \mathcal{F}$ or $S \cap T \notin \mathcal{F}$. We will show that there exists a quasi-supermodular set function on $\mathcal{F}$ whose truncation is not supermodular. Too see this, we define a set function $f : \mathcal{F} \to \mathbb{R}$ by $f(X) = |X| - 1$ for each $X \in \mathcal{F} \setminus \{\emptyset\}$ and $f(\emptyset) = 0$. One can readily see that the function f is quasi-supermodular, but $\hat{f}$ is not supermodular.

Combining Theorem 2 and Lemma 2 yields the following corollary.

Corollary 2. *Let N be a finite set and $\mathcal{F}$ be a family of subsets of N with $N, \emptyset \in \mathcal{F}$ such that $\hat{\mathcal{F}}$ is a distributive lattice. Then, the following two statements are equivalent.*

(i) *$\mathcal{F}$ is an intersecting family.*
(ii) *For every quasi-supermodular function $f : \mathcal{F} \to \mathbb{R}$, its truncation $\hat{f} : \hat{\mathcal{F}} \to \mathbb{R}$ is supermodular.*

4.2 Supermodular function maximization

We introduce several celebrated results related to supermodular functions on intersecting families.

Theorem 3 ([19]). *Let N be a finite set and $\mathcal{F} \subseteq 2^N$. If $\mathcal{F}$ is a distributive lattice represented by the smallest set M and the largest set L in $\mathcal{F}$, together with a preorder $\preceq$ satisfying* (2.1)*, and $f : \mathcal{F} \to \mathbb{R}$ is a supermodular function given by an oracle, one can find $S \in \mathcal{F}$ maximizing $f(S)$ in strongly polynomial time in $|N|$.*

If $\mathcal{F}$ is an intersecting family and $f : \mathcal{F} \to \mathbb{R}$ is intersecting supermodular, the restriction f_a of f to $\mathcal{F}(a)$ is a supermodular function on a distributive lattice; thereby, the previous theorem enables us to find a set maximizing f in strongly polynomial time. Another nice property of an intersecting supermodular function $f : \mathcal{F} \to \mathbb{R}$ is that each value $\hat{f}(S)$ for $S \in \hat{\mathcal{F}}$ can be computed in strongly polynomial time, having a value-giving oracle for f and a compact representation for $\mathcal{F}$, as we shall see below.

Theorem 4 ([11]). *Let N be a finite set and $\mathcal{F} \subseteq 2^N$. If $\mathcal{F}$ is an intersecting family where each $\mathcal{F}(a)$ for $a \in N$ is represented by the smallest set M_a and the largest set L_a in $\mathcal{F}(a)$, together with a preorder $\preceq_a$ satisfying* (2.1)*, and $f : \mathcal{F} \to \mathbb{R}$ is an intersecting supermodular function given by an oracle, one can calculate $\hat{f}(S)$ for each $S \in \hat{\mathcal{F}}$ in strongly polynomial time in $|N|$.*

As we have seen before, the set of connected subsets of a cycle-complete graph (N, E) forms an intersecting family; moreover, given such a graph, one can construct suitable representations for a family $\mathcal{F}_E$ in polynomial time (we omit the proof due to space constraints).

Lemma 3. *Given a connected cycle-complete graph (N, E) and $a \in N$, one can construct the smallest set M_a and the largest set L_a in $\mathcal{F}_E(a)$, together with a preorder $\preceq_a$ on N satisfying* (2.1) *in time polynomial in $|N|$.*

Lemma 4. *Given a connected cycle-complete graph (N, E) and $a, b \in N$ ($a \neq b$), one can construct the smallest set M_{ab} and the largest set L_{ab} in $\mathcal{F}_E(a \setminus b)$, together with a preorder $\preceq_{ab}$ on N satisfying* (2.1) *in time polynomial in $|N|$.*

4.3 Polynomial-time solvability

We are now ready to give polynomial-time solvability results in the following theorems.

Theorem 5. *For a quasi-supermodular graph game (N, v, E) where v is given by an oracle and (N, E) is cycle-complete, an element of the core can be found in strongly polynomial time in $|N|$.*

Proof. Let (N, v, E) be a quasi-supermodular game on a cycle-complete graph where v is given by an oracle. By Theorem 1, it suffices to find an element $\boldsymbol{x} \in \mathcal{C}(N, \hat{v})$ for the supermodular game $(N, \hat{v})$. By Theorem 4 and Lemma 3 that each $\hat{v}(S)$ for $S \subseteq N$ can be computed in strongly polynomial time in $|N|$. Edmonds [6] presented a strongly polynomial-time algorithm to find an element of the core for supermodular games with full coalition structure. Therefore, one can also find $\boldsymbol{x} \in \mathcal{C}(N, v, E)$ in strongly polynomial time in $|N|$.

Theorem 6. *For a quasi-supermodular graph game (N, v, E) where v is given by an oracle and (N, E) is cycle-complete, one can check whether a given imputation $\boldsymbol{x}$ belongs to the core in strongly polynomial time in $|N|$.*

Proof. Given a quasi-supermodular game (N, v, E) on a cycle-complete graph and an imputation $\boldsymbol{x} \in \mathcal{I}(N, v, E)$, checking whether $\boldsymbol{x} \in \mathcal{C}(N, v, E)$ can be reduced to the maximization of a supermodular function f_a on a distributive lattice $\mathcal{F}_E(a)$ where $a \in N$ and $f_a(S) = v(S) - x(S)$ for $S \in \mathcal{F}_E(a)$: the imputation $\boldsymbol{x} \in \mathcal{C}(N, v, E)$ if and only if for each $a \in N$, the maximum of f_a is less than or equal to 0. This can be checked in strongly polynomial time in $|N|$ due to Theorem 3 and Lemma 3.

The similar approach can be used to show the polynomial-time solvability for the least core, the nucleolus and the kernel.

Theorem 7. *For a quasi-supermodular graph game (N, v, E) where v is a rational-valued function given by an oracle, and (N, E) is cycle-complete, an imputation of the nucleolus can be found in time polynomial in $|N|$ and $\langle v \rangle$.*

Proof. Let (N, v, E) be a quasi-supermodular game on a cycle-complete graph where v is given by an oracle. Since v is superadditive and $\mathcal{C}(N, v, E)$ is nonempty, it suffices to find $\boldsymbol{x} \in \mathcal{N}(N, \hat{v}) = \mathcal{N}(N, v, E)$ by Theorem 1. We have seen that $\hat{v}$ is supermodular, and each $\hat{v}(S)$ for $S \subseteq N$ can be computed in strongly polynomial time in $|N|$. Faigle et al. [8] showed that the nucleolus can be computed in time polynomial in $|N|$ and $\langle v \rangle$ for supermodular games (N, v) with full coalition structure. Hence, one can find $\boldsymbol{x} \in \mathcal{N}(N, \hat{v})$ in time polynomial in $|N|$ and $\langle v \rangle$.

As noted in Section 3, the nucleolus always belongs to the least core and the kernel. Hence, Theorem 7 also implies that finding these solutions can be done in time polynomial in $|N|$ and $\langle v \rangle$ for quasi-supermodular graph games (N, v, E) on cycle-complete graphs.

Theorem 8. *For a quasi-supermodular graph game (N, v, E) where v is given by an oracle and (N, E) is cycle-complete, one can check whether a given imputation $\boldsymbol{x}$ belongs to the kernel in strongly polynomial time in $|N|$.*

Proof. Notice that if one can efficiently calculate each surplus $s_{ab}(\boldsymbol{x})$ for each pair of distinct players $a, b \in N$, checking if $\boldsymbol{x} \in \mathcal{K}(N, v, E)$ is easy. Let (N, v, E) be a quasi-supermodular game on a cycle-complete graph where v is given by an oracle, and $\boldsymbol{x} \in \mathcal{I}(N, v, E)$. For $a, b \in N$ $(a \neq b)$, computing the surplus $s_{ab}(\boldsymbol{x})$ is equivalent to the maximization of a supermodular function f_{ab} on a distributive lattice $\mathcal{F}_E(a \setminus b)$ where $f_{ab}(S) = v(S) - x(S)$ for $S \in \mathcal{F}_E(a \setminus b)$, which can be done in strongly polynomial time in $|N|$ by Theorem 3 and Lemma 4.

Theorem 9. *For a quasi-supermodular graph game (N, v, E) where v is a rational-valued function given by an oracle, and (N, E) is cycle-complete, one can check whether a given imputation $\boldsymbol{x}$ belongs to the nucleolus or the least core in time polynomial in $|N|$ and $\langle v \rangle$.*

Proof. By Theorem 7, one can compute $\boldsymbol{x}^* \in \mathcal{N}(N, v, E)$ in time polynomial in $|N|$ and $\langle v \rangle$. Recall that $|\mathcal{N}(N, v, E)| = |\mathcal{N}(N, \hat{v})| = 1$ [18]. Hence, one can check whether a given imputation $\boldsymbol{x}$ belongs to the nucleolus by comparing each element of the vectors $\boldsymbol{x}$ and $\boldsymbol{x}^*$. Notice that an imputation $\boldsymbol{x} \in \mathcal{LC}(N, v, E)$ if and only if $\max\{\, s_{ab}(\boldsymbol{x}) \mid a, b \in N, a \neq b \,\} = \max\{\, s_{ab}(\boldsymbol{x}^*) \mid a, b \in N, a \neq b \,\}$. These values can be computed in strongly polynomial time in $|N|$, since each surplus can be computed in strongly polynomial time, as we have seen in the proof of Theorem 8.

5 Hardness results

So far, we discussed how supermodularity of the characteristinc function enables an efficient computation of cooperative solutions for games on cycle-complete graphs. In this section, let us turn our attention to the complexity questions for non-supermodular games on trees, which are a natural subclass of cycle-complete graphs. It seems natural to conjecture that tree restrictions help to decrease complexity. However, we will see in this section that this may not be the case.

5.1 The core

Our first hardness result is the co-NP-completeness of Core-Existence for games on trees. The result of Demange [4] regarding stability in cooperative games on trees implies the following necessary and sufficient condition for a game on a tree to have a nonempty core.

Lemma 5. *For graph games (N, v, E) whose underlying graph (N, E) is a tree, the core is nonempty if and only if $v : \mathcal{F}_E \to \mathbb{R}$ is cohesive.*

It follows from Lemma 5 that determining the non-emptiness of the core of a game on a tree is equivalent to the problem of deciding whether the given characteristic function is cohesive. By a simple reduction from Sat, one can prove that the problem is co-NP-complete for games on trees. We omit the detail due to space constraints.

Theorem 10. Core-Existence *is co-NP-complete for graph games (N, v, E) whose underlying graph (N, E) is a star.*

Corollary 3. *If one can find an element of the core for graph games (N, v, E) whose underlying graph (N, E) is a star in polynomial time, then P=NP.*

5.2 The least core, the nucleolus and the kernel

We now move on to the complexity analysis for the least core, the nucleolus and the kernel of games on trees.

Theorem 11. In-LeastCore, In-Nucleolus, *and* In-Kernel *are co-NP-hard for cohesive graph games* (N, v, E) *whose underlying graph* (N, E) *is a star.*

Proof. We reduce from Sat. Given a Boolean formula ϕ over the set of variables $\{\alpha_1, \alpha_2, \ldots, \alpha_k\}$, we construct a graph game (N, v, E) as follows. Introduce one player a_h for each variable α_h $(h = 1, 2, \ldots, k)$; and, introduce two other players c and d. Let $A = \{a_1, a_2, \ldots, a_k\}$. We set $N = \{c, d\} \cup A$ and $E = \{\{c, d\}\} \cup \{\, \{c, a_h\} \mid a_h \in A\,\}$. Note that the graph (N, E) is a star with the center player c. Hence, a feasible coalition of this game is either a singleton, or a coalition including c. The value $v(S)$ for $S \in \mathcal{F}_E$ is given as follows. Firstly, $v(S) = 1$ when $S = N$; $v(S) = 0$ when $|S| = 1$ or $d \in S \neq N$; $v(S) = 1$ when $S = T \cup \{c\}$ for a nonempty subset $T \subseteq A$ such that ϕ is satisfied by setting the variables $\{\, \alpha_h \mid a_h \in T\,\}$ to true and the variables $\{\, \alpha_h \mid a_h \in A \setminus T\,\}$ to false; finally, $v(S) = 0$ when $S = T \cup \{c\}$ for a nonempty subset $T \subseteq A$ such that ϕ is not satisfied by setting the variables $\{\, \alpha_h \mid a_h \in T\,\}$ to true and the variables $\{\, \alpha_h \mid a_h \in A \setminus T\,\}$ to false. Observe that the characteristic function v is cohesive, and that given a subset $S \in \mathcal{F}_E$, the value $v(S)$ can be computed in polynomial time. Also, it follows from the definition of v that ϕ is unsatisfiable if and only if $v(S) = 0$, for all $S \in \mathcal{F}_E \setminus \{N\}$. Let $n = |N|$ and $\boldsymbol{x}^* = (1/n, 1/n, \ldots, 1/n)^\top$. Clearly, $\boldsymbol{x}^* \in \mathcal{I}(N, v, E)$. We will now argue that the following statements hold:

(i) If ϕ is unsatisfiable, then $\{\boldsymbol{x}^*\} = \mathcal{LC}(N, v, E) = \mathcal{K}(N, v, E)$.
(ii) If ϕ is satisfiable, then $\boldsymbol{x}^* \notin \mathcal{LC}(N, v, E) \cup \mathcal{K}(N, v, E)$.

(i) : Suppose that ϕ is unsatisfiable. We will first show that any imputation different from $\boldsymbol{x}^*$ belongs to neither the kernel nor the least core. Take any $\boldsymbol{x} \in \mathcal{I}(N, v, E)$ where $\boldsymbol{x} \neq \boldsymbol{x}^*$. Clearly, $x(a) < 1/n < x(b)$ for some $a, b \in N$. Observe that $v(S) = 0$, for all $S \in \mathcal{F}_E \setminus \{N\}$ by definition of v, and that $\boldsymbol{x} \geq \mathbf{0}$ by individual rationality, which follows that $s_{ab}(\boldsymbol{x}) = -x(a) > -1/n > -x(b) = s_{ba}(\boldsymbol{x})$. Notice, however, that $x(b) > 0 = v(\{b\})$. Hence, $\boldsymbol{x} \notin \mathcal{K}(N, v, E)$. Furthermore, we have $e_1(\boldsymbol{x}) \geq e(\boldsymbol{x}, \{a\}) > -1/n = e_1(\boldsymbol{x}^*)$. Thus, $\boldsymbol{x} \notin \mathcal{LC}(N, v, E)$ since $\boldsymbol{x}^*$ gives a smaller maximum excess than $\boldsymbol{x}$. It follows that $\boldsymbol{x}^*$ is the unique imputation that belongs to the least core. Recall that the nucleolus is nonempty whenever the imputation set is nonempty and always lies in the intersection of the least core and the kernel. Therefore, $\boldsymbol{x}^*$ is also in the nucleolus of the game, and thereby belongs to the kernel.

(ii) : Suppose that ϕ is satisfiable. Then, there exists $S^* \in \mathcal{F}_E$ such that $v(S^*) = 1$ and $S^* = T \cup \{c\}$ for some $T \subseteq A$. Let $\boldsymbol{y}$ be an imputation of (N, v, E) such that the central player c receives the whole value $v(N) = 1$ and any other player receives nothing, i.e., $y(c) = 1$ and $y(a) = 0$ for all $a \in N \setminus \{c\}$. Then, it can be easily checked that $e_1(\boldsymbol{y}) \leq 0$. However, the maximum excess with respect to $\boldsymbol{x}^*$ is greater than 0 because $e_1(\boldsymbol{x}^*) \geq v(S^*) - x(S^*) = 1 - |S^*|/n > 0$. Hence,

$\boldsymbol{x}^* \notin \mathcal{LC}(N, v, E)$. We will next show that $\boldsymbol{x}^*$ does not belong to the kernel. Observe that $c \in S^*$ and $d \notin S^*$, which implies that $s_{cd}(\boldsymbol{x}^*) \geq e(\boldsymbol{x}^*, S^*) = 1 - |S^*|/n > 0$. On the other hand, $s_{dc}(\boldsymbol{x}^*) = v(\{d\}) - x^*(d) = -1/n < 0$ since $\{d\}$ is the unique coalition in $\mathcal{F}_E(d \setminus c)$. Furthermore, $x^*(d) = 1/n > 0 = v(\{d\})$. Hence, $\boldsymbol{x}^* \notin \mathcal{K}(N, v, E)$.

It follows from (i) and (ii) that the Boolean formula ϕ is unsatisfiable if and only if $\boldsymbol{x}^*$ belongs to the least core, the nucleolus, or the kernel of the game (N, v, L).

Corollary 4. *If one can find an element of the least core or the kernel for cohesive graph games (N, v, E) whose underlying graph (N, E) is a star in polynomial time, then P=NP.*

Proof. We will show that polynomial-time algorithms for our problems can be used to decide Sat in polynomial time. Given a Boolean formula ϕ, we construct the game (N, v, E) defined in the proof of Theorem 11. Let $n = |N|$ and $\boldsymbol{x}^* = (1/n, 1/n, \ldots, 1/n)^\top$. Take any $\boldsymbol{x} \in \mathcal{LC}(N, v, E) \cup \mathcal{K}(N, v, E)$. By the previous proof for Theorem 11, it is clear that ϕ is unsatisfiable if and only if $\boldsymbol{x} = \boldsymbol{x}^*$. It follows that, by finding an imputation $\boldsymbol{x}$ of the least core or the kernel of the game (N, v, E), and checking whether $\boldsymbol{x} = \boldsymbol{x}^*$, we can decide in polynomial time whether ϕ is satisfiable or not.

Our next theorem shows that Δ_2^p-hardness holds for the least core, the nucleolus, and the kernel of general games on trees. We prove this by a reduction from the problem of deciding whether the least significant variable is true in the lexicographically maximum satisfying assignment, which was shown to be Δ_2^p-complete [16]. The reduction behind the theorem is inspired by an argument of Greco et al. [12], who showed that membership problems for these solutions are Δ_2^p-complete for games with full coalition structure.

Given a Boolean formula ϕ over the variables $\alpha_1, \alpha_2, \ldots, \alpha_k$, recall that for truth assignments $\boldsymbol{u} = (u_1, u_2, \cdots, u_k)$ and $\boldsymbol{v} = (v_1, v_2, \cdots, v_k)$ of ϕ where $u_i, v_i \in \{0, 1\}$ for $i = 1, 2, \ldots, k$, $\boldsymbol{u}$ is lexicographically greater than $\boldsymbol{v}$ if and only if for the minimum index j such that $u_j \neq v_j$ we have $u_j = 1$ and $v_j = 0$.

Definition 7 (LeastLexSat). *An instance of* LeastLexSat *is a satisfiable Boolean formula ϕ over the variables $\alpha_1, \alpha_2, \ldots, \alpha_k$. It is a yes-instance if α_k is true in the lexicographically maximum satisfying assignment of ϕ and a no-instance otherwise.*

Lemma 6. *For $\boldsymbol{u} = (u_1, u_2, \cdots, u_k)$ and $\boldsymbol{v} = (v_1, v_2, \cdots, v_k)$ where $u_i, v_i \in \{0, 1\}$ for $i = 1, 2, \ldots, k$, $\boldsymbol{u}$ is lexicographically greater than $\boldsymbol{v}$ if and only if $\sum_{u_i=1} 2^{k-i+1} \geq \sum_{v_i=1} 2^{k-i+1} + 2$.*

Lemma 6 follows immediately from the definition of lexicographical order.

Theorem 12. In-LeastCore*,* In-Nucleolus*, and* In-Kernel *are Δ_2^p-complete for graph games (N, v, E) whose underlying graph (N, E) is a tree.*

Proof. Membership in Δ_2^p was proved in [12, 13]. We reduce from LeastLexSat.

Let ϕ be a satisfiable Boolean formula over the set of variables $\{\alpha_1, \alpha_2, \dots, \alpha_k\}$. We assume that the formula ϕ is not satisfied by setting all the variables to true, or setting all the variables to false.

We construct a graph game (N, v, E) as follows. Introduce one player a_k for α_k; introduce two players a_h and $\overline{a}_h$ for each of other variables α_h ($h = 1, 2, \dots, k-1$); finally, introduce two other players c and $\overline{c}$. Let $A = \{a_1, a_2, \dots, a_k\}$ and $\overline{A} = \{\overline{a}_1, \overline{a}_2, \dots, \overline{a}_{k-1}, a_k\}$. For a nonempty subset $T \subseteq A$, we denote by $\overline{T}$ the dual of T, i.e., $\overline{T} = (T \cap \{a_k\}) \cup \{\, \overline{a}_h \mid a_h \in A, h \neq k \,\}$. We set $N = A \cup \overline{A} \cup \{c, \overline{c}\}$ and $E = \{\, \{c, a\} \mid a \in A \,\} \cup \{\, \{\overline{c}, a\} \mid a \in \overline{A} \,\}$. Note that the graph (N, E) is connected since $A \cap \overline{A} = \{a_k\}$. A feasible coalition of this game is either a singleton, a coalition including the three players c, a_k, and $\overline{c}$, a coalition of the form $T \cup \{c\}$ where $T \subseteq A$, or a coalition of the form $\overline{T} \cup \{\overline{c}\}$ where $\overline{T} \subseteq \overline{A}$.

The values $v(S)$ for $S \in \mathcal{F}_E$ is given as follows. Firstly, $v(S) = 1$ when $S = N$; $v(S) = 0$ when $|S| = 1$ or $\{c, a_k, \overline{c}\} \subseteq S \neq N$; $v(S) = \sum_{a_i \in T} 2^{k-i+1}$ when $S = T \cup \{c\}$ or $\overline{T} \cup \{\overline{c}\}$ for some nonempty subset $T \subseteq A$ such that ϕ is satisfied by setting the variables $\{\, \alpha_h \mid a_h \in T \,\}$ to true and the variables $\{\, \alpha_h \mid a_h \in A \setminus T \,\}$ to false; finally, $v(S) = 0$ when $S = T \cup \{c\}$ or $\overline{T} \cup \{\overline{c}\}$ for some nonempty subset $T \subseteq A$ such that ϕ is not satisfied by setting the variables $\{\, \alpha_h \mid a_h \in T \,\}$ to true and the variables $\{\, \alpha_h \mid a_h \in A \setminus T \,\}$ to false. Note that given a subset $S \in \mathcal{F}_E$, a polynomial-time oracle for $v(S)$ can be constructed.

Let $T^* \in \operatorname{argmax}\{\, v(T \cup \{c\}) \mid T \subseteq A \,\}$ and $\overline{T^*}$ be the dual coalition of T^* (hence, $v(T^* \cup \{c\}) = v(\overline{T^*} \cup \{\overline{c}\})$). By Lemma 6 and the definition of v, the truth assignment that sets the variables corresponding to the players in T^* to true and the rest to false is the lexicographically maximum satisfying assignment of ϕ. Thus, α_k evaluates to true in the lexicographically maximum satisfying assignment for ϕ if and only if $a_k \in T^*$. Let $S^* = T^* \cup \{c\}$ and $\overline{S^*} = \overline{T^*} \cup \{\overline{c}\}$. Before we proceed, we give the following lemma.

Lemma 7. *For any $\boldsymbol{x} \in \mathcal{I}(N, v, E)$ and $S \in \mathcal{F}_E \setminus \{N, \emptyset, S^*, \overline{S^*}\}$,*

$$e(\boldsymbol{x}, S^*) > e(\boldsymbol{x}, S) \text{ and } e(\boldsymbol{x}, \overline{S^*}) > e(\boldsymbol{x}, S). \tag{5.1}$$

Proof. Take any $\boldsymbol{x} \in \mathcal{I}(N, v, E)$ and $S \in \mathcal{F}_E \setminus \{N, \emptyset, S^*, \overline{S^*}\}$. Observe that $x(S) - x(S^*) \geq -1$, since $\boldsymbol{x} \geq \mathbf{0}$ and since $x(N) = v(N) = 1$. We will now show that $v(S^*) - v(S) \geq 2$. Consider when $|S| = 1$ or $\{c, a_k, \overline{c}\} \subseteq S$. Then, $v(S) = 0$ by definition of v. In addition, $v(S^*) \geq 2$ since ϕ is a satisfiable formula not to be satisfied by setting all the variables to false. Hence, $v(S^*) - v(S) = v(S^*) \geq 2$. Consider when $S = T \cup \{c\}$ or $S = \overline{T} \cup \{\overline{c}\}$ for some $T \subseteq A$ where $T \neq T^*$. By Lemma 6, we have $v(S^*) - v(S) \geq 2$. Thus, $e(\boldsymbol{x}, S^*) - e(\boldsymbol{x}, S) \geq 2 - 1 > 0$. Similarly, one can prove that $e(\boldsymbol{x}, \overline{S^*}) > e(\boldsymbol{x}, S)$. Hence, the inequalities (5.1) hold.

Let $\boldsymbol{x}^* \in \mathbb{R}^N$ be an imputation such that player a_k receives the whole value $v(N) = 1$, that is, $x^*(a_k) = 1$ and $x^*(b) = 0$ for each $b \in N \setminus \{a_k\}$. We will now argue that the following statements hold:

(i) If $a_k \in T^*$, then $\{\boldsymbol{x}^*\} = \mathcal{LC}(N, v, E) = \mathcal{K}(N, v, E)$.
(ii) If $a_k \notin T^*$, then $\boldsymbol{x}^* \notin \mathcal{LC}(N, v, E) \cup \mathcal{K}(N, v, E)$.

(i) : Suppose that $a_k \in T^*$. We will first prove that $\boldsymbol{x}^*$ is the unique imputation that belongs to the kernel of the game (N, v, E). Observe that a_k is the only player that belongs to both S^* and $\overline{S^*}$. By Lemma 7, player a_k has stronger bargaining power towards any other player. Hence, $\boldsymbol{x}^*$ is the unique imputation that belongs to the kernel of the game (N, v, E). We will next prove that $\boldsymbol{x}^*$ is the unique imputation that lies in the least core of the game (N, v, E). Observe that $e(\boldsymbol{x}^*, S^*) = e(\boldsymbol{x}^*, \overline{S^*}) = v(S^*) - 1$. By Lemma 7, $e_1(\boldsymbol{x}^*) = v(S^*) - 1$. It suffices to show that for any imputation $\boldsymbol{x} \neq \boldsymbol{x}^*$,

$$\max\{e(\boldsymbol{x}, S^*), e(\boldsymbol{x}, \overline{S^*})\} > v(S^*) - 1. \tag{5.2}$$

Take any $\boldsymbol{x} \in \mathcal{I}(N, v, E)$ such that $\boldsymbol{x} \neq \boldsymbol{x}^*$. If $x(S^*) = 1$ and $x(\overline{S^*}) = 1$, then this would mean that $x(a_k) = 1$, contradicting the fact that $\boldsymbol{x} \neq \boldsymbol{x}^*$. Hence, we have $x(S^*) < 1$ or $x(\overline{S^*}) < 1$, which implies that $e(\boldsymbol{x}, S^*) > v(S^*) - 1$ or $e(\boldsymbol{x}, \overline{S^*}) > v(S^*) - 1$. This gives (5.2).

(ii) : Suppose that $a_k \notin T^*$. Then, player a_k belongs to neither the coalition S^* nor the coalition $\overline{S^*}$, and hence $x^*(S^*) = x^*(\overline{S^*}) = 0$. It follows from Lemma 7 that any imputation $\boldsymbol{x}$ such that $x(S^*) = x(\overline{S^*}) > 0$ gives a smaller maximum excess than that of $\boldsymbol{x}^*$. Thus, $\boldsymbol{x}^* \notin \mathcal{LC}(N, v, E)$. By Lemma 7, it also holds that for any $\boldsymbol{x} \in \mathcal{I}(N, v, E)$, $s_{ca_k}(\boldsymbol{x}) \geq e(\boldsymbol{x}, S^*) > \max_{S \in \mathcal{F}_E(a_k \setminus c)} e(\boldsymbol{x}, S) = s_{a_k c}(\boldsymbol{x})$, because $a_k \notin S^*$ and $a_k \notin \overline{S^*}$. Thus, if $\boldsymbol{x} \in \mathcal{K}(N, v, E)$, then we must have $x(a_k) = v(\{a_k\}) = 0$. We conclude that $\boldsymbol{x}^* \notin \mathcal{K}(N, v, E)$.

It follows from (i) and (ii) that the least significant player a_k belongs to T^* if and only if $\boldsymbol{x}^*$ belongs to the least core, the nucleolus, or the kernel of the game (N, v, L).

Corollary 5. *If one can find an element of the least core or the kernel for graph games (N, v, E) whose underlying graph (N, E) is a tree in polynomial time, then P=Δ_2^p.*

Proof. The proof is similar to the previous one for Corollary 4. We omit the details due to space restrictions.

Acknowledgments

The author is grateful for very useful comments by the CoopMas reviewers, and thanks Edith Elkind and Yoshitsugu Yamamoto for helpful suggestions.

References

1. Algaba, E., Bilbao, J., and López, J. A unified approach to restricted games. *Theory and Decision*, 50(4), 333–345 (2001).
2. Bilbao, J. M., Lebrón, and E., Jiménez, N. The core of games on convex geometries. *European Journal of Operations Research*, 119, 365–372 (1999).

3. Chalkiadakis, G., Greco, G., and Markakis, E. Characteristic function games with restricted agent interactions: Core-stability and coalition structures. *Artificial Intelligence*, 232, 76–113 (2016).
4. Demange, G. On group stability in hierarchies and networks. *Journal of Political Economy*, 112(4), 754–778 (2004).
5. Deng, X., and Papadimitriou, C. H. On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, 19(2), 257–266 (1994).
6. Edmonds, J. Submodular functions, matroids, and certain polyhedra. In Guy, R., Hanani, H., Sauer, N., Schönheim, J. (eds.): *Combinatorial Structures and Their Applications*, Gordon and Breach (1970).
7. Elkind, E. Coalitional games on sparse social networks. In T. Liu, Q. Qi, and Y. Ye, editors, *WINE'14*, volume 8877 of *Lecture Notes in Computer Science*, pages 308–321. Springer, 2014.
8. Faigle, U. Kern, W., and Kuipers, J. On the computation of the nucleolus of a cooperative game. *International Journal of Game Theory*, 30, 79–98 (2001).
9. Faigle, U. Kern, W., and Paulusma, D. Note on the computational complexity of least core concepts for min-cost spanning tree games. *Mathematical Methods of Operations Research*, 52, 23–38 (2000).
10. Fujishige, S. Structures of polyhedra determined by submodular functions on crossing families. *Mathematical Programming*, 29, 125–141 (1984).
11. Frank, A. and Tardos, É. Generalized polymatroids and submodular flows. *Mathematical Programming*, 42, 489–563 (1988).
12. Greco, G., Malizia, E., Palopoli, L., and Scarcello, F. On the complexity of core, kernel, and bargaining set. *Artificial Intelligence*, 175(12–13), 1877–1910 (2011).
13. Greco, G., Malizia, E., Palopoli, L., and Scarcello, F. The complexity of the nucleolus in compact games. *ACM Transactions on Computation Theory*, 7(1), 3:1–3:52 (2014).
14. Huberman, G. The nucleolus and the essential coalitions. In A. Bensoussan, J. Lions (eds.): *Analysis and Optimization of Systems*, Lecture Notes in Control and Information Sciences, Springer-Verlag, Berlin/Heidelberg, vol. 28, pp. 416–422 (1980).
15. Jamison-Waldner, R.E. Convexity and block graphs. *Congressus Numerantium*, 33, 129–142 (1981).
16. Krentel, M. W. The complexity of optimization problems. *Journal of Computer and System Sciences*, 36(3), 490–509 (1988).
17. Myerson, R. B. Graphs and cooperation in games. *Mathematics of Operations Research*, 2(3), 225–229 (1977).
18. Schmeidler, D. The nucleolus of a characteristic function game. *SIAM Journal on Applied Mathematics*, 17, 1163–1170 (1969).
19. Schrijver, A. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theroy, Series B*, 80: 346–355 (2000).
20. Slikker, M., and van den Nouweland, A. *Social and Economic Networks in Cooperative Game Theory*, Kluwer Academic Publishers (2001).

Altruistic Hedonic Games

Nhan-Tam Nguyen, Anja Rey, Lisa Rey, Jörg Rothe, and Lena Schend

Heinrich-Heine-Universität Düsseldorf, Düsseldorf, Germany
{nguyen, rey, lrey, rothe, schend}@cs.uni-duesseldorf.de

Abstract. Hedonic games are coalition formation games in which players have preferences over the coalitions they can join. All models of representing hedonic games studied so far are based upon selfish players only. Among the known ways of representing hedonic games compactly, we focus on friend-oriented hedonic games and propose a novel model for them that takes into account not only a player's own preferences but also her friends' preferences under three degrees of altruism. We study both the axiomatic properties of these games and the computational complexity of problems related to various stability concepts.

Keywords: hedonic games; coalition formation; stability; altruism

1 Introduction

Hedonic games, proposed by Drèze and Greenberg [17] and later formally modelled by Bogomolnaia and Jackson [10] and Banerjee et al. [8], are coalition formation games in which players have preferences over coalitions (subsets of players) they can be part of. In the context of decentralized coalition formation, several stability concepts and representations have been studied from an axiomatic and a computational complexity point of view; see Woeginger's survey [33] on this topic and the book chapters by Aziz and Savani [6] and Elkind and Rothe [18] for an overview.

Dimitrov et al. [16] proposed a model that allows for compact representation of hedonic games, namely, the friend-and-enemy encoding of the players' preferences, where each player divides the set of players into friends and enemies. Based on such a network of friends, they suggest two models of preference extensions: appreciation of friends and aversion to enemies. In *friend-oriented hedonic games*, a coalition A is preferred to another coalition B if A contains either more friends than B or the same number of friends as B but fewer enemies than B. This setting corresponds to a network of players represented as a graph. Since we study symmetric friendship relations for stability reasons, this graph is undirected. For example, suppose there are four players, 1, 2, 3, and 4, and let 1 be friends with 2 but neither with 3 nor with 4, while 2 and 3 are friends with each other but not with 4. The corresponding network is displayed in Figure 1.

$$1 \text{ --- } 2 \text{ --- } 3 \qquad 4$$

Fig. 1: Example of a network of friends

Now, in the friend-oriented extension model player 2 prefers teaming up with 1 and 3 to forming a coalition with 1 and 4. Player 1, on the other hand, is indifferent between coalitions $\{1,2,3\}$ and $\{1,2,4\}$. Intuitively, however, 1 would have an advantage from being in a coalition with 2 and 3, since 2 and 3—being friends—can be expected to cooperate better than 2 and 4. Also, 1 can be expected to care about her friend 2's interests and thus might prefer a coalition in which 2 is satisfied ($\{1,2,3\}$) to one in which 2 is less satisfied ($\{1,2,4\}$). In order to model these kinds of preferences we will introduce several degrees of altruism, starting from friend-oriented hedonic games. Taking friends' preferences into account does not contradict the idea of hedonic games: In hedonic games player i's utility function depends only on the coalitions that contain i. Since player i is also interested in her friends' satisfaction (with varying degrees), we incorporate this notion into player i's utility function. Note that player i's utility is still a function of the coalitions containing her, which in addition takes her friends' preferences that are in the *same* coalition as player i into account.

Our Contribution and Related Work: Focusing on the friend-oriented encoding of preferences and taking the idea of players caring about their friends' preferences into account, we propose hedonic games with altruistic influences. In particular, we define three degrees of altruism, from being selfish first, over aggregating opinions of a player and her friends equally, to altruistically letting one's friends decide first. The latter is the most altruistic case, as we assume that from a player's perspective only friends can be consulted, while agents further away (such as a friend's friend that is one's enemy) cannot be communicated with or cannot be trusted. In a social network, for example, the whole set of players other than friends might not even be known. The proposed games are compactly representable but not fully expressive. However, they can express other hedonic games than those representable by popular compact representations in the literature. We study both the axiomatic properties of these games and the computational complexity of problems related to common stability concepts.

From a *noncooperative* game-theoretic point of view, the interests of not only selfish, but altruistic agents have been modelled and studied by, for example, Hoefer and Skopalik [24], Chen et al. [14], Apt and Schäfer [2], and Rahn and Schäfer [30]. Salehi-Abari and Boutilier [31] study social choice with empathetic preferences. Their local empathetic model is related to our model. Altruism has also been studied in (experimental) economics [27]. Brânzei and Larson [12] study social distance games: In contrast to degrees of altruism as proposed here, a player's opinion on her friends (players of distance one) has the highest weight while her opinion on players farther away counts less. This is similar (but not equivalent) to our selfish-first model to be defined in Section 3.2.

Furthermore, the study of other agents' influence on opinions has gained increasing interest in collective decision making [22, 23]. In the context of voting scenarios, preference extensions and their properties have been studied by Endriss [20]. An overview of axiomatic properties of preference orders can be found, e.g., in the book chapters by Barberà et al. [9] and Lang and Rothe [26]. The work by Darmann et al. [15] combines aspects of voting theory and the theory of coalition formation games: They define a model for selection scenarios for a number of group activities that can also be represented by hedonic games and they study the complexity of stability concepts in this model. Aziz et al. [5] provide a survey of known results for additively separable hedonic games; in particular, two impressive complexity results—$\mathrm{NP}^{\mathrm{NP}}$-completeness for the existence of (strictly) core-stable coalition structures—are due to Woeginger [34] and Peters [28]. The concept of Pareto optimality has been studied by Aziz et al. [3] for a number of encodings of hedonic games, while recent work of Brandl et al. [11] is concerned with the complexity of various stability concepts in fractional hedonic games [4]. Lang et al. [25] introduce a new type of hedonic game where agents rank their friends and their enemies (and may, in addition, feel "neutral" about some other players), and these preferences over players are extended to preferences over coalitions.

2 Preliminaries

A *hedonic game* is a pair $(N, \succeq)$, where $N = \{1, \ldots, n\}$ is a set of players and $\succeq\ = (\succeq_1, \ldots, \succeq_n)$ is a list of the players' preferences. For $i \in N$, let $\mathcal{N}^i = \{C \subseteq N \mid i \in C\}$ denote the set of coalitions containing i. Player i's preference relation $\succeq_i\ \in \mathcal{N}^i \times \mathcal{N}^i$ induces a complete, weak preference order over $\mathcal{N}^i$. For $A, B \in \mathcal{N}^i$, we say that player i *weakly prefers* A *to* B if $A \succeq_i B$, that i *prefers* A *to* B ($A \succ_i B$) if $A \succeq_i B$ but not $B \succeq_i A$, and that i is *indifferent between* A *and* B ($A \sim_i B$) if $A \succeq_i B$ and $B \succeq_i A$. We call $C \in \mathcal{N}^i$ *acceptable for player* i if $C \succeq_i \{i\}$. A *coalition structure* is a partition $\Gamma = \{C_1, \ldots, C_k\}$ of the players into k coalitions $C_1, \ldots, C_k \subseteq N$ (i.e., $\bigcup_{r=1}^{k} C_r = N$ and $C_r \cap C_s = \emptyset$ for all r and s, $1 \leq r \neq s \leq k$). The unique coalition in Γ containing player $i \in N$ is denoted by $\Gamma(i)$.

In order to avoid exponentially large preference orders in the number of players, a common way to represent players' preferences is to consider a network of friends [16]. Each player $i \in N$ has a set of friends $F_i \subseteq N \setminus \{i\}$ and a set of enemies $E_i = N \setminus (F_i \cup \{i\})$. Visually, let the players in $N = \{1, \ldots, n\}$ be represented by the vertices in a graph $G = (N, H)$, and let a directed edge $(i, j) \in H$ denote that j is i's friend, that is, the open neighborhood of i represents the set of i's friends $F_i = \{j \mid (i, j) \in H\}$. Since in the context of stability it is reasonable to consider symmetric friendship relations [33], we will focus on undirected graphs representing networks of friends. In the *friend-oriented* preference extension [16] more friends are preferred to fewer friends, and in case of an equal number of friends, fewer enemies are preferred. Formally, define

$$
\begin{aligned}
A \succeq_i^F B \iff & |A \cap F_i| > |B \cap F_i| \text{ or} \\
& (|A \cap F_i| = |B \cap F_i| \text{ and } |A \cap E_i| \leq |B \cap E_i|).
\end{aligned} \tag{1}
$$

Note that friend-oriented preferences can be represented additively, by assigning a value of $n = |N|$ to each friend and a value of -1 to each enemy [16]: For any player $i \in N$ and for any coalition $A \in \mathcal{N}^i$, define the *value of a coalition* by

$$v_i(A) = n|A \cap F_i| - |A \cap E_i|.$$

Then, for $A, B \in \mathcal{N}^i$, we have $A \succeq_i^F B \iff v_i(A) \geq v_i(B)$.

Relatedly, other representations and their preference extensions are enemy-oriented preferences [16], additively separable [32] and fractional hedonic games [4], and singleton encodings [13] (which each are compactly representable but not fully expressive or complete), individually rational encodings [7], and hedonic coalition nets [19] (which are fully expressive but not compact in the sense of polynomial-size representation).

See Section 4 for a discussion of how our models differ from representations known from the literature.

2.1 Properties of Preference Extensions

Below we give a selection of properties of preference extensions inspired by various related topics such as voting theory and resource allocation. Let $N = \{1, \dots, n\}$ be a set of players and F_i and E_i the sets of player i's friends and enemies, respectively. Let $G = (N, H)$ be the corresponding network of friends. Consider player i's preference relation $\succeq_i$ on $\mathcal{N}^i$. We say $\succeq_i$ is *reflexive* if $A \succeq_i A$ for each coalition $A \in \mathcal{N}^i$; $\succeq_i$ is *transitive* if for any three coalitions $A, B, C \in \mathcal{N}^i$, $A \succeq_i B$ and $B \succeq_i C$ implies $A \succeq_i C$; $\succeq_i$ is *polynomial-time computable* if for a given player i and two given coalitions $A, B \in \mathcal{N}^i$, it can be decided in polynomial time whether or not $A \succeq_i B$; and $\succeq_i$ is *anonymous* if renaming the players in N does not change $\succeq_i$. Clearly, the first three properties are necessary to have efficiently computable and rational preferences, and anonymity means that only the structure of the friendship network is important. We further define the following properties.

Weak Friend-Orientedness: If coalition A is acceptable for i, then $A \cup \{f\}$ is also acceptable for i, where $f \in F_i \setminus A$.
Favoring Friends: If $x \in F_i$ and $y \in E_i$ then $\{x, i\} \succ_i \{y, i\}$.
Indifference between Friends: If $x, y \in F_i$ then $\{x, i\} \sim_i \{y, i\}$.
Indifference between Enemies: If $x, y \in E_i$ then $\{x, i\} \sim_i \{y, i\}$.

Note that these four properties hold for friend-oriented preferences, see the work of Alcantud and Arlegi [1].

Sovereignty of Players: For a fixed player i and each $C \in \mathcal{N}^i$, there exists a network of friends such that C ends up as i's most preferred coalition.
Monotonicity: Let $j \neq i$ be a player with $j \in E_i$ and $A, B \in \mathcal{N}^i$, and $\succeq_i'$ be the preference relation resulting from $\succeq_i$ when j turns from being i's enemy to being i's friend (all else being equal). We call $\succeq_i$ *type-I-monotonic* if it holds that (1) if $A \succ_i B$, $j \in A \cap B$, and $A \succeq_j^F B$, then $A \succ_i' B$, and (2) if $A \sim_i B$, $j \in A \cap B$, and $A \succeq_j^F B$, then $A \succeq_i' B$. We call $\succeq_i$ *type-II-monotonic* if it holds that (1) if $A \succ_i B$ and $j \in A \setminus B$, then $A \succ_i' B$, and (2) if $A \sim_i B$ and $j \in A \setminus B$, then $A \succeq_i' B$.
Type-I-monotonicity ensures i's preference of A over B not to become worse if an enemy j who is contained in both coalitions, turns into i's friend, while j is weakly preferring A to B. Type-II-monotonicity, on the other hand, requires that j is only in A (hence has no opinion on B), but still i's preference of A over B should not become worse.
Symmetry: Let j and k be two distinct players with $j \neq i \neq k$. We say that $\succeq_i$ is *symmetric* if it holds that if swapping the positions of j and k in G is an automorphism then
$\left(\forall\, C \in \mathcal{N}^i \setminus (\mathcal{N}^j \cup \mathcal{N}^k)\right)\ [C \cup \{j\} \sim_i C \cup \{k\}]$.
Local Friend Dependence: The preference order $\succeq_i$ can depend on the sets of friends $F_1, \dots, F_n$. Let $A, B \in \mathcal{N}^i$. We say that comparison (A, B) is

– *friend-dependent in* $\succeq_i$ if (1) $A \succeq_i B$ is true (false) and (2) can be made false (true) by changing the set of friends of some players (except for i);
– *locally friend-dependent in* $\succeq_i$ if (1) $A \succeq_i B$ is true (false), (2) can be made false (true) by changing the set of friends of some players that are in A or B and are i's friends, and (3) changing the set of friends of all other players in $N \setminus (\{i\} \cup (F_i \cap (A \cup B))$ does not affect the status of the comparison.

We say $\succeq_i$ is *locally friend-dependent* if (1) there are $A, B \in \mathcal{N}^i$ such that (A, B) is friend-dependent in $\succeq_i$ and (2) every (A', B') that is friend-dependent in $\succeq_i$ is locally friend-dependent in $\succeq_i$.

This property says that an agent's preference over some coalition can change if the set of a friend's friends changes. This friend also has to be a member of a coalition that is under consideration. Thus local friend dependence is a crucial property that tries to capture the essence of the proposed approach to altruism in hedonic games.

Friend-Oriented Unanimity: Let $A, B \in \mathcal{N}^i$ with $A \cap F_i = B \cap F_i$. We say that $\succeq_i$ is *friend-orientedly unanimous* if $A \succ_j^F B$ for each $j \in (F_i \cup \{i\}) \cap A$ implies $A \succ_i B$.

Note that the definition of friend-oriented unanimity covers all cases where the same subset of friends is consulted who all have a unanimous opinion in terms of friend-oriented preferences, in particular the case considering all friends' opinions: $F_i \subseteq A \cap B$.

2.2 Stability Concepts

The following stability concepts are commonly studied in hedonic games.

Definition 1. *Let $(N, \succeq)$ be a hedonic game and Γ be a coalition structure. A coalition $C \subseteq N$* blocks Γ *if for each $i \in C$ it holds that $C \succ_i \Gamma(i)$. If there is at least one $i \in C$ with $C \succ_i \Gamma(i)$ while $C \succeq_j \Gamma(j)$ holds for the other players $j \neq i$ in C, we call C* weakly blocking. *A coalition structure Γ is said to be*

1. individually rational *if for all $i \in N$, $\Gamma(i)$ is acceptable;*
2. Nash-stable *if for all $i \in N$ and for each $C \in \Gamma \cup \{\emptyset\}$ with $\Gamma(i) \neq C$, it holds that $\Gamma(i) \succeq_i C \cup \{i\}$;*
3. individually stable *if for all $i \in N$ and for each $C \in \Gamma \cup \{\emptyset\}$, it either holds that $\Gamma(i) \succeq_i C \cup \{i\}$ or there is a player $j \in C$ with $C \succ_j C \cup \{i\}$;*
4. contractually individually stable *if for all $i \in N$ and for each $C \in \Gamma \cup \{\emptyset\}$, it either holds that $\Gamma(i) \succeq_i C \cup \{i\}$, or there is a player $j \in C$ with $C \succ_j C \cup \{i\}$, or there is a player $k \in \Gamma(i)$ with $i \neq k$ and $\Gamma(i) \succ_k \Gamma(i) \setminus \{i\}$;*
5. strictly popular *if it beats every other coalition structure $\Gamma' \neq \Gamma$ in pairwise comparison, that is, if $|\{i \in N \mid \Gamma(i) \succ_i \Gamma'(i)\}| > |\{i \in N \mid \Gamma'(i) \succ_i \Gamma(i)\}|$;*
6. (strictly) core-stable *if there is no (weakly) blocking coalition;*
7. perfect *if for all $i \in N$ and for all $C \in \mathcal{N}^i$, it holds that $\Gamma(i) \succeq_i C$.*

3 Altruistic Hedonic Games

In this section, we introduce our new model that refines friend-oriented hedonic games by taking altruistic influences into account. In this model, each player still wants to be with as many friends and as few enemies as possible, but in addition she wants her friends to be as satisfied as possible. We omit proofs due to space constraints.

3.1 Naïve Approach

A first attempt to formalize this idea (that will turn out to fail) is the following. Consider the scenario where $i \in N$ has a friend-oriented preference extension (according to Equivalence (1)) except that, whenever the number of friends in A and B is the same and so is the number of enemies in A and B (i.e., $A \sim_i^F B$), i now prefers A to B if more of i's friends that are contained in A and B prefer A to B than B to A (according to Equivalence (1)). Formally:

$$\begin{aligned} A \succeq_i^{NA} B \iff & |A \cap F_i| > |B \cap F_i| \text{ or} \\ & (|A \cap F_i| = |B \cap F_i| \text{ and } |A \cap E_i| < |B \cap E_i|) \text{ or} \\ & (|A \cap F_i| = |B \cap F_i| \text{ and } |A \cap E_i| = |B \cap E_i| \text{ and} \\ & \ |\{j \in A \cap B \cap F_i \mid A \succ_j^F B\}| \geq \\ & \ |\{j \in A \cap B \cap F_i \mid B \succ_j^F A\}|). \end{aligned} \tag{2}$$

Intuitively, according to (2), a player is selfish first, but as soon as she is indifferent between two coalitions in the sense of (1), she cares about her friends' preferences. A major disadvantage of this definition, however, is that *irrational* preference orders can arise, i.e., preference orders that are not transitive in general: Consider, e.g., the hedonic game $(N, \succeq^{NA})$ with $N = \{1,2,3,4,5,6,7\}$ and the network of friends shown in Figure 2a. For coalitions $A = \{1,2,3,5\}$, $B = \{1,2,4,7\}$, and $C = \{1,3,4,6\}$, it holds that $A \succ_1^{NA} B$ and $B \succ_1^{NA} C$, yet $C \succ_1^{NA} A$, violating transitivity.

(a) Used, e.g., in the proof of Proposition 3

(b) Illustrating distinct degrees of altruism in Example 1

Fig. 2: Two networks of friends representing hedonic games

In order to ensure transitivity, we have to add an extra condition to Equivalence (2). One idea would be to demand indifference between all coalitions that are involved in a $\succ_i^{NA}$-cycle by (2). This, however, can lead to a comparison of all coalitions containing a player, so determining a relation between two coalitions might comprise an exponential number of steps in the number of players. Then it would have been easier to give an arbitrary preference order as an input in the first place. Another idea would be to include the preferences of all friends, not only of those contained in the considered coalitions, but this would contradict the concept of hedonic game. In the following, we take a different approach.

3.2 Modelling Altruistic Influences

Given the failure of extending friend-oriented preferences by breaking ties with "majority voting," we consider the following model instead: Player $i \in N$ prefers coalition A over B if the average value of i's friends in A is larger than the average value of i's friends in B. In more detail, using the friend-oriented encoding, we obtain a friend j's opinion on a coalition containing both player i and j, which can have an influence on i's preference relation in the following ways. Since we consider friends to be equally important and focus on the average valuation, assigning a weight to player i's own contribution in comparison to her friends' influence on her preference, we will distinguish between three *degrees of altruism*: A player may (a) be selfish first and ask her friends only in case of indifference, (b) treat her friends and herself equally, or (c) be truly altruistic by asking her friends first and deciding herself only in case of indifference. Next to the definition we will show that the preferences capture the intuitive ideas behind them. For $i \in N$ and $A \in \mathcal{N}^i$, let

$$\mathrm{avg}_i^F(A) = \sum_{a \in A \cap F_i} \frac{v_a(A)}{|A \cap F_i|}.$$

In each of the three cases below, a player's *utility* u_i of a coalition is used as a measure of comparison combining the values v_i and v_j for $j \in F_i$. Note that $u_i = v_i$ under friend-oriented extensions.

(a) Selfish First: A player initially decides upon her preference over two coalitions friend-orientedly (i.e., according to (1)) and, if and only if she is indifferent between them, she asks her friends for a vote. For $M \geq n^5$, we define:

$$\begin{aligned} & A \succeq_i^{SF} B \iff \\ & M(n|A \cap F_i| - |A \cap E_i|) + \sum_{a \in A \cap F_i} \frac{n|A \cap F_a| - |A \cap E_a|}{|A \cap F_i|} \geq \\ & M(n|B \cap F_i| - |B \cap E_i|) + \sum_{b \in B \cap F_i} \frac{n|B \cap F_b| - |B \cap E_b|}{|B \cap F_i|}. \end{aligned} \tag{3}$$

Theorem 1. *For $M \geq n^5$, $v_i(A) > v_i(B)$ implies $A \succ_i^{SF} B$.*

(b) Equal Treatment: A player and her friends "vote" friend-orientedly at the same time, equally taking part in the decision:

$$\begin{aligned} A \succeq_i^{EQ} B \iff & \sum_{a \in A \cap (F_i \cup \{i\})} \frac{n|A \cap F_a| - |A \cap E_a|}{|A \cap (F_i \cup \{i\})|} \geq \\ & \sum_{b \in B \cap (F_i \cup \{i\})} \frac{n|B \cap F_b| - |B \cap E_b|}{|B \cap (F_i \cup \{i\})|}. \end{aligned} \tag{4}$$

(c) Altruistic Treatment: A player first asks her friends for their opinion on a coalition they are contained in and adopts their average opinion; if and only if the consensus is indifference, the player decides for herself. For $M \geq n^5$, we define:

$$
\begin{aligned}
& A \succeq_i^{AL} B \iff \\
& \quad n|A\cap F_i| - |A\cap E_i| + M \sum_{a\in A\cap F_i} \frac{n|A\cap F_a| - |A\cap E_a|}{|A\cap F_i|} \geq \\
& \quad n|B\cap F_i| - |B\cap E_i| + M \sum_{b\in B\cap F_i} \frac{n|B\cap F_b| - |B\cap E_b|}{|B\cap F_i|}.
\end{aligned} \tag{5}
$$

Theorem 2. *For $M \geq n^5$, $\mathrm{avg}_i^F(A) > \mathrm{avg}_i^F(B)$ implies $A \succ_i^{AL} B$.*

For consistency we choose $M \geq n^5$. In all three cases in the proof of Theorem 2, normalization by the number of i's friends in a coalition prevents a "tyranny of the many" (otherwise, large coalitions would be preferred merely by the fact that the total number of friends is larger). The following example represents the different approaches to altruism in hedonic games.

Example 1. Consider the game with five players $N = \{1,2,3,4,5\}$ and the network in Figure 2b. Table 1 gives an overview of the relevant values and average values needed to determine player 1's utilities for different acceptable coalitions depending on the degree of altruism. A dash indicates that a value does not exist.

C	$\{1,2,3\}$	$\{1,2,3,4\}$	$\{1,2,3,5\}$	N	$\{1,2\}$	$\{1,3\}$	$\{1,2,4\}$	$\{1,2,5\}$	$\{1,3,4\}$
$v_1(C)$	10	9	9	8	5	5	4	4	4
$v_2(C)$	4	3	9	8	5	–	4	10	–
$v_3(C)$	4	9	3	8	–	5	–	–	10
$\mathrm{avg}_1^F(C)$	4	6	6	8	5	5	4	10	10
EQ: $u_1(C)$	6	7	7	8	5	5	4	7	7

Table 1: Values and average values of the players in Example 1

All four weak preference orders are different. Under the friend-oriented preference extension (1), player 1's weak preference order is given in the first line according to the values of v_1. For the *selfish-first* extension (3), the order remains the same; however, indifferences are dissolved, as is the case here with $\{1,2,5\} \succ_1^{SF} \{1,2,4\}$ by Theorem 1. Under the *equal-treatment* extension (4), the grand coalition is the most preferred one; intuitively, because all friends have a large number of friends at the same time. Finally, under the most altruistic extension (5), player 1's friends consider $\{1,2,5\}$ and $\{1,3,4\}$ the best coalition. As they agree on that, player 1 altruistically adopts this opinion by Theorem 2.

A player's utility of a coalition can also be deduced from the corresponding network of friends itself:

Proposition 1. *Let G be a network of friends, i a player, and $C \in \mathcal{N}^i$ a coalition. Let λ be the number of edges $\{i, j\}$ where $j \in C$, i.e., $\lambda = |F_i \cap C|$. Let μ be the number of edges between i's friends in C, i.e., $\mu = |\{\{j,k\} \mid j,k \in F_i \cap C\}|$, and let ν denote the number of edges between i's friends in C and those friends of j in C who are i's enemies, i.e., $\nu = |\{\{j,k\} \mid j \in F_i \cap C,\ k \in F_j \cap C,\ k \notin F_i\}|$. Then i's utility of C under selfish-first preferences is*

$$M \cdot \lambda(n+1) + M + n + 2 - (M+1)|C| + \frac{(n+1)(2\mu+\nu)}{\lambda};$$

under equal-treatment preferences it is

$$\frac{(2\lambda+2\mu+\nu)(n+1)}{\lambda+1} - |C| + 1;$$

and under altruistic-treatment preferences it is

$$M(n+2) + \lambda(n+1) + 1 - (M+1)|C| + \frac{M(n+1)(2\mu+\nu)}{\lambda}.$$

4 Properties of Hedonic Games with Altruistic Influences

In this section, we show which of the desirable properties from Section 2.1 are satisfied by our model. First, however, we start with a discussion of expressiveness, focusing on model (4):

First, as the original definition of friend-oriented preferences is recovered for coalitions that only consist of enemies, our models are not fully expressive. This follows from indifference between friends and enemies, respectively.

Second, we show that the expressiveness of model (4) is incomparable to (additively) separable hedonic games, fractional hedonic games, hedonic games with $\mathcal{B}$- or $\mathcal{W}$-preferences, and B- and W-hedonic games (see, for example, the book chapter by Aziz and Savani [6] for the definitions of these representations of hedonic games). In all of the above models two players' preference orders are independent but in our model they might depend on each other. Players are independent in choosing friends; however, the induced preferences depend crucially on friends' relations to other players. In other words, a player's preference is constrained by her friends' preferences. Hence, there is a tradeoff between the expressiveness of preferences and the expressiveness of profiles.

Third, model (4) can express preferences that are not separable: Consider a game with players $N = \{i,a,b,j\}$ and $F_i = \{a,b,j\}$, $F_a = \{i,b\}, F_b = \{i,a\}, F_j = \{i\}$. Denote the left-hand side of the inequality in (4) by $u_i^*(A) = \sum_{a' \in A \cap (F_i \cup \{i\})} \frac{n|A \cap F_{a'}| - |A \cap E_{a'}|}{|A \cap (F_i \cup \{i\})|}$. Then $u_i^*(\{i,a,b\}) = 2n$ and $u_i^*(\{i,a,b,j\}) = 2n-1$. Thus $\{i,a,b\} \succ_i^{EQ} \{i,a,b,j\}$ but $\{i,j\} \succ_i^{EQ} \{i\}$, because j is a friend. However, additively separable preferences can express strict preferences over coalitions with a single friend, which is not possible in model (4) because of indifference between friends. Similarly, fractional preferences can express strict preferences over pairs. In addition, they can express nonseparable relations by losing the ability to express indifference between pairs. $\mathcal{B}$- and $\mathcal{W}$-preferences can express indifference between pairs but this constraints the preferences over larger coalitions. In this case, however, depending on the network of friends, model (4) can express every possible relation between some specific coalitions. For B- and W-hedonic games, there is a simple example with two coalitions of size two only one of which is acceptable, where the implied relation over the coalition with both players does not hold under model (4).

Overall, neither is model (4) more expressive than any of the other considered models nor the other way around. Similar examples also exist for the other two models. It is not hard to see that all three degrees of altruism are locally friend-dependent, because (except for player i's own preference) only her friends in the current coalition affect its value. Note that this is a crucial property that distinguishes our model from previous work.

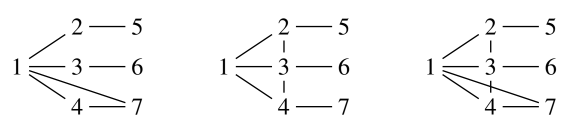
Proposition 2. *Under all three degrees of altruism* (3)–(5)*, the following properties are satisfied: reflexivity, transitivity, polynomial-time computability, as well as anonymity.*

Theorem 3. *Under all three degrees of altruism* (3)–(5)*, weak friend-orientedness, favoring friends, indifference between friends, indifference between enemies, sovereignty of players, symmetry, and friend-oriented unanimity are satisfied.*

Regarding the property of symmetry, note that whenever two interchangeable players have a distance of at most two, then the statement $\left(\forall\, C \in \mathcal{N}^i \setminus (\mathcal{N}^j \cup \mathcal{N}^k)\right)$ $[C \cup \{j\} \sim_i C \cup \{k\}]$ implies that swapping i and j in G is an automorphism.

Theorem 4. *Selfish-first preferences* (3) *are type-I-monotonic and type-II-monotonic.*

Proposition 3. *Equal-treatment preferences* (4) *and altruistic-treatment preferences* (5) *are not type-II-monotonic.*



(a) Network of $\mathcal{G}_1'$ (b) Network of $\mathcal{G}_2$ (c) Network of $\mathcal{G}_2'$

Fig. 3: Network of friends in the proof of Proposition 3

Note that the above result is a desirable outcome since this behavior exactly captures the intuition behind the definition of the equal treatment and the altruistic treatment.

Based on a similar line of thought we note the following: In addition to the axiomatic properties in Section 2.1, one could consider notions of independence (see, e.g., [1] for a characterization of friend-oriented preferences using an independence axiom). Classic independence axioms say that a relation between two coalitions, A and B, continues to hold even if a new (and the same) player is introduced to both coalitions. However, independence axioms of this type are not desirable in our model because the new player can be valued very differently in both coalitions. This would be the case, for example, if the new player were an enemy to most of i's friends in A but were loved by most of i's friends in B. Similarly, $\mathcal{B}$- and $\mathcal{W}$-preferences [13] are natural extensions from singleton encodings that are not independent.

5 Stability

In this section, we study several common stability concepts in our model. Questions of interest are how hard it is to verify whether a given coalition structure satisfies a certain concept in a given hedonic game and whether stable coalition structures for certain concepts always exist. If, for some concept, such a coalition structure does not always exist, we are also interested in the computational complexity of deciding whether or not some such coalition structure exists in a given hedonic game. Recently, Peters and Elkind [29] established metatheorems that help proving NP-hardness results for stability concepts in hedonic games. However, their results do not seem to be immediately applicable.

Observation 1. *Under all three degrees of altruism* (3)–(5)*, a coalition structure* Γ *is individually rational if and only if for each* $i \in N$*,* $\Gamma(i) \cap F_i \neq \emptyset$ *or* $\Gamma(i) = \{i\}$*.*

Proposition 4. *For all three degrees of altruism* (3)–(5)*, it can be tested in polynomial time whether a given coalition structure in a given game is Nash-stable, individually stable, or contractually individually stable.*

Lemma 1. *For all three degrees of altruism the following hold:*

1. For each i, each $j \in F_i$ *assigns a positive value to any coalition* $C \in \mathcal{N}^i \cap \mathcal{N}^j$*.*

2. If a player has at least one friend, her favorite coalition contains at least one friend.

Theorem 5. *For all three degrees of altruism* (3)–(5)*, there always exist Nash-stable, individually stable, and contractually individually stable coalition structures.*

On the other hand, for all three degrees of altruism, there exists a game such that no coalition structure is strictly popular.

Example 2. Consider Example 1 and the coalition structures $\Gamma_1 = \{\{1,2,5\},\{3,4\}\}$, $\Gamma_2 = \{\{1,3,4\},\{2,5\}\}$,$\Gamma_3 = \{\{1,2,3,4\},\{5\}\}$, $\Gamma_4 = \{\{1,2,3,5\},\{4\}\}$, and $\Gamma_5 = \{N\}$.

1. Under selfish-first preferences (3), Γ_1 and Γ_2 are more popular than all other coalition structures, but are in a tie.

2. Under equal-treatment preferences (4), even three coalition structures are in a tie: Γ_3, Γ_4, and Γ_5.

3. Under altruistic-treatment preferences (5), Γ_2 is more popular than Γ_3, which in turn is more popular than Γ_5. Γ_5 and Γ_2 are in a tie. Further, Γ_1 is more popular than Γ_4; the two coalition structures behave analogously to Γ_2 and Γ_3, respectively, due to symmetries. There is no other coalition structure that is not beaten by any of the above-mentioned coalition structures. Hence, no coalition structure is strictly popular.

We now turn to the complexity of the verification and the existence problem for strict popularity in selfish-first hedonic games.

Theorem 6. *Under selfish-first preferences* (3)*, the problem of whether a given coalition structure in a given game is strictly popular is* coNP*-complete and the problem of whether there exists a strictly popular coalition structure in a given game is* coNP*-hard.*

Theorem 7 is inspired by a result of Dimitrov et al. [16].

Theorem 7. *In games with selfish-first preferences* (3)*, there always exists a (strictly) core-stable coalition structure.*

Under selfish-first preferences, it is easy to figure out whether there exists a perfect coalition structure: This is the case if and only if each connected component is a clique.

Lemma 2. *Let C be player i's most preferred coalition in a game with equal-treatment preferences* (4)*. If a friend j is in $F_i \cap C$, then $F_j \setminus F_i \subseteq C$.*

Proposition 5. *Whenever a perfect coalition structure exists under equal-treatment preferences* (4)*, it is unique and consists of all connected components.*

Corollary 1. *If there exists a perfect coalition structure under equal treatment* (4)*, all connected components have a diameter of at most two.*

There do exist networks with a diameter of at most two that do not allow a perfect coalition structure, e.g., stars (i.e., one central vertex connected to a number of leaves).

Proposition 6. *Under equal treatment* (4)*, trees with at least three vertices do not allow a perfect coalition structure.*

6 Conclusions and Future Work

We have introduced and studied hedonic games with altruistic influences where the agents' utility functions depend on their friends' preferences. Axiomatically, we have defined desirable properties and have shown that these are satisfied by our model, depending on the degree of altruism. When tailored to other well-studied preference models, such as friend-oriented, enemy-oriented, additively separable, and $\mathcal{B}$- and $\mathcal{W}$-preferences, we note that all of these five extension principles fulfill the introduced properties of anonymity, symmetry, and type-II-monotonicity, while only the former three satisfy independence.

In terms of stability, hedonic games with altruistic influences always admit, e.g., Nash-stable coalition structures. However, both the verification and the existence problems of strictly popular coalition structures are computationally intractable.

We consider it important future work to completely characterize when certain properties hold or stable coalition structures exist (e.g., to characterize when the grand coalition is perfect). Also, it might be useful to extend the model and normalize by the size of the coalition to consider only relative contributions of friend-of-a-friend relationships. This can be compared to a friend-oriented restriction of a fractional hedonic game [4]. For example, one could define

$$\begin{aligned} A \succeq_i^{EQ_f} B \iff & \sum_{a \in A \cap (F_i \cup \{i\})} \frac{n|A \cap F_a| - |A \cap E_a|}{|A| \cdot |A \cap (F_i \cup \{i\})|} \\ & \geq \sum_{b \in B \cap (F_i \cup \{i\})} \frac{n|B \cap F_b| - |B \cap E_b|}{|B| \cdot |B \cap (F_i \cup \{i\})|}. \end{aligned} \tag{6}$$

This definition clearly extends to the altruistic case. For the selfish-first case, the normalization is without effect. Hence, the fractional variant is equivalent to the selfish-first case we have considered.

In addition, we propose considering restrictions of the input such as constraining networks to special graph classes (such as interval graphs, where the width of an interval represents an agent's "tolerance"), studying problems of strategic influence (e.g., misreporting preferences to friends, pretending to be a friend while one in fact is an enemy, asserting control over the game as a whole). The model can be extended in multiple ways. To model more realistic situations, it would be suitable to allow for different degrees of altruism for distinct players and other representations of preferences and aggregators. So far, a player takes only her friends' preferences into consideration, that is, a player tries to satisfy her friends with respect to their preferences. Since players derive utility based on their own preferences and their friends' preferences, an interesting model would be to consider players that try to maximize their friends utilities (see, e.g., [31]). In a similar vein, the model can be extended to edge-weighted graphs, where the influence of a friend (or of a friend's friend) diminishes with the distance as in, e.g., social distance games [12]. Moreover, a different representation of hedonic game with such a weight function or a priority order over friends may lead to a different utility function and interesting new models of altruism.

Acknowledgements

We thank the CoopMAS and AAMAS reviewers for their helpful comments and suggestions. This work was supported in part by DFG grant RO-1202/14-2.

References

1. J. Alcantud and R. Arlegi. An axiomatic analysis of ranking sets under simple categorization. *SERIEs*, 3(1–2):227–245, 2012.
2. K. Apt and G. Schäfer. Selfishness level of strategic games. In *Proc. SAGT'12*, pages 13–24. Springer-Verlag *LNCS #7615*, 2012.
3. H. Aziz, F. Brandt, and P. Harrenstein. Pareto optimality in coalition formation. *Games and Economic Behavior*, 82:562–581, 2013.
4. H. Aziz, F. Brandt, and P. Harrenstein. Fractional hedonic games. In *Proc. AAMAS'14*, pages 5–12. IFAAMAS, 2014.
5. H. Aziz, F. Brandt, and H. Seedig. Computing desirable partitions in additively separable hedonic games. *Artificial Intelligence*, 195:316–334, 2013.
6. H. Aziz and R. Savani. Hedonic games. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 15. Cambridge University Press, 2016. To appear.
7. C. Ballester. NP-completeness in hedonic games. *Games and Economic Behavior*, 49(1):1–30, 2004.
8. S. Banerjee, H. Konishi, and T. Sönmez. Core in a simple coalition formation game. *Social Choice and Welfare*, 18(1):135–153, 2001.

9. S. Barberà, W. Bossert, and P. Pattanaik. Ranking sets of objects. In S. Barberà, P. Hammond, and C. Seidl, editors, *Handbook of Utility Theory*, volume 2: Extensions, chapter 17, pages 893–977. Kluwer Academic Publishers, 2004.
10. A. Bogomolnaia and M. Jackson. The stability of hedonic coalition structures. *Games and Economic Behavior*, 38(2):201–230, 2002.
11. F. Brandl, F. Brandt, and M. Strobel. Fractional hedonic games: Individual and group stability. In *Proc. AAMAS'15*, pages 1219–1227. IFAAMAS, 2015.
12. S. Brânzei and K. Larson. Social Distance Games. *Proc. IJCAI'11*, pages 91–96, AAAI Press/IJCAI, 2011.
13. K. Cechlárová and A. Romero-Medina. Stability in coalition formation games. *International Journal of Game Theory*, 29(4):487–494, 2001.
14. P. Chen, D. de Keijzer, D. Kempe, and G. Schäfer. The robust price of anarchy of altruistic games. In *Proc. WINE'11*, pages 383–390. Springer-Verlag *LNCS #7090*, 2011.
15. A. Darmann, E. Elkind, S. Kurz, J. Lang, J. Schauer, and G. Woeginger. Group activity selection problem. In *Proc. WINE'12*, pages 156–169. Springer Verlag *LNCS #7695*, 2012.
16. D. Dimitrov, P. Borm, R. Hendrickx, and S. Sung. Simple priorities and core stability in hedonic games. *Social Choice and Welfare*, 26(2):421–433, 2006.
17. J. Drèze and J. Greenberg. Hedonic coalitions: Optimality and stability. *Econometrica*, 48(4):987–1003, 1980.
18. E. Elkind and J. Rothe. Cooperative game theory. In J. Rothe, editor, *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, chapter 3, pages 135–193. Springer-Verlag, 2015.
19. E. Elkind and M. Wooldridge. Hedonic coalition nets. In *Proc. AAMAS'09*, pages 417–424. IFAAMAS, 2009.
20. U. Endriss. Sincerity and manipulation under approval voting. *Theory and Decision*, 74(3):335–355, 2013.
21. M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.
22. S. Ghosh and F. Velázquez-Quesada. Agreeing to agree: Reaching unanimity via preference dynamics based on reliable agents. In *Proc. AAMAS'15*, pages 1491–1499. IFAAMAS, 2015.
23. U. Grandi, E. Lorini, and L. Perrussel. Propositional opinion diffusion. In *Proc. AAMAS'15*, pages 989–997. IFAAMAS, 2015.
24. M. Hoefer and A. Skopalik. Altruism in atomic congestion games. In *Proc. ESA'09*, pages 179–189. Springer-Verlag *LNCS #5757*, 2009.
25. J. Lang, A. Rey, J. Rothe, H. Schadrack, and L. Schend. Representing and solving hedonic games with ordinal preferences and thresholds. In *Proc. AAMAS'15*, pages 1229–1238. IFAAMAS, 2015.
26. J. Lang and J. Rothe. Fair division of indivisible goods. In J. Rothe, editor, *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, chapter 8, pages 493–550. Springer-Verlag, 2015.
27. D. Levine. Modeling altruism and spitefulness in experiments. *Review of Economic Dynamics*, 1(3):593–662, 1998.
28. D. Peters. Σ_2^p-complete problems on hedonic games. Technical Report arXiv:1509.02333v1 [cs.GT], Computing Research Repository (CoRR), Sept. 2015.
29. D. Peters and E. Elkind. Simple causes of complexity in hedonic games. In *Proc. IJCAI'15*, pages 617–623. AAAI Press/IJCAI, 2015.
30. M. Rahn and G. Schäfer. Bounding the inefficiency of altruism through social contribution games. In *Proc. WINE'13*, pages 391–404. Springer-Verlag *LNCS #8289*, 2013.
31. A. Salehi-Abari and C. Boutilier. Empathetic social choice on social networks. In *Proc. AAMAS'14*, pages 693–700. IFAAMAS, 2014.

32. S. Sung and D. Dimitrov. Computational complexity in additive hedonic games. *European Journal of Operational Research*, 203(3):635–639, 2010.
33. G. Woeginger. Core stability in hedonic coalition formation. In *Proc. SOFSEM'13*, pages 33–50. Springer-Verlag *LNCS #7741*, 2013.
34. G. Woeginger. A hardness result for core stability in additive hedonic games. *Mathematical Social Sciences*, 65(2):101–104, 2013.

Local Fairness in Hedonic Games via Individual Threshold Coalitions

Nhan-Tam Nguyen and Jörg Rothe

Heinrich-Heine-Universität Düsseldorf, Düsseldorf, Germany
{nguyen, rothe}@cs.uni-duesseldorf.de

Abstract. Hedonic games are coalition formation games where players only specify preferences over coalitions they are part of. We introduce and systematically study local fairness notions in hedonic games by suitably adapting fairness notions from fair division. In particular, we introduce three notions that assign to each player a threshold coalition that only depends on the player's individual preferences. A coalition structure (i.e., a partition of the players into coalitions) is considered locally fair if all players' coalitions in this structure are each at least as good as their threshold coalitions. We relate our notions to previously studied concepts and show that our fairness notions form a proper hierarchy. We also study the computational aspects of finding threshold coalitions and of deciding whether fair coalition structures exist in additively separable hedonic games. At last, we investigate the price of fairness.

Keywords: coalition formation; hedonic games; fairness; game theory

1 Introduction

Coalition formation plays a crucial role in multiagent systems when agents have to cooperate. A commonly studied model of coalition formation is the model of hedonic game. These are coalition formation games with nontransferable utility, which were first studied by Drèze and Greenberg [15] and later on by Banerjee et al. [7] and Bogomolnaia and Jackson [8]. A key feature of hedonic games is that the players' preferences depend only on coalitions they are part of. Since players specify their preferences over an exponential-size domain (in the number of players), various compact representations have been proposed, which either are fully expressive but may still have an exponential size in the worst case or restrict the preference domain, e.g., [6, 18, 8, 2, 22]. Most of these studies are concerned with stability issues. Intuitively, they capture incentives of (groups of) players to deviate by joining a different coalition so as to increase their individual utility values. Thus stability-related questions address a decentralized aspect of hedonic games.

A more recent approach to hedonic games is welfare maximization [11, 3, 4]. This idea is different because welfare maximization usually presupposes a central authority guiding the maximization process by eliciting preferences and suggesting or enforcing an optimal solution. This enforcement may be necessary because the optimality of a solution is determined by a global criterion, such as utilitarian or egalitarian social welfare, and may affect some players' utility values negatively compared with the status quo.

In this paper we will focus on the concept of local fairness. Fairness is an important aspect besides stability and efficiency (see the related work section and, e.g., [14] for a discussion of fairness in multiagent systems and [9, 23] for fair division of indivisible goods). The only work that we are aware of considering fairness in hedonic games is due to Bogomolnaia and Jackson [8], Aziz et al. [3], Wright and Vorobeychik [26], and Peters [24, 25]. Fairness is related to both centralized approaches and stability issues. On the one hand, the center may want to ensure a certain utility level for each player. This goal can be achieved by a global fairness condition. However, fairness does not per se presuppose the existence of a center. On the other hand, players may not consider their current coalition fair, given their individual preferences. While we agree with Bogomolnaia and Jackson [8] that stability has a " 'restricted fairness' flavor," we add that one can also take the complementary view that lack of fairness can be a major cause of instability.

To make this more concrete, consider a situation where all players except a single player in some coalition consider this coalition their favorite one, yet for that single player this coalition is actually only marginally better than being alone. However, because everyone else prefers this coalition and thus is much better off than that player, she rejects this coalition. This can be considered an unfair situation and is comparable to an ultimatum game situation, where the proposal is very imbalanced and the second player (responder) rejects the proposal because it is below her fair share (see, again, [14]). Note that we would have to contrast the single player's utility to the other players' utility values in order to explain the predicament. This approach of balanced utility values and inequality reduction requires either a center that knows all players' utility values or that players look at coalitions (and even other players' well-being) outside of their own. To some extent, however, this is at odds with the idea of hedonic game because players in such games should only be interested in their own coalition.

The traditional fairness notion of envy-freeness requires players to inspect other coalitions. If there is a large number of coalitions, that is something we would like to avoid. Therefore, we propose and study notions of *local* fairness—restricted fairness notions with the additional constraint that players only compare their current coalition to some bound that *solely* depends on their individual preferences.[1] We feel that this is in the general spirit of the decentralized aspect of hedonic games.

Contribution: In order to achieve this goal of local fairness criteria, we introduce new fairness notions for hedonic games that are inspired by ideas from the field of fair division of indivisible goods. Our main contributions are the following:

1. We introduce the idea of local fairness and three specific such fairness notions in hedonic games. We show that these concepts form a (strict) hierarchy and relate them to previously studied concepts. Surprisingly, the hierarchy strikingly differs from the scale proposed by Bouveret and Lemaître [10] in the context of fair division of indivisible goods.
2. We systematically study the complexity of finding "threshold coalitions" and of determining whether a fair coalition structure exists in an additively separable hedonic game. We also find that two of our notions coincide in such games.

[1] In case of envy-freeness the bound would also depend on the whole coalition structure.

3. We initiate the study of price of fairness in hedonic games. In addition, we strengthen a result by Brânzei and Larson [11] on coalition structures maximizing social welfare in symmetric additively separable hedonic games.

Related Work: Surveys and book chapters on hedonic games are, for example, due to Aziz and Savani [5], and Elkind and Rothe [17]. Bogomolnaia and Jackson [8] already mention envy-freeness in their work, but they focus on studying stability notions. Aziz et al. [3] study the complexity of determining the existence of stable coalition structures in additively separable hedonic games. They also consider the welfare maximization approach and the notion of envy-freeness. The work by Wright and Vorobeychik [26] is related to ours. They study hedonic games under the perspective of mechanism design and propose mechanisms for solving the team formation problem. A key difference is that they consider additively separable hedonic games with nonnegative values only. Since in this case the grand coalition is most preferred by every player, Wright and Vorobeychik introduce cardinality constraints on feasible coalition sizes. They also consider *envy bounded by a single teammate*, which for the aforementioned reasons is not suitable for our goals. In addition, they introduce the *maximin share guarantee for team formation*, which is based on the idea of replacing players. This, however, leads to a provably different notion than ours (see Theorem 6). More recently introduced stability notions include *strong Nash stability*, proposed by Karakaya [21], and *strictly strong Nash stability*, due to Aziz and Brandl [1]. Brânzei and Larson [11] study social welfare maximization and core stability in additively separable hedonic games. Moreover, they consider the so-called stability gap.

Elkind et al. [16] investigate the price of Pareto optimality in various representations of hedonic games. Peters [25] considers restrictions of hedonic games that admit fast algorithms and he models allocating indivisible goods as hedonic game.

Surveys and book chapters on fair division are due, for example, to Lang and Rothe [23], and Bouveret et al. [9]. Budish [12] introduces the max-min fair-share criterion, whereas Bouveret and Lemaître [10] introduce min-max fair share and propose a scale of even more demanding fairness criteria. Caragiannis et al. [13] study the price of fairness in fair division.

Organization of the Paper: In Sect. 2, we formally define hedonic games and relevant notions of stability. In Sect. 3, we introduce our notions of fairness and relate them to other stability, fairness, and optimality concepts. In Sect. 4, we study our notions in additively separable hedonic games under computational aspects. The price of fairness is considered in Sect. 5, followed by a discussion of our findings and the conclusions in Sect. 6. We omit most proofs due to space constraints.

2 Preliminaries

We denote by $N = \{1,\dots,n\}$ the set of *players*. A *coalition* is a subset of N and a *coalition structure* π is a partition of N. The set of all coalition structures over N is $\Pi(N)$. We denote by $\pi(i)$ the unique coalition with player i in coalition structure π and by $\mathscr{N}_i = \{C \subseteq N \mid i \in C\}$ all coalitions that player i is part of. Every player i has a weak and complete preference order $\succeq_i$ over $\mathscr{N}_i$. For $A, B \in \mathscr{N}_i$, we write $A \succeq_i B$ if player i

weakly prefers coalition A to B; we write $A \succ_i B$ if player i *(strictly) prefers* coalition A to B, i.e., $A \succeq_i B$ but not $B \succeq_i A$; and we write $A \sim_i B$ if $A \succeq_i B$ and $B \succeq_i A$ (i.e., i is indifferent between A and B). Let $\succeq$ be the collection of all $\succeq_i$, $i \in N$. A *hedonic game* is a pair $(N, \succeq)$. It is an *additively separable hedonic game* (ASHG) if for every $i \in N$, there is a valuation function $v_i : N \to \mathbb{Q}$ such that $\sum_{j \in A} v_i(j) \geq \sum_{j \in B} v_i(j) \iff A \succeq_i B$. We write (N, v) for an additively separable hedonic game, where v is the collection of all v_i, $i \in N$. We assume normalization of the valuation functions, that is, $v_i(i) = 0$. We overload the notation to mean $v_i(A) = \sum_{j \in A} v_i(j)$ for each coalition $A \in \mathscr{N}_i$.

Now we define previously studied notions of stability that are relevant for this work. We distinguish, as is common, between group deviations, individual deviations, and other notions.

We consider the following notions of group deviations:

(1) A nonempty coalition $C \subseteq N$ *blocks* a coalition structure π if every $i \in C$ prefers C to $\pi(i)$. A coalition structure π is *core-stable* (CS) if no coalition blocks π.

(2) A coalition $C \subseteq N$ *weakly blocks* a coalition structure π if every $i \in C$ weakly prefers C to $\pi(i)$ and there is some $j \in C$ that prefers C to $\pi(j)$. A coalition structure π is *strictly core-stable* (SCS) if no coalition weakly blocks π.

(3) Given a coalition $H \subseteq N$, coalition structure π' is *reachable from coalition structure $\pi \neq \pi'$ by coalition H* if for all $i, j \in N \setminus H$, we have $\pi(i) = \pi(j) \iff \pi'(i) = \pi'(j)$. A nonempty coalition $H \subseteq N$ *weakly Nash-blocks* coalition structure π if there exists some coalition structure π' that is reachable from π by coalition H such that every $i \in H$ weakly prefers $\pi'(i)$ to $\pi(i)$ and there is some $j \in H$ that prefers $\pi'(j)$ to $\pi(j)$. We say π is *strictly strong Nash-stable* (SSNS) if there is no coalition that weakly Nash-blocks π.

As to individual deviations, we need the following definitions:

(1) A coalition structure π is *Nash-stable* (NS) if every $i \in N$ weakly prefers $\pi(i)$ to $C \cup \{i\}$ for every $C \in \pi \cup \{\emptyset\}$.

(2) A coalition structure π is *contractually individually stable* (CIS) if for every $i \in N$, the existence of a coalition $C \in \pi \cup \{\emptyset\}$ with $C \cup \{i\} \succ_i \pi(i)$ implies that there exists some $j \in C$ such that $C \succ_j C \cup \{i\}$ or there exists some $k \in \pi(i)$ such that $\pi(k) \succ_k \pi(k) \setminus \{i\}$.

Of the remaining notions we need the following:

(1) A coalition structure π is *perfect* if every $i \in N$ weakly prefers $\pi(i)$ to C for every $C \in \mathscr{N}_i$. In Figure 1 below, we write PERFECT to refer to this property.

(2) A coalition structure π' *Pareto-dominates* coalition structure π if every $i \in N$ weakly prefers $\pi'(i)$ to $\pi(i)$ and there is some $j \in N$ that prefers $\pi'(j)$ to $\pi(j)$. A coalition structure π is *Pareto-optimal* (PO) if no coalition structure Pareto-dominates it.

(3) A coalition structure π is *envy-free by replacement* (EF-R) if $\pi(i) \succeq_i (\pi(j) \setminus \{j\}) \cup \{i\}$ for every $i, j \in N$.

(4) A coalition $C \in \mathscr{N}_i$ is acceptable for $i \in N$ if $C \succeq_i \{i\}$. A coalition structure π is *individually rational* (IR) if $\pi(i)$ is acceptable for every $i \in N$.

The following notions are defined only for ASHGs (N, v): A coalition structure $\pi \in \Pi(N)$ maximizes

(1) *utilitarian social welfare* (USW) if for every $\pi' \in \Pi(N)$, $\sum_{i \in N} v_i(\pi(i)) \geq \sum_{i \in N} v_i(\pi'(i))$;

(2) *egalitarian social welfare* (ESW) if for every $\pi' \in \Pi(N)$, $\min_{i\in N} v_i(\pi(i)) \geq \min_{i\in N} v_i(\pi'(i))$.

For the last two definitions we make the common assumption (see, for example, [3, 11]) in coalition formation that values are interpersonally comparable.

We also say that a coalition structure π *satisfies some notion* X if π is X or maximizes X.

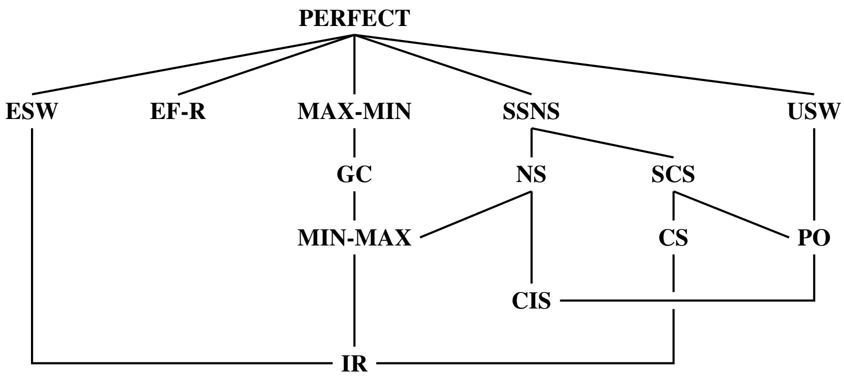


Fig. 1. Relations between notions. A line from notion A to a lower notion B means that every coalition structure that is A is also B. For example, every strictly core-stable (SCS) coalition structure is core-stable (CS) and Pareto-optimal (PO).

Fig. 1 shows the relationships between these notions. We refer to the surveys, book chapters, and papers mentioned in the related work section for more explanations of these definitions and their interrelations. The notions are chosen such that our separation results in the next section also apply to intermediate notions such as *contractual strict core stability* and *individual stability*.

3 Fairness in Hedonic Games

We consider fairness notions of the following form: Given $(N, \succeq)$, a coalition structure π is *f-locally fair* if $\pi(i) \succeq_i f(\succeq_i)$ for every $i \in N$. Note that f only depends on i's preferences. We start with the weakest fairness criterion. Individual rationality is the most basic notion of stability. It is also the weakest fairness criterion. Similarly to the example in the introduction, a player who is in a coalition not acceptable to her is exploited by the other players in that coalition if this coalition is acceptable to them. In other words, this player has to be in a disliked coalition just for other players to benefit. In this case, a coalition structure consisting of singletons is more preferable for this player. Lowering the bound of acceptable coalitions would make the situation even worse. Note that individual rationality and perfectness are examples of local fairness criteria that only propose a threshold coalition a player has to be part of, i.e., $f(\succeq_i) = \{i\}$

and $g(\succsim_i) = \max_{C \in \mathcal{N}_i} C$, respectively, where maximization is with respect to $\succsim_i$. In a sense, we look for criteria situated between these two extreme notions. Because all fairness criteria have to satisfy individual rationality necessarily, we consider such fairness criteria only. Following the definition of envy-freeness by replacement [8], it is not immediately clear which players to replace in the criteria that we will propose and how to motivate this. Therefore, we will focus on definitions that are based on players *joining* coalitions (without replacing any players). This is also another reason for why we do not consider the well-known fairness notion of envy-freeness (based on joining) because it coincides with Nash stability. Similarly, the maximin share guarantee for team formation by Wright and Vorobeychik [26] is defined in terms of replacing a player (and, therefore, is different from max-min fairness considered here).

3.1 Min-Max Fairness

Before we formally define the min-max threshold, we illustrate it with the following situation: A player is arriving late and all other players have already formed a coalition structure without her (where the specific form of the coalition structure is irrelevant for this argument). Because the player could not participate in the coalition formation process, the player is allowed to join any coalition. Clearly, this player joins her most preferred coalition. This describes a fairness criterion because someone who was neglected should be allowed to adapt to the situation in the best possible way.

Definition 1. *The* min-max threshold *of $i \in N$ is defined as*

$$\mathrm{MinMax}_i = \min_{\pi \in \Pi(N \setminus \{i\})} \max_{C \in \pi \cup \{\emptyset\}} C \cup \{i\},$$

where minimization and maximization are with respect to $\succsim_i$. A coalition structure π satisfies min-max fairness *(MIN-MAX) if*

$$\pi(i) \succsim_i \mathrm{MinMax}_i$$

for every $i \in N$.

This notion is the hedonic-games variant of min-max fair share, originally proposed by Bouveret and Lemaître [10] in fair division. We relate min-max fairness to previously known notions of stability. By definition, min-max fairness satisfies individual rationality. Clearly, since USW is not IR, it cannot satisfy min-max fairness. Similarly, EF-R, PO, and CIS cannot imply min-max fairness. Later (in Section 3.3 on max-min fairness) we will see that min-max fairness is independent of most stability notions in the sense that it does not imply them. Now, we check which stability notions except for perfectness imply min-max fairness.

Proposition 1. *A strictly core-stable coalition structure does not necessarily satisfy min-max fairness.*

Corollary 1. *An individually rational or core-stable coalition structure does not necessarily satisfy min-max fairness.*

On the other hand, Nash stability does imply min-max fairness.

Theorem 1. *Every Nash-stable coalition structure satisfies min-max fairness.*

Proof. Let π be a Nash-stable coalition structure and $i \in N$. Then $\pi(i) \succeq_i C \cup \{i\}$ for every $C \in \pi \cup \{\emptyset\}$. Since MinMax_i is a best coalition in a *worst* coalition structure for i, $\pi(i) \succeq_i \mathrm{MinMax}_i$. □

The following example shows that min-max fair coalition structures do not always exist (which is to be expected from any reasonable notion of fairness; envy-freeness is a classic fairness condition in fair division of indivisible goods, but in conjunction with completeness or Pareto optimality such partitions do not always exist either). This also shows that coalition structures that maximize egalitarian social welfare do not necessarily satisfy min-max fairness.

Example 1. Consider the following additively separable hedonic game, defined via the values $v_i(j)$:

$i \backslash j$	1	2	3
1	0	-10	15
2	-100	0	20
3	10	20	0

The individual min-max thresholds are $\mathrm{MinMax}_1 = 5$, $\mathrm{MinMax}_2 = 0$, and $\mathrm{MinMax}_3 = 20$. Therefore, player 1 has to be in coalition $\{1,3\}$ or $\{1,2,3\}$, player 2 in $\{2\}$ or $\{2,3\}$, and player 3 in $\{2,3\}$ or $\{1,2,3\}$. Hence, there is no min-max fair coalition structure.

3.2 Grand-Coalition Fairness

Bogomolnaia and Jackson [8] proposed the grand coalition as a notion of fairness. We recover their idea in the context of local fairness. It can be seen as a special variant of proportionality in the setting of hedonic games.

Definition 2. *The* grand-coalition threshold *of $i \in N$ is defined as*

$$\mathrm{GC}_i = \max\{\{i\}, N\},$$

where we maximize with respect to $\succeq_i$. A coalition structure satisfies grand-coalition fairness *(GC) if*

$$\pi(i) \succeq_i \mathrm{GC}_i$$

for every $i \in N$.

Grand-coalition fairness is a notion of fairness because the grand coalition can be interpreted as an average: Every player has to face both her friends and her enemies. Note that a proportionality threshold is typically defined as the ratio of the valuation for the whole to the number of players. Since players "share" their coalitions, it is not clear which number the valuation of the whole should be compared to. Comparing to the number of coalitions in a coalition structure, however, violates our locality requirement: thresholds that should only depend on a player's own preference.

First, we show that grand-coalition fairness is strictly stronger than min-max fairness.

Theorem 2. *Every grand-coalition fair coalition structure satisfies min-max fairness, yet a min-max fair coalition structure does not necessarily satisfy grand-coalition fairness.*

Proof. Let $i \in N$. Every coalition structure serves as an upper bound of MinMax_i. Consider the coalition structure $\{N\}$. Then $\max\{\{i\}, N\} \succeq_i \text{MinMax}_i$.

Conversely, consider the following hedonic game:

$$\begin{aligned}
&\{1,2\} \succ_1 \{1\} \succ_1 \{1,2,3\} \succ_1 \{1,3\}, \\
&\{1,2,3\} \succ_2 \{2,3\} \succ_2 \{1,2\} \succ_2 \{2\}, \\
&\{1,2,3\} \succ_3 \{2,3\} \succ_3 \{1,3\} \succ_3 \{3\}.
\end{aligned}$$

The players' min-max threshold coalitions are $\text{MinMax}_1 = \{1\}$, $\text{MinMax}_2 = \{2,3\}$, and $\text{MinMax}_3 = \{2,3\}$. Thus $\{\{1\},\{2,3\}\}$ satisfies min-max fairness but not grand-coalition fairness. □

It follows that a coalition structure that satisfies USW, ESW, EF-R, PO, or CIS does not necessarily satisfy grand-coalition fairness (otherwise it would satisfy min-max fairness). Later we will see that grand-coalition fairness is independent of all other considered notions except for perfectness. For now we show that these notions do not imply grand-coalition fairness.

Proposition 2. *A strictly strong Nash-stable coalition structure does not necessarily satisfy grand-coalition fairness.*

Corollary 2. *An individually rational, Nash-stable, core-stable, or strictly core-stable coalition structure does not necessarily satisfy grand-coalition fairness.*

3.3 Max-Min Fairness

We motivate the next fairness notion with the following situation: Suppose some player is allowed to partition all players excluding herself but does not know which coalition she will be part of in the end. Since she had the right to choose a partition, she has to live with all possible consequences. In other words, she could end up in any of these coalitions, even the worst. Therefore, a player would partition all remaining players so that the worst coalition among them is as good as possible for her.

Definition 3. *The* max-min threshold *of $i \in N$ is defined as*

$$\text{MaxMin}_i = \max_{\pi \in \Pi(N \setminus \{i\})} \max\{\{i\}, \min_{C \in \pi} C \cup \{i\}\},$$

where maximization and minimization are with respect to $\succeq_i$. A coalition structure π satisfies max-min fairness *(MAX-MIN) if*

$$\pi(i) \succeq_i \text{MaxMin}_i$$

for every $i \in N$.

Note that we cannot include the acceptability constraint into the minimization because then the definition would be weaker than IR. Max-min fairness is the hedonic-games variant of max-min fair share due to Budish [12]. We show that max-min fairness is strictly stronger than grand-coalition fairness.

Theorem 3. *Every max-min fair coalition structure satisfies grand-coalition fairness, yet a grand-coalition fair coalition structure does not necessarily satisfy max-min fairness.*

Proof. Let $i \in N$. The coalition structure π consisting of the grand coalition without i is the one where $\max_{C\in\pi\cup\{\emptyset\}} C \cup \{i\}$ and $\max\{\{i\}, \min_{C\in\pi} C \cup \{i\}\}$ become equal. Since every coalition structure gives a lower bound for MaxMin_i and an upper bound for MinMax_i, we have

$$\text{MaxMin}_i \succeq_i \text{GC}_i \succeq_i \text{MinMax}_i.$$

Conversely, consider the following hedonic game:

$$\begin{aligned}
&\{1,2\} \succ_1 \{1,3\} \succ_1 \{1,2,3\} \succ_1 \{1\},\\
&\{1,2,3\} \succ_2 \{1,2\} \succ_2 \{1,3\} \succ_2 \{2\},\\
&\{2,3\} \succ_3 \{1,2,3\} \succ_3 \{3\} \succ_3 \{1,3\}.
\end{aligned}$$

Coalition structure $\{\{1,2,3\}\}$ satisfies grand-coalition fairness but not max-min fairness because player 1's max-min threshold coalition is $\{1,3\}$. $\square$

Theorems 2 and 3 give additional motivation of grand-coalition fairness: It is strictly between max-min and min-max fairness. It follows that a USW, ESW, EF-R, PO, or CIS coalition structure does not necessarily satisfy max-min fairness.

Max-min fairness is independent of all other considered notions except for perfectness.

Proposition 3. *A max-min fair coalition structure does not necessarily satisfy contractually individual stability or core stability.*

Corollary 3. *1. A grand-coalition fair or min-max fair coalition structure does not necessarily satisfy contractually individual stability or core stability.*

2. A max-min fair, grand-coalition fair, or min-max fair coalition structure does not necessarily satisfy Nash stability, Pareto optimality, strictly strong Nash stability, strict core stability, utilitarian social welfare, or perfectness.

Proposition 4. *A max-min fair coalition structure does not necessarily satisfy envy-freeness by replacement or egalitarian social welfare.*

Corollary 4. *A grand-coalition fair or min-max fair coalition structure does not necessarily satisfy envy-freeness by replacement or egalitarian social welfare.*

From Proposition 2 we have the following corollary.

Corollary 5. *An individually rational, Nash-stable, core-stable, strictly strong Nash-stable, or strictly core-stable coalition structure does not necessarily satisfy max-min fairness.*

See Fig. 1 for a summary of the results of this section.

4 Local Fairness in ASHGs

In this section we study the existence of fair coalition structures, the complexity of computing fairness thresholds and of deciding whether a hedonic game admits a fair coalition structure. Since additively separable hedonic games are a well-studied (see [5] and the references therein) class of hedonic games, we will focus on this class. In addition, it will be easier to compare our complexity results to some results in fair division with additive utility functions. We begin with min-max fairness.

4.1 Min-Max Fairness

We start by computing min-max fairness thresholds. Since we have valuation functions in ASHGs, we can compare to the *value* of threshold coalitions. In particular, we consider the decision problem MIN-MAX-THRESHOLD: Given a set N of players, a player i's valuation function v, and a rational number k, does it hold that $\text{MinMax}_i \geq k$?

By considering coalition structures consisting of either the grand coalition or only of singletons, we have the following observations that show that MIN-MAX-THRESHOLD is easy to solve for certain restricted valuation functions.

Observation 1. *If $v_i(N) \leq 0$, then* $\text{MinMax}_i = 0$.

Observation 2. *If $v_i(j) \geq 0$ for every $j \in N$, then* $\text{MinMax}_i = \max_{j\in N} v(j)$.

For general valuation functions, however, we have this result:

Theorem 4. MIN-MAX-THRESHOLD *is* coNP-*complete.*

Coming now to the question of existence of fair coalition structures, we first define the decision problem that we study. The input of the problem MIN-MAX-EXIST consists of an additively separable hedonic game. The question is whether a min-max fair coalition structure exists. We start with a simple observation.

Observation 3. *If $v_i(j) \geq 0$ for every $i, j \in N$, then $\{N\}$ satisfies min-max fairness, i.e., min-max fair coalition structures always exist.*

We say an additively separable hedonic game is *symmetric* if $v_i(j) = v_j(i)$ for every $i, j \in N$. Since there always exist Nash-stable coalition structures in symmetric ASHGs [8], we have

Corollary 6. *Symmetric ASHGs always admit min-max fair coalition structures.*

For general additively separable hedonic games, we have NP-hardness as a lower bound and membership in Σ_2^p (the second level of the polynomial hierarchy) as an upper bound. We use a modified version of the game in Example 1 as a gadget.

Theorem 5. MIN-MAX-EXIST *is* NP-*hard and in* Σ_2^p.

4.2 Grand-Coalition & Max-Min Fairness

In this section, we can consider grand-coalition and max-min fairness at the same time because of the following result:

Theorem 6. *In additively separable hedonic games, for every $i \in N$ we have*

$$\mathrm{MaxMin}_i = \mathrm{GC}_i.$$

Define the threshold and existence problems for grand-coalition and max-min fairness analogously to MIN-MAX-THRESHOLD and MIN-MAX-EXIST. Since computing the value of the grand coalition is easy in additively separable hedonic games, we have

Corollary 7. MAX-MIN-THRESHOLD *and* GRAND-COALITION-THRESHOLD *are in* P.

However, checking whether there exists a grand-coalition fair or max-min fair coalition structure is hard.

Theorem 7. *The problems* GRAND-COALITION-EXIST *and* MAX-MIN-EXIST *are* NP-*complete.*

5 Price of Fairness

Now we study the price of fairness in additively separable hedonic games. Informally, the price of fairness captures the loss in social welfare of a worst (best) coalition structure that satisfies some fairness criterion. We denote by $\mathrm{SW}_G(\pi)$ the utilitarian social welfare of coalition structure π in an additively separable hedonic game $G = (N, v)$, that is, $\mathrm{SW}_G(\pi) = \sum_{i \in N} v_i(\pi(i))$. We omit G when it is clear from the context.

Definition 4. *Let $G = (N, v)$ be an additively separable hedonic game and let π^* denote a coalition structure maximizing utilitarian social welfare. Define the* maximum price of min-max fairness *by*

$$\text{Max-PoMMF}(G) = \max_{\pi \in \Pi(N), \pi \textit{ is min-max fair}} \frac{\mathrm{SW}(\pi^*)}{\mathrm{SW}(\pi)}$$

if there is some min-max fair $\pi \in \Pi(N)$ and $\mathrm{SW}(\pi) > 0$ for all min-max fair $\pi \in \Pi(N)$; by $\text{Max-PoMMF}(G) = 1$ *if* $\mathrm{SW}(\pi^*) = 0$ *and* $\mathrm{SW}(\pi) = 0$ *for some min-max fair $\pi \in \Pi(N)$; and by setting* $\text{Max-PoMMF}(G) = +\infty$ *otherwise.*

Define the minimum price of min-max fairness *(*Min-PoMMF*) analogously.*

Note that we have $\mathrm{SW}(\pi^*) \geq 0$ and $\mathrm{SW}(\pi) \geq 0$, where π^* maximizes utilitarian social welfare and π is min-max fair.

Because the grand coalition maximizes utilitarian welfare under nonnegative valuation functions, the minimum and maximum price of grand-coalition fairness is one. Since this bound is not really informative, we now make some suitable assumptions to strengthen our results. Elkind et al. [16] argue that these notions are only sensible if the set of coalition structures that we consider is large enough. Because of that we only consider min-max fairness, the weakest fairness notion, in order to constrain the set of feasible coalition structures as least as possible. In addition, Elkind et al. [16] focus on Pareto optimality because such coalition structures always exist. Similarly, we restrict our study to symmetric additively separable hedonic games so as to guarantee the existence of min-max fair coalition structures.

Unfortunately, the maximum price of min-max fairness is not bounded by a constant value even for nonnegative valuation functions.

Theorem 8. *Let $G = (N, v)$ be a symmetric ASHG of n players with $v_i(j) \geq 0$ for every $i, j \in N$. Then*

$$\text{Max-PoMMF}(G) \leq n-1.$$

In addition, this bound is tight.

Proof. If $\text{SW}(\pi^*) = 0$, then $\text{Max-PoMMF}(G) = 1$. Otherwise, there are $i, j \in N$, $i \neq j$, such that $v_i(j) > 0$. We can upper-bound $\text{SW}(\pi^*)$ by $\sum_{i \in N} v_i(N)$. By Observation 2, we can lower-bound the value of every player i by $\max_{j \in N} v_i(j)$. Thus

$$\begin{aligned}\text{Max-PoMMF}(G) &\leq \frac{\sum_{i \in N} v_i(N)}{\sum_{i \in N} \max_{j \in N} v_i(j)} \\ &\leq \frac{\sum_{i \in N} (n-1) \max_{j \in N} v_i(j)}{\sum_{i \in N} \max_{j \in N} v_i(j)} \\ &= n-1.\end{aligned}$$

To see that this bound is tight, consider a game with n players, n even. Every player values every other player with $a > 0$. Thus the min-max threshold of every player is a. Therefore, the coalition structure that consists of $n/2$ pairs satisfies min-max fairness and has minimum utilitarian social welfare of na among all min-max fair coalition structures. The coalition structure consisting of the grand coalition that maximizes utilitarian social welfare, however, has a utilitarian social welfare of $n(n-1)a$. □

To obtain a meaningful bound in the above result, we need the existence of a min-max fair coalition structure, which is guaranteed in symmetric ASHGs. We need the following result before we can turn to Min-PoMMF.

Theorem 9. *Let $G = (N, v)$ be a symmetric ASHG. Then every coalition structure π that maximizes utilitarian social welfare satisfies min-max fairness.*

Proof. Suppose, for the sake of contradiction, that there is some $i \in N$ with $v_i(\pi(i)) < \text{MinMax}_i$. By a result of Brânzei and Larson [11], we know that every coalition structure maximizing utilitarian social welfare satisfies individual rationality. On the other hand, considering π with respect to MinMax, we have

$$0 \leq v_i(\pi(i)) < \text{MinMax}_i \leq \max_{C_j \in \pi} v_i(C_j \cup \{i\}). \tag{1}$$

Because of the strict inequality $\pi(i) \neq C_\ell$, where C_ℓ is a maximizer of $\max_{C_j \in \pi} v_i(C_j \cup \{i\})$, we have $v_i(\pi(i)) < v_i(C_\ell \cup \{i\})$. Since π maximizes utilitarian social welfare, the social welfare of π should not be lower than the coalition structure where i joins C_ℓ:

$$\begin{aligned}&\sum_{k\in\pi(i)} v_k(\pi(i)) + \sum_{k\in C_\ell} v_k(C_\ell) \geq \\ &\sum_{k\in\pi(i),\ k\neq i} v_k(\pi(i)\setminus\{i\}) + \sum_{k\in C_\ell} v_k(C_\ell) + 2v_i(C_\ell\cup\{i\}).\end{aligned}$$

Using symmetry, this is equivalent to the contradiction

$$v_i(\pi(i)) \geq v_i(C_\ell\cup\{i\}),$$

which completes the proof. □

Since min-max fairness is strictly stronger than individual rationality, Theorem 9 strengthens the result by Brânzei and Larson [11] that we used in the proof. Note that the above theorem also implies that we have an alternative proof of Corollary 6 (that every symmetric additively separable hedonic game admits a min-max fair coalition structure) that does not depend on the guaranteed existence of Nash-stable coalition structures but on the guaranteed existence of coalition structures maximizing utilitarian social welfare. From Theorem 9 we immediately have the following corollary.

Corollary 8. *Let G be a symmetric ASHG. Then*

$$\text{Min-PoMMF}(G) = 1.$$

6 Discussion & Conclusion

We have introduced three new notions of fairness in hedonic games and studied the connection with previously studied notions. Our notions themselves form a strict hierarchy: Every max-min fair coalition structure is grand-coalition fair (but not vice versa), and every grand-coalition fair coalition structure is min-max fair (but not vice versa). Although our fairness criteria are inspired from the field of fair division, our results are very different. Bouveret and Lemaître's scale of fairness criteria for additive utility functions [10] says that an envy-free partition of goods satisfies min-max fair share, which in turn implies proportionality, which in turn implies max-min fair share. So our strongest notion of fairness is the weakest notion in fair division of indivisible goods (according to this scale). In addition, in additively separable hedonic games, we have seen that grand-coalition fairness and max-min fairness coincide. This is not the case in fair division (if one equates grand-coalition fairness with proportionality). Also note that Nash stability (or, equivalently, a definition of envy-freeness based on joining) implies min-max fairness but none of the stronger notions. In the setting of indivisible goods, envy-freeness even implies min-max fair share. So it is one of the strongest notions there. We consider these results surprising, as the intution from fair division of indivisible goods is no longer valid in this different context. The main reasons are the already mentioned difference between the number of allowed subsets of a partition and that players can "share" coalitions. This missing intuition is also a reason of why we have checked in detail whether any known stability notions imply one of our fairness notions.

Then we have studied the complexity of computing threshold coalitions and deciding whether an additively separable hedonic game admits a fair coalition structure. Although nearly all of these problems are intractable, our fairness criteria still have some meaning. They give additional motivation to notions of stability, such as Nash stability. Moreover, in a decentralized setting the hardness of a problem can be "distributed" (of course, the intractability cannot disappear). Giving players a yardstick for fairness that only depends on their own preferences reduces the amount of communication that is necessary to check whether a coalition structure is fair. Our complexity results are also comparable to the results by Bouveret and Lemaître [10] and Heinen et al. [20] with the exception that no lower bound is known for deciding whether a max-min fair-share allocation exists, whereas in ASHGs we know that the corresponding problem is NP-complete. Also note that with min-max fairness we have found a notion that is strictly stronger than individual rationality, but is still satisfied by every coalition structure maximizing utilitarian social welfare in symmetric additively separable hedonic games. At last, we have initiated the study of price of fairness in hedonic games. Our results here are unsatisfactory in the sense that either the price is unbounded or not very informative.

Therefore, we consider finding suitable restrictions to players' valuation functions such that the maximum price of min-max fairness is bounded by a nontrivial constant an interesting research question for the future. Interesting future work would also be identifying (other) sufficient conditions that imply the existence of a fair coalition structure, determining the complexity of searching for a min-max fair coalition structure in symmetric additively separable hedonic games, and showing Σ_2^p-hardness of MIN-MAX-EXIST.

References

1. H. Aziz and F. Brandl. Existence of stability in hedonic coalition formation games. In *Proceedings of the 11th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 763–770. IFAAMAS, June 2012.
2. H. Aziz, F. Brandt, and P. Harrenstein. Fractional hedonic games. In *Proceedings of the 13th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 5–12. IFAAMAS, May 2014.
3. H. Aziz, F. Brandt, and H. Seedig. Computing desirable partitions in additively separable hedonic games. *Artificial Intelligence*, 195:316–334, 2013.
4. H. Aziz, S. Gaspers, J. Gudmundsson, J. Mestre, and H. Täubig. Welfare maximization in fractional hedonic games. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 461–467. AAAI Press/IJCAI, July 2015.
5. H. Aziz and R. Savani. Hedonic games. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 15. Cambridge University Press, 2016. To appear.
6. C. Ballester. NP-completeness in hedonic games. *Games and Economic Behavior*, 49(1):1–30, 2004.
7. S. Banerjee, H. Konishi, and T. Sönmez. Core in a simple coalition formation game. *Social Choice and Welfare*, 18(1):135–153, 2001.

8. A. Bogomolnaia and M. Jackson. The stability of hedonic coalition structures. *Games and Economic Behavior*, 38(2):201–230, 2002.
9. S. Bouveret, Y. Chevaleyre, and N. Maudet. Fair allocation of indivisible goods. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 12. Cambridge University Press, 2016. To appear.
10. S. Bouveret and M. Lemaître. Characterizing conflicts in fair division of indivisible goods using a scale of criteria. *Journal of Autonomous Agents and Multi-Agent Systems*. To appear. A preliminary version appeared in the proceedings of AAMAS-2014.
11. S. Brânzei and K. Larson. Coalitional affinity games and the stability gap. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 79–84. AAAI Press/IJCAI, 2009.
12. E. Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.
13. I. Caragiannis, C. Kaklamanis, P. Kanellopoulos, and M. Kyropoulou. The efficiency of fair division. *Theory of Computing Systems*, 50(4):589–610, 2012.
14. S. de Jong, K. Tuyls, and K. Verbeeck. Fairness in multi-agent systems. *The Knowledge Engineering Review*, 23(2):153–180, 2008.
15. J. Drèze and J. Greenberg. Hedonic coalitions: Optimality and stability. *Econometrica*, 48(4):987–1003, 1980.
16. E. Elkind, A. Fanelli, and M. Flammini. Price of Pareto optimality in hedonic games. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*. AAAI Press, Feb. 2016. To appear.
17. E. Elkind and J. Rothe. Cooperative game theory. In J. Rothe, editor, *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, chapter 3, pages 135–193. Springer-Verlag, 2015.
18. E. Elkind and M. Wooldridge. Hedonic coalition nets. In *Proceedings of the 8th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 417–424. IFAAMAS, May 2009.
19. M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.
20. T. Heinen, N. Nguyen, and J. Rothe. Fairness and rank-weighted utilitarianism in resource allocation. In *Proceedings of the 4th International Conference on Algorithmic Decision Theory*, pages 521–536. Springer-Verlag *Lecture Notes in Artificial Intelligence #9346*, Sept. 2015.
21. M. Karakaya. Hedonic coalition formation games: A new stability notion. *Mathematical Social Sciences*, 61(3):157–165, 2011.
22. J. Lang, A. Rey, J. Rothe, H. Schadrack, and L. Schend. Representing and solving hedonic games with ordinal preferences and thresholds. In *Proceedings of the 14th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1229–1238. IFAAMAS, May 2015.
23. J. Lang and J. Rothe. Fair division of indivisible goods. In J. Rothe, editor, *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, chapter 8, pages 493–550. Springer-Verlag, 2015.
24. D. Peters. Complexity of hedonic games with dichotomous preferences. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*. AAAI Press, Feb. 2016. To appear.
25. D. Peters. Graphical hedonic games of bounded treewidth. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*. AAAI Press, Feb. 2016. To appear.
26. M. Wright and Y. Vorobeychik. Mechanism design for team formation. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 1050–1056. AAAI Press, Jan. 2015.

Structural Control in Weighted Voting Games

Anja Rey and Jörg Rothe

Heinrich-Heine-Universität Düsseldorf, Düsseldorf, Germany
{rey, rothe}@cs.uni-duesseldorf.de

Abstract. Inspired by the study of control scenarios in elections and complementing manipulation and bribery settings in cooperative games with transferable utility, we introduce the notion of structural control in weighted voting games. We model two types of influence, adding players to and deleting players from a game, with goals such as increasing a given player's Shapley–Shubik or probabilistic Penrose–Banzhaf index in relation to the original game. We study the computational complexity of the problems of whether such structural changes can achieve the desired effect.

1 Introduction

A major task in computational social choice [42, 12, 13] is the complexity analysis of the question of whether a certain form of influence is possible in an election under some voting rule (see, e.g., [42, 13]). Bartholdi et al. [5] introduced and analyzed the notion of manipulation in elections, where one or more voters strategically change their true preference in order to make a distinguished candidate a winner. In a bribery scenario, on the other hand, an external agent tries to pay voters for them to change their votes such that a certain candidate becomes a winner, and the question is whether the briber can be successful within a given budget. This idea has been introduced and analyzed by Faliszewski et al. [24, 25]. In a third model, control, the chair of an election changes the structure of an election by adding, deleting, or partioning either voters or candidates, with the aim of making a distinguished candidate a winner [6]. In addition to these constructive types of control, destructive control—the problem of whether a given candidate can be prevented from being a winner—has also been introduced and studied by Hemaspaandra et al. [32]. Manipulation, bribery, and control have been studied for many voting systems, and we refer the reader to the book chapters by Baumeister and Rothe [10], Conitzer and Walsh [15], and Faliszewski and Rothe [27] for an overview of numerous related results. In a nutshell, whenever successful manipulative actions are generally possible, a high computational complexity may provide some protection against them, or at least against detecting whether such actions are possible or not for an election.

Similar ideas have been adapted to other fields, such as manipulation in preference aggregation [21] and manipulation, bribery, and control in judgement aggregation [23, 8, 7] (see the book chapters by Endriss [22] and Baumeister et al. [9] for an overview). In algorithmic game theory, the question of influencing the outcome of a game has also been studied extensively. In particular, for weighted voting games, manipulation by merging a coalition of players to a single player, or by splitting a player into several players in order to increase a player's power, have been introduced by Elkind et al. [2]. Here, "power" refers to the notion of power indices, such as the normalized and the probabilistic Penrose–Banzhaf [37, 4, 17] and the Shapley–Shubik index [43], measuring the significance of a player in a game (formal definitions will be given in Section 2).

The computational complexity of beneficial merging, splitting, and annexation[1] have been studied by Aziz et al. [1] who show, e.g., NP-hardness for the Shapley–Shubik and the normalized Penrose–Banzhaf power index. An upper bound of PP for the merging problem for the Shapley–Shubik index is provided by Faliszewski and Hemaspaandra [26]. Rey and Rothe [40] prove PP-hardness for both beneficial merging and splitting in case of the Shapley-Shubik and the probabilistic Penrose–Banzhaf index.

Weighted voting games are a well-known class of simple cooperative games that are compactly representable by expressing a distribution of the players' weights, which then is additively extended to coalitions winning exactly if a given quota is reached or exceeded. Besides merging, splitting, and annexation, other forms of manipulation have been studied in weighted voting games. For example, Zuckerman et al. [48] study manipulation of the quota. From an algorithmic point of view, this is different from our model: In their model, the number of players and thus the denominator in a power index (see Equations (1) and (2) in Section 2) remains the same but the same coalitions can have different success due to different quotas, whereas with structural control the number of players varies but all coalitions that remain in the game are equally successful before and after the change. Relatedly, Zick et al. [46, 47] study algorithmic properties of the quota. In dynamic weighted voting games, as presented by Elkind et al. [18], the quota is changed as well, but dynamically over time.

The notion of bribery has been adapted from voting theory to a model in cooperative game theory, bribery in so-called path-disruption games [3], where an external player tries to bribe a coalition of players in order to reach a target vertex from a source vertex in a graph [41]. Another perspective of persuation for weighted voting games has been studied by Freixas and Pons [30].

Inspired by the notion of control in elections, we consider control scenarios in weighted voting games. We define the problems of whether it is possible to change the structure of a game by either *adding* or *deleting* players in order to achieve certain goals. One could, for instance, think of a committee that needs a certain quota of votes so as to decide upon an issue. In order to increase the significance of some participant, an organizer might invite further participants or might choose a certain meeting schedule to make sure that originally existing participants are excluded. These structural changes could also be viewed as a change of the players' participation over time without malicious intentions.

Goals include *increasing* and *decreasing* the power of a distinguished player, in relation to the player's power in the original game. Increasing and decreasing power in a game by adding or deleting players can be seen as analogues of, respectively, construc-

[1] While merging is an action of a manipulative coalition, annexation describes one manipulative player who takes over other players. This goes back to the bloc paradox [28].

tive and destructive control in elections by adding or deleting either candidates or voters. Moreover, if an *exact* number of players is to be added, it might be desirable to *maintain* an original player's power index (or to keep it upper-bounded or lower-bounded by the original value—we will say the index is *nonincreasing* or *nondecreasing*).

We show that all defined control types are possible in weighted voting games, and we therefore analyze the computational complexity of whether control by structural changes can be exerted successfully in a given game. The complexity depends on the control type, the goal, and on whether the number of players that can be added or deleted is fixed or given in the problem instance.

Table 1. Overview of complexity results of structural control problems in weighted voting games with respect to the Shapley–Shubik and the probabilistic Penrose–Banzhaf index. Key: k is the number of players to be added or deleted, respectively; *PI* stands for *power index* (either *SS* or *PB*); *SS* (respectively, *PB*) indicates that these results are only known to hold for the Shapley–Shubik index (respectively, for the probabilistic Penrose–Banzhaf index); the other results each hold for both indices.

Goal	Control type		
	Adding players		Deleting players
	k fixed	k given	$k = 1$
Increase *PI*	PP-complete (Thm. 3)	PP-hard (Thm. 2)	NP-hard *(SS)* (Thm. 4)
Nondecrease *PI*	PP-complete (Thm. 3)	PP-hard (Thm. 2)	
Decrease *PI*	PP-complete (Thm. 3)	PP-hard (Thm. 2)	coNP-hard *(PB)* (Thm. 5)
Nonincrease *PI*	PP-complete (Thm. 3)	PP-hard (Thm. 2)	coNP-hard *(PB)* (Thm. 7)
Maintain *PI*	coNP-hard, in PP (Thm. 6)	PP-hard (Thm. 2)	coNP-hard (Thm. 7)

The paper is organized as follows. In Section 2, we introduce the needed notions and notation. Section 3 deals with the main model of control in weighted voting games. The goals of increasing or decreasing an index are studied in Section 4 and those of maintaining (relatedly, of nonincreasing or nondecreasing) an index are studied in Section 5. In Section 6, we conclude and point out some open questions. Most proofs are omitted due to space constraints.

2 Preliminaries

A *cooperative game with transferable utility* $\mathscr{G} = (N, v)$ consists of a set of players N and a coalitional function $v : 2^N \to \mathbb{R}$ assigning a value to each subset of players, called a *coalition*. $\mathscr{G}$ is called *simple* if v it is *monotonic* (i.e., $v(C) \leq v(D)$ whenever $C \subseteq D \subseteq N$) and if a coalition C is either *winning* ($v(C) = 1$) or *losing* ($v(C) = 0$).

Power indices are a common concept to measure a player's significance in a simple game. Two popular indices are the Penrose–Banzhaf index [37, 4] and the Shapley–Shubik index [43].

Letting $\text{PenroseBanzhaf}^*(\mathscr{G}, i) = \sum_{C \subseteq N \smallsetminus \{i\}} (v(C \cup \{i\}) - v(C))$ denote player i's *raw* index, we consider the *probabilistic* Penrose–Banzhaf power index proposed by Dubey and Shapley [17]:

$$\text{PenroseBanzhaf}(\mathscr{G}, i) = \frac{\text{PenroseBanzhaf}^*(\mathscr{G}, i)}{2^{n-1}}. \tag{1}$$

We say that i is *pivotal for a coalition C* if the marginal contribution $v(C \cup \{i\}) - v(C)$ of player i to coalition C in the definition above is 1, i.e., if C is losing but, after i has joined, $C \cup \{i\}$ is winning. On the other hand, $v(C \cup \{i\}) - v(C) = 0$ means that i is not pivotal for C.

The *raw* Shapley–Shubik power index is $\text{ShapleyShubik}^*(\mathscr{G}, i) = \sum_{C \subseteq N \smallsetminus \{i\}} \|C\|!(n - 1 - \|C\|)(v(C \cup \{i\}) - v(C))$, which is then normalized by

$$\text{ShapleyShubik}(\mathscr{G}, i) = \frac{\text{ShapleyShubik}^*(\mathscr{G}, i)}{n!} \tag{2}$$

to obtain the Shapley–Shubik power index.

Some simple games $\mathscr{G} = (N, v)$ can be compactly represented as *weighted voting games* $(w_1, \dots, w_n; q)$, where w_i, $1 \leq i \leq n$, is player i's *weight* and q is a *quota*, and a coalition $C \subseteq N$ *wins* if $\sum_{i \in C} w_i \geq q$ and otherwise it *loses*. Note that this representation is not fully expressive, i.e., there are simple games that cannot be represented by weighted voting games. For further background on cooperative game theory, see, e.g., the textbooks by Shoam and Leyton-Brown [44] and Peleg and Sudhölter [36] and, for computational aspects, the book by Chalkiadakis et al. [14] and the book chapters by Elkind et al. [19, 20].

For some background on computational complexity, see, e.g. the textbook by Papadimitriou [35]. We use the standard notions of *hardness* and *completeness* for a complexity class with respect to *many-one polynomial-time reducibility*. NP is the class of decision problems that can be solved in nondeterministic polynomial time, and coNP is the class of problems whose complements are in NP.

PARTITION is the following well-known NP-complete problem:

PARTITION	
Given:	A set $A = \{1, \dots, n\}$ and a function $a : A \to \mathbb{N} \smallsetminus \{0\}$, $i \mapsto a_i$, such that $\sum_{i=1}^{n} a_i$ is even.
Question:	Does there exist a partition into two subsets of equal weight, that is, does there exist a subset $A' \subseteq A$ such that $\sum_{i \in A'} a_i = \sum_{i \in A \smallsetminus A'} a_i$?

SUBSETSUM is also a well-known NP-complete problem:

SUBSETSUM	
Given:	A set $A = \{1, \dots, n\}$, a function $a : A \to \mathbb{N} \smallsetminus \{0\}$, $i \mapsto a_i$, and a positive integer q.
Question:	Is there a subset $A' \subseteq A$ such that $\sum_{i \in A'} a_i = q$?

Let $(a_1,\dots,a_n;q)$ and $(a_1,\dots,a_n)$ denote SUBSETSUM and PARTITION instances, respectively. A third well-known NP-complete problem that we will need is

EXACT COVER BY 3-SETS (X3C)	
Given:	A set $B=\{1,\dots,3k\}$, $k>0$, and a collection $\mathscr{S}=\{S_1,\dots,S_n\}$ of subsets $S_i\subseteq B$ with $\|S_i\|=3$ for $1\le i\le n$.
Question:	Is there an exact cover of B in $\mathscr{S}$, that is, is there a subcollection $\mathscr{S}'\subseteq\mathscr{S}$ such that $\bigcup_{S\in\mathscr{S}'}S=B$ and $S_i\cap S_j=\emptyset$, for each $S_i,S_j\in\mathscr{S}'$, $i\neq j$?

We furthermore consider the function class #P, the class of functions that give the number of solutions of NP problems. A function is #P-*many-one-hard* if there exists a polynomial-time reduction from each function in #P; it is #P-*parsimonious-hard* if this reduction preserves the number of solutions. If a #P function is #P-many-one-hard (#P-parsimonious-hard) it is said to be #P-*many-one-complete* (#P-*parsimonious-complete*). For instance, #SUBSETSUM and #X3C are known to be #P-parsimonious-complete functions. #P is closed under addition, i.e., if $f,g\in$ #P then $f+g\in$ #P.

From the literature [38, 16, 26] we obtain the following lemma.

Lemma 1 ([38, 16, 26]). *Computing the raw Penrose–Banzhaf index is* #P-*parsimonious-complete. Computing the raw Shapley–Shubik index is* #P-*many-one-complete.*

The complexity class PP (*probabilistic polynomial time*) was introduced by Gill [31] via probabilistic Turing machines; equivalently, it can be defined as the class of all decision problems X for which there exist a function $f\in$ #P and a polynomial p such that for all instances x: $x\in X$ if and only if $f(x)\ge 2^{p(|x|)-1}$. PP is considered to be a rather large complexity class, since it contains both NP and coNP (and even $\mathrm{P}^{\mathrm{NP}}_{\|}$ as shown by Beigel et al. [11]) and since it is known to be at least as hard (in terms of polynomial-time Turing reductions) as the polynomial hierarchy (i.e., $\mathrm{PH}\subseteq\mathrm{P}^{\mathrm{PP}}$) by Toda's theorem [45]. Further, PP is closed under complement (which is easy to see) and, far from being trivial, it is also closed under union and intersection [29]. We make use of the following lemma by Faliszewski and Hemaspaandra [26, Lemma 2.3] in the context of comparing a player's power in weighted voting games with respect to the probabilistic Penrose–Banzhaf and the Shapley–Shubik index.

Lemma 2 ([26]). *Let F be a* #P-*parsimonious-complete function. Then, the problem* COMPARE-$F=\{(x,y)\mid F(x)>F(y)\}$ *is* PP-*complete.*

Since #X3C and #SUBSETSUM are #P-parsimonious-complete, COMPARE-#SUBSETSUM and COMPARE-#X3C are PP-complete.

Moreover, we will use the following lemma that is due to Faliszewski and Hemaspaandra [26] and has been slightly adapted by Rey and Rothe [40].

Lemma 3. *Every* X3C *instance* $(B',\mathscr{S}')$ can be transformed into an X3C instance $(B,\mathscr{S})$ where $\|B\|=3k$ and $\|\mathscr{S}\|=n$, such that $k/n=2/3$ without changing the number of solutions (i.e., #X3C$(\mathrm{B},\mathscr{S})=$ #X3C$(\mathrm{B}',\mathscr{S}')$).

Consequently, we can assume that the size of each solution in a SUBSETSUM instance is $2n/3$, that is, each subsequence summing up to the given quota contains the same number of elements.

We consider a restricted variant of the COMPARE-#SUBSETSUM problem, namely COMPARE-#SUBSETSUM-RR as defined in [40]: Given a set $A = \{1,\dots,n\}$ and a function $a : A \to \mathbb{N} \setminus \{0\}$, $i \mapsto a_i$, is the number of subsets of A with values summing up to $(\alpha/2) - 2$, where $\alpha = \sum_{i=1}^{n} a_i$, greater than the number of subsets of A with values summing up to $(\alpha/2) - 1$, i.e., is it true that

$$\#\text{SUBSETSUM}((a_1,\dots,a_n;\,(\alpha/2)-2)) \;>\; \#\text{SUBSETSUM}((a_1,\dots,a_n;\,(\alpha/2)-1))? \tag{3}$$

Let $(a_1,\dots,a_n)$ denote an instance of COMPARE-#SUBSETSUM-RR. From [40, Lemma 4.5] we obtain the following lemma.

Lemma 4 ([40]). COMPARE-#SUBSETSUM-RR *is* PP-*hard.*

Likewise, the analogous problem of whether $<$ holds in (3), denoted by COMPARE-#SUBSETSUM-ЯЯ, is PP-hard [40].

The following lemma differentiates between players that are not part of a weighted voting game and those who are part of the game but do not have any weight.

Lemma 5. *For both the probabilistic Penrose–Banzhaf index and the Shapley–Shubik index, given a weighted voting game, adding a player with weight zero does not change the original players' power indices, and the new player's power index is zero.*

3 Control Types and Goals

We define control by adding and by deleting players in weighted voting games. For each control type, we consider goals, such as increasing or decreasing a distinguished player's power, in relation to the original game. We first define how adding and deleting a player affects the coalitional function for weighted voting games: For control by adding players, from a given weighted voting game $\mathcal{G} = (w_1,\dots,w_n;q)$ with $N = \{1,\dots,n\}$ and a set $M = \{n+1,\dots,n+m\}$ of m unregistered players with weights $w_{n+1},\dots,w_{n+m}$, we obtain a new game $\mathcal{G}_{\cup M} = (w_1,\dots,w_{n+m};q)$.

For example, we consider the following decision problem for a power index PI:

CONTROL BY ADDING PLAYERS TO INCREASE PI	
Given:	A weighted voting game $\mathcal{G}$ with players $N = \{1,\dots,n\}$, a set M of unregistered players with weights $w_{n+1},\dots,w_{n+m}$, a distinguished player $p \in N$, and a positive integer k.
Question:	Can at most k players $M' \subseteq M$ be added to $\mathcal{G}$ such that for the new game $\mathcal{G}_{\cup M'}$ it holds that $\mathrm{PI}(\mathcal{G}_{\cup M'},p) > \mathrm{PI}(\mathcal{G},p)$?

Analogously, we can ask whether the game can be controlled so as to gain the opposite effect, and decrease a certain player's index. In these cases, hardness in terms of complexity can be seen as a shield to prevent a game from being controlled to improve a player's significance or to worsen a player's significance. On the other hand, we also consider the following control question: Is it possible to add players to a game without changing the distribution of power among the original players?

We can ask analogous questions with the same aims for removing players from the game. Deleting a subset $M \subseteq N$ of m players from a weighted voting game $\mathscr{G} = (w_1, \dots, w_n; q)$ yields a weighted voting game $\mathscr{G}_{\smallsetminus M} = (w_{j_1}, \dots, w_{j_{n-m}}; q)$ with $\{j_1, \dots, j_{n-m}\} = N \smallsetminus M$.[2]

For instance, we define the following decision problem for a power index PI:

CONTROL BY DELETING PLAYERS TO INCREASE PI	
Given:	A weighted voting game $\mathscr{G}$ with players $N = \{1, \dots, n\}$, a distinguished player $p \in N$, and a positive integer $k < \|N\|$.
Question:	Can at most k players $M' \subseteq N \smallsetminus \{p\}$ be deleted from $\mathscr{G}$ such that in the new game $\mathscr{G}_{\smallsetminus M'}$ it holds that $\mathrm{PI}(\mathscr{G}_{\smallsetminus M'}, p) > \mathrm{PI}(\mathscr{G}, p)$?

Again, we can analogously define the variations of this problem where the goal is not to increase some player's power index but to decrease or to maintain it.

Example 1. Let $\mathscr{G} = (N, v)$ be a weighted voting game with six players in $N = \{1,2,3,4,5,6\}$ represented by

$$(1,2,2,3,4,5;10).$$

Let $k = 1$, that is, one player can be removed from the game. Consider the Penrose–Banzhaf index. Table 2 lists the players' probabilistic Penrose–Banzhaf and Shapley–Shubik power indices for the resulting games.

Player 1, 4, 5 and 6 with indices of $1/8$, $5/16$, $3/8$, and $9/16$, respectively, cannot improve from any other player being deleted. However, e.g., player 1's index can be decreased to $1/16$ when removing player 5 and is maintained in the other cases. Players 2 and 3 can benefit from the other one being removed, as the index increases from $3/16$ to $1/4$.

For the Shapley–Shubik index, due to normalization over the permutations of participating players, an increase of power is expected when deleting a player. As an example, player 5 has an index of $13/60$ in $\mathscr{G}$ which increases to $1/4$ if either one of the players 2, 3, or 4 is deleted, and even to $7/20$ if 6 is deleted. However, players can also have a disadvantage, if a player leaves the game. For instance, player 1 loses power if 5 is deleted, 2 and 3 lose power if 4 is deleted, 4 loses power if 2 or 3 are deleted, and 5 loses power if 1 is deleted. This suggests a symmetric dependence of the players. In the same way, the power of players 2 and 3 remains the same if 6 is removed, and the other way around.

From the opposite view, consider the weighted voting game represented by $(2,3,4,5;10)$, two unregistered players with weights 1 and 2, and $k = 2$ (see the bottom rows for the two indices). Note that adding them both ends up in $\mathscr{G}$. Here, the four players have probabilistic Penrose–Banzhaf indices of $1/4$, $1/4$, $1/4$, and $1/2$. The first player (with weight 2) can only be worse off when adding any of the two players. The player with weight 3 as well as the player with weight 5 can benefit from adding both players or only the one with weight 2. The former keeps the same index, while the latter loses power if the player with weight 1 is added. Finally, the player with weight 4 improves

[2] One might also think of different ways to reasonably model the new game, and we will eloborate on that in Section 6. Here, we focus on the notion just presented.

Table 2. Power distribution in the games of Example 1

Player i		1	2	3	4	5	6
PenroseBanzhaf$(\mathscr{G}, i)$	$\cdot 32$	4	6	6	10	12	18
PenroseBanzhaf$(\mathscr{G}_{\setminus\{1\}}, i)$	$\cdot 32$		6	6	10	10	18
PenroseBanzhaf$(\mathscr{G}_{\setminus\{2\}}, i)$	$\cdot 32$	4		8	8	12	16
PenroseBanzhaf$(\mathscr{G}_{\setminus\{3\}}, i)$	$\cdot 32$	4	8		8	12	16
PenroseBanzhaf$(\mathscr{G}_{\setminus\{4\}}, i)$	$\cdot 32$	4	4	4		12	16
PenroseBanzhaf$(\mathscr{G}_{\setminus\{5\}}, i)$	$\cdot 32$	2	6	6	10		14
PenroseBanzhaf$(\mathscr{G}_{\setminus\{6\}}, i)$	$\cdot 32$	4	4	4	8	8	
PenroseBanzhaf$(\mathscr{G}_{\setminus\{1,2\}}, i)$	$\cdot 32$			8	8	8	16
ShapleyShubik$(\mathscr{G}, i)$	$\cdot 60$	4	6	6	11	13	20
ShapleyShubik$(\mathscr{G}_{\setminus\{1\}}, i)$	$\cdot 60$		7	7	12	12	22
ShapleyShubik$(\mathscr{G}_{\setminus\{2\}}, i)$	$\cdot 60$	5		10	10	15	20
ShapleyShubik$(\mathscr{G}_{\setminus\{3\}}, i)$	$\cdot 60$	5	10		10	15	20
ShapleyShubik$(\mathscr{G}_{\setminus\{4\}}, i)$	$\cdot 60$	5	5	5		15	30
ShapleyShubik$(\mathscr{G}_{\setminus\{5\}}, i)$	$\cdot 60$	3	8	8	13		28
ShapleyShubik$(\mathscr{G}_{\setminus\{6\}}, i)$	$\cdot 60$	6	6	6	21	21	
ShapleyShubik$(\mathscr{G}_{\setminus\{1,2\}}, i)$	$\cdot 60$			10	10	10	30

in every situation when adding one or two players. The first and the fourth player (with weight 2 and 5, respectively) cannot benefit from adding players with respect to the Shapley–Shubik index. The other two can take advantage in the same cases as for the probabilistic Penrose–Banzhaf index.

In particular, the example shows that these types of control are each possible. We therefore turn to the question of how hard it is to find out whether they can be exerted successfully in a given game.

Next to goals in relation to the old game, we can also compare an index either in relation to the other players' power, or in relation to a constant number. See Section 6 for initial results for this idea.

If a player i is deleted from a weighted voting game, any other player j gains the same amount of power that i would gain if j were deleted [33].

The changes of power indices are bounded as follows.

Proposition 1. *For both the Penrose–Banzhaf index and the Shapley–Shubik index, a player's power can increase by at most* $1/2$ *and decrease by at least* $-1/2$ *when deleting another player.*

Theorem 1. *After deleting the players of a subset* $M \subseteq N \setminus \{i\}$ *of size* $m \geq 1$ *from a weighted voting game* $\mathscr{G} = (N, v)$*, the difference of player i's*

1. *Penrose–Banzhaf index is at most* $1 - 2^{-m}$ *and is at least* $-1 + 2^{-m}$*;*
2. *Shapley–Shubik index is at most* $1 - (n-m+1)!/2n!$ *and is at least* $-1 + (n-m-1)!/2(n-2)!$*.*

4 Increasing or Decreasing an Index

Similarly to control by adding or deleting voters or candidates in elections, adding and deleting players are not merely inverse operations. This is due to the fact that when adding players all original players are guaranteed to be part of the game before and after the structural change, whereas when deleting players each player except the distinguished one can be removed from the game. Hardness in terms of complexity can be seen as a shield to prevent a game from being controlled to improve or worsen a player's significance.

4.1 Control by Adding Players

From a computational complexity point of view, we distinguish the cases where an upper bound of new players is given as defined above and where the number of new players is fixed.

Theorem 2. *Control by adding a given number of players in order to increase (decrease) a distinguished player's probabilistic Penrose–Banzhaf or Shapley–Shubik index in a weighted voting game is* PP*-hard.*

Remark 1. An upper bound of $\mathrm{NP}^{\mathrm{PP}}$ can be established whenever the number of players to be added is given. We can guess the subset of new players to be added nondeterministically. Verifying whether the different goals are satisfied is encoded in the PP-oracle. We conjecture that this problem is complete for this class.

Theorem 3. *Control by adding a fixed number of players in order to increase (decrease) a distinguished player's probabilistic Penrose–Banzhaf or Shapley–Shubik index in a weighted voting game is* PP*-complete.*

4.2 Control by Deleting Players

Recall that although deleting a previously added player results in the same game, the possibility to fulfill a certain goal by adding a player is not the complement of the possiblity to fulfill the complement goal by deleting a player. Initially, we obtain the following two results.

Theorem 4. *Control by deleting players to increase a distinguished player's Shapley–Shubik index in a weighted voting game is* NP*-hard (even if only one player is deleted).*

Theorem 5. *Control by deleting players to decrease a distinguished player's Penrose–Banzhaf index in a weighted voting game is* coNP*-hard (even if only one player can be deleted).*

Proof We show coNP-hardness by means of a reduction from the complement of PARTITION, denoted by $\overline{\text{PARTITION}}$. Letting $(a_1,\ldots,a_n)$ be a PARTITION instance with $\alpha = \sum_{i=1}^{n} a_i$, we construct the following control instance: Let $\mathscr{G}$ be represented by $(1, a_1, \ldots, a_n, \alpha/2; \alpha/2 + 1)$ and consider distinguished player $p = 1$ and deletion limit

$k = 1$. For $\xi = \#\textsc{Partition}((a_1, \ldots, a_n))$, we show that $\xi = 0$ if and only if there exists a player whose removal from the game causes player 1's Penrose–Banzhaf power to decrease.

Only if: Assume that $\xi = 0$. Then $\text{PenroseBanzhaf}^*(\mathcal{G}, 1) = 1$. However, if player $n+2$ with weight $\alpha/2$ is removed, there is no coalition left player 1 is pivotal for. Therefore, control in order to decrease player 1's Penrose–Banzhaf index is possible.

If: Assume that $\xi \geq 0$. Then $\text{PenroseBanzhaf}^*(\mathcal{G}, 1) = \xi + 1$. If player $n+2$ is deleted, $\text{PenroseBanzhaf}^*(\mathcal{G}_{\smallsetminus\{n+2\}}, 1) = \xi$ and

$$\begin{aligned}
&\text{PenroseBanzhaf}(\mathcal{G}_{\smallsetminus\{n+2\}}, 1) - \text{PenroseBanzhaf}(\mathcal{G}, 1) \\
&= \frac{\xi}{2^n} - \frac{\xi+1}{2^{n+1}} = \frac{2\xi - \xi - 1}{2^{n+1}} = \frac{\xi - 1}{2^{n+1}} \geq 0.
\end{aligned}$$

Note that this difference is even greater than 0, since ξ is even. If a player j, $2 \leq j \leq n+1$, is deleted, $\text{PenroseBanzhaf}^*(\mathcal{G}_{\smallsetminus\{j\}}, 1) = 1 + \xi/2$ and

$$\begin{aligned}
&\text{PenroseBanzhaf}(\mathcal{G}_{\smallsetminus\{n+2\}}, 1) - \text{PenroseBanzhaf}(\mathcal{G}, 1) \\
&= \frac{1 + \frac{\xi}{2}}{2^n} - \frac{\xi+1}{2^{n+1}} = \frac{2 + \xi - \xi - 1}{2^{n+1}} = \frac{1}{2^{n+1}} > 0.
\end{aligned}$$

Consequently, a decrease of player 1's Penrose–Banzhaf index is not possible by deleting any other player than 1. ❑

5 Maintaining an Index

In addition to constructive or destructive goals, we now consider situations in which an exact number of players is to be added and the goal is to either maintain a distinguished player's power index in this new game, or at least to ensure that this player's power does not increase or decrease, compared with this player's power in the old game. For a real-world example, note that weighted voting games can be used to model decision processes in legislative bodies such as the EU Commission, national parliaments, or the United Nations Security Council. However, it may happen that such bodies have to be expanded by adding a certain number of new members. In such a case, an old member may be interested in maintaining the same power as before the new members have joined the game. For instance, control by adding players with the goal to maintain a given player's power index PI is defined as follows. The other goals of nonincreasing or nondecreasing a given player's power by adding or deleting players can be defined analogously.

Control by Adding Players to Maintain PI	
Given:	A weighted voting game $\mathcal{G}$ with players $N = \{1, \ldots, n\}$, a set M of unregistered players with weights $w_{n+1}, \ldots, w_{n+m}$, a distinguished player $p \in N$, and a positive integer k.
Question:	Can exactly k players $M' \subseteq M$ be added to $\mathcal{G}$ such that for the new game $\mathcal{G}_{\cup M'}$ it holds that $\text{PI}(\mathcal{G}_{\cup M'}, p) = \text{PI}(\mathcal{G}, p)$?

Note that in the context of maintaining an index, we require that an *exact* (either fixed or given) number of players is to be added to or to be deleted from the game. Otherwise (i.e., if adding *at most* some given number of players—and thus, for instance, no player at all—were allowed), the problem of whether it is possible to maintain an index would be trivial.

At first glance, it might not make much sense to distinguish all five goals of control. However, they each are differently motivated and they all are also different in their algorithmic nature: While for all five goals in relation to the original game, for a fixed number of players to be added, these problems are contained in PP (see Theorem 3), a complexity lower bound for, say, the case of deleting players to nondecrease a given player's Shapley–Shubik index cannot immediately be obtained from NP-hardness for the same case of deleting players so as to increase a given player's Shapley–Shubik index (see Theorem 4).

5.1 Control by Adding Players

Analogously to Theorem 2, since PP is closed under complement and by an alternative reduction from the complement of COMPARE-#SUBSETSUM-ЯЯ, control by adding a given number of players in order to maintain a distinguished player's probabilistic Penrose–Banzhaf or Shapley–Shubik index in a weighted voting game is PP-hard. Similarly, whenever the number of players to be added is given in unary, these problems are in $\mathrm{NP}^{\mathrm{PP}}$.

Analogously to Theorem 3, control by adding a fixed number of players in order to nonincrease or nondecrease a distinguished player's probabilistic Penrose–Banzhaf index or Shapley–Shubik index in a weighted voting game is PP-complete.

Theorem 6. *Control by adding a fixed number of players to maintain a distinguished player's probabilistic Penrose–Banzhaf or Shapley–Shubik index in a weighted voting game is* coNP*-hard and in* PP*.*

5.2 Control by Deleting Players

Note again that deleting players in order to increase a power index is not the inverse of adding players in order to nonincrease the same index.

Theorem 7. *Control by deleting a player in order to maintain a distinguished player's probabilistic Penrose–Banzhaf index in a weighted voting game is* coNP*-hard (even if only one player can be deleted).*

In particular, if $\xi \geq 1$ in the above proof, then deleting any player cannot lead to a nonincrease. Therefore, it also holds that $\xi \geq 1$ if and only if deleting any player but 1 does not nonincrease the probabilistic Penrose–Banzhaf index of player 1. Therefore, we get coNP-hardness for the problem where the goal is to nonincrease the probabilistic Penrose–Banzhaf by essentially the same proof.

Observe that from these constructions we cannot draw further conclusion about the complexity of structural control by deleting players for neither the Shapley–Shubik nor the probabilistic Penrose–Banzhaf index.

6 Conclusions and Future Work

For weighted voting games, we have studied two types of control, combined with the following variants of goals: Strictly increasing or strictly decreasing a player's power index by adding or deleting at most a given number of players as well as maintaining, nondecreasing, and nonincreasing a player's power index by adding or deleting an exact number of players. As a measure of a player's power we have analyzed the well-known Shapley–Shubik power index and the probabilistic Penrose–Banzhaf power index.

If the number of players to be added is given, the problems of adding players in order to obtain a change in a player's index (or at least allow a change in one direction) is PP-hard. And if the number of players to be added is fixed, a corresponding PP upper bound is valid, so we have PP-completeness. In the case of deleting players, we have established NP- and coNP-hardness lower bounds, even for the case of deleting exactly one player. The complexity results are summed up in Table 1.

The complexity of some control problems is left open; For instance, interesting gaps remain, e.g., between NP-hardness and PP membership as well as PP-hardness and $\mathrm{NP}^{\mathrm{PP}}$ membership, and we do not know the complexity of control by deleting players in order to nondecrease a player's index. Also, considering other measures of voter power may provide further insights into the problem of structurally controlling a game.

So far we have only obtained results for goals in *relation to the original game*. Alternatively, one might think of a situation where the goal is to increase a player's significance in *comparison to the other players*, which can also be achieved if players are added or deleted; the distinguished player's power index remains the same, but all remaining players' indices are distributed so that they are below this value.

Besides this, we can also model a scenario where a player is required to exceed a certain *constant power index*, and we ask whether it is possible to control a game by adding or deleting players in order to reach this index. So far, we can tell that if the number of players to be added or deleted is $k = 0$, our value is $1/2$, and the considered power index is the Penrose–Banzhaf index, the problem is PP-complete. This might change if $k \geq 0$ is required. We might also study the variant of obtaining an exact value.

There seems to be a close connection to the notion of *synergies* in cooperative games (see, e.g., [39]), and it will be interesting to have a closer look at related results here.

In addition to weighted voting games, other classes of cooperative games with transferable utility might of course be affected by control scenarios as well. In each case, adding and deleting players has to be well-defined. As an example, consider general (weighted) majority games. Let $\mathscr{G} = (w_1, w_2, \ldots, w_n; \alpha(n))$ be a majority game, that is, $v(C) = 1$ if $\sum_{i \in C} w_i \geq \lfloor \alpha(n) \rfloor + 1$, and $v(C) = 0$ otherwise, for each $C \subseteq N$. Now, if a player is deleted, the number of players n is decremented, which changes the threshold $\alpha(n)$. The new coalitional function is computed as above. Adding a player requires a set of unregistered players given by their weights, and n is increased. As we have seen, for (weighted) threshold games, the new coalitional function is determined similarly, with the only difference that the threshold does not change. One could alternatively think of weights as a percentage, and change the weights of the remaining players proportionally. Thus the new game $\mathscr{G}_{\cup M}$ is defined differently, by normalizing the sum of weights to the original value. Similarly to majority games, players here do not make an absolute but a relative contribution to the game.

Adding and deleting players can be viewed in some sense as a change over time, which has previously been analyzed only for changing the quota over time [18]. Similarly, studying a change of players dynamically over time is an interesting task for future work.

Other games that will be interesting to study in this context include games in which the Shapley–Shubik index is easy to compute, such as weighted graph games [16]. In such games, two indices in two games can be compared in polynomial time and, therefore, if the coalition that is added to or removed from a game is known, the possibility of control is easy to detect, rendering the problems trivial. If, on the other hand, there are several possible coalitions to be added, this problem might become interesting again. Eventually, if players correspond to an edge in a game, deleting an edge may be interesting in the context of Braess's paradox for noncooperative congestion games (see, e.g., [34, pp. 464–465]) where, informally, an extra fast lane might lead to congestion, whereas without this lane traffic may split up to equally slower paths. Can we find a connection to control by deleting a player in a cooperative game with transferable utility?

Acknowledgment: A two-page extended abstract of this paper is to appear in the proceedings of AAMAS-2016, and we thank the AAMAS and CoopMAS reviewers for useful comments. This work was supported in part by DFG grant RO-1202/14-2.

References

1. H. Aziz, Y. Bachrach, E. Elkind, and M. Paterson. False-name manipulations in weighted voting games. *Journal of Artificial Intelligence Research*, 40:57–93, 2011.
2. Y. Bachrach and E. Elkind. Divide and conquer: False-name manipulations in weighted voting games. In *Proc. AAMAS'08*, pages 975–982. IFAAMAS, 2008.
3. Y. Bachrach and E. Porat. Path disruption games. In *Proc. AAMAS'10*, pages 1123–1130. IFAAMAS, 2010.
4. J. Banzhaf III. Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Review*, 19:317–343, 1965.
5. J. Bartholdi III, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.
6. J. Bartholdi III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical Computer Modelling*, 16(8/9):27–40, 1992.
7. D. Baumeister, G. Erdélyi, O. Erdélyi, and J. Rothe. Control in judgment aggregation. In *Proc. STAIRS'12*, pages 23–34. IOS Press, 2012.
8. D. Baumeister, G. Erdélyi, O. Erdélyi, and J. Rothe. Complexity of manipulation and bribery in judgment aggregation for uniform premise-based quota rules. *Mathematical Social Sciences*, 76:19–30, 2015.
9. D. Baumeister, G. Erdélyi, and J. Rothe. Judgment aggregation. In J. Rothe, editor, *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, chapter 6, pages 361–391. Springer-Verlag, 2015.
10. D. Baumeister and J. Rothe. Preference aggregation by voting. In J. Rothe, editor, *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, chapter 4, pages 197–325. Springer-Verlag, 2015.

11. R. Beigel, L. Hemachandra, and G. Wechsung. On the power of probabilistic polynomial time: $\mathrm{P}^{\mathrm{NP}[\log]} \subseteq \mathrm{PP}$. In *Proc. Structures'89*, pages 225–227. IEEE Computer Society Press, 1989.
12. F. Brandt, V. Conitzer, and U. Endriss. Computational social choice. In G. Weiß, editor, *Multiagent Systems*, pages 213–283. MIT Press, second edition, 2013.
13. F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, editors. *Handbook of Computational Social Choice*. Cambridge University Press, 2016. To appear.
14. G. Chalkiadakis, E. Elkind, and M. Wooldridge. *Computational Aspects of Cooperative Game Theory*. Morgan & Claypool, 2011.
15. V. Conitzer and T. Walsh. Barriers to manipulation in voting. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 6. Cambridge University Press, 2016. To appear.
16. X. Deng and C. Papadimitriou. On the complexity of comparative solution concepts. *Mathematics of Operations Research*, 19(2):257–266, 1994.
17. P. Dubey and L. Shapley. Mathematical properties of the Banzhaf power index. *Mathematics of Operations Research*, 4(2):99–131, 1979.
18. E. Elkind, D. Pasechnik, and Y. Zick. Dynamic weighted voting games. In *Proc. AAMAS'13*, pages 515–522. IFAAMAS, 2013.
19. E. Elkind, T. Rahwan, and N. Jennings. Computational coalition formation. In G. Weiß, editor, *Multiagent Systems*, pages 329–380. MIT Press, second edition, 2013.
20. E. Elkind and J. Rothe. Cooperative game theory. In J. Rothe, editor, *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, chapter 3, pages 135–193. Springer-Verlag, 2015.
21. U. Endriss. Sincerity and manipulation under approval voting. *Theory and Decision*, 74(3):335–355, 2013.
22. U. Endriss. Judgment aggregation. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 17. Cambridge University Press, 2016. To appear.
23. U. Endriss, U. Grandi, and D. Porello. Complexity of judgment aggregation. *Journal of Artificial Intelligence Research*, 45:481–514, 2012.
24. P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. How hard is bribery in elections? *Journal of Artificial Intelligence Research*, 35:485–532, 2009.
25. P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research*, 35:275–341, 2009.
26. P. Faliszewski and L. Hemaspaandra. The complexity of power-index comparison. *Theoretical Computer Science*, 410(1):101–107, 2009.
27. P. Faliszewski and J. Rothe. Control and bribery in voting. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 7. Cambridge University Press, 2016. To appear.
28. D. Felsenthal and M. Machover. Postulates and paradoxes of relative voting power – A critical re-appraisal. *Theory and Decision*, 38(2):195–229, 1995.
29. L. Fortnow and N. Reingold. PP is closed under truth-table reductions. *Information and Computation*, 124(1):1–6, 1996.
30. J. Freixas and M. Pons. Circumstantial power: Optimal persuadable voters. *European Journal of Operational Research*, 186:1114–1126, 2008.
31. J. Gill. Computational complexity of probabilistic Turing machines. *SIAM Journal on Computing*, 6(4):675–695, 1977.
32. E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5-6):255–285, 2007.

33. R. Myerson. Conference structures and fair allocation rules. *International Journal of Game Theory*, 9(3):169–182, 1980.
34. N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.
35. C. Papadimitriou. *Computational Complexity*. Addison-Wesley, second edition, 1995.
36. B. Peleg and P. Sudhölter. *Introduction to the Theory of Cooperative Games*. Springer-Verlag, second edition, 2007.
37. L. Penrose. The elementary statistics of majority voting. *Journal of the Royal Statistical Society*, 109(1):53–57, 1946.
38. K. Prasad and J. Kelly. NP-completeness of some problems concerning voting games. *International Journal of Game Theory*, 19(1):1–9, 1990.
39. T. Rahwan, T. Michalak, and M. Wooldridge. A measure of synergy in coalitions. Technical Report arXiv:1404.2954.v1 [cs.GT], CoRR, Apr. 2014.
40. A. Rey and J. Rothe. False-name manipulation in weighted voting games is hard for probabilistic polynomial time. *Journal of Artificial Intelligence Research*, 50:573–601, July 2014.
41. A. Rey, J. Rothe, and A. Marple. Path-disruption games: Bribery and a probabilistic model. *Theory of Computing Systems*. To appear.
42. J. Rothe, editor. *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*. Springer-Verlag, 2015.
43. L. Shapley and M. Shubik. A method of evaluating the distribution of power in a committee system. *American Political Science Review*, 48(3):787–792, 1954.
44. Y. Shoham and K. Leyton-Brown. *Multiagent Systems. Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009.
45. S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20(5):865–877, 1991.
46. Y. Zick. On random quotas and proportional representation in weighted voting games. In *Proc. IJCAI'13*, pages 432–438. AAAI Press, 2013.
47. Y. Zick, A. Skopalik, and E. Elkind. The Shapley value as a function of the quota in weighted voting games. In *Proc. IJCAI'11*, pages 490–496. AAAI Press, 2011.
48. M. Zuckerman, P. Faliszewski, Y. Bachrach, and E. Elkind. Manipulating the quota in weighted voting games. *Artificial Intelligence*, 180–181:1–19, 2012.

Strategy-Proofness in the Stable Matching Problem with Couples

Andrew Perrault*, Joanna Drummond, and Fahiem Bacchus

Department of Computer Science
University of Toronto, Toronto, Canada
{perrault,jdrummond,fbacchus}@cs.toronto.edu

Abstract. Stable matching problems (SMPs) arising in real-world markets often have complementarities in the participants' preferences, which break many of the theoretical properties of SMP and make it computationally hard to find a stable matching. In this paper, we provide some key insights into the issue of strategy-proofness in SMP-C, which is SMP with one such complementarity, the presence of couples. We relate the set of resident Pareto optimal stable matchings ($\mathcal{RP}_{opt}$) admitted by an SMP-C instance to the ability of the residents to manipulate, showing that an $\mathcal{RP}_{opt}$ mechanism is immune to certain forms of manipulation. We provide an algorithm for finding an $\mathcal{RP}_{opt}$ matching when one exists. And finally, we study empirically the frequency of multiple stable and multiple $\mathcal{RP}_{opt}$ matchings in a variety of settings, finding that SMP-C becomes less susceptible to manipulation as both the market size increases and the fraction of couples in the market decreases.

Keywords: Stable Matching, Complementarities, Strategy-proofness, SAT

1 Introduction

The Stable Matching Problem (SMP), where two sides of a market wish to be matched with each other, is one of the most widely-studied economics problems. SMP has extensive real-world applications of which residency matching is one of the most well-known [18, 23, 1]. The National Resident Matching Program (NRMP) in the US, and the Scottish Foundation Allocation Scheme (SFAS) in Scotland, are two such real-world instantiations of such matching markets [17, 9]. The SMP framework however does not allow for complementarities between the participants' preferences. An important complementarity in resident matching arises from couples where a resident's preferences can depend on the match received by their partner. To address this need for couples to coordinate their placement, residency matching markets (including the NRMP and SFAS) began allowing couples to express their preferences over residency programs jointly.

* We acknowledge the support of NSERC. Drummond and Perrault were supported by OGS. We thank the AAMAS and CoopMAS reviewers for helpful suggestions.

This extension to SMP is called the Stable Matching Problem with Couples (SMP-C).

As in SMP, the goal of SMP-C is to find a match such that no pair (one from each side of the market) has an incentive to defect from their assignment. A matching with this property is said to be *stable*. In standard SMP, there always exists a resident-optimal stable matching ($\mathcal{R}_{opt}$) in which every resident is as least as well off as in any other stable matching; and the simple Deferred Acceptance (DA) algorithm [7] can find this $\mathcal{R}_{opt}$ matching in polynomial time. These properties no longer hold for SMP-C. Finding a stable matching in SMP-C becomes NP-complete [20]; a stable matching is not guaranteed to exist; and even when one does exist, an $\mathcal{R}_{opt}$ matching may not exist.

While there are many strategy-proofness results for SMP, little work has investigated analogous results for SMP-C. For SMP, no stable matching mechanism is strategy-proof for both sides of the market [22]. There are, however, mechanisms that are approximately stable and approximately strategy-proof where the approximation becomes better as the market size grows [10]. In SMP, an $\mathcal{R}_{opt}$ matching always exists and a mechanism returning it is known to be strategy-proof against manipulation by the residents [7].

Since SMP-C extends SMP there can be no stable mechanism for SMP-C that is strategy-proof for both sides of the market. Furthermore, the limit result of [10] is not known to hold for SMP-C. To the authors' knowledge, no investigation of strategy-proofness against residents in SMP-C has been published. This is a gap in the literature since strategy-proofness on one side of the market is frequently cited as an important property for real world matching mechanisms; e.g., there is evidence that mechanisms lacking this property are often abandoned in favor of strategy-proof ones [2].

In this paper we investigate the question of strategy-proofness against residents in SMP-C. We show that in general, no resident strategy-proof mechanism exists for SMP-C, but there is a mechanism that is resident strategy-proof with respect to truncations when the SMP-C instance has an $\mathcal{R}_{opt}$ matching. Furthermore, we provide an algorithm for implementing this mechanism. When no $\mathcal{R}_{opt}$ matching exists this mechanism will return a resident Pareto optimal ($\mathcal{RP}_{opt}$) matching when the instance has at least one stable matching. We show that residents who have a unique match among the set of $\mathcal{RP}_{opt}$ matchings cannot manipulate this mechanism by truncating their preferences. We also show, however, that no stable mechanism can be strategy-proof when the residents can use the more general manipulation of reordering their preferences, even in the setting where a unique stable matching exists.

We empirically evaluate the set of stable and $\mathcal{RP}_{opt}$ matchings in instances drawn from two different synthetic markets (previously developed by Kojima et al. and Biró et al. [13, 15]). For problem instances of similar size and percentage of couples as the NRMP data investigated by Roth and Peranson, we find that nearly all problem instances admit an $\mathcal{R}_{opt}$ matching, and the DA-style algorithm presented by Roth and Peranson almost always finds that $\mathcal{R}_{opt}$ matching. Thus, in conjunction with our theoretical work, we provide an alternate hypothesis for behaviour seen by Roth and Peranson: any mechanism that returns a $\mathcal{R}_{opt}$ matching will be resident strategy-proof with respect to truncations. We additionally find that the theoretical result of Immorlica and Mahdian [8] for large markets appears to hold for SMP-C, even though that result was only proved for SMP.

Section 2 formally defines SMP-C and $\mathcal{RP}_{opt}$ matches. Section 3 discusses strategy-proofness in SMP-C and provides some theoretical results for this question. Section 4 describes our algorithms for finding $\mathcal{RP}_{opt}$ matchings in SMP-C. These algorithms are extensions of a prior SAT based algorithm for SMP-C [6]. Section 5 provides our empirical results, investigating properties of the set of $\mathcal{RP}_{opt}$ matchings as the size and percentage of couples in the market varies. We conclude in Section 6.

2 Background

The stable matching problem with couples (SMP-C) can be formalized in terms of the residency matching problem [21]. In this problem doctors wish to be matched with a hospital residency program, and programs wish to accept some number of residents. Both doctors and hospitals have preferences over who they are matched with, expressed as ranked order lists (ROLs). Some doctors are members of a couple, and these couples provide a joint ROL. Both doctors and hospitals ROLs can be incomplete: any alternative not listed in their ROL is considered to be unacceptable. That is, they would rather not be matched at all than be matched to an alternative not on their ROL. The SMP-C problem is to find a *stable* matching, such that no doctor-hospital pair has an incentive to defect from the assigned matching.

2.1 SMP-C

More formally, let D be a set of doctors and P be a set of programs. Since there is a preference to be unmatched over an unacceptable match, we use *nil* to denote this alternative: matching a program p to *nil* indicates that p has an unfilled slot while matching a doctor d to *nil* indicates that d was not placed into any program. Let D^+ and P^+ denote the sets $D \cup \{nil\}$ and $P \cup \{nil\}$ respectively.

The doctors are partitioned into two subsets, $S \subseteq D$ and $D \setminus S$. S is the set of single doctors and $D \setminus S$ is the set of doctors that are in couple relationships. Couples are specified by a set of pairs $C \subseteq (D \setminus S) \times (D \setminus S)$. If $(d_1, d_2) \in C$ we say that d_1 and d_2 are each other's partner. We require that every doctor who is not single (i.e., every doctor in $D \setminus S$) have one and only one partner in C. Each program $p \in P$ has an integer capacity $cap_p > 0$ specifying the maximum number of doctors p can accept.

Everyone participating in the matching market has preferences over their possible matches. Each participant a specifies their preferences in a ROL, rol_a, which lists a's preferred matches from most preferred to least preferred. The ROLs of single doctors $d \in S$ contain programs from P^+; the ROLs of couples

$c \in C$ contain pairs of programs from $P^+ \times P^+$; and the ROLs of programs $p \in P$ contain doctors from D^+. Every ROL is terminated by *nil* (couple ROLs are terminated by (nil, nil)) since being unmatched is always the least preferred option, but is preferred to any option not on the ROL.

The order of items on a's ROL defines a partial ordering relation where $x \succeq_a y$ indicates that x appears before y on a's ROL or $x = y$. We define $\succ_a$, $\preceq_a$, and $\succ_a$ in terms of $\succeq_a$ and equality in the standard way. We say that x is *acceptable* to a if $x \succeq_a nil$. (Note that unacceptable matches are not ordered by $\succeq_a$ and that *nil* is always acceptable.)

We define a choice function $Ch_p()$ for programs $p \in P$. Given a set of doctors R, $Ch_p(R)$ returns the subset of R that p would prefer to accept. In our setting $Ch_p(R)$ returns the maximal subset of R such that for all $d \in Ch_p(R)$, $d \succ_p nil$, for all $d' \in R - Ch_p(R)$, $d \succ_p d'$, and $|Ch_p(R)| \leq cap_p$. That is, all doctors in $Ch_p(R)$ are acceptable, are strictly preferred to all doctors not chosen, and p's capacity is not violated. It is also convenient to give the null program a choice function as well: $Ch_{nil}(O) = O$, i.e., *nil* will accept any and all matches.

We use the notation $ranked(a)$ to denote the set of options that a could potentially be matched with. For single doctors d and programs p this is simply the ROLs of d (rol_d) and p (rol_p). For a doctor that is part of a couple (d_1, d_2), $ranked(d_1) = \{p_1 | \exists p_2.(p_1, p_2) \in rol_{(d_1,d_2)}\}$ and similarly for d_2. Note that $nil \in ranked(a)$.

Finally, we will sometimes use rol_a as an indexable vector (zero-based). We define the function $rank_a(x)$ to find the index i of x in a's rol: $rank_a(x) = i$ iff $rol_a[i] = x$. When $x \notin rol_a$ we let $rank_a(x) = |rol_a|$ (an out of bounds index).

2.2 Stable Matchings

Definition 1 (Matching) A **matching** μ is a mapping from D to P^+. We say that a doctor d is matched to a program p under μ if $\mu(d) = p$, and that p is matched to d if $d \in \mu^{-1}(p)$.

We want to find a **stable** matching where no doctor-program pair has an incentive to defect. We call the pairs that do have an incentive to defect blocking pairs. First we define the condition $willAccept(p, R, S)$ to mean that given the set of doctors $R \cup S$, p will accept a set that includes R: $willAccept(p, R, S) \equiv R \subseteq Ch_p(S \cup R)$.

Definition 2 (Blocking Pairs) Let μ be a matching.

1. A single doctor $d \in S$ and a program $p \in P$ is a **blocking pair** for μ if $p \succ_d \mu(d)$ and $willAccept(p, \{d\}, \mu^{-1}(p))$.
2. A couple $c = (d_1, d_2) \in C$ and a program pair $(p_1, p_2) \in P^+ \times P^+$ with $p_1 \neq p_2$ is a **blocking pair** for μ if and only if $(p_1, p_2) \succ_{(d_1,d_2)} (\mu(d_1), \mu(d_2))$, $willAccept(p_1, \{d_1\}, \mu^{-1}(p_1))$, and $willAccept(p_2, \{d_2\}, \mu^{-1}(p_2))$.
3. A couple $c = (d_1, d_2)$ and a program $p \in P$ is a **blocking pair** for μ if $(p, p) \succ_{(d_1,d_2)} (\mu(d_1), \mu(d_2))$ and $willAccept(p, \{d_1, d_2\}, \mu^{-1}(p))$.

Definition 3 (Individually Rational) A matching μ is **individually rational** if (a) for all $d \in S$, $\mu(d) \succeq_d nil$, (b) for all $c = (d_1, d_2) \in C$, $(\mu(d_1), \mu(d_2)) \succeq_c (nil, nil)$, and (c) for all $p \in P$, $|\mu^{-1}(p)| \leq cap_p$ and for all $d \in \mu^{-1}(p)$, $d \succeq_p nil$.

Definition 4 (Stable Matching) A matching μ is **stable** if it is individually rational and no blocking pairs for μ exist.

2.3 Resident Preferred Matchings

The set of stable matchings can be quite large. In SMP, where there are no couples this set is always non-empty [7] and has a nice structure: it forms a lattice under the partial order $\succeq_{\mathcal{R}}$ defined as follows [11].

Definition 5 (Resident Preferred) A matching μ_1 is **resident preferred** to another matching μ_2, denoted by $\mu_1 \succeq_{\mathcal{R}} \mu_2$, if for all $a \in S \cup C$ we have that $\mu_1(a) \succeq_a \mu_2(a)$. We also define $\succ_{\mathcal{R}}$, $\prec_{\mathcal{R}}$, and $\preceq_{\mathcal{R}}$ in terms of $\succeq_{\mathcal{R}}$ and equality in the standard way. In particular, $\mu_1 \succ_{\mathcal{R}} \mu_2$ whenever $\mu_1 \succeq_{\mathcal{R}} \mu_2$ and for at least one $a \in S \cup C$ we have that a strictly prefers μ_1 to μ_2 ($\mu_1(a) \succ_a \mu_2(a)$). When $\mu_1 \succ_{\mathcal{R}} \mu_2$ we say that μ_1 **dominates** μ_2.

Definition 6 (Resident Optimal) We say that a matching μ is **resident optimal**, written $\mathcal{R}_{opt}(\mu)$ if μ is stable and it dominates all other stable matches. That is, for all stable matches μ' with $\mu' \neq \mu$ we have that $\mu \succ_{\mathcal{R}} \mu'$.

Note that we restrict the resident-optimal matching to be stable. It can be observed that when a matching is resident optimal ($\mathcal{R}_{opt}(\mu)$) then for any $a \in S \cup C$, $\mu(a)$ is the best match for a offered by any stable matching.

Resident optimality is generally cited as an important property for stable matching algorithms (e.g., [7, 21]). In SMP the fact that the stable matchings form a lattice under $\succeq_{\mathcal{R}}$ implies that a resident-optimal matching always exists. In the presence of couples however, stable matchings may not exist and even when they do an $\mathcal{R}_{opt}$ matching might not exist. However, the $\succeq_{\mathcal{R}}$ relation is still well defined, and for SMP-C leads to potentially multiple (resident) *Pareto-optimal* matchings.

Definition 7 (Resident Pareto Optimal) We say that a matching μ is **resident Pareto optimal**, written $\mathcal{RP}_{opt}(\mu)$ if μ is stable and there does not exist another stable matching μ' such that $\mu' \succ_{\mathcal{R}} \mu$.

It is easy to see that in SMP-C every stable matching μ is either an $\mathcal{RP}_{opt}$ or is dominated by an $\mathcal{RP}_{opt}$ matching. This also means that an SMP-C instance has an $\mathcal{R}_{opt}$ matching if and only if it has a unique $\mathcal{RP}_{opt}$ matching.

2.4 Mechanisms and Strategy-Proofness

Definition 8 (Mechanisms) A **mechanism** for SMP-C is any algorithm that takes an SMP-C instance as input and returns a matching. A **stable mechanism** is a mechanism that always returns a stable matching if one exists and otherwise returns the empty matching (i.e., everyone is matched to *nil*).[1] A $\mathcal{P}$ **mechanism** is one that always returns a matching satisfying property $\mathcal{P}$ if one exists and the empty matching otherwise.

Definition 9 (Manipulation) Let rol^*_α be the complete, true preferences of resident or couple α. rol^*_α consists of a number of programs or program pairs, then the *nil* program, and then the remaining programs or program pairs. α is **truthful** if their reported rol_α is the same as rol^*_α up to and including the *nil* program. α **manipulates** if they are not truthful. Any manipulation is a **reordering** of rol^*_α. A manipulation is a **truncation at rank i** if the reported rol_α is the first i elements of rol^*_α followed by *nil*.

Definition 10 (Strategy-Proof) A mechanism m for SMP-C is **resident strategy-proof** when for every SMP-C instance being truthful is a dominant strategy for every resident and couple. That is, no resident or couple in any SMP instance can improve their matching under m by manipulating. When no resident or couple can improve their matching using only truncation manipulations, we say that m is **resident strategy-proof against truncation**.

Definition 11 ($\mathcal{P}$ programs) Let α be a resident or couple in an SMP-C instance I. A program p (pair of programs) is a $\mathcal{P}$ **program** for α in I if there exists a matching μ for I such that μ satisfies property $\mathcal{P}$ and $p = \mu(\alpha)$.

3 Strategy-proofness in SMP-C

The current NRMP mechanism was developed partly in response to concerns that the previous mechanism could be manipulated by residents by misreporting their ranking of programs [21]. In one-to-one SMP, it is possible to achieve *strategy-proofness* for either side of the market by using a mechanism that returns the resident or program-optimal matching[2] [26]. The administrators of the NRMP perceived that residents had a much higher average difference in utility between adjacent elements on their *rol*; thus, the NRMP decided that they wanted a mechanism that was as close to resident strategy-proof as possible.

Roth and Peranson hypothesized that since there is a relatively small fraction of couples in the NRMP (4% in their historical data), the resident strategy-

[1] An SMP-C instance might have no stable matchings, so we cannot define a stable mechanism to be one that **always** returns a stable matching as in [22].

[2] Strategy-proofness for both sides of the market is only possible for a small number of instances of SMP [22]. In many-to-one SMP, only strategy-proofness on the side of the residents is possible [24].

proof mechanism for SMP that returns the $\mathcal{R}_{opt}$ matching could be generalized to SMP-C and the result would be similar [21]. Although their mechanism was not in fact an $\mathcal{R}_{opt}$ mechanism it was quite successful empirically—less than 10 residents (out of 30,000) had an incentive to manipulate via truncation. While not all manipulations are truncations, they tested for truncations for several reasons: i) any outcome that can be achieved by manipulation in SMP can be achieved by a truncation [27] ii) profitable truncations are computationally inexpensive to check for, and iii) truncations are the kind of manipulations that can be potentially identified with the least information about others' preferences [25].

To our knowledge, no theoretical strategy-proofness results exist for SMP-C. We hypothesized that if we restricted our attention to SMP-C instances in which an $\mathcal{R}_{opt}$ matching exists, then an $\mathcal{R}_{opt}$ mechanism would be resident strategy-proof. As shown in Thm. 2 below, we found that this is not the case. We did find, however, that such a mechanism is resident strategy-proof against truncations. This is useful in practice. In particular, as we will show in Sec. 5, an $\mathcal{R}_{opt}$ matching frequently exists in SMP-C instances that have a low proportion of couples. Furthermore, truncations are an attractive way to manipulate because of the computational and informational properties mentioned above. Hence, an $\mathcal{R}_{opt}$ mechanism can at least block this simpler form of manipulation in many practical cases.

We begin with a lemma that establishes a limit on the ability of truncating agents to benefit from manipulation. All of the proofs from this section are included in Appendix A.

Lemma 1 *Let α be an agent who truncates their preferences at rank i in an SMP-C instance $\mathcal{I}$. Let Ω be the set of stable matchings before the truncation and let Ω' be the set of stable matchings after the truncation. For any $\mu \in \Omega'$, either i) $\mu \in \Omega$ or ii) $\mu(\alpha) = nil$.*

Theorem 1 *Let α be a resident or couple in an SMP-C instance $\mathcal{I}$. For any property $\mathcal{P}$, if α has a unique $\mathcal{P}$ program in $\mathcal{I}$, then for any $\mathcal{P}$ mechanism $y_{\mathcal{P}}$, α cannot improve its matching under $y_{\mathcal{P}}$ using only truncation manipulations (α has no incentive to manipulate under $y_{\mathcal{P}}$).*

A corollary is that in an instance of SMP-C that has an $\mathcal{R}_{opt}$ matching, a $\mathcal{R}_{opt}$ mechanism is resident strategy-proof against truncation. However, such mechanisms are not resident strategy-proof, as reordering manipulations can still be beneficial.

Theorem 2 *Let $y_{\mathcal{R}_{opt}}$ be an $\mathcal{R}_{opt}$ mechanism for SMP-C. Then $y_{\mathcal{R}_{opt}}$ is strategy-proof against residents who manipulate via truncation. However, $y_{\mathcal{R}_{opt}}$ is not strategy-proof against residents who manipulate via reordering.*

We can strengthen the latter result of Thm. 2.

Theorem 3 *(Biró and Klijn [4]) No stable mechanism for SMP-C is resident strategy-proof.*

Thus, reorderings are a powerful way of manipulating in SMP-C. Reorderings can remove old stable matchings and create new ones allowing great scope for beneficial manipulations. In fact, for SMP-C, strategy-proofness is a very strong requirement due to diversity of instances. Hence, we can further strengthen Thm. 3 to show that even less general manipulations by truncations suffice to foil strategy-proofness for stable mechanisms.

Theorem 4 *No stable mechanism for SMP-C is resident strategy-proof against truncations.*

In this situation it is unclear what mechanism to use for SMP-C when an $\mathcal{R}_{opt}$ matching does not exist. By extending the analogy with SMP, it would be intuitive for $\mathcal{RP}_{opt}$ mechanisms to be harder to manipulate via truncations than other stable mechanisms. However, Lemma 1 states that a truncating agent a may create new stable matchings where they are matched to *nil*. A stable mechanism might return one of these matching. Therefore, a might have a disincentive to manipulate by truncation because the mechanism might now return a matching in which a is unmatched. However, these new matches of a to *nil* might not be $\mathcal{RP}_{opt}$. In this case they would not be returned by an $\mathcal{RP}_{opt}$ mechanism. Hence, with an $\mathcal{RP}_{opt}$ mechanism a might no longer have a disincentive to manipulate. An instance where this occurs is shown in Figure 7 in the appendix. The instance shown has stable mechanisms that are strategy-proof against truncations, but some $\mathcal{RP}_{opt}$ mechanisms are not.

Despite this difficulty in comparing the strategy-proofness of $\mathcal{RP}_{opt}$ mechanisms with other stable mechanisms, we will show in Sec. 5 that empirically Thm. 1 can be used to show that often a larger number of residents will have no incentive to manipulate an $\mathcal{RP}_{opt}$ mechanism than a stable mechanisms.

4 Algorithms for SMP-C

The standard approach to finding a stable matching in SMP-C has been to extend the deferred acceptance algorithm so that it can handle couples (e.g., [21, 12, 3]). However, these extensions are incomplete: they are unable to determine whether or not a stable matching exists, and even when a stable matching does exist they might not be able to find one.

Since SMP-C is known to be NP-Complete [20] it is also possible to encode it as another NP-Complete problem. For example, it can be encoded as a SAT (satisfiability) problem or as IP (integer program) problem [6, 5]. The advantage of doing this is that SAT and IP solvers have become very advanced and are routinely able to solve large practical problems. Another advantage of these solvers is that when given an instance for which no stable matching exists, they are often able to prove this.

4.1 DA-Style Algorithms for SMP-C

The basic principle of DA algorithms [7] is that members of one side of the market propose down their ROLs while the other side either rejects those proposals or holds them until they see a better proposal: once all proposals have been made the non-rejected proposals are accepted forming a match.

Roth and Peranson develop a DA algorithm, **RP99**, capable of dealing with couples [21]. This algorithm has been used with considerable success in practice, including most famously for finding matches for the NRMP which typically involves about 30,000 doctors [16]. RP99 employs an iterative scheme. After computing a stable matching for all single doctors, couples are added one at a time and a new stable matching computed after each addition. The algorithm uses DA at each stage to find these stable matchings. Matching a couple can make previously made matches unstable and in redoing these matches the algorithm might start to cycle. Hence, cycle checking (or a timeout) is sometimes needed to terminate the algorithm.

Kojima et al. develop a simple "sequential couples algorithm," which they use to show that the probability of a stable matchings existing goes to one under certain assumptions [12]. However, this simple algorithm is not useful in practice as it declares failure under very simple conditions. Kojima et al. also provide a more practical DA algorithm, **KPR**, that they use in their experiments. The main difference between KPR and RP99 is that KPR deals with all couples at the same time—it does not attempt to compute intermediate stable matchings. Drummond et al. found that KPR was much more efficient than RP99 [6].

Finally, Ashlagi et al. extend the analysis of Kojima et al., developing a more sophisticated "Sorted Deferred Acceptance Algorithm" and analyzing its behaviour [3]. This algorithm is designed mainly to be amenable to theoretical analysis rather than for practical application.

4.2 Solving SMP-C via SAT

Drummond et al. developed SAT-E, an effective encoding of SMP-C into SAT, and evaluated its performance on synthesized SMP-C problem instances [6]. They found that their encoding, used in conjunction with a state-of-the art SAT solver, scaled well, outperformed an IP encoding they also developed, and could find solutions to problem instances that the DA-style algorithms could not.

Given an SMP-C instance $\langle D, C, P, ROLs\rangle$, where $ROLs$ is the set of all participant ROLs, SAT-E($\langle D, C, P, ROLs\rangle$) can be viewed as a function that returns a SAT encoding in CNF (conjunctive normal form), which is the input format taken by modern SAT solvers.

We highlight three important things about SAT-E:

1. For any SMP-C instance $\mathcal{I} = \langle D, C, P, ROLs\rangle$, the satisfying models of SAT-E stand in a one-to-one correspondence with the stable models of $\mathcal{I}$.
2. SAT-E includes the set of propositional variables $m_d[p]$. In any satisfying model, π, $m_d[p]$ is true if and only if $\mu(d) = p$ in the stable matching μ corresponding to π.

ALGORITHM 1: SAT-$\mathcal{RP}_{opt}$. Given an SMP-C instance return a $\mathcal{RP}_{opt}$ matching or the empty match if none exists.

Input: $\mathcal{I} = \langle D, C, P, ROLs \rangle$ an **SMP-C** instance.
Output: An $\mathcal{RP}_{opt}$ matching for $\mathcal{I}$.

CNF $\leftarrow$ SAT-E($\mathcal{I}$)
$\mu \leftarrow \emptyset$
while *true* **do**
    (sat?,π) $\leftarrow$ SatSolve(CNF)
    /* *SatSolve returns the status (sat or unsat) and a satisfying model π if sat* */
    if *not* *sat?* **then**
        return μ // *Return last match found.*
    $\mu \leftarrow$ stable matching corresponding to π
    $c \leftarrow \{\neg m_d[p] \,|\, \mu(d) = p\}$ // *Block this match*
    CNF $\leftarrow$ CNF $\cup \{c\}$
    for $d \in S$ **do** //d must get an equally good match
        $c_d \leftarrow \{m_d[p] \,|\, p \succeq_d \mu(d)\}$
        CNF $\leftarrow$ CNF $\cup \{c_d\}$
    for $c \in C$ **do** //c must get an equally good match
        $c_c \leftarrow \{m_c[i] \,|\, i = rank_c(\mu(c))\}$
        CNF $\leftarrow$ CNF $\cup \{c_c\}$

3. SAT-E also includes the set of propositional variables $m_c[i]$. In any satisfying model, π, $m_c[i]$ is true if and only if in μ, the stable matching corresponding to π, c is matched to a program pair they rank i or above in their ROL.

4.3 Finding $\mathcal{RP}_{opt}$ matchings

To implement an $\mathcal{RP}_{opt}$ mechanism we need to find an $\mathcal{RP}_{opt}$ matching. Using SAT-E and the above three facts, we provide SAT-$\mathcal{RP}_{opt}$ (Algorithm 1) for finding an $\mathcal{RP}_{opt}$ matching. This algorithm will return an $\mathcal{R}_{opt}$ matching if one exists (as an $\mathcal{R}_{opt}$ matching exists if and only if there is there is only one $\mathcal{RP}_{opt}$ matching).

SAT-$\mathcal{RP}_{opt}$ takes an SMP-C instance as input and constructs the SAT-E encoding for that instance. It finds a stable matching μ and then tries to find a new matching that dominates μ. This is accomplished by adding a *blocking clause* c to the SAT encoding. This clause c is a disjunction that says that no future solution is allowed to return the same stable model (one of the mappings $\mu(d) = p$ of all future solutions must be different, i.e., one of the variables $m_d[p]$ made true by μ must be false in every future matching). Along with the blocking clause, other clauses are added to ensure that in the next matching no doctor or couple receives a worse matching. For single doctors d a clause is added that says that d must be matched to a program it ranks at least as high as $\mu(d)$, and for every couple c a (unit) clause that says that c must be matched to a pair of programs it ranks at least as highly as $\mu(c)$. This causes the new match to be

$\succeq_{\mathcal{R}} \mu$, and since the new match cannot be equal it must be $\succ_{\mathcal{R}} \mu$. That is, the new match must dominate μ. If no dominating match can be found, the current match is $\mathcal{RP}_{opt}$ and we return it. Note that an instance has no $\mathcal{RP}_{opt}$ matching if and only if it has no stable matching. So if the very first SAT call fails, there is no $\mathcal{RP}_{opt}$ matching and the empty matching will be returned.

In our experiments (Sec. 5) we also need to find all the stable matchings of an instance. This can be accomplished with SAT-ENUM, which uses a similar technique. The details are given in Appendix B.

5 Empirical Results

We use SAT-ENUM to find all stable matchings for two distributions of synthetic data, which we call: i) impartial culture with geography (IC-GEOG), and ii) SFAS-tuned with geography (SFAS-GEOG). IC-GEOG generates heterogeneous preference profiles; each resident has an equal chance of drawing any program as her first choice. SFAS-GEOG generates more homogeneous profiles; certain programs are much more likely to be in a given resident's top 10. The models have slightly different procedures for generating couples preferences, but both models assign each program a geographic region and require that, for any program pair in a couple's ROL, the programs are in the same region. We describe the models in detail in Appendix C. In both models, we set the number of positions as 87% of the number of residents to match the 2015 NRMP data [17]. Each data point represents the average of 50 instances. In the next section, we evaluate how frequently these instances admit $\mathcal{R}_{opt}$ and unique stable matchings for various parameters of the markets.

5.1 Results and Discussion

We begin by investigating the ability of residents to benefit by truncating their preferences in SMP-C as the size of the market gets large. Roth and Peranson found that, when using RP99 on historical NRMP data, very few residents had an incentive to truncate [21]. They conjectured that this was due to two reasons: i) as market size grows large relative to ROL length, the size of the set of all stable matches gets smaller, and ii) as in SMP, residents have no incentive to manipulate when a market has a unique stable matching. In Theorem 2, we prove the latter point for truncations, but show that it is false for reorderings. Immorlica and Mahdian proved the former point for SMP [8], and we address it empirically for SMP-C in this section.

Figure 1 shows the average number of *additional* stable programs per resident (i.e., in excess of one) for 10% and 30% residents in couples in IC-GEOG and SFAS-GEOG models as the market size increases.[3] Note that every resident has at least one stable program because we restrict our attention to SMP-C instances that

[3] Standard error in this graph is 0.003-0.005% for the 30% couples models with market size under 1000 programs and less than 0.001% otherwise.

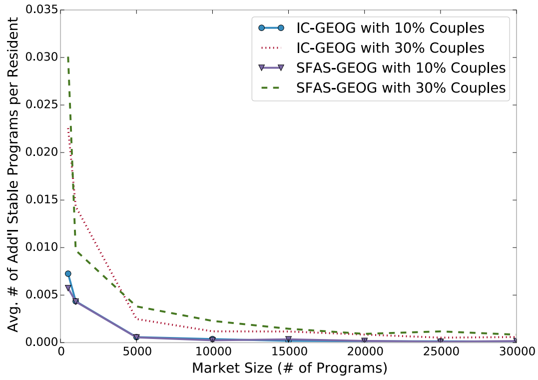


Fig. 1. Average number of additional stable programs per resident as market size increases, only including satisfiable instances.

are satisfiable. The models with a larger percentage of couples have more stable programs per resident, but the differences between IC-GEOG and SFAS-GEOG are small. We see that the average number of stable programs per resident drops rapidly as the market size increases, irrespective of model, which shows that an analog of Immorlica and Mahdian's result may apply in this setting.

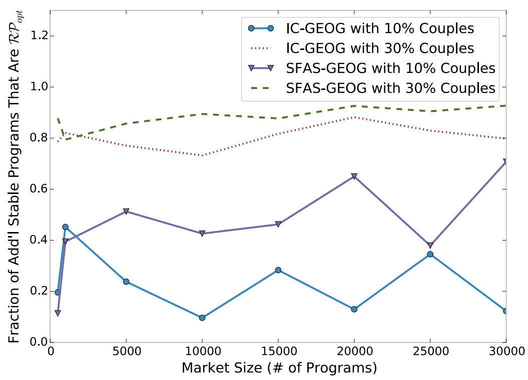


Fig. 2. Average number of additional $\mathcal{RP}_{opt}$ matchings per resident divided by average number of additional stable matchings per resident as market size increases, only including satisfiable instances.

In addition, we find that the residents' ability to truncate is even more strongly curtailed under an $\mathcal{RP}_{opt}$ mechanism. Figure 2 shows the average number of additional $\mathcal{RP}_{opt}$ programs per resident relative to the average number of additional stable programs per resident. We see that generally residents have much fewer additional $\mathcal{RP}_{opt}$ programs than additional stable programs. Since Figure 1 shows that the average number of additional stable programs is well below 1, this means that many residents have a unique stable program and even more will have a unique stable $\mathcal{RP}_{opt}$ program. Hence, by Theorem 1 there will be more residents with no incentive to manipulate an $\mathcal{RP}_{opt}$ mechanism than there are residents with no incentive to manipulate a stable mechanism.

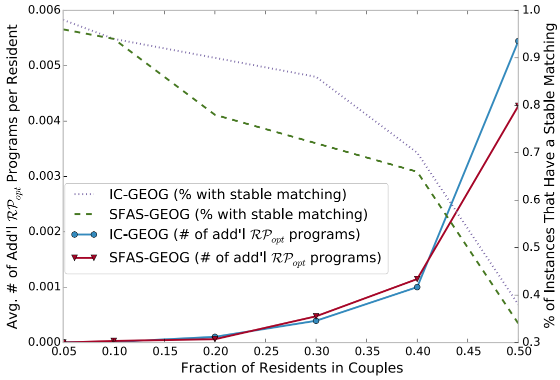


Fig. 3. Average number of additional $\mathcal{RP}_{opt}$ programs per resident, and overall percentage of satisfiable instances, as fraction of residents in couples increases. 20,000 programs.

Figure 3 provides a more detailed picture of what happens to the number of additional $\mathcal{RP}_{opt}$ programs per resident as the fraction of residents in couples increases for a fixed market size of 20,000. We see that the two models behave similarly, showing a superlinear growth. On the other axis, we see that although the average number of $\mathcal{RP}_{opt}$ programs per resident increases rapidly as the fraction of couples increases, the percentage of instances with a stable matching decreases. With 5% couples, almost all instances had at least one stable matching for IC-GEOG and SFAS-GEOG, while with 50% couples, fewer than 40% do. Thus, mechanism designers in environments with a high fraction of couples will have to contend with both the lack of existence of stable matchings and high incentives to truncate among residents.

We can conclude from Figure 2 that, under an $\mathcal{RP}_{opt}$ mechanism, residents will have substantially fewer programs they can potentially be matched to as the result of a truncation. Combining this result with Figure 3, this effect is particularly strong for low percentages of couples in the market. We find that both RP99 and KPR behave like $\mathcal{RP}_{opt}$ mechanisms in instances with a fraction of couples similar to that found in the NRMP in the 1990s (4% [21]). In particular, for instances with 20,000 programs and 5% couples and multiple stable matchings, KPR found an $\mathcal{RP}_{opt}$ matching 100% of the time and RP99 found an

$\mathcal{RP}_{opt}$ matching more than 93% of the time.[4] Thus, i) RP99 is nearly an $\mathcal{RP}_{opt}$ mechanism, ii) $\mathcal{RP}_{opt}$ mechanisms can rarely be manipulated by resident truncations on markets with a low percentage of couples, and iii) when they can, only an extremely small number of residents can benefit. We thus hypothesize that the number of residents who could benefit by truncating observed by Roth and Peranson was made lower by their exploitation of $\mathcal{RP}_{opt}$ matchings rather than a small number of stable matchings as they conjectured.

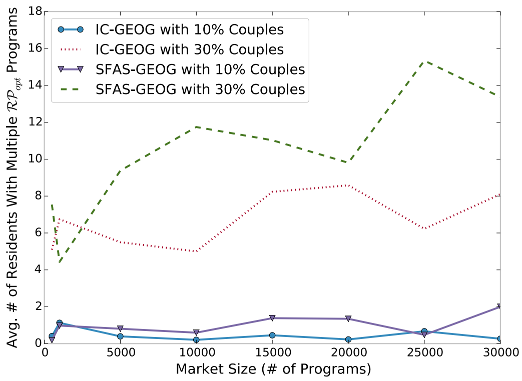


Fig. 4. Average number of residents with multiple $\mathcal{RP}_{opt}$ programs as market size increases, only including satisfiable instances.

However, there is a peculiar caveat to these results: the number of residents per instance with multiple stable or $\mathcal{RP}_{opt}$ programs does not seem to be affected by market size. Figure 4 shows the average number of residents with multiple $\mathcal{RP}_{opt}$ programs for 10% and 30% residents in couples in IC-GEOG and SFAS-GEOG models as the market size increases.[5] We see that the number of residents with multiple $\mathcal{RP}_{opt}$ programs does not decrease as market size increases. We observed similar behaviour for the number of residents with multiple stable matchings. Comparing IC-GEOG and SFAS-GEOG, we see that SFAS-GEOG has more residents with multiple $\mathcal{RP}_{opt}$ programs, which is consistent with the higher fraction of $\mathcal{RP}_{opt}$ programs in SFAS-GEOG that we previously observed. Unsurprisingly, the number of residents with multiple stable or $\mathcal{RP}_{opt}$ programs is higher when the fraction of couples is increased, since an instance with a higher fraction of couples tends to have a larger number of stable and $\mathcal{RP}_{opt}$ matchings.

[4] There was a problem instance where an $\mathcal{R}_{opt}$ matching existed, and RP99 did not find it.

[5] Standard error in this graph is 0.3-0.5 for the 10% couples models and 2.5-3.5 for the 30% couples models.

Since it only requires a single resident with multiple $\mathcal{RP}_{opt}$ programs to prevent the existence of a $\mathcal{R}_{opt}$ matching, we likewise find that the frequency of instances that have $\mathcal{R}_{opt}$ matchings is not affected by market size. $\mathcal{R}_{opt}$ matchings exist 94.2% and 70.1% of the time for IC-GEOG with 10% and 30% couples and 91.1% and 62.9% of time for SFAS-GEOG with 10% and 30% couples.

These results suggest that Roth and Peranson's observation that few residents could benefit by truncating, was not affected by the size of the market, further suggesting their observation was only dependent on the percentage of residents in couples. While there may be some residents that can benefit from truncating irrespective of market size, we expect the fact that the number of them is so small to be a strong disincentive to truncating in practice. It requires considerable effort (i.e., learning the preferences of others) to be able to truncate effectively, and these efforts will be very unlikely to yield any benefit in sufficiently large instances. In addition, truncating carries the risk that a resident will become unmatched even under an $\mathcal{RP}_{opt}$ mechanism.

6 Conclusions

In this paper we examined strategy-proofness for SMP-C, showing that in general, no resident strategy-proof stable mechanism exists. We do show that, for certain problem instances, a stable matching mechanism that always returns an $\mathcal{RP}_{opt}$ matching is strategy-proof w.r.t. residents truncating their preferences, and we extend a previously developed SAT encoding to provide an $\mathcal{RP}_{opt}$ mechanism. We empirically show, on two very different preference distributions based on real-world markets, that when a low percentage of couples exist in the market, the $\mathcal{RP}_{opt}$ mechanism is frequently resistant to truncations, suggesting a new hypothesis for why there is little incentive for residents to truncate their preferences in the NRMP.

Furthermore, our empirical results suggest new avenues for theoretical research: we hypothesize that an analogous SMP-C result exists for the Immorlica and Mahdian large markets result for SMP [8]; we additionally hypothesize that if the percentage of couples in the market grows slower than the size of the market, an $\mathcal{R}_{opt}$ matching exists with high probability (an analogous result to the Ashlagi et al. large markets result [3]). We also wish to further use our extension to SAT-E to allow us to empirically explore different properties of SMP-C, providing further insight into possible theoretical results.

References

1. Abdulkadiroglu, A., Pathak, P., Roth, A.E., Sönmez, T.: The Boston public school match. American Economic Review 95(2), 368–371 (2005)
2. Abdulkadiroglu, A., Pathak, P., Roth, A.E., Sonmez, T.: Changing the Boston school choice mechanism. Tech. rep., National Bureau of Economic Research (2006)
3. Ashlagi, I., Braverman, M., Hassidim, A.: Stability in large matching markets with complementarities. Operations Research 62(4), 713–732 (2014)

4. Biró, P., Klijn, F.: Matching with couples: a multidisciplinary survey. International Game Theory Review 15(02), 1340008 (2013)
5. Biró, P., Manlove, D.F., McBride, I.: The hospitals/residents problem with couples: Complexity and integer programming models. In: Experimental Algorithms, pp. 10–21. Springer (2014)
6. Drummond, J., Perrault, A., Bacchus, F.: SAT is an effective and complete method for solving stable matching problems with couples. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI-15) (2015)
7. Gale, D., Shapley, L.S.: College admissions and the stability of marriage. American Mathematical Monthly 69(1), 9–15 (1962)
8. Immorlica, N., Mahdian, M.: Marriage, honesty, and stability. In: Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 53–62 (2005)
9. Irving, R.: Matching practices for entry-labor markets - scotland (2011), `http://www.matching-in-practice.eu/the-scottish-foundation-allocation-scheme-sfas/`, accessed: 2015-11-14
10. Kannan, S., Morgenstern, J., Roth, A., Wu, Z.S.: Approximately stable, school optimal, and student-truthful many-to-one matchings (via differential privacy). In: Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 1890–1903 (2015)
11. Knuth, D.E.: Marriages stables. Les Presses de I'Universite de Montreal (1976)
12. Kojima, F., Pathak, P., Roth, A.E.: Matching with couples: Stability and incentives in large markets. Quarterly Journal of Economics 128(4), 1585–1632 (2013)
13. Kojima, F., Pathak, P., Roth, A.E.: Online appendix matching with couples: Stability and incentives in large markets. Quarterly Journal of Economics 128(4) (2013), `http://economics.mit.edu/files/9029`
14. Luce, R.D.: Individual Choice Behavior: A Theoretical Analysis. Wiley (1959)
15. McBride, I., Manlove, D.F.: The hospitals / residents problem with couples: Complexity and integer programming models. CoRR abs/1308.4534 (2013), `http://arxiv.org/abs/1308.4534`
16. National Resident Matching Program: National resident matching program, results and data: 2013 main residency match® (2013)
17. National Resident Matching Program: National resident matching program, results and data: 2015 main residency match® (2015)
18. Niederle, M., Roth, A.E., Sonmez, T.: Matching and market design. In: Durlauf, S.N., Blume, L.E. (eds.) The New Palgrave Dictionary of Economics (2nd Ed.), vol. 5, pp. 436–445. Palgrave Macmillan, Cambridge (2008)
19. Plackett, R.: The analysis of permutations. Applied Statistics 24, 193–202 (1975)
20. Ronn, E.: NP-complete stable matching problems. Journal of Algorithms 11(2), 285–304 (1990)
21. Roth, A.E., Peranson, E.: The redesign of the matching market for American physicians: Some engineering aspects of economic design. The American Economic Review 89(1), 748–780 (September 1999)
22. Roth, A.E.: The economics of matching: Stability and incentives. Mathematics of Operations Research 7(4), 617–628 (1982)
23. Roth, A.E.: The evolution of the labor market for medical interns and residents: A case study in game theory. Journal of Political Economy 92(6), 991–1016 (1984)
24. Roth, A.E.: The college admissions problem is not equivalent to the marriage problem. Journal of economic Theory 36(2), 277–288 (1985)
25. Roth, A.E., Rothblum, U.G.: Truncation strategies in matching markets—in search of advice for participants. Econometrica 67(1), 21–43 (1999)

26. Roth, A.E., Sotomayor, M.: Chapter 16. two-sided matching. In: Aumann, R.J., Hart, S. (eds.) Handbook of Game Theory Volume 1, pp. 485–541. Elsevier (1992)
27. Roth, A.E., Vande Vate, J.H.: Random paths to stability in two-sided matching. Econometrica 58(6), 1475–1480 (1990)

A Proofs

Lemma 1 *Let α be an agent who truncates their preferences at rank i in an SMP-C instance $\mathcal{I}$. Let Ω be the set of stable matchings before the truncation and let Ω' be the set of stable matchings after the truncation. For any $\mu \in \Omega'$, either i) $\mu \in \Omega$ or ii) $\mu(\alpha) = nil$.*

Proof. Let μ' be a matching in Ω' where α is not matched to *nil*. It must be the case that $\mu'(\alpha)$ is ranked above i; otherwise $\mu'(\alpha)$ would be unacceptable to α and μ' would be unstable. Since rol_α is the same as rol^*_α above rank i and the rank order lists of all of the other agents and programs in the instance are the same, μ' must also be stable before α truncates. Thus $\mu' \in \Omega$.

Theorem 1 *Let α be a resident or couple in an SMP-C instance $\mathcal{I}$. For any property $\mathcal{P}$, if α has a unique $\mathcal{P}$ program in $\mathcal{I}$, then for any $\mathcal{P}$ mechanism $y_{\mathcal{P}}$, α cannot improve its matching under $y_{\mathcal{P}}$ using only truncation manipulations (α has no incentive to manipulate under $y_{\mathcal{P}}$).*

Proof. The theorem says that there exists a program (or pair of programs) p such that for all $\mathcal{P}$ matchings μ we have that $\mu(\alpha) = p$. Before truncation $y_{\mathcal{P}}$ returns p as α's match. After truncation, by Lemma 1, $y_{\mathcal{P}}$ must return either p or *nil* as α's match. Neither of these improve α's match.

Theorem 2 *Let $y_{\mathcal{R}_{opt}}$ be an $\mathcal{R}_{opt}$ mechanism for SMP-C. Then $y_{\mathcal{R}_{opt}}$ is strategy-proof against residents who manipulate via truncation. However, $y_{\mathcal{R}_{opt}}$ is not strategy-proof against residents who manipulate via reordering.*

Proof. Part I: For any SMP-C instance $\mathcal{I}$, if $\mathcal{I}$ has an $\mathcal{R}_{opt}$ matching then every resident has a unique $\mathcal{R}_{opt}$ program. Then by Theorem 1 no resident has an incentive to manipulate $y_{\mathcal{R}_{opt}}$ using truncation. Otherwise $y_{\mathcal{R}_{opt}}$ always returns the empty match, and again no resident has an incentive to manipulate.

Part II: Reordering is not strategy-proof. We provide a counterexample in Figure 5. The preferences provided show residents' and programs' true preferences. Couples are identified by their resident-resident pair, and give joint preferences as expected. Given these true preferences, only one stable matching exists: $\mu(r_0) = c$, $\mu((r_1, r_2)) = (b, e)$, and $\mu((r_3, r_4)) = (a, d)$. However, even though only one stable matching exists (and thus this matching is $\mathcal{R}_{opt}$), the single resident r_0 has an incentive to manipulate via reordering. Instead of reporting $a \succ b \succ c \succ d$ (r_0's true preferences) r_0 can be matched to b instead of c by reporting $b \succ d \succ c \succ a$. (All other participants in the market report their true preferences.) The resulting matching is $\mu'(r_0) = b$, $\mu'((r_1, r_2)) = (a, d)$, and $\mu'((r_3, r_4)) = (c, e)$. μ' was not stable under the original preferences, and single resident r_0 is better off than when they reported their true preferences.

Resident preferences	Program preferences
$r_0 : a \succ b \succ c \succ d$	$a : r_3 \succ r_0 \succ r_1$
$(r_1, r_2) : (b, e) \succ (a, d)$	$b : r_1 \succ r_0$
$(r_3, r_4) : (a, d) \succ (c, e)$	$c : r_3 \succ r_0$
	$d : r_0 \succ r_2 \succ r_4$
	$e : r_4 \succ r_2$

Fig. 5. An instance that is manipulable by r_0 using reordering under any mechanism. Each program has capacity 1.

Theorem 3 *No stable mechanism for SMP-C is resident strategy-proof.*

Proof. Since the instance of Figure 5 has only one stable matching before and after r_0 reorders, any stable mechanism is manipulable by r_0.

Resident preferences	Program preferences
$(r_0, r_1) : (d, b) \succ (a, c)$	$a : r_0 \succ r_2 \succ r_4$
$(r_2, r_3) : (e, c) \succ (b, d) \succ (a, c)$	$b : r_2 \succ r_1$
$(r_4, r_5) : (a, c) \succ (e, nil)$	$c : r_1 \succ r_3 \succ r_5$
	$d : r_0 \succ r_3$
	$e : r_4 \succ r_2$

Fig. 6. An instance that is manipulable for truncating residents under any stable mechanism. Each program has capacity 1.

Theorem 4 *No stable mechanism for SMP-C is resident strategy-proof against truncations.*

Proof. Figure 6 shows an instance that can be manipulated by couple (r_0, r_1) or couple (r_2, r_3). This instance has two stable matchings, μ and μ':

$$\mu((r_0, r_1)) = (a, c), \mu((r_2, r_3)) = (b, d), \mu((r_4, r_5)) = (e, nil)$$
$$\mu'((r_0, r_1)) = (d, b), \mu'((r_2, r_3)) = (a, c), \mu'((r_4, r_5)) = (e, nil)$$

(r_0, r_1) prefers μ' to μ and (r_2, r_3) prefers μ to μ'. By truncating their rank-order list at rank 1, each couple can guarantee that their preferred stable matching is selected by the mechanism. Thus, any stable mechanism is manipulable by at least one of the two couples.

In this situation it is unclear what mechanism to use for SMP-C when an $\mathcal{R}_{opt}$ matching does not exist. By extending the analogy with SMP, it would be intuitive for $\mathcal{RP}_{opt}$ mechanisms to be harder to manipulate via truncations than other stable mechanisms. However, Lemma 1 states that a truncating agent a may create new stable matchings where they are matched to *nil*. A stable mechanism might return one of these matching. Therefore, a might have a disincentive to manipulate by truncation because the mechanism might now return a matching in which a is unmatched. However, these new matches of a to *nil* might not be $\mathcal{RP}_{opt}$. In this case they would not be returned by an $\mathcal{RP}_{opt}$ mechanism. Hence, with an $\mathcal{RP}_{opt}$ mechanism a might no longer have a disincentive to manipulate. Figure 7 shows an instance where this occurs. Initially, there are four stable matchings and two $\mathcal{RP}_{opt}$ matchings (not shown in the figure). By truncating, (r_0, r_1) can eliminate one of the $\mathcal{RP}_{opt}$ matchings while also creating a new stable matching μ where they are matched to (*nil*, *nil*). However, μ is not $\mathcal{RP}_{opt}$, and so (r_0, r_1) can freely manipulate an $\mathcal{RP}_{opt}$-mechanism without fear of being matched to (*nil*, *nil*). Hence there are stable mechanisms that are strategy-proof against truncations for this instance, but some $\mathcal{RP}_{opt}$ mechanisms are not.

Resident preferences	Program preferences
$(r_0, r_1) : (d, b) \succ (a, c) \succ (g, h)$	$a : r_0 \succ r_2$
$(r_2, r_3) : (a, f) \succ (b, e)$	$b : r_8 \succ r_2 \succ r_2 \succ r_7 \succ r_1$
$(r_4, r_5) : (c, e) \succ (d, f)$	$c : r_1 \succ r_4$
$(r_6, r_7) : (d, b) \succ (g, h)$	$d : r_4 \succ r_0 \succ r_6$
$(r_8, r_9) : (b, g)$	$e : r_3 \succ r_5$
	$f : r_5 \succ r_3$
	$g : r_0 \succ r_6 \succ r_9$
	$h : r_7 \succ r_1$

Fig. 7. An instance where the $\mathcal{RP}_{opt}$ mechanism is weak to truncations. (r_0, r_1) manipulates and each program has capacity 1.

B Algorithm to Enumerate All Stable Matchings

In our experiments (Sec. 5) we also need to find all the stable matchings of an instance. This can be accomplished with SAT-ENUM (Algorithm 2). SAT-ENUM uses a sequence of SAT calls each one returning a stable match. After each stable match is found we ensure that no future stable match is the same by adding a blocking clause to the SAT encoding as described for SAT-$\mathcal{RP}_{opt}$.

ALGORITHM 2: SAT-ENUM. Find all stable models of an inputted SMP-C instance

Input: $\mathcal{I} = \langle D, C, P, ROLs\rangle$ an **SMP-C** instance
Output: Find all stable models of $\mathcal{I}$

CNF $\leftarrow$ SAT-E($\mathcal{I}$)
while *true* **do**
    (sat?,π) $\leftarrow$ SatSolve(CNF)
    if *sat?* **then**
        $\mu \leftarrow$ stable matching corresponding to π
        output(μ)
        $c \leftarrow \{\neg m_d[p] \mid \mu(d) = p\}$
        CNF $\leftarrow$ CNF $\cup \{c\}$ // *Block* μ
    else
        return // *All stable matchings found.*

C Statistical Models of SMP-C Instances

IC-GEOG is the model presented Kojima et al. [12, 13]. It draws residents' and programs' preferences from the impartial culture model (i.e., i.i.d. uniform). All residents draw i.i.d. a ROL of size 10. For singles, this is their final ROL after appending *nil*. To generate a couple's joint ROL, all 11^2 pairs are scored via a Borda-like scoring function, breaking ties according to an arbitrarily chosen member of the couple's preferences. However, as a couple's preferences are constrained by geography, a pair of programs is only added to the joint ROL if both programs are in the same geographic region (or one is *nil*). Each program is randomly assigned one of 5 regions. Each program ranks each resident that ranks them, again with preferences i.i.d. uniform.

SFAS-GEOG is a variant of the model presented by Biró et al. to mimic SFAS [15]. All residents draw i.i.d. a ROL of size 10 from a Plackett-Luce model [19, 14], where the most popular program (resp., resident) is five times more popular than the least popular program (resp., resident) as seen in the SFAS market. All other programs' popularity are linearly interpolated between the most and least popular programs. Formally, given a reference ranking over all programs (resp., residents), the value of the ith element in the corresponding scoring vector is $m + 4i$, where m is the number of programs (resp., residents) in the match. For singles, this is their final ROL. For couples, all program-program pairs in the same geographic region are ranked in reverse lexicographic ordering, with arbitrary tie-breaking. Each program is randomly assigned one of 5 regions.

In order to ensure that the underlying preference distribution was responsible for any variance shown in the experiments, SFAS-GEOG is modified from Biró et al.'s original description to mirror IC-GEOG in the following ways: i) we fix all residents' ROL to be length 10; ii) we allow programs to rank all residents who rank them (again, drawing from a Plackett-Luce model with the scoring vector as described above); and iii) we impose the same geographical restriction on couples' joint ROLs as Kojima et al. do for IC-GEOG. In addition to being more directly comparable to IC-GEOG, imposing the geographical restriction on the Biró et al. model allows us to better capture couples' real-world preferences in larger markets. (The SFAS market has roughly 800 program positions; we test up to 30,000 program positions.)

In both models, we set the number of positions as 87% of the number of residents to match the 2015 NRMP data [17]. Each data point represents the average of 50 instances.

# The balanced contribution property for equal contributors*

Koji Yokote† Takumi Kongo‡ Yukihiko Funaki§

February 22, 2016

Abstract

In this paper we introduce a weaker version of the balanced contribution property by Myerson (1980), and explore the class of solutions characterized by the axiom. Our new axiom, the balanced contribution property for equal contributors, states the if two players' contributions to the grand coalition are the same, then their contributions to each other's payoffs are the same. We prove that this axiom, together with efficiency and weak strategic invariance, characterizes the class of **r**-egalitarian Shapley values. This class includes the egalitarian Shapley values (Joosten (1996)) and the generalized solidarity values (Casajus and Huettner (2014)) as special cases. We also provide a non-cooperative implementation of the solution.

1 Introduction

The balanced contribution property by Myerson (1980) has been extensively discussed in the literature. The property well captures characteristics of the Shapley value in that the Shapley value is the unique efficient solution that satisfies the property. Moreover, the property enables us to interpret the Shapley value in terms of objections and counterobjections (see Osborne and Rubinstein (1994)). The balanced contribution property, however, is "too strong" and tells us little about the difference between the Shapley value and other solutions. To circumvent this difficulty, Kamijo and Kongo (2010) introduced a weaker axiom, called the balanced cycle contributions property.

In this paper we introduce a new weaker version of the property and characterize a new class of solutions. To clarify how to weaken the property, we revisit the original

*Preliminary version prepared for Fifth World Congress of the Game Theory Society.

†JSPS Research Fellow. Graduate School of Economics, Waseda University, 1-6-1, Nishi-Waseda, Shinjuku-ku, Tokyo 169-8050, Japan (sidehand@toki.waseda.jp)

‡Faculty of Economics, Fukuoka University, 8-19-1 Nanakuma, Jonan-ku, Fukuoka 814-0180, Japan

§Faculty of Political Science and Economics, Waseda University, 1-6-1, Nishi-Waseda, Shinjuku-ku, Tokyo 169-8050, Japan (funaki@waseda.jp)

statement. We say that a solution satisfies the balanced contribution property if for any two players i and j, i's contribution to j's payoff is the same as j's contribution to i's payoff. We argue that the requirement "for any two players" is a severe condition, because two players are often in different bargaining positions in a game.

Motivated by the above argument, we introduce a new axiom, the *balanced contribution property for equal contributors*. This axiom states that if two players' marginal contributions to the grand coalition are the same, then their contributions to each other's payoffs are the same. We prove that this property, together with efficiency and weak strategic invariance, characterizes the class of $\mathbf{r}$-*egalitarian Shapley values*. This class includes the egalitarian Shapley values (Joosten (1996)) and the generalized solidarity values Casajus and Huettner (2014) as special cases. Our characterization indicates that the difference between the two variants of the Shapley value is pinpointed to the response to strategically equivalent changes.

By slightly changing the definition of the $\mathbf{r}$-egalitarian Shapley value, we define a new solution, the $\mathbf{r}$-discounted Shapley values, and provide an axiomatization. This solution includes the discounted Shapley value by Joosten (1996) as a special case.

Finally, we provide a non-cooperative implementation of the new solutions. We generalize the bidding mechanisms by van den Brink et al (2013) and van den Brink and Funaki (2015), who adapted Pérez-Castrillo and Wettstein's (2001) bidding mechanism to the egalitarian Shapley values and the discounted Shapley values, respectively. Our result extends the scope for the implementable variants of the Shapley value.

The reminder of the paper is organizes as follows. Section 2 is preliminary. In Section 3, we introduce two weaker versions of the balanced contribution property and axiomatize two classes of solutions. Section 4 presents new implementations. All proofs are provided in Section 5.

2 Preliminary

Let $\mathbb{N}$ denote the set of natural numbers. In this paper, $\mathbb{N}$ represents the *universe of players*. A *player set* N is a subset of $\mathbb{N}$. For a player set N, a *characteristic function on* N is a function $v : 2^N \to \mathbb{R}$ satisfying $v(\emptyset) = 0$. A pair of a player set N and a characteristic function v on N is called a *game*. Let Γ denote the set of all games, i.e.,

$$\Gamma = \big\{(N, v) : N \subset \mathbb{N}, v \text{ is a characteristic function on } N\big\}.$$

For notational simplicity, let n denote the size of a player set N, and let s denote the size of coalition S.

A solution is a function that assigns an n-dimensional payoff vector to each game

$(N,v) \in \Gamma$. The Shapley value (Shapley (1953)) is defined by

$$Sh_i(N,v) = \sum_{S \subseteq N \setminus i} \frac{(n-|S|-1)!|S|!}{n!}\big\{v(S \cup \{i\}) - v(S)\big\} \text{ for all } (N,v) \in \Gamma, i \in N.$$

The equal division value is defined by

$$ED_i(N,v) = \frac{v(N)}{n} \text{ for all } (N,v) \in \Gamma, i \in N.$$

For $\alpha \in [0,1]$, the α-egalitarian Shapley value (Joosten (1996)) is defined by

$$Sh_i^{\alpha}(N,v) = \alpha Sh_i(N,v) + (1-\alpha)ED_i(N,v) \text{ for all } (N,v) \in \Gamma, i \in N.$$

For $\delta \in [0,1]$ and $(N,v) \in \Gamma$, we define the game (N, v^{δ}) by

$$v^{\delta}(S) = \delta^{|N \setminus S|} v(S) \text{ for all } S \subseteq N.$$

For $\delta \in [0,1]$, the δ-discounted Shapley value (Joosten (1996)) is defined by

$$Sh_i^{\delta}(N,v) = Sh_i(N,v^{\delta}) \text{ for all } (N,v) \in \Gamma, i \in N.$$

For $\xi \in [0,1]$, the ξ-generalized solidarity value (Casajus and Huettner (2014)) is defined by

$$So_i^{\xi}(N,v) = \xi_n \cdot \frac{v(N)}{n} + \sum_{S \subseteq N \setminus i} \frac{s!(n-s-1)!}{n!} \cdot \Big[(1-\xi_{s+1}) \cdot v(S \cup i) - (1-\xi_s) \cdot v(S)\Big]$$
$$\text{for all } (N,v) \in \Gamma, i \in N,$$

where

$$\xi_1 = \xi,\ \xi_k = \frac{k \cdot \xi}{(k-1) \cdot \xi + 1} \text{ for all } k = 2, 3, \cdots n.$$

For a player set $N \subset \mathbb{N}$ and $i \in N$, we define (N, u_i) by

$$u_i(S) = \begin{cases} 1 & \text{if } i \in S, \\ 0 & \text{otherwise.} \end{cases}$$

Adding u_i to a game is often called a strategically equivalent change.

We define axioms satisfied by ψ:

Efficiency (E). $\sum_{i \in N} \psi_i(N,v) = v(N)$ for all $(N,v) \in \Gamma$.

Weak Strategic Invariance (WSI). For any $(N,v) \in \Gamma$, $i \in N$ and λ, $\psi(N, v + \lambda u_i) = \psi(N,v) + \lambda \psi(N, u_i)$.

Strict Desirability (SD). Let $(N, v) \in \Gamma$ and $i, j \in N$. If $v(S \cup i) > v(S \cup j)$ for all $S \subseteq N \backslash \{i, j\}$, then $\psi_i(N, v) > \psi_j(N, v)$.

WSI states that ψ is linear with respect to the addition of u_i. SD is a strict version of Desirability by Maschler and Peleg (1966).

Let $\mathcal{S}$ denote the set of sequences of real numbers, i.e.,

$$\mathcal{S} = \left\{ \{r_k\}_{k=1}^{\infty} : r_k \in \mathbb{R} \text{ for all } k = 1, 2, \cdots \right\}.$$

Let $\mathcal{S}_+$ denote the set of sequences of positive real numbers.

3 Axiomatizations

3.1 r-egalitarian Shapley values

In this section we introduce a weaker version of the balanced contribution property. Let us first revisit the original statement of the property. We say that a solution ψ satisfies the balanced contribution property if for any $(N, v) \in \Gamma$, $n \geq 2$, and $i, j \in N$, $i \neq j$,

$$\psi_i(N, v) - \psi_i(N \backslash j, v) = \psi_j(N, v) - \psi_j(N \backslash i, v). \tag{1}$$

In words, we require that the two players' contributions to each other's payoffs are the same. This property, however, is a strong requirement when two players i and j are in different bargaining positions in the game (N, v). For example if the worths of coalitions containing i are much larger than the worths of coalitions containing j, then their contributions to the payoffs likely to differ.

Motivated by the above argument, we restrict the requirement (1) to two players who are in the same bargaining position under a certain criterion.

Axiom 1. We say that a solution ψ satisfies the *balanced contribution property for the equal contributors* (BCEC) if the following condition is satisfied: let $(N, v) \in \Gamma$ and $i, j \in N$. If $v(N \backslash i) = v(N \backslash j)$, then

$$\psi_i(N, v) - \psi_i(N \backslash j, v) = \psi_j(N, v) - \psi_j(N \backslash i, v).$$

This axiom states that if i and j's contributions to the worth of the grand coalition are the same, i.e., $v(N) - v(N \backslash i) = v(N) - v(N \backslash j)$, then their contributions to the payoffs are the same.

To describe the class of solutions characterized by this axiom, we introduce some notations. For $\mathbf{r} \in \mathcal{S}$ and $(N, v) \in \Gamma$, we define $v^{\mathbf{r}}$ by

$$v^{\mathbf{r}}(S) = r_s v(S) \text{ for all } S \subseteq N.$$

We define the **r-egalitarian Shapley value** $ESh^{\mathbf{r}}$ by

$$ESh^{\mathbf{r}}(N,v) = (1-r_n)\cdot\frac{v(N)}{n} + Sh(N,v^{\mathbf{r}}) \text{ for all } (N,v)\in\Gamma. \tag{2}$$

Instead of directly applying the Shapley value, we first rescale the worth of each coalition by multiplying r_s. We interpret this rescaling as a generalization of per-capita measure or discounting. Note that $v^{\mathbf{r}}(N)$ is different from $v(N)$ unless $r_n = 1$. To achieve efficiency, we equally divide the difference between them, $(1-r_n)\cdot v(N)$.

We are now in a position to state our main theorem.

Theorem 1. *A solution ψ satisfies E, WSI and BCEC if and only if there exists $\mathbf{r}\in\mathcal{S}$ such that $\psi = ESh^{\mathbf{r}}$.*

The class of **r**-egalitarian Shapley values includes several variants of the Shapley value, and the above characterization provides a new insight into the difference between them. Note first that the α-egalitarian Shapley value is an **r**-egalitarian Shapley value with the sequence $r_k = \alpha$ for all $k = 1, 2, \cdots$. Moreover, the ξ-generalized solidarity value is also an **r**-egalitarian Shapley value with the sequence

$$r_1 = 1-\xi,\ r_k = 1-\frac{k\cdot\xi}{(k-1)\cdot\xi+1} \text{ for all } k = 2,3,\cdots.$$

Although the two solutions are defined from different perspectives, Theorem 1 indicates that they share the same axiom, BCEC, and the difference is pinpointed to the response to strategically equivalent changes.

In Theorem 1 we allow any sequence of real numbers. However, positive sequences are more suited to the idea of rescaling the worths of coalitions. We can characterize the class with positive sequences by adding one more axiom.

Corollary 1. *A solution ψ satisfies E, WSI, SD and BCEC if and only if there exists $\mathbf{r}\in\mathcal{S}_+$ such that $\psi = ESh^{\mathbf{r}}$.*

3.2 r-discounted Shapley values

In this subsection we introduce a new class of solutions, **r**-discounted Shapley values, which is a building block for implementing the **r**-egalitarian Shapley value in the next section.

Let us first go back to the definition (2) of the **r**-egalitarian Shapley value. Given an infinite sequence **r** and a game (N,v), we first apply the sequence **r** to the game (N,v), and then adjust the reminder $(1-r_n)\cdot v(N)$ in order to achieve efficiency.

There is another possible approach to define an efficient solution. Fix the number of players n and, instead of the sequence $r_1,\cdots,r_n$, consider the following sequence:

$$\frac{r_1}{r_n},\frac{r_2}{r_n},\cdots,\frac{r_{n-1}}{r_n},\frac{r_n}{r_n}=1.$$

That is, we "normalize" the sequence so that the n-th entry, r_n, is equal to 1. If we apply this sequence to an n-person game (N, v), then the derived game has the same worth of the grand coalition as v, and there is no need for adjusting the remainder.

We formally describe the above idea. For $\mathbf{r} \in \mathcal{S}_+$ and $(N, v) \in \Gamma$, we define $\bar{v}^{\mathbf{r}}$ by

$$\bar{v}^{\mathbf{r}}(S) = \frac{r_s}{r_n} v(S) \text{ for all } S \subseteq N.$$

We define the **r-discounted Shapley value** $DSh^{\mathbf{r}}$ by

$$DSh^{\mathbf{r}}(N, v) = Sh(N, \bar{v}^{\mathbf{r}}) \text{ for all } (N, v) \in \Gamma.$$

We remark that for $\delta \in (0, 1]$, the δ-discounted Shapley value is a special case of the $\mathbf{r}$-discounted Shapley value with the sequence $r_k = \delta^{-k}$ for all $k = 1, 2, \cdots$.

Like the $\mathbf{r}$-egalitarian Shapley value, we can axiomatize the $\mathbf{r}$-discounted Shapley value in a parallel manner by considering a weaker version of the balanced contribution property.

Axiom 2. We say that a solution ψ satisfies the balanced contribution for equal recipients (BCER) if the following condition is satisfied: let $(N, v) \in \Gamma$ and $i, j \in N$. If $\psi_i(N \backslash j, v) = \psi_j(N \backslash i, v)$, then

$$\psi_i(N, v) - \psi_i(N \backslash j, v) = \psi_j(N, v) - \psi_j(N \backslash i, v).$$

This axiom states that if two players receive the same in the absence of the other player, then their contributions to the payoffs are the same.[1]

Theorem 2. *A solution ψ satisfies E, WSI, SD and BCER if and only if there exists $\mathbf{r} \in \mathcal{S}_+$ such that $\psi = DSh^{\mathbf{r}}$.*

4 Implementation

This section is devoted to non-cooperative implementation of the new solutions provided in Section 3. Our approach is based on the bidding mechanism developed by Pérez-Castrillo and Wettstein (2001). The mechanism is subsequently adapted to the egalitarian Shapley value and the discounted Shapley value by van den Brink et al (2013) and van den Brink and Funaki (2015), respectively. We further extend previous mechanisms and analyze the scope for the implementable variants of the Shapley value.

Fix a natural number $m \in \mathbb{N}$ and we restrict our attention to games with no more

[1]Note that BCEC is equivalent to the following condition: for any $(N, v) \in \Gamma$ and $i, j \in N$, if $\psi_i(N \backslash j, v) = \psi_j(N \backslash i, v)$, then $\psi_i(N, v) = \psi_j(N, v)$.

than m players. Namely, we consider the class Γ^m defined by

$$\Gamma^m = \{(N, v) \in \Gamma : n \leq m\}.$$

We define $\bar{\mathcal{S}}$ by

$$\bar{\mathcal{S}} = \big\{\{r_k\}_{k=1}^m : r_k \in (0, 1] \text{ for all } k = 1, \cdots, m\big\}.$$

In this section we restrict the domain of a solution to Γ^m. With a slight abuse notation for $\mathbf{r} \in \bar{\mathcal{S}}$, we define the $\mathbf{r}$-egalitarian Shapley value and the $\mathbf{r}$-discounted Shapley value as we did in Section 3.

van den Brink and Funaki (2015) extended the bidding mechanism by incorporating a discount factor δ at each round. The $\mathbf{r}$-discounted Shapley value can be implemented by generalizing the discount factor. We formally describe the bidding mechanism by mimicking van den Brink and Funaki (2015). Let N_t be the player set with which round $t \in \{1, \cdots, m\}$ starts. We define $N_1 = N$. We also define

$$r'_t = \frac{r_t}{r_{t+1}}, \text{ for all } t = 1, \cdots, m-1$$
$$\tilde{r}_0 = 1, \ \tilde{r}_t = \prod_{j=1}^{t} r'_{m-j} \text{ for all } t = 1, \cdots, m-1.$$

Definition 1 (Bidding mechanism)**.**
Round t**,** $t \in \{1, \cdots, k-1\}$.

Stage 1 Each player $i \in N_t$ makes bids $b^i_j \in \mathbb{R}$ for every $j \neq i$. For each $i \in N_t$, let $B^i = \sum_{j \in N \setminus i}(b^i_j - b^j_i)$ be the net bid of player i. Let α_t be the player with the highest net bid of round t (In case of a non-unique maximizer we choose any of these maximal bidders to be the "winner" with equal probability). Player α_t pays every other player $j \in N_t \setminus \alpha_t$, its offered bid $b^{\alpha_t}_j$. Player α_t becomes the proposer in the next stage. Go to Stage 2.

Stage 2 Player α_t proposes an offer $y^{\alpha_t}_j \in \mathbb{R}$ to every player $j \in N_t \setminus \alpha_t$ (This offer is additional of the bids paid at Stage 1). Go to Stage 3.

Stage 3 The players other than α_t, sequentially, either accept or reject the offer. If at least one player rejects it, then the offer is rejected. Otherwise, the offer is accepted. Go to Stage 4.

Stage 4 If the offer is accepted, then each player $j \in N_t \setminus \alpha_t$ receives $y^{\alpha_t}_j$ and player α_t obtains the remainder

$$\tilde{r}_{t-1} v(N_t) - \sum_{j \in N_t \setminus \alpha_t} y^{\alpha_t}_j.$$

If the offer is rejected then player α_t leaves the game and obtains $\tilde{r}_{t-1}r'_{n-j}v(\alpha_t)$, while the players in $N_t \backslash \{\alpha_t\}$ proceed to round $t+1$ to bargain over $\tilde{r}_t v(N_t \backslash \{\alpha_t\})$.

Theorem 3. *The bidding mechanism implements $DSh^{\mathbf{r}}$ in any subgame perfect equilibrium.*

van den Brink et al (2013) proved that the α-egalitarian Shapley value can be implemented by incorporating the probability α of breakdown of negotiation at Round 1. By using this idea, we can implement the $\mathbf{r}$-egalitarian Shapley value. For $\mathbf{r} \in \bar{\mathcal{S}}$ and $(N, v) \in \Gamma^m$, the following equation holds:

$$ESh^{\mathbf{r}}(N,v) = (1-r_m)\cdot \frac{v(N)}{n} + r_m \cdot DSh^{\mathbf{r}}(N,v).$$

Namely, like the egalitarian Shapley value, $ESh^{\mathbf{r}}$ can be represented by a convex combination of the ED value and $DSh^{\mathbf{r}}$. Thus, by incorporating the probability of breakdown r_m to the mechanism defined above, we implement $ESh^{\mathbf{r}}$.

5 Proof of Theorem 1

If part: Since $ESh^{\mathbf{r}}$ is linear, $ESh^{\mathbf{r}}$ satisfies WSI. We prove that $ESh^{\mathbf{r}}$ satisfies E. For any $(N, v) \in \Gamma$,

$$\begin{aligned}\sum_{i\in N} ESh_i^{\mathbf{r}}(N,v) &= (1-r_n)v(N) + \sum_{i\in N}\phi_i(N,v^{\mathbf{r}})\\ &= (1-r_n)v(N) + r_n v(N)\\ &= v(N),\end{aligned}$$

where the second equality follows from efficiency of the Shapley value.

We prove that $ESh^{\mathbf{r}}$ satisfies BCEC. Let $(N,v) \in \Gamma^n$, $n \geq 2$, $i, j \in N$, $i \neq j$.

$$\begin{aligned}&[ESh_i^{\mathbf{r}}(N,v) - ESh_i^{\mathbf{r}}(N\backslash j,v)] - [ESh_j^{\mathbf{r}}(N,v) - ESh_j^{\mathbf{r}}(N\backslash i,v)]\\ =&\Big[(1-r_n)\frac{v(N)}{n} + Sh_i(N,v^{\mathbf{r}}) - (1-r_{n-1})\frac{v(N\backslash j)}{n-1} - Sh_i(N\backslash j, v^{\mathbf{r}})\Big]\\ -&\Big[(1-r_n)\frac{v(N)}{n} + Sh_j(N,v^{\mathbf{r}}) - (1-r_{n-1})\frac{v(N\backslash i)}{n-1} - Sh_j(N\backslash i, v^{\mathbf{r}})\Big]\\ =&[Sh_i(N,v^{\mathbf{r}}) - Sh_i(N\backslash j,v^{\mathbf{r}})] - [Sh_j(N,v^{\mathbf{r}}) - Sh_j(N\backslash i,v^{\mathbf{r}})]\\ &+ (1-r_{n-1})\cdot\frac{v(N\backslash i) - v(N\backslash j)}{n-1}\\ =&(1-r_{n-1})\cdot \frac{v(N\backslash i) - v(N\backslash j)}{n-1},\end{aligned}$$

where the last equality follows from the balanced contribution property of the Shapley value. Thus, if $v(N\backslash i) = v(N\backslash j)$, we obtain

$$ESh_i^{\mathbf{r}}(N,v) - ESh_i^{\mathbf{r}}(N\backslash j,v) = ESh_j^{\mathbf{r}}(N,v) - ESh_j^{\mathbf{r}}(N\backslash i,v).$$

Only if part: We first prove auxiliary lemmas.

Lemma 1. *Let $n \geq 2$. Suppose that ψ satisfies BCEC and A on Γ^{n-1}. Then ψ satisfies S on Γ^n.*

Proof. Let $(N,v) \in \Gamma^n$ and $i,j \in N$, $i \neq j$, $i \sim_v j$. Since $v(N\backslash i) = v(N\backslash j)$, by BCEC,

$$\psi_i(N,v) - \psi_j(N,v) = \psi_i(N\backslash j,v) - \psi_j(N\backslash i,v). \tag{3}$$

Since $i \sim_v j$, $(N\backslash j,v)$ is equivalent to $(N\backslash i,v)$. By A on Γ^{n-1}, $\psi_i(N\backslash j,v) = \psi_j(N\backslash i,v)$. Together with (3), we obtain $\psi_i(N,v) = \psi_j(N,v)$. □

Lemma 2. *Let $n \geq 2$. Suppose that ψ satisfies WSI and S on Γ^n. Then for any $N \subset \mathbb{N}$ with $|N| = n$ and $i,j \in N$, $\psi_i(N,u_i) = \psi_j(N,u_j)$.*

Proof. Let $x = \psi_i(N,u_i)$, $y = \psi_j(N,u_j)$. Then,

$$\begin{aligned}\psi_i(N,u_i) + \psi_i(N,u_j) &\overset{\text{E,S}}{=} x + \frac{1-y}{n-1},\\ \psi_j(N,u_i) + \psi_j(N,u_j) &\overset{\text{E,S}}{=} \frac{1-x}{n-1} + y.\end{aligned}$$

Together with

$$\psi_i(N,u_i) + \psi_i(N,u_j) \overset{\text{WSI}}{=} \psi_i(N,u_i+u_j) \overset{\text{S}}{=} \psi_j(N,u_i+u_j) \overset{\text{WSI}}{=} \psi_j(N,u_i) + \psi_j(N,u_j),$$

we obtain

$$x + \frac{1-y}{n-1} = \frac{1-x}{n-1} + y.$$

This equation implies $x = y$. □

Lemma 3. *Let $n \geq 2$. Suppose that ψ satisfies BCEC and S on Γ^n. Then for any $N \subset \mathbb{N}$, $|N| = n$, $i,j \in N$, $i \neq j$, and $k \in \mathbb{N}\backslash N$,*

$$\psi_i(N,u_i) = \psi_i\big((N\backslash j) \cup \{k\}, u_i\big).$$

Proof. Let $M = N \cup \{k\}$. Define $v = u_i + u_{M\backslash i}$ and consider the $n+1$-person game (M,v). Note that $v(M\backslash l) = v(M\backslash l')$ for all $l,l' \in M$.

Let $x = \psi_i(M\backslash k,u_i)$, $y = \psi_i(M\backslash j,u_i)$. Then,

$$\psi_i(M,v) - \psi_j(M,v) \overset{\text{BCEC}}{=} \psi_i(M\backslash j,v) - \psi_j(M\backslash i,v) = y - \psi_j(M\backslash i,u_{M\backslash i}), \tag{4}$$

$$\psi_i(M,v) - \psi_k(M,v) \overset{\text{BCEC}}{=} \psi_i(M\backslash k,v) - \psi_k(M\backslash i,v) = x - \psi_k(M\backslash i,u_{M\backslash i}). \tag{5}$$

By taking (4) − (5),

$$\psi_k(M,v) - \psi_j(M,v) \overset{\text{BCEC}}{=} y - \psi_j(M\backslash i, u_{M\backslash i}) - x + \psi_k(M\backslash i, u_{M\backslash i}) \overset{\text{S}}{=} y - x. \tag{6}$$

On the other hand,

$$\begin{aligned}\psi_j(M,v) - \psi_k(M,v) &\overset{\text{BCEC}}{=} \psi_j(M\backslash k, u_i) - \psi_k(M\backslash j, u_i)\\ &\overset{\text{S}}{=} \frac{1-x}{n-1} - \frac{1-y}{n-1}\\ &= \frac{y-x}{n-1}.\end{aligned} \tag{7}$$

By (6) and (7), we obtain $x = y$. □

For each $n, s \in \mathbb{N}_0$, $n \geq 1$, $0 \leq s \leq n-1$, we define

$$p_{n,s} = \frac{s!(n-s-1)!}{n!}.$$

To prove the only-if part, it suffices to prove that exists an infinite sequence of real numbers $\{r_k\}_{k=1}^{\infty}$ such that, for any $(N,v) \in \Gamma$ and $i \in N$,

$$\psi_i(N,v) = r_n \cdot \frac{v(N)}{n} + \sum_{S \subseteq N\backslash i} p_{n,s}\Big[(1 - r_{s+1})v(S \cup i) - (1 - r_s)v(S)\Big]. \tag{8}$$

For 1-person games, ψ is uniquely determined by E. We focus on 2-person games. Let $\{i,j\} \subset \mathbb{N}$. Since ψ satisfies A on Γ^1, by Lemma 1, ψ satisfies S on Γ^2. By E and S on Γ^2, for any $\{i,j\} \subset \mathbb{N}$ and $\lambda \in \mathbb{R}$,

$$\psi(\{i,j\}, \lambda u_{ij}) = \Big(\frac{\lambda}{2}, \frac{\lambda}{2}\Big). \tag{9}$$

For each $\{i,j\} \in \mathbb{N}$, by E and Lemma 2,

$$\psi_i(\{i,j\}, u_j) = \psi_j(\{i,j\}, u_i).$$

For each $\{i,j\} \subset \mathbb{N}$, let $r(\{i,j\}) \in \mathbb{R}$ denote the above equal value. For any $\{i,j\} \subset \mathbb{N}$ and $k \in \mathbb{N}\backslash\{i,j\}$,

$$r(\{i,j\}) \overset{\text{E}}{=} 1 - \psi_i(\{i,j\}, u_i) \overset{\text{L3}}{=} 1 - \psi_i(\{i,k\}, u_i) \overset{\text{E}}{=} r(\{i,k\}). \tag{10}$$

For any $\{i,j\} \subset \mathbb{N}$, by replacing i and j with an outside player, we can obtain an arbitrary 2-person player set $\{i',j'\} \subset \mathbb{N}$. This observation and (10) imply that $r(\{i,j\}) \in \mathbb{R}$ does not depend on the choice of $\{i,j\} \subset \mathbb{N}$. Choose an arbitrary $\{i,j\} \subset \mathbb{N}$ and we define r_1

by

$$r_1 = 2r(\{i,j\}). \tag{11}$$

By WSI and E, for any $\lambda \in \mathbb{R}$,

$$\psi(\{i,j\}, \lambda u_j) = \Big(\frac{\lambda r_1}{2}, \lambda - \frac{\lambda r_1}{2}\Big), \tag{12}$$
$$\psi(\{i,j\}, \lambda u_i) = \Big(\lambda - \frac{\lambda r_1}{2}, \frac{\lambda r_1}{2}\Big). \tag{13}$$

Let $(\{i,j\}, v) \in \Gamma^2$. For each $S \subseteq \{i,j\}$, $S \neq \emptyset$, let d_S denote the dividend of S in $(\{i,j\}, v)$. Then,

$$\begin{aligned}
\psi_i(\{i,j\}, v) &\overset{\text{WSI}}{=} \psi_i(\{i,j\}, d_{ij}u_{ij}) + \psi_i(\{i,j\}, d_i u_i) + \psi_i(\{i,j\}, d_j u_j) \\
&\overset{(9),(12),(13)}{=} \frac{d_{ij}}{2} + d_i - d_i \cdot \frac{r_1}{2} + d_j \cdot \frac{r_1}{2} \\
&= \frac{1}{2}\big\{v(ij) - v(i) - v(j)\big\} + v(i) - v(i) \cdot \frac{r_1}{2} + v(j) \cdot \frac{r_1}{2}.
\end{aligned}$$

Thus for any $r_2 \in \mathbb{R}$, we get

$$\psi_i(\{i,j\}, v) = r_2 \cdot \frac{v(ij)}{2} + \frac{1}{2}(1 - r_1)v(i) + \frac{1}{2}\big[(1 - r_2)v(ij) - (1 - r_1)v(j)\big].$$

Thus, when r_1 is given by (11) and r_2 is arbitrary, ψ coincides with (8) for 2-person games.

We proceed by the induction on the number of players. It suffices to prove that for each $t \in \mathbb{N}$, $t \geq 3$, the following claim holds:

Claim t . *Suppose that there exist real numbers $\{r_1, \cdots, r_{t-2}\}$ such that, for any $(N, v) \in \Gamma$, $1 \leq n \leq t-1$, and any $r_{t-1} \in \mathbb{R}$, $\psi(N, v)$ coincides with (8).*

Then, there exists $r_{t-1} \in \mathbb{R}$ such that, for any $(N, v) \in \Gamma$, $n = t$, and any $r_t \in \mathbb{R}$, $\psi(N, v)$ coincides with (8).

The proof of Claim t consists of two steps. In Step 1, we endogenously derive r_{t-1}. In Step 2, we prove that the real number r_{t-1} derived in Step 1 satisfies the desired condition.

Step 1: By the induction hypothesis, ψ satisfies A on Γ^{n-1}. Thus, by Lemma 1, ψ satisfies S on Γ^n. For any $N \subset \mathbb{N}$ with $|N| = n$, $i, j \in N$, $i \neq j$, and $i', j' \in N$, $i' \neq j'$,

$$\psi_j(N, u_i) \overset{\text{E,S}}{=} \frac{1 - \psi_i(N, u_i)}{n-1} \overset{\text{L2}}{=} \frac{1 - \psi_{i'}(N, u_{i'})}{n-1} \overset{\text{E,S}}{=} \psi_{j'}(N, u_{i'})$$

For each $N \subset \mathbb{N}$, let $r(N) \in \mathbb{R}$ denote the above equal value. For any $N \subset \mathbb{N}$ with

$|N| = n$, $i, j \in N$ and $k \in \mathbb{N} \backslash N$,

$$r(N) \overset{\text{E,S}}{=} \frac{1 - \psi_i(N, u_i)}{n-1} \overset{\text{L3}}{=} \frac{1 - \psi_i\big((N\backslash j) \cup \{k\}, u_i\big)}{n-1} \overset{\text{E,S}}{=} r\big((N\backslash j) \cup \{k\}\big). \tag{14}$$

For any $N \subset \mathbb{N}$, by repeatedly replacing an player in N with an outside player, we can obtain an arbitrary n-person player set $N' \subset \mathbb{N}$. This observation and (14) imply that $r(N) \in \mathbb{R}$ does not depend on the choice of $N \subset \mathbb{N}$. Choose an arbitrary player set $N \subset \mathbb{N}$ and we define r_{t-1} by

$$r_{t-1} = n(n-1)r(N) - \sum_{m=1}^{t-2} r_m. \tag{15}$$

Step 2: We prove that, when r_{t-1} is given by (15), $\psi(N, v)$ coincides with (8) for all $(N, v) \in \Gamma$, $n = t$.

For each game $(N, v) \in \Gamma$, $n = t$, we define the binary relation $\sim^*$ on N by

$$i \sim^* j \;\Leftrightarrow\; v(N\backslash\{i\}) = v(N\backslash\{j\}).$$

The binary relation $\sim^*$ is an equivalent relation and induces a partition on N. Let $\mathcal{N}(N, v)$ denote the partition and set

$$\#(N, v) = \max_{S \in \mathcal{N}(N,v)} |S|.$$

We proceed by the induction on $\#(N, v)$.

Induction base: Consider a game $(N, v) \in \Gamma$ such that $n = t$ and $\#(N, v) = n$, i.e.,

$$v(N\backslash\{i\}) = v(N\backslash\{j\}) \text{ for all } i, j \in N.$$

Fix $i \in N$. Then,

$$\begin{aligned}
&\psi_i(N, v) - \psi_j(N, v) \overset{\text{BCEC}}{=} \psi_i(N\backslash j, v) - \psi_j(N\backslash i, v) \text{ for all } j \in N\backslash\{i\},\\
&\psi_i(N, v) - \psi_i(N, v) = 0.
\end{aligned}$$

By taking the sum of both sides of the above equations, together with E, we get

$$
\begin{aligned}
&n\psi_i(N,v) - v(N)\\
&= \sum_{j\in N\setminus i}\Big[\psi_i(N\setminus j, v) - \psi_j(N\setminus i, v)\Big]\\
&\overset{\text{IH}}{=} \sum_{j\in N\setminus i}\Big[\sum_{S\subseteq N\setminus\{i,j\}} p_{n-1,s}\Big\{(1-r_{s+1})v(S\cup i) - (1-r_s)v(S)\Big\}\\
&- \sum_{S\subseteq N\setminus\{i,j\}} p_{n-1,s}\Big\{(1-r_{s+1})v(S\cup j) - (1-r_s)v(S)\Big\}\Big]\\
&= \sum_{j\in N\setminus i}\Big[\sum_{S\subseteq N\setminus\{i,j\}} p_{n-1,s}(1-r_{s+1})\{v(S\cup i) - v(S\cup j)\}\Big]\\
&= \sum_{S\subsetneq N\setminus i}(n-s-1)p_{n-1,s}(1-r_{s+1})v(S\cup i) - \sum_{S\subseteq N\setminus i: S\neq\emptyset} s\cdot p_{n-1,s-1}(1-r_s)v(S)\\
&= \sum_{S\subsetneq N\setminus i}\frac{s!(n-s-1)!}{(n-1)!}(1-r_{s+1})v(S\cup i) - \sum_{S\subseteq N\setminus i}\frac{s!(n-s-1)!}{(n-1)!}(1-r_s)v(S)\\
&= n\cdot\sum_{S\subseteq N\setminus i}\frac{s!(n-s-1)!}{n!}\Big[(1-r_{s+1})v(S\cup i) - (1-r_s)v(S)\Big] - (1-r_n)v(N),
\end{aligned}
$$

where r_t is an arbitrary real number. It follows that, for any $r_t \in \mathbb{R}$,

$$
n\psi_i(N,v) = r_n v(N) + n\sum_{S\subseteq N\setminus i}\frac{s!(n-s-1)!}{n!}\Big[(1-r_{s+1})v(S\cup i) - (1-r_s)v(S)\Big]. \quad (16)
$$

Induction step: Suppose that the result holds for any $(N,v)\in\Gamma$, $n=t$, $\#(N,v)=l+1$, and we prove the result for $(N,v)\in\Gamma$, $n=t$, $\#(N,v)=l$, where $1\le l\le n-1$.

Let $(N,v)\in\Gamma$, $n=t$, $\#(N,v)=l$. Choose a coalition $T\in\mathcal{N}(N,v)$, $|T|=l$. Since $l\le n-1$, $N\setminus T\neq\emptyset$. Choose players $i\in T$ and $j\in N\setminus T$. Define $\delta = v(N\setminus j) - v(N\setminus i)$ and consider the game $(N, v+\delta u_j)$. In this game,

$$
\begin{aligned}
&(v+\delta u_j)(N\setminus j) = v(N\setminus j),\\
&(v+\delta u_j)(N\setminus k) = v(N\setminus j) \text{ for all } k\in T.
\end{aligned}
$$

It follows that $\#(N, v+\delta u_j) = l+1$. Thus, we can apply the induction hypothesis. For any player $k \in N\backslash j$,

$$\begin{aligned}
&\psi_k(N, v + \delta u_j)\\
&= r_n \cdot \frac{v(N)+\delta}{n} + \sum_{S \subseteq N\backslash\{j,k\}} p_{n,s}\Big[(1-r_{s+1})v(S\cup k) - (1-r_s)v(S)\Big]\\
&+ \sum_{S\subseteq N\backslash k: S\ni j} p_{n,s}\Big[(1-r_{s+1})\big(v(S\cup k)+\delta\big) - (1-r_s)\big(v(S)+\delta\big)\Big]\\
&= r_n \cdot \frac{v(N)}{n} + \sum_{S\subseteq N\backslash k} p_{n,s}\Big[(1-r_{s+1})v(S\cup k) - (1-r_s)v(S)\Big]\\
&+ \delta\Big[\frac{r_n}{n} + \sum_{S\subseteq N\backslash k, S\ni j} p_{n,s}(-r_{s+1}+r_s)\Big].
\end{aligned}$$

Since

$$\begin{aligned}
&\frac{r_n}{n} + \sum_{S\subseteq N\backslash k, S\ni j} p_{n,s}(-r_{s+1}+r_s)\\
&= \frac{r_n}{n} + \sum_{q=2}^{n-1}\Big\{\binom{n-2}{q-1}p_{n,q} - \binom{n-2}{q-2}p_{n,q-1}\Big\}r_q + \frac{r_1}{n(n-1)} - \frac{r_n}{n}\\
&= \sum_{q=2}^{n-1}\Big\{\frac{q}{n(n-1)} - \frac{q-1}{n(n-1)}\Big\}r_q + \frac{r_1}{n(n-1)}\\
&= \frac{1}{n(n-1)}\sum_{q=1}^{n-1} r_q,
\end{aligned}$$

we obtain

$$\begin{aligned}
\psi_k(N, v+\delta u_j) = r_n \cdot \frac{v(N)}{n} &+ \sum_{S\subseteq N\backslash k} p_{n,s}\Big[(1-r_{s+1})v(S\cup k) - (1-r_s)\cdot v(S)\Big]\\
&+ \frac{\delta}{n(n-1)}\sum_{q=1}^{n-1} r_q. \qquad (17)
\end{aligned}$$

By WSI,

$$\begin{aligned}
\psi_k(N,v) &= \psi_k(N, v+\delta u_j) - \psi_k(N, \delta u_j)\\
&\overset{(15),(17)}{=} r_n \cdot \frac{v(N)}{n} + \sum_{S\subseteq N\backslash k} p_{n,s}\Big[(1-r_{s+1})v(S\cup k) - (1-r_s)\cdot v(S)\Big].
\end{aligned}$$

Thus, the desired equation holds for all $k \in N\backslash j$. E completes the proof.

References

van den Brink R, Funaki Y (2015) Implementation and axiomatization of discounted shapley values. Social Choice and Welfare 45(2):329–344

van den Brink R, Funaki Y, Ju Y (2013) Reconciling marginalism with egalitarianism: consistency, monotonicity, and implementation of egalitarian shapley values. Social Choice and Welfare 40(3):693–714

Casajus A, Huettner F (2014) On a class of solidarity values. European Journal of Operational Research 236(2):583–591

Joosten R (1996) Dynamics, equilibria and values. dissertation Maastricht University

Kamijo Y, Kongo T (2010) Axiomatization of the shapley value using the balanced cycle contributions property. International Journal of Game Theory 39(4):563–571

Maschler M, Peleg B (1966) A characterization, existence proof and dimension bounds for the kernel of a game. Pacific Journal of Mathematics 18(2):289–328

Myerson R (1980) Conference structures and fair allocation rules. International Journal of Game Theory 9(3):169–182

Osborne M, Rubinstein A (1994) A course in game theory. MIT press

Pérez-Castrillo D, Wettstein D (2001) Bidding for the surplus: a non-cooperative approach to the shapley value. Journal of Economic Theory 100(2):274–294

Shapley L (1953) A value for n-person games. In: Roth AE (ed) The Shapley value Cambridge University Press, Cambridge, pp 41-48