On the Complexity of Exchanging[☆]

Xavier Molinero[a,1], Martin Olsen[b], Maria Serna[c,2]

[a]*Department of Applied Mathematics III. Universitat Politècnica de Catalunya, Manresa, Spain*
[b]*AU Herning, Aarhus University, Denmark*
[c]*Department of Computer Science, Universitat Politècnica de Catalunya, Barcelona, Spain*

Abstract

We analyze the computational complexity of the problem of deciding whether, for a given simple game, there exists the possibility of rearranging the participants in a set of j given losing coalitions into a set of j winning coalitions. We also look at the problem of turning winning coalitions into losing coalitions. We analyze the problem when the simple game is represented by a list of wining, losing, minimal winning or maximal loosing coalitions.

Keywords: Tradeness of Simple Games, Computational Complexity

1. Introduction

Simple games cover voting systems in which a single alternative, such as a bill or an amendment, is pitted against the status quo. In these systems, each voter responds with a vote of yea and nay. Democratic societies and international organizations use a wide variety of complex rules to reach decisions. Examples, where it is not always easy to understand the consequences of the way voting is done, include the Electoral College to elect the President of the United States, the United Nations Security Council, the governance structure of the World Bank, the International Monetary Fund, the European Union Council of Ministers, the national governments of many countries, the councils in several counties, and the system to elect the major in cities or villages of many countries. Another source of examples comes from economic enterprises whose owners are shareholders of the society and divide profits or losses proportionally to the numbers of stocks they posses, but make decisions by voting according to a pre-defined rule (i.e., an absolute majority rule or a qualified majority rule). See [11, 12] for a thorough presentation of theses and other examples. Such systems have been analyzed as *simple games*.

Definition 1. A *simple game* Γ is a pair $(N, \mathcal{W})$ in which $N = \{1, 2, \ldots, n\}$ and $\mathcal{W}$ is a collection of subsets of N that satisfies: *(1)* $N \in \mathcal{W}$, *(2)* $\emptyset \notin \mathcal{W}$ and *(3)* the *monotonicity* property: $S \in \mathcal{W}$ and $S \subseteq T \subseteq N$ implies $T \in \mathcal{W}$.

The subsets of N are called *coalitions*, the coalitions in $\mathcal{W}$ are called *winning coalitions*, and the coalitions that are not winning are called *losing coalitions* (noted by $\mathcal{L}$). Moreover, we say that a coalition is minimal winning (maximal losing) if it is a winning (losing) coalition all of whose proper subsets (supersets) are losing (winning). Because of monotonicity, any simple game is completely determined by its set of minimal winning (maximal losing) coalitions denoted by $\mathcal{W}^m$ ($\mathcal{L}^M$). Note that a description of a simple game Γ can be given by $(N, \mathcal{X})$, where $\mathcal{X}$ is $\mathcal{W}$, $\mathcal{L}$, $\mathcal{W}^m$ or $\mathcal{L}^M$, see [12]. We focus on the process of exchanging or trading where a motivating example is the following:

Example 1. Consider two English football clubs that are in trouble and in danger of leaving Premier League. Maybe the two clubs could trade with each

other and exchange players so they both could avoid relegation. We consider the complexity of figuring out if such an exchange is possible for various ways of knowing what it takes to form a strong team that is able to stay in Premier League. This can be viewed as a simple game where a winning coalition corresponds to a strong team of players.

The considered property is the so called j-trade property for simple games. Loosely speaking, a simple game is j-trade if it is possible to rearrange the players in a set of j winning (losing) coalitions into a set of j losing (winning) coalitions, in such a way that the total number of occurrences of each player is the same in both sets. Thus, it is possible to go from one set to the other via participant trades. This notion was introduced by Taylor and Zwicker [12] in order to obtain a characterization of the weighted games, a subfamily of simple games. Recall that any simple game can be expressed as the intersection of weighted simple games. This leads to the definition of the *dimension* concept, the minimum number of required weighted games whose intersection represents the simple game [2, 6, 3]. Due to this fact, the problem of deciding whether a simple game is weighted has been of interest in several contexts. With respect to tradeness, it is known that a simple game is weighted if and only if it is not j-trade for any non-negative integer j [12]. Freixas *et al.* [5] studied the computational complexity of deciding whether a simple game is weighted among other decision problems for simple games. In particular, they showed that deciding whether a simple game is weighted is polynomial time solvable when the game is given by an explicit listing of one of the families $\mathcal{W}$, $\mathcal{L}$, $\mathcal{W}^m$, $\mathcal{L}^M$. On the other hand, the j-trade concept was also redefined as j-invariant-trade of simple games [4] and extended as (j, k)-simple games [7].

Here we provide a definition of j-trade that uses a formalism that differ from the classic one for j-trade *robustness* applied to a simple game (see [1, 12, 4]) in order to ease the proofs of our new results.

Definition 2. Given a simple game Γ, a *j-trade application* is a set of $2j$ coalitions $(S_1, \ldots, S_{2j})$ such that $\exists I \subseteq \{1, \ldots, 2j\}$ that satisfies:

1. $|I| = j$
2. $\forall i \in \{1, \ldots, 2j\}, S_i \in \mathcal{W} \iff i \in I$
3. $\forall p \in N, |\{i \in I : p \in S_i\}| = |\{i \in \{1, \ldots, 2j\} \setminus I : p \in S_i\}|$

Definition 3. A simple game Γ is *j-trade* if it admits a j-trade application.

Example 2. The simple game defined by $(N, \mathcal{W}^m) = (\{1, 2, 3, 4\}, \{\{1, 3\}, \{2, 4\}\})$ is 2-trade because it admits a 2-trade application. For instance, we can consider the following set of coalitions $(\{1, 3\}, \{2, 4\}, \{1, 2\}, \{3, 4\})$ where $\{1, 3\}, \{2, 4\} \in \mathcal{W}$, but $\{1, 2\}, \{3, 4\} \in \mathcal{L}$.

Example 3. It is easy to generate a simple game that will be $2j$-trade, for an integer j. For instance, we can take the simple game $(N, \mathcal{W}^m)$ where $N = \{1, \ldots, 2j\}$ and $\mathcal{W}^m = \{\{i, i+1\} \mid i \in 1, 3, 5, \ldots, 2j-1\}$. It is clear that coalitions $L_i = \{i, i+1\}$, for all $i \in \{2, 4, 6, \ldots, 2j-2\}$, and $L_{2j} = \{1, 2j\}$ are losing. Thus, the set of $2j$ coalitions $\mathcal{W}^m \cup \left(\cup_{i=1}^{j} L_{2i} \right)$ generates a j-trade application.

Definition 4. A simple game Γ is *j-trade robust* if it is not *j-trade*.

Before formally defining the *decision problems* we focus on, we consider two functions α and β associating games with various types of sets of coalitions. The allowed types are the following $\alpha(\Gamma) \in \{\mathcal{W}, \mathcal{L}, \mathcal{W}^m, \mathcal{L}^M\}$ and $\beta(\Gamma) \in \{\mathcal{W}, \mathcal{L}\}$, respectively. Moreover, given the β application we consider the function $\overline{\beta}$ that provides the *complementary* type with respect to the function β.

$$\overline{\beta}(\Gamma) = \begin{cases} \mathcal{W}, & \textit{if } \beta(\Gamma) = \mathcal{L} \\ \mathcal{L}, & \textit{if } \beta(\Gamma) = \mathcal{W} \end{cases}$$

Now we can state the definition of the considered computational problems, observe that the value of α provides the type of coalitions used in the representation of the input game while the β function indicates the type of the coalitions to be exchanged.

Definition 5. The (α, β, j)-*trade* problem, where $j \in \mathbb{N}$, is

Input: A simple game Γ given by $(N, \alpha(\Gamma))$ and j coalitions $S_1, \ldots, S_j \in \beta(\Gamma)$.

Question: Do there exist $S_{j+1}, \ldots, S_{2j} \in \overline{\beta}(\Gamma)$ such that $(S_1, \ldots, S_{2j})$ is a j-trade application?

Definition 6. The (α, β)-Trade problem is the $(\alpha, \beta, 2)$-trade problem.

$\alpha(\Gamma) \setminus \beta(\Gamma)$	$\mathcal{W}$
$\mathcal{W}$	polynomial
$\mathcal{W}^m$	NP-complete
$\mathcal{L}$	polynomial
$\mathcal{L}^M$	polynomial

$\alpha(\Gamma) \setminus \beta(\Gamma)$	$\mathcal{L}$
$\mathcal{W}$	polynomial
$\mathcal{W}^m$	polynomial
$\mathcal{L}$	polynomial
$\mathcal{L}^M$	NP-complete

Table 1: Complexity for the (α, β)-`Trade` problem, where `polynomial` means polynomially time solvable.

In the remaining part of the paper we analyze the computational complexity of the above problems. Table 1 summarizes all results about the (α, β)-`Trade` problem. We present first the results for the (α, β)-`Trade` problem and then the results for the general case. We finalize with some conclusions and open problems.

2. The computational complexity of trading two given coalitions

We present first the types for which the (α, β)-`Trade` problems are polynomial time solvable.

Proposition 1. The (α, β)-`Trade` problem is polynomially time solvable when $\alpha(\Gamma) \in \{\mathcal{W}, \mathcal{W}^m, \mathcal{L}\}$ and $\beta(\Gamma) = \mathcal{L}$.

Proof. We analyze each case separately. Let S_1, S_2 be two coalitions and assume that both are of type $\beta(\Gamma) = \mathcal{L}$.

- Case $\alpha(\Gamma) = \mathcal{W}$. Observe that we only need to check whether there are two coalitions $S_3, S_4 \in \mathcal{W}$ such that (S_1, S_2, S_3, S_4) is a 2-trade application. This property can be trivially checked in polynomial time by considering all the pairs of coalitions in $\mathcal{W}$. Therefore in polynomial time with respect to the input size.

- Case $\alpha(\Gamma) = \mathcal{W}^m$. The algorithm is the following. First, we look for the existence of two coalitions $S_3, S_4 \in \mathcal{W}^m$ such that, $\forall p \in N$, $|\{i \in \{3, 4\} : p \in S_i\}| \leq |\{i \in \{1, 2\} : p \in S_i\}|$. Observe that if such a pair of coalitions exists we can add the missing players (if any) in such a way that, $\forall p \in N$, $|\{i \in \{3, 4\} : p \in S_i\}| = |\{i \in \{1, 2\} : p \in S_i\}|$ and obtain a 2-trade application.

- Case $\alpha(\Gamma) = \mathcal{L}$. Now we compute $\mathcal{W}^m$ from $\mathcal{L}$ using the polynomial time algorithm shown in [5] and reduce the problem to the previous case.

$\square$

The same result can be proven when $\beta(\Gamma) = \mathcal{W}$.

Proposition 2. The (α, β)-`Trade` problem is polynomially time solvable when $\alpha(\Gamma) \in \{\mathcal{L}, \mathcal{L}^M, \mathcal{W}\}$ and $\beta(\Gamma) = \mathcal{W}$.

Proof. Arguments are symmetric to Proposition 1. Let S_1, S_2 be two coalitions and assume that both are of type $\beta(\Gamma) = \mathcal{W}$.

- Case $\alpha(\Gamma) = \mathcal{L}$. Here it is enough to check all pairs of losing coalitions. The reasoning is symmetric to the first case of Proposition 1.

- Case $\alpha(\Gamma) = \mathcal{L}^M$. The algorithm is symmetric to the second case of Proposition 1. We check whether there are two maximal loosing coalitions S_3 and S_4 so that, $\forall p \in N$, $|\{i \in \{3, 4\} : p \in S_i\}| \geq |\{i \in \{1, 2\} : p \in S_i\}|$. If this is the case, by removing the additional players we get a 2-trade application.

- If $\alpha(\Gamma) = \mathcal{W}$, we compute $\mathcal{L}^M$ from $\mathcal{W}$ using the polynomial time algorithm given in [5] and use the algorithm for the previous case.

$\square$

In the following results we isolate the types giving rise to computationally hard cases.

Proposition 3. The (α, β)-`Trade` problem is NP-complete when $\alpha(\Gamma) = \mathcal{L}^M$ and $\beta(\Gamma) = \mathcal{L}$.

Proof. The considered (α, β)-`Trade` problem is easily seen to be a member of NP. We show that it is also NP-hard providing a reduction from the SAT problem. Recall that the SAT asks whether a given boolean formula ϕ given in conjunctive normal form is satisfiable or not. The SAT problem is a famous NP-complete problem [8]. We let $X = \{x_1, \neg x_1, x_2, \neg x_2, \ldots, x_n, \neg x_n\}$ be the literals of ϕ and let X_i be the set of literals in the i'th clause of ϕ. Let m denote the number of clauses of ϕ. Our reduction transforms ϕ into an equivalent instance of the considered (α, β)-`Trade` problem in polynomial time.

The set of players of the associated game $\Gamma = \Gamma(\phi)$ contains the literals and two extra players a and b: $N = X \cup \{a, b\}$. A set of players Y can win if and only if at least one of the following two conditions are met:

$$a \in Y \wedge \forall i = \{1, 2, \ldots, m\} : Y \cap X_i \neq \emptyset \quad (1)$$

$$b \in Y \wedge \forall j = \{1, 2, \ldots, n\} : x_j \in Y \vee \neg x_j \in Y (2)$$

It is not hard to see that this is indeed a simple game since any superset of a winning set is also winning. We now have to show how to construct the set of maximal loosing coalitions L^M for this game in polynomial time. A set of players S is loosing if and only if (1) and (2) are both violated. This happens if and only if at least one of the following four conditions are met:

$$S \subseteq N \setminus \{a, b\} \quad (3)$$

$$\exists i : S \subseteq N \setminus (X_i \cup \{b\}) \quad (4)$$

$$\exists j : S \subseteq N \setminus \{a, x_j, \neg x_j\} \quad (5)$$

$$\exists i, j : S \subseteq N \setminus (X_i \cup \{x_j, \neg x_j\}) \quad (6)$$

If we consider all possible combinations of i and j then the sets on the right hand side of these expressions form a set of loosing sets. Any loosing set is contained in at least one of those sets. If we pick the maximal sets of this family – which can be done in polynomial time – we get $\mathcal{L}^M$ for the game Γ. The sets S_1 and S_2 are constructed as follows: $S_1 = \{a, b\}$ and $S_2 = X$.

Now assume that ϕ is a yes-instance to SAT. Let S_3 be the set formed by the player a and all literals corresponding to a truth-assignment satisfying ϕ and let S_4 be the set formed by the remaining literals and the player b. It is easy to see that coalitions (S_1, S_2, S_3, S_4) are a 2-trade application of Γ, where $S_1, S_2 \in \mathcal{L}$ and $S_3, S_4 \in \mathcal{W}$.

On the other hand, note that such 2-trade application only exists if one of the winning sets contain a and a set of literals defining a truth-assignment satisfying ϕ. Thus, the instances to the SAT-problem and the considered (α, β)-Trade problem are equivalent. $\square$

Using a symmetric construction to the previous one we have.

Proposition 4. The (α, β)-Trade problem is NP-complete when $\alpha(\Gamma) = \mathcal{W}^m$ and $\beta(\Gamma) = \mathcal{W}$.

We conclude this section by isolating a parameter for which one of the hard cases is fixed parameter tractable. We do so by providing a parameterized reduction to the SetSplitting problem.

Proposition 5. The (α, β)-Trade problem, being $\alpha(\Gamma) = \mathcal{L}^M$ and $\beta(\Gamma) = \mathcal{L}$, is fixed parameter tractable when considering the parameter $k = |\mathcal{L}^M|$.

Proof. Recall that according to [8] the SetSplitting problem is the following *decision problem*: Given a family $\mathcal{F}$ of subsets of a finite set U and an integer k, decide whether there exists a partition of U into two subsets U_1 and U_2 such that at least k elements of $\mathcal{F}$ are split by this partition. Lokshtanov and Saurabh [9] show that the SetSplitting problem is fixed parameter tractable when the parameter is the integer k.

Now we provide a fixed parameter reduction from our case of the (α, β)-Trade problem to the SetSplitting problem.

Given a simple game $(N, \mathcal{L}^M)$ and $S_1, S_2 \in \mathcal{L}$. Let $Z_i = (S_1 \cup S_2) \setminus L_i$ for $L_i \in \mathcal{L}^M$ such that $S_1 \cap S_2 \subseteq L_i$. Let $\mathcal{F}$ be the family of sets consisting of all the Z_i's. We construct the input to the SetSplitting problem given by $U = S_1 \cup S_2$, $\mathcal{F}$ and $k = |\mathcal{F}|$.

Notice that there must be at least one member in $\mathcal{F}$ since S_1 and S_2 are loosing coalitions. Moreover, the number of sets in $\mathcal{F}$ is always less than or equal to $|\mathcal{L}^M|$.

Now we prove the following claim: Given a simple game $(N, \mathcal{L}^M)$ and $S_1, S_2 \in \mathcal{L}$, there exists $S_3, S_4 \in \mathcal{W}$ where (S_1, S_2, S_3, S_4) is a 2-trade application if and only if $\mathcal{F}$ is a yes-instance to the SetSplitting problem.

If: Now assume that $\mathcal{F}$ is a yes-instance to the SetSplitting problem with the sets U_1 and U_2 splitting the members of $\mathcal{F}$. We will now prove that the sets $S_3 = U_1 \cup (S_1 \cap S_2)$ and $S_4 = U_2 \cup (S_1 \cap S_2)$ are winning coalitions. Now consider an arbitrary member $L_i \in \mathcal{L}^M$. If $S_1 \cap S_2 \subseteq L_i$ then $U_1 \cap Z_i \neq \emptyset$ implying $S_3 \not\subseteq L_i$. If $S_1 \cap S_2 \not\subseteq L_i$ then we also have $S_3 \not\subseteq L_i$. This holds for any $L_i \in \mathcal{L}^M$ implying that S_3 is winning. The same goes for S_4. Note that (U_1, U_2) is a partition of $S_1 \cup S_2$, so every player appears the same number of times in S_3 and S_4 as in S_1 and S_2.

Only if: Assume that we have a 2-trade application (S_1, S_2, S_3, S_4) where $S_3, S_4 \in \mathcal{W}$ and $S_1, S_2 \in \mathcal{L}$. Let $R = \{L \in \mathcal{L}^M : S_1 \cap S_2 \subseteq L\}$ and $U_i = S_i \setminus (\cap_{L \in R} L)$ for $i \in \{3, 4\}$. It is not hard

to see that U_3 and U_4 are disjoint. What remains is to show that U_3 and U_4 split all the members of $\mathcal{F}$.

Now consider an arbitrarily chosen set Z_i. The coalition S_3 is winning. Therefore it contains at least one player that is not a member of $L_i \in \mathcal{L}^M$. This player is a member of U_3 and it is also a member of Z_i, so $U_3 \cap Z_i \neq \emptyset$. Finally, the same argument can be used for U_4.

Using the FPT Algorithm for the `SetSplitting` problem with complexity $f(k)+p(n)$, where $k = |\mathcal{F}|$ and $n = |N|$, for some function f and a polynomial p, we get that the total complexity of the composed algorithm is $f(|\mathcal{L}^M|)+p(n)+O(n|\mathcal{L}^M|)$ as the time of computing the associated instance is $O(n|\mathcal{L}^M|)$. Thus, the claim follows. □

3. The computational complexity of trading j coalitions

Note that it is enough to check combinations of j-coalitions in $\mathcal{L}^M$ or $\mathcal{W}^m$ to seek for a j-trade application. This is so, because, if needed, we can remove players from a maximal losing coalition or add players to a minimal wining coalition getting loosing or winning coalitions that matches the requirements for the j-trade application.

Theorem 1. *For a fixed j, given a simple game $\Gamma = (N, \mathcal{W})$, we can decide whether such a simple game is j-trade in polynomial time. Furthermore, if the game is j-trade, a j-trade application can be efficiently computed.*

Proof. We start by computing $\mathcal{L}^M$ from $\mathcal{W}$ using the polynomial time algorithm given in [5]. Then, we try all the possible combinations of j members of $\mathcal{W}$ and j members of $\mathcal{L}^M$ (repetitions are allowed) to find a j-trade application. This requires $\binom{|\mathcal{W}|}{j}$ combinations.

The algorithm works because there is such j-trade application if and only if there is a j-trade application where all losing coalitions are maximal losing coalitions. □

Next we present a similar result but for the case in which the game is given by $\Gamma = (N, \mathcal{L})$.

Theorem 2. *For a fixed j, given a simple game $\Gamma = (N, \mathcal{L})$, we can decide whether such a simple game is j-trade in polynomial time. Furthermore, if the game is j-trade, a j-trade application can be efficiently computed.*

Proof. We start by computing $\mathcal{W}^m$ from $\mathcal{L}$ in polynomial time [5] and proceed as Theorem 1. □

Now we adapt the result of Proposition 3 for the (α, β)-`Trade` problem to the (α, β, j)-trade problem

Proposition 6. *The (α, β, j)-trade problem is NP-complete when $\alpha(\Gamma) = \mathcal{L}^M$ and $\beta(\Gamma) = \mathcal{L}$.*

Proof. The argument is quite similar to the one used in the proof of Proposition 3, but considering that the set of players of the game Γ contains the literals and $2 + 2 \cdot (j - 2)$ extra players, i.e., a, b, and c_{1i} and c_{2i} for $i \in \{3, \ldots, j\}$. Thus, we have that $N = X \cup \{a, b\} \cup_{i=3}^{j} \{c_{1i}, c_{2i}\}$. A set of players Y can win if and only if at least one of the following conditions are met:

$$a \in Y \wedge \forall i = \{1, 2, \ldots, m\} : Y \cap X_i \neq \emptyset \qquad (7)$$

$$b \in Y \wedge \forall j = \{1, 2, \ldots, n\} : x_j \in Y \vee \neg x_j \in Y \quad (8)$$

$$\{c_{1i}, c_{2i}\} \subseteq Y, \forall i \in \{3, 4, \ldots, j\} \ . \qquad (9)$$

As winning coalitions we have

$$\begin{aligned} S_{j+1} = \{&a, \textit{all true literals corresponding to} \\ &\textit{a truth-assignment satisfying } \phi\}, \\ S_{j+2} = \{&b, \textit{the remaining literals that are} \\ &\textit{not in } S_1\}, \\ S_{j+i} = \{&c_{1i}, c_{2i}\} \ , \ \forall i \in \{3, 4, \ldots, j\}. \end{aligned}$$

Finally, as losing coalitions we distinguish two cases:

- If j is even,

$$\begin{aligned} S_1 &= X, \\ S_2 &= \{a, b\}, \\ S_i &= \{c_{1i}, c_{1,i+1}\} \ , \ \forall i \in \{3, 5, 7, \ldots, j-1\} \\ S_i &= \{c_{2,i-1}, c_{2i}\} \ , \ \forall i \in \{4, 6, 8, \ldots, j\} \ . \end{aligned}$$

- If j is odd

$$\begin{aligned} S_1 &= X, \\ S_2 &= \{a, c_{1j}\}, \\ S_i &= \{c_{1i}, c_{1,i+1}\} \ , \ \forall i \in \{3, 5, 7, \ldots, j-1\} \\ S_i &= \{c_{2,i-1}, c_{2i}\} \ , \ \forall i \in \{4, 6, 8, \ldots, j-2\} \\ S_{2j} &= \{b, c_{2j}\} \ . \end{aligned}$$

□

4. Related Remaining Problems

In this paper we have focused on the computational complexity of trade robustness problems for simple games. Our results are summarized in Table 1. Nevertheless, there remain many related open questions. Let us highlight some of them.

Because constructing $\mathcal{W}$ or $\mathcal{L}$ from $\mathcal{W}^m$ or $\mathcal{L}^M$, respectively, is not polynomially solvable, we post the following conjecture.

Conjecture 1. *For a fixed j, to decide whether a game given by either $(N, \mathcal{W}^m)$ or $(N, \mathcal{L}^M)$ is j-trade is* `coNP`*-complete.*

We recall that a simple game is weighted if and only if it is j-trade robust for any non-negative integer j, see the characterization given by Taylor and Zwicker [12]. This leads us to the following problem

> Trade robustness problem.
>
> `Input:` A simple game Γ and a non-negative integer j.
>
> `Question:` Is Γ a j-trade robust game?

Whose computational complexity remains open for the different forms of representations of simple games considered in this paper.

In a recent paper Molinero *et al.* [10] introduced *influence games*. Influence games provide a succinct form of representation for simple games based on graphs. It would be of interest to analyze the complexity of the (α, β, j)-trade and the trade robustness problems when the simple game is given as an influence game.

References

[1] F. Carreras and J. Freixas. Complete simple games. *Mathematical Social Sciences*, 32:139–155, 1996.

[2] V.G. Deĭneko and G.J. Woeginger. On the dimension of simple monotonic games. *European Journal of Operational Research*, 170:315–318, 2006.

[3] J. Freixas. The dimension for the European Union Council under the Nice rules. *European Journal of Operational Research*, 156(2):415–419, 2004.

[4] J. Freixas and X. Molinero. Simple games and weighted games: A theoretical and computational viewpoint. *Discrete Applied Mathematics*, 157(7):1496–1508, April 2009.

[5] J. Freixas, X. Molinero, M. Olsen, and M. Serna. On the complexity of problems on simple games. *RAIRO - Operations Research*, 45:295–314, 9 2011.

[6] J. Freixas and M.A. Puente. Dimension of complete simple games with minimum. *European Journal of Operational Research*, 188(2):555–568, 2008.

[7] J. Freixas and W.S. Zwicker. Weighted voting, abstention, and multiple levels of approval. *Social Choice and Welfare*, 21:399–431, 2003.

[8] M. R. Garey and D. S. Johnson. *Computers and intractability, a guide to the theory of NP–Completness.* W.H. Freeman and Company, New York, USA, 1999. Twenty-first printing.

[9] D. Lokshtanov and S. Saurabh. Even faster algorithm for set splitting! In Jianer Chen and FedorV. Fomin, editors, *Parameterized and Exact Computation*, volume 5917 of *Lecture Notes in Computer Science*, pages 288–299. Springer Berlin Heidelberg, 2009.

[10] X. Molinero, F. Riquelme, and M. Serna. Cooperation through social influence. *European Journal of Operational Research*, 242(3):960–974, 2015.

[11] A.D. Taylor. *Mathematics and Politics.* Springer Verlag, New York, USA, 1995.

[12] A.D. Taylor and W.S. Zwicker. *Simple games: desirability relations, trading, and pseudoweightings.* Princeton University Press, New Jersey, USA, 1999.