RAIRO-Oper. Res. 45 (2011) 295–314
DOI: 10.1051/ro/2011115

RAIRO Operations Research
www.rairo-ro.org

ON THE COMPLEXITY OF PROBLEMS ON SIMPLE GAMES

Josep Freixas[1], Xavier Molinero[2], Martin Olsen[3] and Maria Serna[4]

Abstract. Simple games cover voting systems in which a single alternative, such as a bill or an amendment, is pitted against the status quo. A simple game or a yes-no voting system is a set of rules that specifies exactly which collections of "yea" votes yield passage of the issue at hand. Each of these collections of "yea" voters forms a winning coalition. We are interested in performing a complexity analysis on problems defined on such families of games. This analysis as usual depends on the game representation used as input. We consider four natural explicit representations: winning, losing, minimal winning, and maximal losing. We first analyze the complexity of testing whether a game is simple and testing whether a game is weighted. We show that, for the four types of representations, both problems can be solved in polynomial time. Finally, we provide results on the complexity of testing whether a simple game or a weighted game is of a special type. We analyze strongness, properness, weightedness, homogeneousness, decisiveness and majorityness, which are desirable properties to be fulfilled for a simple game. Finally, we consider the possibility of representing a game in a more succinct and natural way and show that the corresponding recognition problem is hard.

Keywords. Simple, weighted, majority games, NP-completeness.

Mathematics Subject Classification. 68Q, 91A.

Received March 9, 2011. Accepted September 1, 2011.

[1] Universitat Politècnica de Catalunya, EPSEM-DMA3, 08240 Manresa, Spain. josep.freixas@upc.edu

[2] Universitat Politècnica de Catalunya, EPSEM-DMA3, 08240 Manresa, Spain. xavier.molinero@upc.edu

[3] Center for Innovation and Business Development, Institute of Business and Technology, Aarhus University, Birk Centerpark 15, 7400 Herning, Denmark. martino@hih.au.dk

[4] Universitat Politècnica de Catalunya, LSI, 08034 Barcelona, Spain. mjserna@lsi.upc.edu

Article published by EDP Sciences

1. Introduction

Simple game theory is a very dynamic and expanding field. Taylor and Zwicker [22] pointed out that "*few structures arise in more contexts and lend themselves to more diverse interpretations than do simple games*". Indeed, 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" or "nay". A simple game or a yes-no voting system is a set of rules that specifies exactly which collections of "yea" votes yield passage of the issue at hand. Each of these collections of "yea" voters forms a winning coalition.

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).

There are several alternative ways to introduce a simple game; the most natural is by giving the list of winning coalitions in which case the complementary set is the set of losing coalitions and the simple game is fully described. A considerable reduction in terms of use of space in introducing a simple game can be obtained by considering only the list of minimal winning coalitions, *i.e.* winning coalitions which are minimal by the inclusion operation. Coalitions containing minimal winning coalitions are also winning. Analogously, one may present a simple game by using either the set of losing coalitions or the set of maximal losing coalitions.

We are interested in performing a complexity analysis of problems on simple games, in the case that the number of players is large, as pointed out in [5], "*from a computational point of view, the key issues relating to coalitional games are, first, how such games should be represented (since the obvious representation is exponentially large in the number of players); and second, the extent to which cooperative solution concepts can be efficiently computed*". We undertake here this task and to the best of our knowledge this is the first paper that addresses such a study.

Previous results have focused on problems where the input is a subclass of the class of simple games, the so called *weighted games*. A way to describe a weighted game is to assign a (positive) real number weight to each voter, and declare a coalition to be winning precisely when its total weight meets or exceeds some predetermined quota. Not every simple game is weighted but every simple game can be decomposed as an intersection of some weighted games. Work with the complexity

TABLE 1. Complexity of changing the representation form of a simple game.

Input → Output ↓	(N,W)	(N,L)	(N,W^m)	(N,L^M)
(N,W)	–	EXP	EXP	EXP
(N,L)	EXP	–	EXP	EXP
(N,W^m)	P	P	–	EXP
(N,L^M)	P	P	EXP	–

TABLE 2. Our results on the complexity of problems on simple games.

Input →	(N,W)	(N,W^m)	(N,L)	(N,L^M)	$(q;w)$
IsSimple	P	P	P	P	–
IsStrong	P	co-NPC	P	P	co-NPC
IsProper	P	P	P	co-NPC	co-NPC
IsWeighted	P	P	P	P	–
IsHomogeneous	P	?	P	?	?
IsDecisive	P	?	P	?	co-NPC
IsMajority	P	?	P	?	co-NPC

of problems on weighted games dates back to [19], where Prasad and Kelly provide NP-completeness results on determining properties of weighted voting games. For instance, they show that computing standard political power indices, such as absolute Banzhaf, Banzhaf-Coleman and Shapley-Shubik, are all NP-hard problems. More recent work is related to the notion of *dimension* considered by Taylor and Zwicker [21, 22]. The dimension of a simple game is the minimum number of weighted games whose intersection coincides with the game. The computational effort to weigh up the dimension of a simple game, given as the intersection of d weighted games, was determined by Deĭneko and Woeginger [2]: computing the dimension of a simple game is a NP-hard problem. More results on solution concepts for weighted games can be found in [3, 5, 6, 14]. There also exist works related to economics [4, 10].

Our first objective is to fix some natural game representations. After doing so, as usual, we analyze the complexity of transforming one representation into another and the complexity of the problem of recognizing simple games. Our second aim is to classify the complexity of testing whether a simple game is of a special type. Apart from weighted games there are some other subclasses of simple games which are very significant in the literature of voting systems. Strongness, properness, decisiveness and homogeneity are, among others, desirable properties to be fulfilled for a simple game. Our results are summarized in Tables 1 and 2.

Table 1 shows the complexity of passing from a given form to another one. All *explicit* forms are represented by a pair (N, C) in which $N = \{1, \ldots, n\}$ for some positive integer n, and C is the set of winning, minimal winning, losing or maximal losing coalitions. Note that it is possible to pass from winning (or losing) coalitions to minimal winning (or maximal losing) coalitions in polynomial time, but the other swaps require exponential time. On the other hand, given a game in a specific form, Table 2 shows the complexity on determining whether it is simple, strong, proper, weighted, homogeneous, decisive or majority. Here $(q; w)$ denotes an *integer representation* of a weighted game where q is the quota and w are the weights. Observe that there are some problems that still remain open.

Finally, we refer the reader to Papadimitriou [17] for the definitions of the complexity classes P, NP, co-NP, and their subclasses of complete problems NPC and co-NPC, and the counting class #P.

2. Recognizing simple games

We start stating some basic definitions on simple games (we refer the interested reader to [22] for a thorough presentation).

Simple games can be viewed as models of voting systems in which a single alternative, such as a bill or an amendment, is pitted against the status quo.

Definition 2.1. A simple game Γ is a pair (N, W) in which $N = \{1, \ldots, n\}$ for some positive integer n, and W is a collection of subsets of N that satisfies $N \in W$, $\emptyset \notin W$, and the *monotonicity* property: $S \in W$ and $S \subseteq R \subseteq N$ implies $R \in W$.

Any set of voters is called a *coalition*, the set N is called the *grand coalition*, and the empty set $\emptyset$ is called the *null coalition*. Members of N are called *players* or *voters*, and the subsets of N that are in W are called *winning coalitions*. The intuition here is that a set S is a winning coalition *iff* the bill or amendment passes when the players in S are precisely the ones who vote for it. A subset of N that is not in W is called a *losing coalition*. The collection of losing coalitions is denoted by L. The set of *minimal winning coalitions* (*maximal losing coalitions*) is denoted by W^m (L^M), where a minimal winning coalition (a maximal losing coalition) 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 coalitions. A voter i is null if $i \notin S$ for all $S \in W^m$.

From a computational point of view a simple game can be given under different representations. In this paper we essentially consider the following options:

- Explicit or extensive winning form: the game is given as (N, W) by providing a listing of the collection of subsets W.
- Explicit or extensive minimal winning form: the game is given as (N, W^m) by providing a listing of the family W^m. Observe that this form requires less space than the explicit winning form whenever $W \neq \{N\}$.

When we consider descriptions of a game in terms of winning coalitions in this paper, we also consider the corresponding representations for losing coalitions, replacing minimal by maximal. Thus, in addition we also consider the explicit or extensive losing, and explicit or extensive maximal losing forms.

We analyze first the computational complexity of obtaining a representation of a game in a given form when a representation in another form is given.

Theorem 2.2. *Given a simple game:*

(i) *passing from the explicit winning (losing) form to the explicit minimal winning and maximal losing (maximal losing and minimal winning) form can be done in polynomial time;*
(ii) *passing from the explicit minimal winning (maximal losing) form to the explicit winning (losing) form requires exponential time;*
(iii) *passing from the explicit minimal winning (maximal losing) form to the explicit maximal losing (minimal winning) form requires exponential time;*
(iv) *passing from the explicit minimal winning (maximal losing) form to the explicit losing (winning) form requires exponential time;*
(v) *passing from the explicit winning (losing) form to the explicit losing (winning) form requires exponential time.*

This theorem gives us all the results presented in Table 1. The polynomial time results are obtained from simple properties of monotonic sets. For the exponential time transformations we provide examples in which the size of the representation increases exponentially. The transformations are similar to the ones used to show that computing a CNF[5] from a given DNF[6] requires exponential time. The difference relies in that now instead of transforming the same formula we have to get a different maximal normal form for a formula and its negation.

Before proving Theorem 2.2 in detail, we introduce some notations and definitions together with some preliminary technical results.

Given a family of subsets C of a set N, $\overline{C}$ denotes the closure of C under $\subseteq$, and $\underline{C}$ the closure of C under $\supseteq$.

Definition 2.3. A subset C of a set N is closed under $\subseteq$ ($\supseteq$) if $C = \overline{C}$ ($\underline{C}$).

The following lemma is proved in [17].

Lemma 2.4. *Given a family of subsets C of a set N, we can check whether it is closed under $\subseteq$ or $\supseteq$ in polynomial time.*

Lemma 2.5. *Given a family of subsets C of a set N, the families $\overline{C}^m$ and $\underline{C}^M$ can be obtained in polynomial time.*

Proof. Observe that, for any set S in C we have to check whether there is a subset (superset) of S that forms part of C, and keep those S that do not have this property. Therefore, the complete computation can be done in polynomial time on the input length of C. □

Now we define the minimal and maximal subset families.

[5]A Boolean formula is in *Conjunctive Normal Form* (CNF) *iff* it is a conjunction of disjuntion of literals.

[6]A Boolean formula is in *Disjunctive Normal Form* (DNF) *iff* it is a standardization (or normalization) of a logical formula which is a disjuntion of conjunction of literals.

Definition 2.6. Given a family of subsets C of a set N, we say that it is minimal if $C = \overline{C}^m$.

Definition 2.7. Given a family of subsets C of a set N, we say that it is maximal if $C = \underline{C}^M$.

As a consequence of Lemma 2.5 we have the following corollary.

Corollary 2.8. *Given a family of subsets C of a set N, we can check whether it is maximal or minimal in polynomial time.*

The proof of Theorem 2.2 is split into five lemmas. We start with our first result for simple games given in explicit winning or losing form.

Lemma 2.9. *Given a simple game Γ in explicit winning (or losing) form, the representation of Γ in explicit minimal winning or maximal losing (maximal losing or minimal winning) form can be obtained in polynomial time.*

Proof. Given a simple game $\Gamma = (N, W)$, consider the set

$$R = \bigcup_{i=1}^{n} W_{-i}$$

where $W_{-i} = \{S \setminus \{i\} : i \in S \in W\}$. Observe that all the sets in $R \backslash W$ are losing coalitions, $R \backslash W \subseteq L$. We claim that $(R \backslash W)^M = L^M$. We are going to prove that in two steps:

- $(R \backslash W)^M \subseteq L^M$: now suppose that $T \in (R \backslash W)^M$ and that $T \notin L^M$. Consequently, we have that $T \in L$ and that $T \cup \{i\} \in W$ for some $i \in N$. We also have that $T \subset U$ for some $U \in L$. Due to the monotonicity we conclude that $U \cup \{i\} \in W$. This means that $U \in R \backslash W$ which contradicts that T is maximal in $R \backslash W$;
- $L^M \subseteq (R \backslash W)^M$: we will show this inclusion in two steps:
  (i) $L^M \subseteq R \backslash W$: if $T \in L^M$ then $T \cup \{i\} \in W$ for any $i \notin T$. Thus T can be obtained from a winning coalition $(T \cup \{i\})$ from removing an element (i). This means that $T \in R \backslash W$ since T is a losing coalition;
  (ii) maximal elements in a set will also be maximal in any subset they appear in. From $L^M \subseteq R \backslash W \subseteq L$ we conclude that $L^M \subseteq (R \backslash W)^M$.

For the cost of the algorithm, observe that, given (N, W), the set R has cardinality at most $|N| \cdot |W|$, and thus R can be obtained in polynomial time. Using Lemma 2.5, from W and $R \backslash W$, we can compute W^m and L^M in polynomial time.

Analogously, when the game is given by the family of losing coalitions a symmetric argument provides the proof for explicit maximal losing or minimal winning form. □

Now we focus on simple games given in explicit minimal winning or explicit maximal losing form.

Lemma 2.10. *Given a simple game Γ in explicit minimal winning (maximal losing) form, computing the representation of Γ in explicit winning (losing) form requires exponential time.*

Proof. The following two examples show that the size of the computed family can be exponential in the size of the given one. Therefore, any algorithm that solves the problem requires exponential time.

Consider $N = \{1, \ldots, n\}$, then:

(i) the simple game defined by $W^m = \bigcup_{i=1}^n \{\{i\}\}$ has $W = \{T \subseteq N : T \neq \emptyset\}$. Therefore, $|W^m| = n$ and $|W| = 2^n - 1$;
(ii) the simple game defined by $L^M = \{T \subseteq N : |T| = n - 1\}$ has $L = \{T \subset N\}$. Therefore, $|L^M| = n$ and $|L| = 2^n - 1$. □

Lemma 2.11. *Given a simple game Γ in explicit minimal winning (maximal losing) form, computing the representation of Γ in explicit maximal losing (minimal winning) form requires exponential time.*

Proof. In a similar way as we did in the previous Lemma, we show two examples where size of the computed family can be exponential in the size of the given one.

Consider $N = \{1, \ldots, 2n\}$ and coalitions $S_i = \{2i - 1, 2i\}$, for all $i = 1, \ldots, n$. Then,

(i) the simple game defined by $W^m = \bigcup_{i=1}^n \{S_i\}$ has

$$L^M = \{T \subseteq N : |T \cap S_i| = 1, \text{for all } i = 1, \ldots, n\}.$$

Therefore, $|W^m| = n$ and $|L^M| = 2^n$;

(ii) the simple game defined by

$$W^m = \{T \subseteq N : |T \cap S_i| = 1, \text{for all } i = 1, \ldots, n\}$$

has $L^M = \bigcup_{i=1}^n \{N \setminus S_i\}$. Therefore, $|W^m| = 2^n$ and $|L^M| = n$. □

As a consequence of Lemmas 2.9 and 2.11 we have Corollary 2.12.

Corollary 2.12. *Given a simple game Γ in explicit minimal winning (maximal losing) form, computing the representation of Γ in explicit losing (winning) form requires exponential time.*

The remaining cases of Theorem 2.2 are again computationally hard.

Lemma 2.13. *Given a simple game Γ in explicit winning (losing) form, computing the representation of Γ in explicit losing (winning) form requires exponential time.*

Proof. We present two examples where the size of the computed family is exponential in the size of the given one. Let (N, W) be the game, where $W = \{N\}$, then $|W| = 1$ and $|L| = 2^{|N|} - 1$. Similarly, let (N, W) be the game, where $L = \{\emptyset\}$, then $|W| = 2^{|N|} - 1$ and $|L| = 1$. □

Lemmas (2.9)–(2.13) make up Theorem 2.2.

The next step is to analyze the computational complexity of the following recognition problems:

Name: IsSimpleE
Input: (N, C).
Question: Is (N, C) a correct explicit representation of a simple game?

We have in total four different problems depending on the input description: winning, minimal winning, losing and maximal losing. However, the recognition problem becomes polynomial time solvable in all these cases.

Theorem 2.14. *The* IsSimpleE *problem belongs to* P *for any explicit form* F*: winning, minimal winning, losing, or maximal losing.*

Proof. The proof follows from the fact that given a family of subsets C of a set N, the families of minimal or maximal sets of its closure can be obtained in polynomial time. It is a direct consequence of Lemmas 2.4 and 2.5 and Corollary 2.8, stating that whether the family is monotonic[7] or minimal/maximal can be tested in polynomial time. □

Observe that, as the recognition problem can be solved in polynomial time, we can use any of the proposed representations in further complexity analysis.

3. Problems on simple games

In this section we consider a set of decision problems related to properties that define some special types of simple games (again we refer the reader to [22]). In general we will state a property P for simple games and consider the associated decision problem which has the form:

Name: IsP
Input: A simple game Γ.
Question: Does Γ satisfy property P?

Further considerations on the complexity of such problems will be stated in terms of the input representation.

3.1. Recognizing strong and proper games

Now we study the complexity of determining if a given simple game (in explicit form) is strong, weak, proper or improper.

Definition 3.1. A simple game (N, W) is *strong* if $S \notin W$ implies $N \setminus S \in W$. A simple game that is not strong is called *weak*.

[7]We say that a family of sets is *monotonic iff* it satisfies the monotonicity property.

Intuitively speaking, if a game is weak it has too few winning coalitions, because adding sufficiently many winning coalitions would make the game strong. Note that the addition of winning coalitions can never destroy strongness.

Definition 3.2. A simple game (N, W) is *proper* if $S \in W$ implies $N \setminus S \notin W$. A simple game that is not proper is called *improper*.

An improper game has too many winning coalitions, in the sense that deleting sufficiently many winning coalitions would make the game proper. Note that the deletion of winning coalitions can never destroy properness.

When a game is both proper and strong, a coalition wins *iff* its complement loses. Therefore, in this case we have $|W| = |L| = 2^{n-1}$.

A related concept to the properness and strongness is the dualityness.

Definition 3.3. Given a simple game (N, W), its *dual game* is (N, W^*), where $S \in W^*$ if and only if $N \setminus S \notin W$.

That is, winning coalitions in the dual game are just the "blocking" coalitions in the original game. Thus, (N, W) is proper *iff* (N, W^*) is strong, and (N, W) is strong *iff* (N, W^*) is proper.

Theorem 3.4. *The* IsStrong *problem, when the input game is given in explicit losing or maximal losing form, and the* IsProper *problem, when the game is given in explicit winning or minimal winning form, can be solved in polynomial time.*

Proof. To prove this result we provide an adequate formalization of the strong and proper properties in terms of simple properties of the set of minimal winning or maximal losing coalitions respectively. Those properties can be checked in polynomial time when the games are given in the specified forms.

First observe that, given a family of subsets F, we can check, for any set in F, whether its complement is not in F in polynomial time. Therefore, taking into account the definitions, we have that the IsStrong problem, when the input is given in explicit losing form, and the IsProper problem, when the input is given in explicit winning form, are polynomial time solvable.

Thus, taking into account that:

- a simple game is weak *iff*

$$\exists S \subseteq N : S \in L \ \wedge \ N \setminus S \in L$$

which is equivalent to

$$\exists S \subseteq N : \exists L_1, L_2 \in L^M : S \subseteq L_1 \ \wedge \ N \setminus S \subseteq L_2.$$

The last assertion is equivalent to the fact that there are two maximal losing coalitions L_1 and L_2 such that $L_1 \cup L_2 = N$;

- a simple game is *improper iff*

$$\exists S \subseteq N : S \in W \ \wedge \ N \setminus S \in W$$

  which is equivalent to

$$\exists S \subseteq N : \exists W_1, W_2 \in W^m : W_1 \subseteq S \ \wedge \ W_2 \subseteq N \setminus S.$$

  This last assertion is equivalent to the fact that there are two minimal winning coalitions W_1 and W_2 such that $W_1 \cap W_2 = \emptyset$.

Observe that, given a family of subsets F, checking whether any one of the two conditions hold can be done in polynomial time. Thus, the theorem holds also when the set of maximal losing (or minimal winning) coalitions is given. □

As a consequence of Theorems 2.2 and 3.4 we have:

Corollary 3.5. *The* IsStrong *problem, when the input game is given in explicit winning form, and the* IsProper *problem, when the game is given in explicit losing form, can be solved in polynomial time.*

Our next result states the complexity of the IsStrong problem when the game is given in the remaining form.

Theorem 3.6. *The* IsStrong *problem is* co-NP*-complete when the input game is given in explicit minimal winning form.*

Proof. The membership proof follows from an adequate formalization. To prove hardness we consider the *set splitting* problem in which we are asked whether it is possible to partition N into two subsets P and $N \backslash P$ such that no subset in a given collection C is entirely contained in either P or $N \backslash P$. It is known that this problem is NP-complete [9]. We provide a polynomial time reduction from *set splitting* to the IsWeak problem. In other words we have to decide whether $P \subseteq N$ exists such that

$$\forall S \in C : S \nsubseteq P \wedge S \nsubseteq N \setminus P. \tag{3.1}$$

We transform a set splitting instance (N, C) into the simple game in explicit minimal winning form (N, C^m). This transformation can be computed in polynomial time according to Lemma 2.5. We will now show that (N, C) has a set splitting *iff* (N, C^m) is a weak game:

- now assume that $P \subseteq N$ satisfying (3.1) exists. This means that P and $N \backslash P$ are losing coalitions in the game (N, C^m);
- let P and $N \backslash P$ be losing coalitions in the game (N, C^m). As a consequence we have that $S \nsubseteq P$ and $S \nsubseteq N \backslash P$ for any $S \in C^m$. This implies that $S \nsubseteq P$ and $S \nsubseteq N \backslash P$ holds for any $S \in C$ since any set in C contains a set in C^m. □

Finally we prove a similar complexity result for the remaining version of the IsProper problem.

Theorem 3.7. *The* IsProper *problem is* co-NP*-complete when the input game is given in extensive maximal losing form.*

Proof. The hardness of the IsProper problem is obtained by using duality and providing a polynomial time reduction from the IsStrong problem.

From Definition 3.2, a game is *improper* if and only if there exists a coalition $S \subseteq N$ such that neither S nor $N \setminus S$ is contained in a member of L^M. For a given coalition S we can easily perform this check in polynomial time. Therefore the problem IsImproper belongs to NP, and the problem IsProper belongs to co-NP.

To complete the proof we provide a reduction from the IsStrong problem for games given in extensive minimal winning form. First observe that, if a family C of subsets of N is minimal then the family $\{N \setminus L : L \in C\}$ is maximal. Given a game $\Gamma = (N, W^m)$, in minimal winning form, let us consider its dual game $\Gamma' = (N, \{N \setminus L : L \in W^m\})$ given in maximal losing form. Of course Γ' can be obtained from Γ in polynomial time. Thus Γ is weak *iff*

$$\exists S \subseteq N : S \in L(\Gamma) \ \wedge \ N \setminus S \in L(\Gamma)$$

which is equivalent to

$$\exists S \subseteq N : N \setminus S \in W(\Gamma') \ \wedge \ S \in W(\Gamma')$$

iff Γ' is improper.

Thus, the IsProper problem belongs to co-NP and it is co-NP-hard – in other words it is co-NP-complete. □

3.2. Recognizing weighted games

In this subsection we study the complexity of determining if a given simple game (in explicit form) is weighted, trade robust or invariant trade robust.

Definition 3.8. A simple game (N, W) is *weighted* if there exist a "*quota*" $q \in \mathbb{R}$ and a "*weight function*" $w : N \to \mathbb{R}$ such that each coalition S is winning exactly when the sum of weights of S meets or exceeds q.

Weighted games are probably the most important kind of simple games. Any specific example of a weight function w and quota q is said to *realize* G as a weighted game. A particular realization of a weighted game is denoted $(q; w_1, \ldots, w_n)$, or briefly $(q; w)$. By $w(S)$ we denote $\sum_{i \in S} w_i$.

Observe also that, from the *monotonicity property*, it is obvious that a simple game (N, W) is *weighted iff* there exist a "*quota*" $q \in \mathbb{R}$ and a "*weight function*" $w : N \to \mathbb{R}$ such that

$$w(S) \geq q \qquad \forall \ S \in W^m$$

$$w(S) < q \qquad \forall \ S \in L^M.$$

Theorem 3.9. *The* IsWeighted *problem can be solved in polynomial time when the input game is given in explicit winning, losing, minimal winning and maximal losing forms.*

Proof. A simple polynomial time reduction from the IsWeighted problem to the *Linear Programming* problem, which is known to be solvable in polynomial time [12, 13], gives the result for the cases of explicit winning and explicit losing forms.

Taking into account Lemma 2.5, in both cases we can obtain W^m and L^M in polynomial time. Once this is done we can write, again in polynomial time, the following *Linear Programming* instance Π:

$$\begin{array}{lll} \min q & & \\ \text{subject to} & w(S) \geq q & \text{if } S \in W^m \\ & w(S) < q & \text{if } S \in L^M \\ & 0 \leq w_i & \text{for all } 1 \leq i \leq n \\ & \sum_i w_i = 1 & \\ & 0 \leq q. & \end{array}$$

The game (N, W) is weighted *iff* Π has a solution and the proposed construction is a polynomial time reduction.

For the minimal winning form we provide a reduction to the *threshold function* problem for monotonic DNF formulas which is known to be polynomial time solvable [11, 18]. For the maximal losing form we make use of duality and provide a reduction to the problem when the input is described in minimal winning form.

Given (N, W^m) we are going to prove that we can decide in polynomial time whether the simple game is weighted. For $C \subseteq N$ we let $x_C \in \{0,1\}^n$ denote the vector with the i'th coordinate equal to 1 if and only if $i \in C$. In polynomial time we transform W^m into the Boolean function Φ_{W^m} given by the DNF formula:

$$\Phi_{W^m}(x) = \bigvee_{S \in W^m} (\wedge_{i \in S} x_i).$$

By construction we have the following:

$$\Phi_{W^m}(x_C) = 1 \Leftrightarrow C \text{ is winning in the game given by } (N, W^m) \tag{3.2}$$

Note that Φ_{W^m} is a threshold function if and only if the game given by (N, W^m) is weighted:

- **only if** ($\Rightarrow$): assume that Φ_{W^m} is a threshold function. Let $w \in \mathbb{R}^n$ be the weights and $q \in \mathbb{R}$ the threshold value. Thus we have that

  $$\Phi_{W^m}(x_C) = 1 \Leftrightarrow \langle w, x_C \rangle \geq q$$

  where $\langle \cdot, \cdot \rangle$ denotes the usual inner product. By using (3.2) we conclude that the game given by (N, W^m) is weighted;

- **if** ($\Leftarrow$): now assume that the game given by (N, W^m) is weighted and that $(q; w)$ is a realization of such game. In this case we have the following:

$$C \text{ is winning in the game given by } (N, W^m) \Leftrightarrow \langle w, x_C \rangle \geq q.$$

  Again we use (3.2) and conclude that Φ_{W^m} is a threshold function.

The Boolean function Φ_{W^m} is monotonic (*i.e. positive*) so according to the papers [11, 18] (pp. 211 and 59, respectively) we can in polynomial time decide whether Φ_{W^m} is a threshold function. Consequently, we can also decide in polynomial time whether the game given by (N, W^m) is weighted.

On the other hand, we can prove a similar result given (N, L^M) just taking into account that a game Γ is weighted *iff* its dual game Γ' is weighted. Then, we can use the technique from the proof of Theorem 3.7. □

It is important to remark that it is known that "*a simple game is weighted* iff *it is trade robust* iff *it is invariant-trade robust*" [5, 7, 22]. Thus, according to Theorem 3.9, checking whether a simple game is trade robust or invariant-trade robust can be done in polynomial time.

Corollary 3.10. *The* IsTradeRobust *and the* IsInvariantTradeRobust *problem can be solved in polynomial time when the input game is given in explicit winning, minimal winning, losing or maximal losing form.*

3.3. Recognizing homogeneous, decisive and majority games

In this section we define the homogeneous, decisive and majority games and, afterwards, we analyze the complexity of the IsHomogeneous, IsDecisive and IsMajority problems.

Definition 3.11. A weighted game (N, W) is *homogeneous* if there exists a realization $(q; w)$ such that $q = w(S)$ for all $S \in W^m$.

That is, a weighted game is homogeneous *iff* the sum of the weights of any minimal winning coalition is equal to the quota.

Theorem 3.12. *The* IsHomogeneous *problem can be solved in polynomial time when the input game is given in explicit winning or losing form.*

Proof. The polynomial time reduction from the IsHomogeneous problem to the *Linear Programming* problem, is done in the same way as in the proof of Theorem 3.9, but considering the instance Π' obtained by replacing $w(S) \geq q$, in the first set of inequalities of Π, by $w(S) = q$. It is immediate to see that (N, W) is homogeneous *iff* Π' has a solution. This modification provides the proof of Theorem 3.12. □

Now we introduce the remaining subclasses of simple games.

Definition 3.13. A simple game is *decisive* (or *self–dual*, or *constant sum*) if it is proper and strong. A simple game is *indecisive* if it is not decisive.

Note that the decisiveness is related to the dualityness. As stated earlier in this paper, (N, W) is proper *iff* (N, W^*) is strong, and (N, W) is strong *iff* (N, W^*) is proper. As a consequence, we have that a simple game (N, W) is decisive *iff* $W = W^*$. On the other hand, W is closed under $\subseteq$ or $\supseteq$ *iff* W^* is closed under $\subseteq$ or $\supseteq$, respectively.

In the seminal work on game theory by Von Neumann and Morgenstern [23] only decisive simple games were considered. Nowadays, many governmental institutions make their decisions through voting rules that are in fact decisive games. If abstention is not allowed (see [8] for voting games with abstention) ties are not possible in decisive games.

Another interesting subfamily of simple games are the so-called majority games:

Definition 3.14. A simple game is a *majority game* if it is weighted and decisive.

Observe that, although a simple game can fail to be proper and fail to be strong, this cannot happen with weighted games (the proof appears in [22]).

Proposition 3.15. *Any weighted game is either proper or strong.*

From Proposition 3.15, it follows that there are three kinds of weighted games: proper but not strong, strong but not proper, and both strong and proper.

Finally, we use Theorem 3.4, Corollary 3.5 and Theorem 3.9 and obtain the following result:

Theorem 3.16. *The* IsMajority *and the* IsDecisive *problems can be solved in polynomial time when the input game is given in explicit winning or losing form.*

4. Problems on weighted games

In this section we consider weighted games which are described by an integer realization $(q; w)$. Observe that it is well-known that any weighted game admits an integer realization (see for instance [1]), that is, a weight function with nonnegative integer values, and a positive integer as quota. Integer realizations naturally arise; just consider the seats distributed among political parties in any voting system. In consequence we assume an integer realization as representation of a weighted game. We analyze the complexity of problems of the type:

Name: IsP
Input: An integer realization $(q; w)$ of a weighted game Γ.
Question: Does Γ satisfy P?

We are interested in problems associated with the properties of being strong, proper, homogeneous, and majority[8]. Observe that for weighted games majority and decisive are just the same property, so we consider only the majority games.

[8]Note that the definition of majority weighted games given in [2] is equivalent to our definition of weighted games.

From now on some of the proofs are based on reductions from the NP-complete problem PARTITION [9], which is defined as:

Name: PARTITION
Input: n integer values, $x_1, \ldots, x_n$.
Question: Is there $S \subseteq \{1, \ldots, n\}$ for which $\sum_{i \in S} x_i = \sum_{i \notin S} x_i$.

Observe that, for any instance of the PARTITION problem in which the sum of the n input numbers is odd, the answer must be NO.

Theorem 4.1. *The* ISSTRONG*,* ISPROPER *and* ISMAJORITY *(here, equivalent to* ISDECISIVE*) problems, when the input is described by an integer realization of a weighted game* $(q; w)$*, are* co-NP*-complete.*

Proof. From the definitions of strong, proper and majority games, it is straightforward to show that the three problems belong to co-NP.

Observe that the weighted game with integer representation $(2; 1, 1, 1)$ is proper and strong, and thus decisive.

We transform an instance $x = (x_1, \ldots, x_n)$ of the PARTITION problem into a realization of a weighted game according to the following schema

$$f(x) = \begin{cases} (q(x); x) & \text{when } x_1 + \cdots + x_n \text{ is even,} \\ (2; 1, 1, 1) & \text{otherwise.} \end{cases}$$

The function f can be computed in polynomial time provided q can, and we will use a different q for each problem.

Nevertheless, independently of q, when $x_1 + \cdots + x_n$ is *odd*, x is a NO input for partition, but $f(x)$ is a YES instance of ISSTRONG, ISPROPER, and ISMAJORITY, and thus a NO instance of the complementary problems.

Therefore, we have to take care only of the case in which $x_1 + \cdots + x_n$ is *even*. Assume that this is the case and let $s = (x_1 + \cdots + x_n)/2$ and $N = \{1, \ldots, n\}$. We now provide the proof that f reduces PARTITION to the respective complementary problem.

(a) ISSTRONG *problem*

For the case of strong games, taking $q(x) = s + 1$, we have:

- if there is a $S \subset N$ such that $\sum_{i \in S} x_i = s$, then $\sum_{i \notin S} x_i = s$, thus both S and $N \backslash S$ are losing coalitions and $f(x)$ is weak;
- now assume that S and $N \backslash S$ are both losing coalitions in $f(x)$. If $\sum_{i \in S} x_i < s$ then $\sum_{i \notin S} x_i \geq s + 1$, which contradicts that $N \backslash S$ is losing. Thus we have that $\sum_{i \in S} x_i = \sum_{i \notin S} x_i = s$, and there exists a partition of x.

Therefore, f is a polytime reduction from PARTITION to ISWEAK.

(b) IsProper *problem*

For the case of proper games we take $q(x) = s$. Then, if there is a $S \subset N$ such that $\sum_{i \in S} x_i = s$, then $\sum_{i \notin S} x_i = s$, thus both S and $N \backslash S$ are winning coalitions and $f(x)$ is improper. When $f(x)$ is improper

$$\exists S \subseteq N : \sum_{i \in S} x_i \geq s \wedge \sum_{i \notin S} x_i \geq s,$$

and thus $\sum_{i \in S} x_i = s$. Thus, we have a polytime reduction from Partition to IsImproper.

(c) IsMajority *problem*

For the case of majority games we take again $q(x) = s$. Observe that $f(x)$ cannot be weak, as in such a case there must be some $S \subseteq N$ for which,

$$\sum_{i \in S} x_i < s \wedge \sum_{i \notin S} x_i < s,$$

contradicting the fact that $s = (x_1 + \cdots + x_n)/2$. Therefore, the game is not majority *iff* it is improper, and the claim follows. □

Before finishing this section we introduce the following related problem:

> Name: IsHomogeneousRealization
> Input: An integer realization $(q; w)$ of a weighted game Γ.
> Question: Is $(q; w)$ a homogeneous realization?

Given the weights w, Rosenmüller [20] solves the problem of computing all q such that $(q; w)$ is a homogeneous realization. Although in [20] the analysis on the complexity is omitted, it is easy to check that the dynamic programming algorithm given in Section 3 of [20] runs in polynomial time. Thus, given an integer realization $(q; w)$ it can be checked whether it is a homogeneous realization in polynomial time.

Theorem 4.2. *The* IsHomogeneousRealization *problem can be solved in polynomial time.*

Note that, given an integer realization $(q; w)$ of a weighted game, we cannot yet check whether this game is homogeneous, only whether a given realization is a homogeneous one. We want to remark that the previous result does not imply that the IsHomogeneous problem belongs to NP. Consider the problem

> Name: IsAnotherRealization
> Input: Two integer realizations $(q; w)$ and $(q'; w')$.
> Question: Is $(q'; w')$ another realization of the game (q, w)?

In [6] it is shown that the IsAnotherRealization problem is co-NP-complete: it is easy to see that $(x_1, \ldots, x_n)$ is a no instance of Partition if and only if $(s + 1; x)$ is another realization of $(s; x)$.

5. Succinct representations

We finish the analysis of simple games introducing a natural succinct representation of families of sets by means of Boolean formulas. A Boolean formula Φ on n variables provides a compact description of a family of subsets C of a set N with n elements in the following way: we associate to each truth assignment $x = (x_1, \ldots, x_n)$ the set $A_x = \{i \mid x_i = 1\}$. Therefore Φ describes the family of subsets $\{A_x \mid \Phi(x) = 1\}$ in a compact way. In consequence we consider the following succinct representations

- Succinct winning form: the game is given by (N, Φ) where Φ is a Boolean formula on $|N|$ variables providing a compact description of the sets in W.
- Succinct minimal winning form: the game is given by (N, Φ) but now Φ describes the family W^m. Observe again that this form might require less space than the previous one whenever $W \neq \{N\}$.

In addition we consider the succinct losing and maximal losing forms. Our first objective again is to analyze the complexity of the recognition problem.

Name: IsSimpleS
Input: (N, Φ).
Question: Is (N, Φ) a correct succinct representation of a simple game?

As it happened with IsSimpleE problem, we have in total four different problems depending on the input description: winning, minimal winning, losing and maximal losing.

Unfortunately, we can show that the recognition problem is hard in all the proposed succinct forms thus forbidding a posterior use of such representations.

Theorem 5.1. *The* IsSimpleS *problem belongs to* co-NP*-complete for any succinct form* F *: winning, minimal winning, losing, or maximal losing.*

Proof. Observe that, from the Definition 2.1 of the *monotonicity property*, a set $W(L)$ is not monotonic *iff* there are two sets S_1 and S_2 such that $S_1 \subseteq S_2$ but $S_1 \in W$ and $S_2 \notin W$ ($S_1 \notin L$ and $S_2 \in L$). When the game is given in succinct winning or losing form, these tests can be done by guessing two truth assignments x_1 and x_2 and checking that $x_1 < x_2$, $\Phi_W(x_1) = 1$ and $\Phi_W(x_2) = 0$ ($\Phi_L(x_1) = 0$ and $\Phi_L(x_2) = 1$). Both properties can be checked in polynomial time once S_1 and S_2 are given. Thus the problems belong to co-NP.

In the case that Φ represents $W^m(L^M)$ we have to check that the represented set is minimal (maximal). Observe that Φ does not represent a minimal (maximal) set if there are $\alpha, \beta \in \{0,1\}^n$ with $\alpha < \beta$ such that $\Phi(\beta) = 1$ and $\Phi(\alpha) = 1$.

Therefore, all the problems of recognizing succinct forms belong to co-NP.

A Boolean formula is *monotonic* if for any pair of truth assignments x, y, such that $x \leq y$ in canonical order (*i.e.*, $x_i \leq y_i$ for all i), we have that $\Phi(x) \leq \Phi(y)$ (assuming that false $<$ true). The latter problem (*i.e.*, to know whether a Boolean formula is monotonic or not) is co-NP-complete (even for DNF formulas) [15]. Consider the following reduction: given a Boolean formula Φ on n variables we construct Φ' on $n+2$ variables as follows

$$\Phi'(\alpha\beta x) = \begin{cases} 1 & \alpha = \beta = 1 \\ 0 & \alpha = \beta = 0 \\ \Phi(x) & \alpha \neq \beta. \end{cases}$$

Now we have that Φ is monotonic iff Φ' is monotonic. Furthermore we have that Φ' is monotonic iff (N, Φ') is a simple game in the explicit winning form since $\Phi'(1^n) = 1$ and $\Phi'(0^n) = 0$. This shows that IsSimpleS for the explicit winning form is co-NP-complete. Observe that (N, Φ_L) is an explicit loosing representation of a simple game iff $(N, \neg\Phi_L)$ is an explicit winning representation of a simple game. Then the IsSimpleS for the explicit loosing form is co-NP-complete.

Recall now that the SAT problem asks whether a given Boolean formula has a satisfying assignment. SAT is a well known NP-complete problem. Consider the following reduction: given a Boolean formula ϕ on n variables we construct Φ for minimal winning forms on $n+2$ variables as follows

$$\Phi(\alpha\beta x) = \begin{cases} 1, & \text{if } \alpha = \beta = 1 \text{ and } x = 1^n \\ 0, & \text{if } \alpha = \beta = 1 \text{ and } x \neq 1^n \\ \phi(x), & \text{if } \alpha \neq \beta \\ 0, & \text{if } \alpha = \beta = 0. \end{cases}$$

We have that ϕ does not have satisfying assignment *iff* Φ describes a non empty minimal winning set. Similarly for maximal losing forms, now we should consider

$$\Phi(\alpha\beta x) = \begin{cases} 0, & \text{if } \alpha = \beta = 1 \\ \phi(x), & \text{if } \alpha \neq \beta \\ 0, & \text{if } \alpha = \beta = 0 \text{ and } x \neq 0^n \\ 1, & \text{if } \alpha = \beta = 0 \text{ and } x = 0^n. \end{cases}$$

Thus the minimal winning and the maximal losing problems are co-NP-hard. □

6. Conclusions and open problems

We have analyzed different representations for simple games: explicit and succinct representations. All explicit forms that we have considered are represented by a pair (N, C) in which $N = \{1, \ldots, n\}$ for some positive integer n, and C is the set of winning, minimal winning, losing or maximal losing coalitions.

For the four proposed explicit representations of a simple game, we have studied the complexity of deciding whether the given game is strong, proper, weighted, homogeneous, decisive or majority. In the same vein, given a weighted game described by an integer representation $(q; w)$, we have also considered the complexity of deciding whether the game is strong, proper, homogeneous or majority.

As this is the first time in which problems on simple games are analyzed there are still many interesting open question as there are many other interesting properties on simple games. With respect to the unclassified problems on Table 2 we conjecture the following:

Conjecture 6.1. The IsDecisive problem is co-NP-complete when the input game is given in explicit minimal winning or maximal losing form.

Conjecture 6.2. The IsMajority problem is co-NP-complete when the input game is given in explicit minimal winning or maximal losing form.

We would also like to remark that our study can be enlarged by considering new explicit forms to present a simple game. For example, blocking coalitions and minimal blocking coalitions provide an alternative way to fully describe a simple game. Precisely, a blocking coalition wins whenever its complementary loses. From the point of view of succinct representations, there are other proposals for representing a simple game, which make use of Boolean functions or weighted representations. For example the multilinear extension of a simple game [16], *succinct representations* [15], or the intersection of a collection of weighted games [2]. It will be of interest to perform a similar complexity analysis on such representations.

Interestingly enough, we have shown in Theorem 3.9 that we can decide in polynomial time whether a simple game is weighted. This result opens the possibility of analyzing the complexity of problems on weighted games described in an explicit form. In particular, as weighted games are games with dimension 1, our results imply that we can decide in polynomial time whether a simple game has dimension 1. Recall that the results in [2] show that computing the dimension of a simple game is NP-hard. The latter result is obtained when the game is described as the intersection of some weighted games. It will be of interest to determine whether the dimension of a simple game given in explicit form can be computed in polynomial time. The same questions can also be formulated for other parameters and solution concepts on simple games.

Acknowledgements. Josep Freixas was partially supported by Grant MTM 2009-08037 from the Spanish Science and Innovation Ministry, SGR 2009-1029 of "Generalitat de Catalunya", and 9-INCREC-11 of "(PRE/756/2008) Ajuts a la iniciació/reincorporació a la recerca (Universitat Politècnica de Catalunya)".

Xavier Molinero was partially supported by Grant MTM 2009-08037 from the Spanish Science and Innovation Ministry, ALBCOM-SGR 2009-1137 of "Generalitat de Catalunya", and 9-INCREC-11 of "(PRE/756/2008) Ajuts a la iniciació/reincorporació a la recerca (Universitat Politècnica de Catalunya)".

Maria Serna was partially supported by the Spanish project TIN-2007-66523 (FORMALISM) and ALBCOM-SGR 2009-1137 of "Generalitat de Catalunya".

We would also like to acknowledge Gerth S. Brodal from University of Aarhus for valuable comments and constructive criticism.

The authors thank the Reviewers for carefully reading an earlier version of this paper and for pointing out interesting and useful comments.

References

[1] F. Carreras and J. Freixas, Complete simple games. *Math. Soc. Sci.* **32** (1996) 139–155.

[2] V.G. Deĭneko and G.J. Woeginger, On the dimension of simple monotonic games. *Eur. J. Oper. Res.* **170** (2006) 315–318.

[3] X. Deng and C.H. Papadimitriou, On the complexity of cooperative solution concepts. *Math. Oper. Res.* **19** (1994) 257–266.

[4] E. Elkind and D. Pasechnik, Computing the nucleolus of weighted voting games, in *SODA '09: Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics. Philadelphia, PA, USA (2009) 327–335.

[5] E. Elkind, L.A. Goldberg, P.W. Goldberg and M. Wooldridge, Computational complexity of weighted threshold games, in *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*. Vancouver, British Columbia, Canada (2007) 718–723.

[6] E. Elkind, L.A. Goldberg, P.W. Goldberg and M. Wooldridge, On the dimensionality of voting games, in *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence.* Hyatt Regency McCormick Place, Chicago (2008) 69–74.

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

[8] J. Freixas and W.S. Zwicker, Weighted voting, abstention, and multiple levels of approval. *Soc. Choice Welfare* **21** (2003) 399–431.

[9] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completness*, edited by W.H. Freeman. San Francisco, New York, USA (1979).

[10] G.W. Harrison and T. McDaniel, Voting games and computational complexity. *Oxford Econ. Papers* **60** 2008 546–565.

[11] T. Hegedüs and N. Megiddo, On the geometric separability of Boolean functions. *Discrete Appl. Math.* **66** (1996) 205–218.

[12] N. Karmarkar, A new polynomial-time algorithm for linear programming. *Combinatorica* **4** (1984) 373–395.

[13] L.G. Khachiyan, A polynomial algorithm for linear programming. *Dokl. Akad. Nauk. SSSR* **244** (1979) 1093–1096; English translation *Soviet Math. Dokl.* **20** (1979) 191–194.

[14] Y. Matsui and T. Matsui, NP-completeness for calculating power indices of weighted majority games. *Theor. Comput. Sci.* **263** (2001) 305–310.

[15] D. Mehta and V. Raghavan, Decision tree approximations of Boolean functions. *Theor. Comput. Sci.* **270** (2002) 609–623.

[16] G. Owen, *Game Theory*, 3th edition. Academic Press, San Diego, USA (1995).

[17] C.H. Papadimitriou, *Computational Complexity.* Addison Wesley (1994).

[18] U.N. Peled and B. Simeone, Polynomial-time algorithms for regular set-covering and threshold synthesis. *Discrete Appl. Math.* **12** (1985) 57–69.

[19] K. Prasad and J.S. Kelly, NP-completeness of some problems concerning voting games. *Int. J. Game Theory* **19** (1990) 1–9.

[20] J. Rosenmüller, An algorithm for the construction of homogeneous games, in *Ökonomie und Mathematik*, edited by O. Opitz and B. Rauhut. Springer-Verlag (1987) 63–74.

[21] A.D. Taylor and W.S. Zwicker, Simple games and magic squares. *J. Comb. Theory, Ser. A* **71** (1995) 67–68.

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

[23] J. Von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, New Jersey, USA (1944).