Minimal representations for majority games*

Josep Freixas[1], Xavier Molinero[2], and Salvador Roura[3]

[1,2] Escola Politècnica Superior d'Enginyeria de Manresa. E-08242 Manresa, Spain.
[3] Universitat Politècnica de Catalunya. CN-Ω, E-08034 Barcelona, Catalonia, Spain.

[1] Dept. de Matemàtica Aplicada 3. E-mail: josep.freixas@upc.edu.
[2,3] Dept. de Llenguatges i Sistemes Informàtics.
E-mail: {molinero,roura}@lsi.upc.edu.

Abstract This paper presents some new results about majority games. Isbell (1959) was the first to find a majority game without a minimum normalized representation; he needed 12 voters to construct such a game. Since then, it has been an open problem to find the minimum number of voters of a majority game without a minimum normalized representation. Our main new results are:

1. All majority games with less than 9 voters have a minimum representation.
2. For 9 voters there are 14 majority games without a minimum integer representation, but these games admit a minimal normalized integer representation.
3. For 10 voters exist majority games with neither a minimum integer representation nor a minimal normalized integer representation.

Keywords Simple games, Weighted games, Majority games, Minimum and minimal weighted representations/realizations, Computing games.

Math. Subj. Class. (2000) 03B45, 68U01, 68W01, 91A12, 94C10.

1 Introduction

We start by giving some basic definitions on simple games. We refer the interested reader to [16] for a thoroughly presentation of simple games. 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 1. *A simple game G is a pair (N, W) in which $N = \{1, \ldots, n\}$ and W is a collection of subsets of N that satisfies: $N \in W$, $\emptyset \notin W$, and* monotonicity *$[S \in W$ and $S \subseteq R \subseteq N \Rightarrow R \in W]$.*

* This research was partially supported by Grant MTM 2006–06064 of "Ministerio de Ciencia y Tecnología y el Fondo Europeo de Desarrollo Regional" and SGRC 2005-00651 of "Generalitat de Catalunya", and by the Spanish "Ministerio de Ciencia y Tecnología" programme TIN2005-05446 (ALINEX).

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 voted for it. A subset of N that is not in W is called a *losing coalition*, denoted by L. A *minimal winning coalition* (*maximal losing coalition*), denoted by W^m (L^M), 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 $S \in W^m$.

Before proceeding, we introduce a real–world example (see [15] for an extensive illustration of many real–world examples and [4] for the European Union Council).

Example 1. We shall consider here the composition of the Catalonia Parliament today. Six parties got elected members in the last elections, giving rise to the distribution of the 135 seats:

Party	Name of the Party	Seats
1	Convergència i Unió	48
2	PSC-Ciutadans pel Canvi	37
3	Esquerra Republicana de Catalunya	21
4	Partit Popular	14
5	Iniciativa per Catalunya Verds - EUIA	12
6	Ciutadans-Partido de la Ciudadanía	3

Voters are here parties since they vote by using party whip. Most of the proposals require absolute majority to be passed, i.e., a minimum of 68 votes are needed. Then $W^m = \{\{1,2\},\{1,3\},\{1,4,5\},\{2,3,4\},\{2,3,5\}\}$. Some others proposals need a 2/3-qualified majority to be passed, that is, at least 90 votes. In this case we have $W^m = \{\{1,2,3\},\{1,2,4\},\{1,2,5\},\{1,3,4,5\}\}$. Notice that party 6 is null in both games.

Now, let us consider some special types of simple games.

Definition 2. *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*. Intuitively speaking, if a game is weak it has too few winning coalitions, because adding sufficiently many winning coalitions would make the game strong. Notice that the addition of winning coalitions can never destroy strongness.

Definition 3. *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. Notice 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 $|W| = |L| = 2^{n-1}$.

Definition 4. *A simple game is* decisive *(or self–dual, or constant sum) if it is proper and strong.*

Definition 5. *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. Notice that (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 $W = W^*$.

In the seminal work on game theory by von Neumann and Morgenstern [14] only decisive simple games were considered. Nowadays, many governmental institutions make their decisions by means of voting rules that are in fact decisive games. If abstention is not allowed (see [6] for voting games with abstention) ties are not possible in decisive games.

This paper is organized as follows. Section 2 is devoted to weighted games, its main properties and to formally introduce several kinds of minimal integer realizations (*representations* in a more general context) for them. Notice that all proofs have been omitted because of either they are quite simple or the lack of the space. It specially focuses in majority games and several open problems related to minimal realizations. Section 3 explains the methodology and sketchy the main algorithms used to obtain the results presented below. Section 4 provides the most representative cases of minimal realizations. A conclusion section ends the paper.

2 Weighted games and majority games

Weighted simple games (or *weighted games*, for short) are probably the most important kind of simple games.

Definition 6. *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 precisely when the sum of weights of* S *meets or exceeds* q*.*

Any specific example of such a weight function w and quota q are 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 $\sum_{i \in S} w_i$. Three parameters can be defined for any realization $(q; w)$ of a weighted game (N, W):

$$T = w(N), \qquad a = \min_{S \in W} w(S), \qquad b = \max_{S \in L} w(S).$$

From the definition of simple game, we have $0 < q \leq T$.

Although a simple game can fail to be proper and fail to be strong, this cannot happen with weighted games. Notice that Example 1 with absolute majority is a proper and strong weighted game; for the 2/3-qualified majority, it is a proper and weak weighted game.

Proposition 1. *Any weighted game is either proper or strong.*

It is well–known that any weighted game admits an integer realization (see for instance [3]), 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.

Henceforth we will only consider the set $\mathcal{I}$[1] of all integer realizations for every weighted game (N, W). Notice that, if $(q; w) \in \mathcal{I}$, then $(c \cdot q; c \cdot w) \in \mathcal{I}$ for every positive c, so $\mathcal{I}$ is an unbounded *cone of integer values*.

The desirability order, which goes back at least to Isbell [7] and was later generalized in [9] (see also [10, 12]), is useful to define a natural order.

Definition 7. *Let* (N, W) *be a simple game.*

(*i*) *Player* i *is* more desirable *than* j *(*$i \succeq j$*, for short) in* (N, W) *if*

$$S \cup \{j\} \in W \;\Rightarrow\; S \cup \{i\} \in W, \qquad \textit{for all } S \subseteq N \setminus \{i, j\}.$$

(*ii*) *Players* i *and* j *are* equally desirable *(*$i \sim j$*, for short) in* (N, W) *if*

$$S \cup \{i\} \in W \Leftrightarrow S \cup \{j\} \in W, \qquad \textit{for all } S \subseteq N \setminus \{i, j\}.$$

(*iii*) *Player* i *is* strictly more desirable *than player* j *(*$i \succ j$*, for short) in* (N, W) *if* i *is more desirable than* j*, but* i *and* j *are not equally desirable.*

If a weighted game admits $(q; w_1, \ldots, w_n)$ as a realization, then $w_i \geq w_j$ implies $i \succeq j$. Therefore, the desirability relation of weighted games is *complete*. (See [1] for a classification theorem of complete simple games.) Notice that $w_i = w_j$ implies $i \sim j$. However, $i \sim j$ does not necessarily imply $w_i = w_j$. But $i \succ j$ implies $w_i > w_j$. All these comments suggest the following definition.

Definition 8. *A realization* $(q; w)$ *of a weighted game is said to* preserve types *or to be* normalized *if* $w_i = w_j$ *whenever* $i \sim j$*. By* $\mathcal{N}$ *we denote the set of all normalized realizations of a game.*

From now on, assume w.l.o.g. that $i \succeq i + 1$ for every $i = 1, \ldots, n - 1$. The set of voters admits a partition in t classes $N_1, \ldots, N_t$ such that two voters i and j belong to the same class iff $i \sim j$. The extreme cases arise when $t = 1$ (all voters are equally desirable) and $t = n$ (each class reduces to a singleton). We sort the classes from the most desirable (N_1) to the least desirable (N_t).

In the following definitions, let $w \leq w'$ mean $w_i \leq w'_i$ for all $1 \leq i \leq n$.

Definition 9. *A realization* $(q; w)$ *of a weighted game is called* minimum *if* $w \leq w'$ *for all realization* $(q'; w') \in \mathcal{I}$*. By* $\mathcal{MI}$ *we denote the set of all minimum realizations of a game.*

---

1 Since there is no confusion about the used game we will write $\mathcal{I}$ instead of $\mathcal{I}(N, W)$, also for other sets defined later.

For instance, one may easily check that the *minimum* realization $(6; 4, 3, 2, 1, 1, 0)$ with just 11 seats is equivalent to the one given in Example 1 with absolute majority.

Definition 10. *A realization $(q; w)$ of a weighted game has* minimum sum *if $w(N) \leq w'(N)$ for all $(q'; w') \in \mathcal{I}$. By $s\mathcal{MI}$ we denote the set of all minimum sum realizations of a game.*

Definition 11. *A realization $(q; w)$ of a weighted game is a* minimum normalized realization *if $w \leq w'$ for all $(q'; w') \in \mathcal{N}$. By $\mathcal{MN}$ we denote the set of all minimum normalized realizations of a game.*

Definition 12. *A realization $(q; w)$ of a weighted game is a* minimum sum normalized realization *if $w(N) \leq w'(N)$ for all $(q'; w') \in \mathcal{N}$. By $s\mathcal{MN}$ we denote the set of all minimum sum normalized realizations of a game.*

Notice that the quota q for minimum sum realizations must be a. This is why there is no restriction for the quota in above definitions.

The next proposition summarizes some properties of these types of realizations.

Proposition 2. *For every weighted game,*

(i) *$\mathcal{MI}$ has at most one element.*
(ii) *$s\mathcal{MI}$ is never empty.*
(iii) *$\mathcal{MI} \subseteq s\mathcal{MI}$. Moreover, $\mathcal{MI} = s\mathcal{MI}$ iff $\mathcal{MI}$ reduces to a singleton.*
(iv) *$\mathcal{MN}$ has at most one element.*
(v) *$s\mathcal{MN}$ is never empty.*
(vi) *$\mathcal{MN} \subseteq s\mathcal{MN}$. Moreover, $\mathcal{MN} = s\mathcal{MN}$ iff $\mathcal{MN}$ reduces to a singleton.*
(vii) *$\mathcal{MI} \subseteq \mathcal{MN}$.*

Regarding Proposition 2 it will be interesting to find the required *minimum* number of voters for a game in order to obtain: weighted games without a minimum integer realization, i.e., with $\mathcal{MI} = \emptyset$ and, distinguishing among, $|s\mathcal{MI}| = 1$ or $|s\mathcal{MI}| = 2$ or $|s\mathcal{MI}| > 2$. In the same way, it would also be meaningful determining the needed minimum number of voters for a game in order to get weighted games without a minimum integer normalized realization, i.e. $\mathcal{MN} = \emptyset$.

In this topic, one of the goals during the last five decades has been to find examples for each of these cases. The construction of all the weighted games for $n < 7$ goes as early as 1962 [11]. In 1970 [13] it was found that $\mathcal{MI} \neq \emptyset$ for each weighted game with $n < 8$. Quite recently [5] it has been proved that, for $n = 8$, there are 154 non–isomorphic weighted games such that $\mathcal{MI} \neq \emptyset$, although none of them decisive. However, all these games have a unique normalized realization. Isbell [8] exhibited a remarkable example of a decisive weighted game with 12 voters in which the two affected voters with different weight are not equi–desirable: $(99; 38, 31, 31, 28, 23, 12, 11, 8, 6, 5, 3, 1)$, which also admits the equivalent realization $(99; 37, 31, 31, 28, 23, 12, 11, 8, 7, 5, 3, 1)$. Since 1979 [2], Isbell's example has been very useful in game theory. See [5] for an example of a non–decisive weighted game with $n = 10$ and $M\mathcal{N} = \emptyset$.

Now we introduce the class of games which are the aim of study in this paper.

Definition 13. *A simple game is a* majority game *if it is weighted and decisive.*

From Proposition 1 it follows that there are three kind of weighted games: proper but not strong, strong but not proper, and decisive. In what follows we will only consider decisive games.

We are interested in finding minimal realizations for majority games. The following result provides conditions for a realization to be minimal.

Proposition 3. *Consider a majority game. Any element $(q; w)$ in $M\mathcal{I}$ satisfies: $a = q$, $b = q - 1$ and $T = 2q - 1$.*

Therefore, the total weight T of every minimal realization of a majority game is an odd number, and its quota is $q = (T + 1)/2$. Henceforth we will represent such a minimal realization omitting the quota, which becomes redundant.

As we claimed, all majority games with $n < 9$ have a minimum realization. On the other hand, Isbell's game (which has $n = 12$) does not have a minimum normalized realization. The goal of the next sections is to find out, with the help of algorithms, majority games without a unique minimal realization.

3 Algorithm to classify majority games

We implemented several algorithms about majority games, to exhaustively study all the games with $n < 10$, and also to find examples with 10 voters depending on its minimal realization. An exhaustive study of games with $n = 10$ was beyond our CPU time possibilities, because the huge number of games (see Table 1 below). All the experiments were done with a processor AMD64X2 4400 (two cores at 2.2 GHz) with 4 Gb of DDR memory with ECC.

Because of the lack of space, we briefly sketch one of the algorithms (see Figure 1 in Appendix). The C++ code can be obtained by asking the authors.

The program uses an integer array $m[n]$, filled with empirical data, with the maximum possible weight for each player when there are n of them. The values for $n = 1, \ldots, 8$ are $1, 1, 2, 3, 5, 9, 18, 42$, respectively.

The main loop considers in increasing order the number of players n. For every such n, and for every sum of weights s between n and $n \cdot m[n]$ also in increasing order, a backtracking searches for all the combinations of n weights with total sum s.

The backtracking searches the combinations in lexicographical order, choosing the weights in increasing order from left to right, and cutting every useless branch. A branch is declared useless when it is known that the weight of every combination reachable from it will be strictly smaller or larger than s.

When we found a combination (we know that its sum of weights is s), we call a routine to check its properties. There, we first compute the minimal winning coalition W^m (also called *representative function*), that is, all the minimum subsets whose sum of weights reaches at least $\lfloor s/2 \rfloor + 1$. Each of these subsets is codified as a bitmap in one integer. Them all, which together form the representative function of the game, are stored in a specific order in a `vector<int>`.

We then check with `M.find(RF)` if the current vector is a new one or if it was already found before. We achieve this by storing every new vector in a `map` (similar to a `set`) data structure. Then,

- For every game with a new representative function, we check if the game verifies $|s\mathcal{MI}| = 2$. Notice that, here, we do not have enough information to check whether $|s\mathcal{MN}| = 1$ or $|s\mathcal{MN}| > 1$.
- For every game with an already found representative function, we check if the game verifies: $|s\mathcal{MI}| = 1$, $|s\mathcal{MI}| > 1$, $|s\mathcal{MN}| = 1$, $|s\mathcal{MN}| > 1$, etcetera.

4 Realizations of majority games

All possible majority games with less than 10 voters were considered. Table 1 shows the results obtained.

n	1	2	3	4	5	6	7	8	9
CG	1	3	8	25	117	1171	44313	16175188	–
WG	1	3	8	25	117	1111	29373	2730164	–
DG	1	1	2	3	7	21	135	2470	319124
MG	1	1	2	3	7	21	135	2470	175428

Table 1. Number of Complete Simple Games (CG), Weighted Complete Simple Games (WG), Decisive Games (DG) and Majority Games (MG) with n voters.

Some observations are worth noting for $n = 9$. In particular, there are just 14 majority games (see Table 2) such that $\mathcal{MI} = \emptyset$ but $|s\mathcal{MI}| > 1$. Muroga *et. al* [13] already found these realizations. Notice that $|s\mathcal{MI}| = 2$ for all them. Of course, by adding null voters to these games we obtain new games with these properties.

In Table 3 we give examples of 10 voters with $|s\mathcal{MI}| > 2$, so that 10 voters is sharp to get games with $\mathcal{MI} = \emptyset$ and $|s\mathcal{MI}| > 2$. We have only listed here a small sample because their properties are similar. Thus, things become more compelling when there are more than 9 voters. Unfortunately, it was not possible for us to study all (majority) games because of the huge number of games and the limitation of our computers, but we could study a sufficiently large subset of such games to find conspicuous examples.

All the examples given in Tables 2 and 3 satisfy $\mathcal{MN} \neq \emptyset$. Table 4 lists some majority games with 10 voters without minimum integer normalized realization.

#	T	q	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9
1	47	24	11	9	6	6	4	4	4	②	①
2	49	25	13	7	6	6	4	4	4	③	②
3	53	27	14	9	⑦	⑥	5	5	3	2	2
4	55	28	13	9	7	7	6	4	4	③	②
5	55	28	13	11	7	7	5	5	4	②	①
6	55	28	13	11	8	6	6	4	4	②	①
7	59	30	17	9	8	⑦	⑥	5	3	2	2
8	63	32	15	13	9	7	7	5	4	②	①
9	63	32	15	13	10	8	6	4	4	②	①
10	65	33	13	11	10	8	6	6	⑤	④	2
11	65	33	17	12	8	8	⑦	⑥	3	2	2
12	67	34	16	14	11	9	6	4	4	②	①
13	71	36	17	15	11	9	7	5	4	②	①
14	75	38	18	16	12	10	7	5	4	②	①

Table 2. All majority games with 9 voters such that $\mathcal{MI} = \emptyset$ and $|s\mathcal{MI}| = 2$. The other minimum sum integer realization are obtained by interchanging encircled weights.

T, T'	q, q'	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w'_1	w'_2	w'_3	w'_4	w'_5	w'_6	w'_7	w'_8	w'_9	w'_{10}
95	48	22	18	12	12	8	8	8	③	③	1	22	18	12	12	8	8	8	④	②	1
97	49	25	13	12	10	8	8	8	⑥	⑤	2	25	13	12	10	8	8	8	⑦	④	2
99	50	26	14	12	12	8	8	8	⑤	⑤	1	26	14	12	12	8	8	8	⑥	④	1
103	52	24	20	13	13	9	9	8	③	③	1	24	20	13	13	9	9	8	④	②	1
107	54	25	17	13	13	10	8	8	⑥	⑤	2	25	17	13	13	10	8	8	⑦	④	2
111	56	25	17	15	13	12	8	8	⑥	⑤	2	25	17	15	13	12	8	8	⑦	④	2
111	56	26	18	14	14	12	8	8	⑤	⑤	1	26	18	14	14	12	8	8	⑥	④	1
111	56	26	22	14	14	10	10	8	③	③	1	26	22	14	14	10	10	8	④	②	1
111	56	26	22	16	12	12	8	8	③	③	1	26	22	16	12	12	8	8	④	②	1
113	57	24	21	16	16	13	7	6	6	②	②	24	21	16	16	13	7	6	6	③	①
119	60	28	24	17	13	13	9	8	③	③	1	28	24	17	13	13	9	8	④	②	1
125	63	27	24	19	16	16	7	6	6	②	②	27	24	19	16	16	7	6	6	③	①
...						...										...					

Table 3. Some majority games with 10 voters such that $\mathcal{MI} = \emptyset$ and $|s\mathcal{MI}| > 2$. The other possible minimum sum integer realizations are obtained by interchanging encircled weights.

Therefore, 10 is *the least* number of voters required to get such majority games. This contribution concludes the open problem left by Isbell in 1959 [8] when he provided his famous example for 12 voters. All examples in Table 4 share the properties $\mathcal{MN} = \emptyset$ and $|s\mathcal{MN}| = 2$ with Isbell's example.

T, T'	q, q'	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w'_1	w'_2	w'_3	w'_4	w'_5	w'_6	w'_7	w'_8	w'_9	w'_{10}
73	37	19	14	(11)	(7)	5	5	5	3	2	2	19	14	(12)	(6)	5	5	5	3	2	2
75	38	(15)	13	13	7	6	6	4	4	4	(3)	(16)	13	13	7	6	6	4	4	4	(2)
77	39	(16)	15	11	9	6	6	4	4	4	(2)	(17)	15	11	9	6	6	4	4	4	(1)
77	39	20	13	(10)	(9)	7	7	5	2	2	2	20	13	(11)	(8)	7	7	5	2	2	2
79	40	19	13	10	10	8	(5)	4	4	4	(2)	19	13	10	10	8	(6)	4	4	4	(1)
79	40	23	(12)	8	8	7	7	(5)	3	3	3	23	(13)	8	8	7	7	(4)	3	3	3
81	41	(15)	13	13	9	7	7	6	4	4	(3)	(16)	13	13	9	7	7	6	4	4	(2)
81	41	17	(14)	13	11	6	6	4	4	4	(2)	17	(15)	13	11	6	6	4	4	4	(1)
81	41	(18)	17	11	9	6	6	4	4	4	(2)	(19)	17	11	9	6	6	4	4	4	(1)
81	41	21	11	10	10	8	(6)	4	4	4	(3)	21	11	10	10	8	(7)	4	4	4	(2)
81	41	21	(12)	11	10	(8)	5	5	3	3	3	21	(13)	11	10	(7)	5	5	3	3	3
81	41	(20)	14	14	9	(7)	5	5	3	2	2	(21)	14	14	9	(6)	5	5	3	2	2
83	42	20	11	11	(8)	7	7	7	5	5	(2)	20	11	11	(9)	7	7	7	5	5	(1)
83	42	24	14	(11)	10	(7)	5	5	3	2	2	24	14	(12)	10	(6)	5	5	3	2	2
85	43	19	(16)	13	11	6	6	4	4	4	(2)	19	(17)	13	11	6	6	4	4	4	(1)
85	43	21	(13)	12	9	9	(7)	5	5	2	2	21	(14)	12	9	9	(6)	5	5	2	2
85	43	22	13	(10)	9	7	7	7	4	4	(2)	22	13	(11)	9	7	7	7	4	4	(1)
85	43	26	(12)	10	8	8	7	(5)	3	3	3	26	(13)	10	8	8	7	(4)	3	3	3
...						...										...					

Table 4. Some majority games with 10 voters *without* minimum integer normalized realization, $\mathcal{MN} = \emptyset$.

5 Conclusion

In this paper we have focused in majority games, with the following conclusions:

1. For less than 9 voters all majority games have a minimum integer realization.
2. For 9 voters there are exactly 14 majority games without a minimum integer realization, but all these games have a minimum normalized integer realization.
3. For more than 9 voters one may find majority games without a minimum realization and with more than two minimum sum integer realizations.
4. For 10 voters there exist majority games without a minimum normalized integer realization.

Notice that Conclusion *4.* allows us to conclude that 10 voters are enough for majority games without a minimum integer normalized realization.

References

1. F. Carreras and J. Freixas. Complete simple games. *Mathematical Social Sciences*, 32:139–155, 1996.
2. P. Dubey and L.S. Shapley. Mathematical properties of the Banzhaf power index. *Mathematics of Operations Research*, 4(2):99–131, 1979.
3. J. Freixas. *Structure of simple games.* PhD thesis, Technical University of Catalonia, Manresa (Barcelona), Spain, Oct 1994. In Spanish.
4. J. Freixas. The dimension for the European Union Council under the Nice rules. *European Journal of Operational Research*, 156(2):415–419, 2004.
5. J. Freixas and X. Molinero. On the existence of a minimum integer representation for weighted voting systems. *Annals of Operations Research*, October 2006. Submitted.
6. J. Freixas and W.S. Zwicker. Weighted voting, abstention, and multiple levels of approval. *Social Choice and Welfare*, 21:399–431, 2003.
7. J.R. Isbell. A class of simple games. *Duke Mathematics Journal*, 25:423–439, 1958.
8. J.R. Isbell. On the enumeration of majority games. *Mathematical Tables and Other Aids to Computation*, 13(65):21–28, 1959.
9. M. Maschler and B. Peleg. A characterization, existence proof, and dimension bounds for the kernel of a game. *Pacific Journal of Mathematics*, 18:289–328, 1966.
10. S. Muroga. *Threshold logic and its applications.* Wiley-Interscience, New York, USA, 1971.
11. S. Muroga, I. Toda, and M. Kondo. Majority decision functions of up to six variables. *Mathematics of Computation*, 16(80):459–472, 1962.
12. S. Muroga, I. Toda, and S. Takasu. Theory of majority decision elements. *J. Franklin Inst.*, 271(5):376–418, May 1961.
13. S. Muroga, T. Tsuboi, and C.R. Baugh. Enumeration of threshold funcitons of eight variables. *IEEE Trans. Computers*, C-19(9):818–825, Sep 1970.
14. J. Von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior.* Princeton University Press, Princeton, New Jersey, USA, 1944.
15. A.D. Taylor. *Mathematics and Politics.* Springer Verlag, New York, USA, 1995.
16. A.D. Taylor and W.S. Zwicker. *Simple games: desirability relations, trading, and pseudoweightings.* Princeton University Press, New Jersey, USA, 1999.

Appendix

Figure 1 sketches the used algorithms to compute all given examples.

```
const int N = 11;  // maximum n

typedef vector<int> VI;

map<VI, VI> M;
vector<int> weight(N);
VI RF;
int m[] = { 0, 1, 1, 2, 3, 5, 9, 18, 42, ... };

...

void check(int sum) {

    int quota = sum/2 + 1;
    RF = compute_representative_function(quota);

    if (M.find(RF) == M.end()) {  // RF is new

        M[RF] = weight;  // we store RF in M
        if (is_sMI_2()) print_game("sMI_2");  // case |sMI| = 2

    } else {

        if (is_sMI()) print_game("sMI");  // case |sMI| = 1
        else if (is_Not_sMI()) print_game("Not_sMI");  // case |sMI| > 1

        if (is_sMN()) print_game("sMN");  // case |sMN| = 1
        else if (is_Not_sMN()) print_game("Not_sMN");  // case |sMN| > 1

        ...
    }
}

// Fill the array weight[0..n-1] from the position i,
// using weights less than or equal to m,
// knowing that sum = weight[0] + ... + weight[i-1],
// and in such a way that the total sum of weights will be s.
void backtracking(int n, int i, int m, int sum, int s) {

    if (sum > s) return;
    if (i == n) check(sum);
    else {
        if (sum == s) return;
        int minimum = (s - sum - 1)/(n - i) + 1;
        int maximum = min(m, s - sum - (n - i - 1));
        for (int x = minimum; x <= maximum; ++x) {
            weight[i] = x;
            backtracking(i + 1, x, sum + x, s);
        }
    }
}

int main() {
    ...

    for (int n = 1; n <= N; ++n)
        for (int s = n; s <= n*m[n]; ++s)
            backtracking(n, 0, m[n], 0, s);

    ...
}
```

Figure 1. Sketch of the used algorithms.