On the Dimensionality of Voting Games

Edith Elkind
Electronics & Computer Science
University of Southampton
Southampton SO17 1BJ, UK

Leslie Ann Goldberg
Computer Science
University of Liverpool
Liverpool L69 3BX, UK

Paul Goldberg
Computer Science
University of Liverpool
Liverpool L69 3BX, UK

Michael Wooldridge
Computer Science
University of Liverpool
Liverpool L69 3BX, UK

Abstract

In a *yes/no voting game*, a set of voters must determine whether to accept or reject a given alternative. *Weighted voting games* are a well-studied subclass of yes/no voting games, in which each voter has a weight, and an alternative is accepted if the total weight of its supporters exceeds a certain threshold. Weighted voting games are naturally extended to k-vector weighted voting games, which are intersections of k different weighted voting games: a coalition wins if it wins in every component game. The dimensionality, k, of a k-vector weighted voting game can be understood as a measure of the complexity of the game. In this paper, we analyse the dimensionality of such games from the point of view of complexity theory. We consider the problems of *equivalence*, (checking whether two given voting games have the same set of winning coalitions), and *minimality*, (checking whether a given k-vector voting game can be simplified by deleting one of the component games, or, more generally, is equivalent to a k'-weighted voting game with $k' < k$). We show that these problems are computationally hard, even if $k = 1$ or all weights are 0 or 1. However, we provide efficient algorithms for cases where both k is small and the weights are polynomially bounded. We also study the notion of *monotonicity* in voting games, and show that monotone yes/no voting games are essentially as hard to represent and work with as general games.

Introduction

Computational aspects of social choice theory have been increasingly studied over the past decade (Endriss & Lang 2006). This growth of interest is in part due to the intriguing possibility that computational complexity may provide a barrier to the strategic manipulation of voting systems, and hence provide a "solution" to negative results such as the Gibbard-Satterthwaite theorem (Bartholdi, Tovey, & Trick 1989). *Yes/no voting games* are one of the most important classes of social choice mechanisms. A yes/no voting game (hereafter, "yes/no game") is one in which a set of voters must determine whether to accept a particular alternative (e.g., a new law or a change to tax regulations) or whether to continue with the status quo (Taylor & Zwicker 1993; 1999). The decision making processes in most governments and political bodies can be understood as yes/no voting systems. Despite their self-evident real-world importance, yes/no games have received comparatively little attention from the multi-agent systems and computational social choice communities. This is perhaps because, strategically, they are rather simple: for example, since there are only two outcomes in such a game ("yea" or "nay"), they do not fall prey to strategic manipulation of the form characterised by Gibbard-Satterthwaite. Nevertheless, they present several interesting challenges for multi-agent systems research, perhaps the most importance of which is the problem of finding representations for such games that strike a useful balance between *succinctness* and *tractability*.

Weighted voting games are one widely-used and well-studied subclass of yes/no games, in which each voter has a weight, and an alternative is accepted if the total weight of its supporters exceeds a certain threshold. The complexity of problems associated with this representation were studied in (Deng & Papadimitriou 1994; Elkind *et al*. 2007). However, this representation is not *complete*: there are yes/no games that cannot be represented as weighted voting games. A natural generalisation of weighted voting games that *is* complete in this sense are *k-vector weighted voting games*. A k-vector weighted voting game is an intersection of k different weighted voting games: a coalition wins if it wins in every component game. Many real world political decision-making bodies can be understood as k-vector weighted voting games, including the European Union (Bilbao *et al*. 2002), and the US Federal Legislature (Taylor & Zwicker 1999, p.13). The dimensionality, k, of a k-vector weighted voting game can be understood as a measure of the inherent complexity of the game. In this paper, we analyse dimensionality from a complexity-theoretic perspective. Specifically, we consider the problems of *equivalence*, (checking whether two given voting games have the same set of winning coalitions), and *minimality* (checking whether a given k-vector weighted voting game can be simplified by deleting one of the component games, or, more generally, is equivalent to a k'-vector weighted voting game with $k' < k$). We show that these problems are computationally hard, even if $k = 1$ or all weights are 0 or 1. However, we provide efficient algorithms for these problems for cases where both k is small and the weights are polynomially bounded. We also study the notion of *monotonicity* in voting games. Our results imply that monotone voting games are essentially as hard to represent and work with as general voting games.

Yes/No Voting Games

Formally, we can understand a yes/no voting game as a pair $Y = \langle N, W \rangle$, where $N = \{1, \ldots, n\}$ is the set of voters, and $W \subseteq 2^N$ is the set of *winning coalitions*, with the intended interpretation that, if $C \in W$, then C would be able to determine the outcome (either "yea" or "nay") to the question at hand, should they collectively choose to.

In some (though not all) domains, it is natural to require that a game is *monotone*, i.e., if $C_1 \subseteq C_2$ and $C_1 \in W$ then $C_2 \in W$. In what follows, we explicitly mention which of our results are for monotone games, and which ones are for general games. As a rule, our hardness results still hold when restricted to the class of monotone games, whereas our algorithms work correctly for general games.

We identify each $C \in 2^N$ with a string $x_C \in \{0,1\}^n$ in the natural way — the j'th element of x_C is a 1 iff $j \in C$. Similarly, C_x denotes the coalition associated with string x under this bijection. We will often abuse notation and use W to denote the set of such binary strings corresponding to the set of winning coalitions.

An obvious issue now arises: the naive representation of W (explicitly listing all winning coalitions) is of size $O(2^{|N|})$, which is not realistic in practice. So, then, how do we *succinctly represent* any given yes/no game, and in particular, the set W of winning coalitions. That is, can we find representations of W whose size is polynomial in $|N|$?

Weighted Voting Games

It *is* possible to represent *certain types* of yes/no voting game succinctly. *Weighted voting games* are a well known example (Taylor & Zwicker 1993). A weighted voting game M is a structure $M = \langle N, w_1, \ldots, w_n, q \rangle$ where N is the set of voters, $w_i \in \mathbb{R}$ is the *weight* of voter $i \in N$, and $q \in \mathbb{R}$ is the *quota* of the game. A coalition C is then deemed to be winning if $\sum_{i \in C} w_i \geq q$. Given a weighted voting game $M = \langle N, w_1, \ldots, w_n, q \rangle$, the corresponding yes/no game Y_M is thus $\langle N, \{C : C \subseteq N \ \& \ \sum_{i \in C} w_i \geq q\} \rangle$. The complexity of weighted voting games was originally studied in (Deng & Papadimitriou 1994), and more recently in (Elkind *et al*. 2007).

Weighted voting games are mathematically simple objects, and are widely used in the real world. However, they are not a *complete* representation for yes/no games: there exist yes/no games Y for which there exists no weighted voting game M such that $Y = Y_M$. (Consider a yes/no game in which the winning coalitions are exactly those containing an odd number of voters; it is easy to prove that such a game cannot be represented by a weighted voting game.) However, *k-vector weighted voting games* are a natural generalisation of weighted voting games which *are* complete in this sense (Taylor & Zwicker 1993; 1999).

k-Vector Weighted Voting Games

If $k \in \mathbb{N}$, then let $\mathbb{R}^k$ denote the set of k-element vectors of real numbers. We overload notation, and write $\sum$ for vector summation; for elements $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^k$, we write $\mathbf{v}_1 \geq \mathbf{v}_2$ to mean that each element in $\mathbf{v}_1$ is greater than or equal to the corresponding element in $\mathbf{v}_2$. A k-weighted voting game S is then a tuple $S = \langle N, \mathbf{w}_1, \ldots, \mathbf{w}_n, \mathbf{q} \rangle$, where N is the set of voters as above, $\mathbf{w}_i \in \mathbb{R}^k$ is a vector of k real weights for voter $i \in N$, and $\mathbf{q} \in \mathbb{R}^k$ is a vector of k real quotas. We then say a coalition C is winning if $\sum_{i \in C} \mathbf{w}_i \geq \mathbf{q}$; given a k-vector weighted voting game $S = \langle N, \mathbf{w}_1, \ldots, \mathbf{w}_n, \mathbf{q} \rangle$, the corresponding yes/no game Y_S is defined in the obvious way: $Y_S = \langle N, \{C : C \subseteq N \text{ and } \sum_{i \in C} \mathbf{w}_i \geq \mathbf{q}\} \rangle$. We say that S is a vector voting game if it is a k-vector voting game for some k.

We can also think of a k-vector weighted voting game as a $k \times n$ real matrix A (representing weights) and a length-k real vector b (quotas), so that $W = \{x \in \{0,1\}^n : Ax \geq b\}$. We will use both forms of notation interchangeably. The game (A, b) is a *non-negative* vector weighted voting game if the elements of A and b are non-negative; clearly, any such game is monotone.

Now, the good news is as follows:

Theorem 1 (Follows from (Taylor & Zwicker 1993))**.** *k-vector weighted voting games are an expressively complete representation for yes/no voting games. More precisely, for every yes/no voting game Y, there exists a $k \in \mathbb{N}$ and a k-vector weighted voting game S such that $Y = Y_S$.*

This result does not tell us anything about the *size*, k, of the weight/quota vectors needed to represent a yes/no game; it simply tells us that for any yes/no game, there *is* some equivalent vector weighted voting game. From a representational point of view, it would seem that the smaller the value of k, the better. Let us say a yes/no game Y is of *dimension k* if there exists a k-vector weighted voting game that corresponds to Y, but there does *not* exist any $(k-1)$-vector weighted voting game corresponding to Y (Taylor & Zwicker 1993, p.174). The dimension of a voting game is one way of measuring the *inherent complexity* of the game; the higher the dimension, the greater the intrinsic complexity. An obvious question is whether we can bound the dimension of any yes/no game; or at least, whether the dimension of a game can be guaranteed to be small, as a function of the number of voters. The answer to this question is no:

Theorem 2 (Follows from (Taylor & Zwicker 1993))**.** *There exists a countably infinite sequence $Y_0, Y_1, Y_2, \ldots$ of distinct yes/no voting games such that for all $i \in \mathbb{N}$, Y_i is of dimension $\Omega(2^{n_i - 1})$, where n_i is the number of voters in Y_i.*

There are in fact very simple examples of yes/no games with exponential dimension: yes/no games in which the winning coalitions are those containing an odd number of voters are one example. Although this result is negative – it tells us that we cannot always rely upon k-vector weighted voting games to be succinct – we can at least match it with an upper bound.

Theorem 3 (Follows from (Taylor & Zwicker 1993))**.** *Every yes/no voting game $Y = \langle N, W \rangle$ is of dimension $O(|2^N \setminus W|)$.*

Proof. Here is one construction. For every $x \in \{0,1\}^n - W$, construct a row A_x. If $x_j = 1$ then $A_{x,j} = -1$. If $x_j = 0$ then $A_{x,j} = 1$. Set $b_x = -|C_x| + 1/2$. □

Equivalence

We have seen that every yes/no game can be represented as a k-vector weighted voting game for some value of k. However, such a representation is not necessarily unique. We will say that two vector weighted voting games (assumed to be over the same set of voters) are *equivalent* if they have the same set of winning coalitions. An obvious question suggests itself: how hard is it to check the *equivalence* of any two games.

To reason about complexity-theoretic aspects of vector weighted voting games, we need to fix a finitary representation of such games. Therefore for the rest of the paper, unless specified otherwise, we assume that all weights are integers (though possibly non-positive) and given in binary, and "polynomial" means "polynomial in the representation size of the input". Clearly, a representation that uses rational weights can be scaled up so that all weights become integer. Moreover, there is no loss of generality in restricting ourselves to integer weights, as it is known (Muroga 1971) [1] that every weighted voting game is equivalent to a weighted voting game with integer weights in which the sum of the absolute values of the weights is $2^{O(n \log n)}$. This result can be generalised to k-vector weighted voting games by replacing component games with equivalent "compact" games one by one. Hence, any k-vector weighted voting game has a representation of size polynomial in n and k.

We will first show that the problem of checking whether two vector weighted voting games are equivalent is co-NP-complete. This result holds even if the sizes of the games k_1 and k_2 are unbounded and all weights are 0 or 1, or if $k_1 = k_2 = 1$ and the weights can be large. However, if both $k_1, k_2 < C$ for some constant C and the weights are polynomially bounded, the hardness result no longer holds. In fact, we show that in this case the problem becomes polynomial-time solvable. We first present our hardness results, followed by the polynomial-time algorithm for the special case mentioned above.

Theorem 4. *The problem of checking whether any two given vector weighted voting games are equivalent is co-NP-complete, even if all weights are equal to 0 or 1 (and hence the games are monotone).*

Proof. We work with the complement problem: that of checking whether two vector weighted voting games have different sets of winning coalitions. Membership in NP is straightforward: A non-deterministic polynomial-time algorithm guesses a coalition C and verifies that exactly one of $Ax_C \geq b$ and $A'x_C \geq b'$ is true. We show hardness by reduction from VERTEX COVER (Garey & Johnson 1979, p.190). Consider an instance G, j of VERTEX COVER in which G has n vertices and m edges and $0 \leq j \leq n$. (This is a "yes" instance iff G has a vertex cover of size j.) We construct (A', b') as follows. The number of columns of A' is n. The number of rows of A' is $k = m$. For each edge e of G we construct a row A'_e. All entries of A'_e are 0 except $A'_{e,v} = 1$ for each endpoint v of e. The quota $b'_e = 1$. W' corresponds to the set of vertex covers of G. The game (A, b) is constructed from (A', b') by adding one more row. Every entry of the new row of A is 1. The corresponding entry in b is $j + 1$. Thus, W corresponds to the set of vertex covers of G of size at least $j + 1$. Now, G has a vertex cover of size j iff the two games have different sets of winning coalitions. Note that the reduction yields games with weights in $\{0, 1\}$. □

If we do not require the weights to be in $\{0, 1\}$, the problem becomes hard even for $k = 1$ (and hence for any larger values of k).

Theorem 5. *The problem of checking whether any two given weighted voting games are equivalent is co-NP-complete, even if both games are non-negative (and hence monotone).*

Proof. The membership is as in the previous result. To prove co-NP-hardness, we will show that the complementary problem, i.e., checking whether two weighted voting games are non-equivalent, is NP-hard. The reduction is from SUBSET SUM (Garey & Johnson 1979, p.223). Recall that an instance of SUBSET SUM is given by a list of integers $L = \{a_1, \ldots, a_n\}$ and a quota T. It is a "yes"-instance, if there is a subset of L that sums up to T, and a "no"-instance otherwise. Given an instance of SUBSET SUM, we define our two weighted voting games M_1 and M_2 as $M_1 = \langle N, a_1, \ldots, a_n, T \rangle$ and $M_2 = \langle N, a_1, \ldots, a_n, T + 1 \rangle$. Clearly, the sets of winning coalitions under M_1 and M_2 are distinct if and only if there is a coalition whose weight under M_1 (and M_2) is exactly T, i.e., we started with a "yes"-instance of SUBSET SUM. □

The vector voting games used in practice often have small dimension ($k \leq 3$) and use weights that are integer and at most polynomial in n. In this case, there is an efficient algorithm for checking whether two such games are equivalent.

Theorem 6. *Given a k_1-vector weighted voting game $S = \langle N, A, b \rangle$ and a k_2-vector weighted voting game $\hat{S} = \langle N, \hat{A}, \hat{b} \rangle$, there is an algorithm that checks whether S and $\hat{S}$ are equivalent and which runs in time $\mathrm{poly}((nW)^{k_1+k_2})$, where $W = \max(A_{\max}, \hat{A}_{\max})$, and $A_{\max}$ (respectively, $\hat{A}_{\max}$) is the element of A (respectively, $\hat{A}$) with the maximum absolute value.*

Proof. We will use dynamic programming to check whether there is a coalition J that wins under S, but not under $\hat{S}$. We then apply the same algorithm with the roles of S and $\hat{S}$ reversed. If in both cases the algorithm finds no such coalition, then S and $\hat{S}$ are equivalent.

First, note that the weight of each coalition under a weight vector that corresponds to a row of A or $\hat{A}$ is between $-nW$ and nW. Now, for any two integer vectors $\mathbf{w} \in [-nW, nW]^{k_1}$, $\hat{\mathbf{w}} \in [-nW, nW]^{k_2}$, set $X(k, \mathbf{w}, \hat{\mathbf{w}}) = 1$ if there is a subset J of the first k voters with the characteristic vector x_J such that $Ax_J = \mathbf{w}$, $\hat{A}x_J = \hat{\mathbf{w}}$, and set $X(k, \mathbf{w}, \hat{\mathbf{w}}) = 0$ otherwise. These quantities can be computed as follows. We have $X(1, \mathbf{w}, \hat{\mathbf{w}}) = 1$ if and only if the first column of A coincides with $\mathbf{w}$ and the first column

[1](Muroga 1971) shows this for linear threshold functions rather than for weighted voting games, but there is a natural isomorphism between the former and the latter.

of $\hat{A}$ coincides with $\hat{\mathbf{w}}$. Now, suppose that we have computed $X(i, \mathbf{w}, \hat{\mathbf{w}})$ for all $i < k$ and all $\mathbf{w} \in [-nW, nW]^{k_1}$, $\hat{\mathbf{w}} \in [-nW, nW]^{k_2}$. We can now compute $X(k, \mathbf{w}, \hat{\mathbf{w}})$ as follows: $X(k, \mathbf{w}, \hat{\mathbf{w}}) = 1$ if and only if $X(k-1, \mathbf{w}, \hat{\mathbf{w}}) = 1$ or $X(k-1, \mathbf{w} - A^{(k)}, \hat{\mathbf{w}} - \hat{A}^{(k)}) = 1$, where $A^{(k)}$ denotes the kth column of the matrix A, and $\hat{A}^{(k)}$ denotes the kth column of the matrix $\hat{A}$. Finally, after all $X(n, \mathbf{w}, \hat{\mathbf{w}})$ have been computed, we check if there is a pair of vectors $\mathbf{w}, \hat{\mathbf{w}}$ such that $X(n, \mathbf{w}, \hat{\mathbf{w}}) = 1$ and $\mathbf{w} \geq b$, but not $\hat{\mathbf{w}} \geq b'$ (i.e., there is at least one entry of $\hat{\mathbf{w}}$ that is smaller than the corresponding entry of b'). By construction, such a pair corresponds to a coalition that wins under S, but not under $\hat{S}$. It is easy to see that the running time of our algorithm is $\mathrm{poly}((nW)^{k_1+k_2})$. □

For weighted voting games ($k = 1$) the requirement that all weights are polynomially bounded can be relaxed: it suffices if this is the case for one of the two games.

Theorem 7. *Given two weighted voting games $S = \langle N, w_1, \ldots, w_n, q\rangle$ and $S' = \langle N, w'_1, \ldots, w'_n, q'\rangle$, there is an algorithm that checks whether S and S' are equivalent and which runs in time $\mathrm{poly}(n, w_{\max})$, where $w_{\max} = \min(\max\{w_i : i = 1, \ldots, n\}, \max\{w'_i : i = 1, \ldots, n\})$.*

Proof. (Sketch) Given a coalition $J \subseteq N$, let $w(J)$ (respectively, $w'(J)$) denote the total weight of the members of J under S (respectively, S'). As in the previous proof, it suffices to describe an algorithm that checks whether there is a coalition J such that $w(J) \geq q$, $w'(J) < q'$ and then makes the corresponding check, reversing the roles of J and J' . It is easy to see that this problem is an instance of KNAPSACK, where $w'_1, \ldots, w'_n$ play the role of weights, and $w_1, \ldots, w_n$ play the role of values. It is well known that KNAPSACK can be solved in polynomial time if either the weights or the values are polynomially bounded. □

Dimensionality

One of the key parameters of interest in yes/no games is the dimension of the game, since, as we noted above, this is an obvious measure of the inherent complexity of the voting game. We saw that, in the worst case, yes/no games are of dimension exponential in the number of voters. An obvious question is whether restrictions – and in particular, monotonicity – lead to simpler games, i.e., games with a smaller dimension. In the case of monotonicity, the answer is no:

Theorem 8. *For every n satisfying $n = 2 \pmod 4$, there exists a monotone yes/no voting game with n voters of dimension at least $\binom{n}{n/2}/2$.*

Proof. The set of all winning coalitions W includes every set of size $s > n/2$ and excludes every set of size $s < n/2$. Let C be the coalition consisting of the first $n/2$ voters. Note that for any set C' of size $n/2$, the Hamming distance between C and C', $\mathrm{Ham}(C, C')$, is even. Include C' in W if and only if $\mathrm{Ham}(C, C') = 2 \pmod 4$.

We will show that exactly half of all sets of size $n/2$, i.e., $\binom{n}{n/2}/2$ sets of size $n/2$, are included in W. To see this, consider any coalition C' of size $n/2$, and let C'' be its complement. Let j be the number of voters in $\{1, \ldots, n/2\}$ that are in C' and let j' be the number of voters in $\{n/2 + 1, \ldots, n\}$ that are in C'. Then $\mathrm{Ham}(C, C') = n/2 - j + j'$ and $\mathrm{Ham}(C, C'') = n/2 - j' + j$. Since $\mathrm{Ham}(C, C') + \mathrm{Ham}(C, C'') = n$, which is equal to 2 modulo 4, exactly one of $\mathrm{Ham}(C, C')$ and $\mathrm{Ham}(C, C'')$ is equal to 2 modulo 4, so exactly one of C' and C'' is in W.

Now consider any binary string x with $n/2$ 1's such that $x \notin W$. Suppose that x violates the ℓth inequality of the weighted voting game, i.e., $a^\ell \cdot x < b^\ell$. Fix arbitrary i, j in $1, \ldots, n$ such that $x_i = 1$, $x_j = 0$ and consider the string x' obtained from x by switching x_i with x_j. The Hamming distance between x and x' is 2, so $x' \in W$. Consequently, $a^\ell \cdot x' \geq b^\ell$, which implies $a^\ell_i < a^\ell_j$. Now, consider any other binary string y with $n/2$ 1's such that $y \notin W$. Suppose that y also violates the ℓth constraint. By a similar argument we can show that for any i, j in $1, \ldots, n$, if $y_i = 0$, $y_j = 1$ then $a^\ell_i > a^\ell_j$. Since $x \neq y$, and both of them have $n/2$ 1's, we can choose i, j so that $x_i = y_j = 1$, $x_j = y_i = 0$. We have $a^\ell_i < a^\ell_j$, $a^\ell_i > a^\ell_j$. The contradiction shows that x and y necessarily violate different constraints, i.e., the matrix A has at least $\binom{n}{n/2}/2$ rows. □

Minimality and Relevance

In this section, we focus on the problem of determining the dimensionality of a given voting game. The first question we would like to address is whether a k-vector weighted voting game is *redundant*, i.e., whether a particular component of a k-vector weighted voting game could be deleted without affecting the overall set of winning coalitions of the game. This leads to two different decision problems, depending on whether we ask this question about a specific row of the weight matrix or would like to check if such a row exists:

> RELEVANCE:
> *Input:* A weighted voting game (A, b) and a row i of A.
> *Question:* Is row i relevant to the game? More formally: Construct (A', b') from (A, b) by deleting row i. Do (A, b) and (A', b') have different sets of winning coalitions?
> MINIMALITY:
> *Input:* A weighted voting game (A, b).
> *Question:* Is it the case that for every row i of A, the voting game (A', b') constructed from (A, b) by deleting row i is not equivalent to (A, b)?

It is easy to see that both problems are in NP. Moreover, a polynomial-time algorithm for RELEVANCE would imply a polynomial-time algorithm for MINIMALITY, but not vice versa. As in the previous section, we will be interested in the complexity of these problems under natural restrictions, i.e., when k is small or all weights are 0 or 1.

Theorem 9. MINIMALITY *and* RELEVANCE *are* NP*-complete even if all weights are in $\{0, 1\}$ (and hence the game is monotone).*

Proof. Membership is as before. We show hardness by reduction from VERTEX COVER, as in the proof of Theorem 4. Let G, j be an instance of VERTEX COVER. Assume without loss of generality that $j \leq n-3$. Construct the game (A, b) as in the proof of Theorem 4. As argued in that proof, the last row of A is relevant iff G has a vertex cover of size j. Hence (A, b, m) is a "yes"-instance of RELEVANCE iff G, j is a "yes"-instance of VERTEX COVER. Now, let W be the set of vertex covers of G of size at least $j+1$. For each row i of A, let (A^i, b^i) be the game obtained from (A, b) by deleting row i and let W^i be its winning coalitions. First, suppose that i is a row corresponding to an edge (v, w) of G. Note that every winning coalition of (A, b) is still a winning coalition of (A^i, b^i), so $W \subseteq W^i$. Furthermore, $W \subset W^i$, since $V - \{v, w\}$ is a winning coalition of (A^i, b^i) of size $n-2 \geq j+1$. So all rows i of (A, b) corresponding to edges of G are relevant. We conclude that (A, b) is a "yes" instance of MINIMALITY iff the final row of A is relevant, i.e., iff G has a vertex cover of size j. □

Theorem 10. MINIMALITY *and* RELEVANCE *are* NP-*complete even if $k = 2$ and all weights are non-negative.*

Proof. The reduction is from SUBSET SUM. We start with the construction used in the proof of Theorem 5, i.e., given an instance $(a_1, \ldots, a_n, T)$ of SUBSET SUM, we construct a voting game M with $N = \{1, \ldots, n\}, A_1^i = A_2^i = a_i$ for $i = 1, \ldots, n$, and $b_1 = T, b_2 = T+1$. Clearly, under this voting game the first row is always redundant, and the second row is relevant if and only if $(a_1, \ldots, a_n, T)$ is a "yes"-instance of SUBSET SUM. We will now modify this game as follows: set $X = 2\sum_{i=1}^n a_i$, add a player $n+1$ to N, set $A_1^{n+1} = X, A_2^{n+1} = 0$, and set $b_1 = X + T, b_2 = T + 1$. Denote the resulting voting game by M'. Clearly, a coalition C wins under M' if and only if $n+1 \in C$ and $C \setminus \{n+1\}$ wins under M. Hence, under M' the first row of A is no longer redundant, as it ensures that $n+1$ is present in the winning coalition. Therefore M' is minimal (and the second row of A is relevant) if and only if we started with a "yes"-instance of SUBSET SUM. □

If k is bounded by a constant, and all weights are polynomially bounded, we can use the algorithm from the proof of Theorem 6 to check whether the voting games (A, b) and (A', b') obtained from (A, b) by deleting the ith row are equivalent. Hence we have the following result.

Corollary 1. *Given a k-vector weighted voting game (A, b), where k is bounded by a constant and all weights are polynomially bounded, there is a polynomial-time algorithm that decides* RELEVANCE *(and hence* MINIMALITY*).*

Even if the k-vector weighted voting game (A, b) is *minimal*, there may still exist a k'-vector weighted voting game (A', b') with $k' < k$ that has the same set of winning coalitions as (A, b): the point is that even though no row of A can be deleted, it can still be possible to construct a completely different set of weights that describes the corresponding set of winning coalitions more compactly. We say that (A, b) is *minimum* if it is not equivalent to *any* k'-vector weighted voting game for $k' < k$. We will now show that deciding whether (A, b) is minimum is NP-hard even if $k = 2$.

Theorem 11. *Given a 2-vector weighted voting game $S = \langle N, \mathbf{w}_1, \ldots, \mathbf{w}_n, \mathbf{q} \rangle$, where $\mathbf{w}_i = (w_i^1, w_i^2)$ for $i = 1, \ldots, n$, and $\mathbf{q} = (q^1, q^2)$, it is* NP-*hard to decide whether S is minimum, i.e., there is no weighted voting game $T = \langle N, u_1, \ldots, u_n, r \rangle$ such that S and T are equivalent.*

Proof. We reduce from BALANCED PARTITION, a version of the classical NP-complete PARTITION problem (Garey & Johnson 1979, p.223), which is also known to be NP-complete. An instance of BALANCED PARTITION is a list of integers $L = (a_1, \ldots, a_n)$, where n is even, that satisfies $\sum_{i=1}^n a_i = 2K$ for some $K \in \mathbb{N}$. It is a "yes"-instance if there is a subset $J \subseteq L$ that satisfies $\sum_{a_i \in J} a_i = K$, $|J| = n/2$, and a "no"-instance otherwise. Given an instance of BALANCED PARTITION, we will construct a 2-vector weighted voting game S as follows. We set $N = \{1, \ldots, n\}$, and for $i = 1, \ldots, n$ we set $w_i^1 = 4Kn + a_i, w_i^2 = 4Kn - a_i$. Also, we define $q^1 = 2Kn^2 + K, q^2 = 2Kn^2 - K$. Clearly, under these rules every coalition of size at least $n/2+1$ wins, and any coalition of size at most $n/2 - 1$ loses. Now, suppose that we started with a "yes"-instance of BALANCED PARTITION, i.e., there exists a subset J of size $n/2$ that satisfies $\sum_{a_i \in J} a_i = K$. It is easy to see that the corresponding coalition wins. Note that we also have $\sum_{a_i \notin J} a_i = K$, so the coalition that corresponds to $N \setminus J$ wins as well.

On the other hand, suppose that we started with a "no"-instance of BALANCED PARTITION. Then there are no winning coalitions of size exactly $n/2$. Indeed, if for $J \subseteq N$, $|J| = n/2$, we have $\sum_{a_i \in J} a_i < K$, then $\sum_{a_i \in J} w_i^1 < q^1$, and if $\sum_{a_i \in J} a_i > K$, then $\sum_{a_i \in J} w_i^2 < q^2$.

In the latter case, S is equivalent to a very simple weighted voting game, where the weight of each player is 1, and the quota is $n/2+1$. It remains to show that in the former case, there is no weighted voting game that is equivalent to S.

To see this, note that any weighted voting game has the following property: given any two disjoint winning coalitions C_1 and C_2 and any two elements $x \in C_1, y \in C_2$ with weights w_x and w_y, it cannot be the case that swapping x and y turns both C_1 and C_2 into losing coalitions, i.e., at least one of $C_1 \setminus \{x\} \cup \{y\}$ and $C_2 \setminus \{y\} \cup \{x\}$ must be winning (Taylor & Zwicker 1993). Indeed, if $w_x \leq w_y$, then replacing x with y cannot decrease the weight of C_1, and if $w_x > w_y$, then replacing y with x cannot decrease the weight of C_2. On the other hand, for the 2-vector weighted voting game constructed above, this is not the case.

Indeed, consider a set J that satisfies $\sum_{a_i \in J} a_i = K$, and the corresponding winning coalitions J and $N \setminus J$. We can assume that not all a_i are equal, so without loss of generality pick $a_j \in J$, $a_k \in N \setminus J$ so that $a_j < a_k$. We claim that swapping the corresponding players j and k will turn both J and $N \setminus J$ into losing coalitions: we have $w_j^1 < w_k^1, w_j^2 > w_k^2$, so $\sum_{a_i \in J} w_i^2 - w_j^2 + w_k^2 < q^2, \sum_{a_i \in N \setminus J} w_i^1 - w_j^1 + w_k^1 < q^1$. Hence, for a "yes"-instance of BALANCED PARTITION, the resulting 2-vector weighted voting game S is not equivalent to any weighted voting game, i.e., S is minimum. □

Note that we do not claim that this problem is NP-complete. Indeed, to show that a given k-vector weighted voting game S is minimum, we would have to check that *for any* k'-weighted voting game, with $k' < k$ *there exists* a coalition on which this game differs from S. As any k'-vector weighted voting game is equivalent to a k'-vector weighted voting game with exponentially bounded integer weights (Muroga 1971) it suffices to perform this check for weighted voting games that can be represented using $\text{poly}(n, k)$ bits. This shows that the problem is in Π_2^p, but not necessarily in NP or co-NP.

Unlike for other problems considered in this paper, we do not know if checking whether a given voting game is minimum is NP-hard even if all weights are in $\{0, 1\}$. Also, we do not have a pseudopolynomial time algorithm for this problem for the case when k is bounded by a constant and all weights are polynomially bounded. However, we can show that in the latter case it is possible to check in polynomial time whether a given k-vector weighted voting game S is equivalent to *some* weighted voting game. To do so, we construct a linear program $\mathcal{L}$ with variables $w_1, \dots, w_n, q$ and constraints that for each J specify whether $\sum_{i \in J} w_i \geq q$ or $\sum_{i \in J} w_i < q$, depending on whether J is a winning coalition under S. While this linear program has exponential size, it has a separation oracle that can be implemented in polynomial time using dynamic programming (see (Elkind *et al*. 2007) and the proof of Theorem 6). Hence, $\mathcal{L}$ can be solved in polynomial time using ellipsoid method.

This algorithm can be modified to check if a given k-vector weighted voting game S is equivalent to a *monotone* weighted voting game. To show this, we first need the following lemma.

Lemma 1. *Any monotone weighted voting game can be represented using non-negative weights.*

Proof. Consider a monotone weighted voting game in which some of the weights are negative. Replace these weights with 0's one by one. We claim that after each step, the resulting game is equivalent to the original game. Indeed, suppose that replacing $w_i < 0$ with 0 changed the set of winning coalitions. Then it has to be the case that in the new game there is a winning coalition C, $i \in C$, that did not win in the original game: the weight of any coalition did not decrease, and it only changed for coalitions containing i. In the new game, $C \setminus \{i\}$ is also a winning coalition, as it has the same weight as C. It does not contain i, so it is a winning coalition in the original game as well. But this violates monotonicity, as we assumed that C was not a winning coalition in the original game. Hence, given a monotone weighted voting game, we can construct an equivalent non-negative representation. □

Now, consider a linear program $\mathcal{L}'$ obtained from $\mathcal{L}$ by adding constraints of the form $w_i \geq 0, i = 1, \dots, n$. Clearly, $\mathcal{L}$ has a solution if and only if the input game is equivalent to a non-negative weighted voting game, and hence, by Lemma 1, to a monotone weighted voting game. Moreover, it is easy to modify the separation oracle to take into account the new constraints, so $\mathcal{L}$ can also be solved in polynomial time.

Conclusions

Weighted voting games are an important, widely used, and mathematically appealing class of yes/no voting games, whose computational properties have been closely studied (Deng & Papadimitriou 1994; Elkind *et al*. 2007). However, k-vector weighted voting games have received much less attention from a computational point of view, although they also play a very significant role in real-world political systems. We have investigated a number of questions relating to the dimensionality of k-vector weighted voting games. The closest related work we know of is (Deineko & Woeginger 2006), which shows that it is NP-complete to check whether, given a collection of d_2 weighted voting games, it is possible to represent it as a collection of d_1 weighted voting games, $1 \leq d_2 < d_1$. Our work differs in many ways: for example, we consider monotone and $0, 1$ weight cases, and consider more general problems; moreover we prove completeness rather than hardness. Several issues suggest themselves for future work. One is the relationship to other representations for yes/no games. For example, *logical* representations of coalitional games have received some attention recently (e.g., (Ieong & Shoham 2005)). It would be interesting to consider the relations between such representations and (k-vector) weighted voting games.

References

Bartholdi, J. J.; Tovey, C. A.; and Trick, M. A. 1989. The computational difficulty of manipulating an election. *Social Choice and Welfare* 6:227–241.

Bilbao, J. M.; Fernández, J. R.; Jiminéz, N.; and López, J. J. 2002. Voting power in the European Union enlargement. *European Jnl of Oper. Res*. 143:181–196.

Deineko, V. G., and Woeginger, G. 2006. On the dimension of simple monotonic games. *European Jnl of Oper. Res*. 170(1):315–318.

Deng, X., and Papadimitriou, C. H. 1994. On the complexity of cooperative solution concepts. *Math. of Oper. Res*. 19(2):257–266.

Elkind, E.; Goldberg, L.; Goldberg, P.; and Wooldridge, M. 2007. Computational complexity of weighted threshold games. In *Proceedings AAAI-2007*.

Endriss, U., and Lang, J., eds. 2006. *Proceedings of COMSOC-2006*.

Garey, M. R., and Johnson, D. S. 1979. *Computers and Intractability*. W. H. Freeman.

Ieong, S., and Shoham, Y. 2005. Marginal contribution nets: A compact representation scheme for coalitional games. In *Proceedings of EC'05*.

Muroga, S. 1971. *Threshold Logic and its Applications*. John Wiley & Sons.

Taylor, A., and Zwicker, W. 1993. Weighted voting, multicameral representation, and power. *Games and Economic Behavior* 5:170–181.

Taylor, A. D., and Zwicker, W. S. 1999. *Simple Games*. Princeton UP.