Combinatorial and computational aspects of multiple weighted voting games

Haris Aziz, Mike Paterson and Dennis Leech

No 823

WARWICK ECONOMIC RESEARCH PAPERS

DEPARTMENT OF ECONOMICS

THE UNIVERSITY OF WARWICK

Combinatorial and computational aspects of multiple weighted voting games

Haris Aziz
Computer Science Department
University of Warwick
Coventry, UK, CV4 7AL
Email: haris.aziz@warwick.ac.uk

Mike Paterson
Computer Science Department
University of Warwick
Coventry, UK, CV4 7AL
Email: msp@dcs.warwick.ac.uk

Dennis Leech
Economics Department
University of Warwick
Coventry, UK, CV4 7AL
Email: d.leech@warwick.ac.uk

Abstract

Weighted voting games are ubiquitous mathematical models which are used in economics, political science, neuroscience, threshold logic, reliability theory and distributed systems. They model situations where agents with variable voting weight vote in favour of or against a decision. A coalition of agents is winning if and only if the sum of weights of the coalition exceeds or equals a specified quota. We provide a mathematical and computational characterization of *multiple weighted voting games* which are an extension of weighted voting games[1]. We analyse the structure of multiple weighted voting games and some of their combinatorial properties especially with respect to dictatorship, veto power, dummy players and Banzhaf indices. An illustrative Mathematica program to compute voting power properties of multiple weighted voting games is also provided.

Keywords: multi-agent systems, multiple weighted voting games, game theory, algorithms and complexity, voting power.

I. Introduction

A. Motivation

Weighted voting games are mathematical models which are used to analyze voting bodies in which the voters have different number of votes. In weighted voting games, each voter is assigned a non-negative weight and makes a vote in favour of or against a decision. The decision is made if and only if the total weight of those voting in favour of the decision is equal to or greater than the quota. Weighted voting games are also encountered in threshold logic, reliability theory, neuro-science and logical computing devices [1]. Nordmann et al. [2] deal with reliability and cost evaluation of weighted dynamic-threshold voting-systems. Systems of this type are used in various areas such as target and pattern recognition, safety monitoring and human organization systems.

Weighted voting games have also been applied in various political and economic organizations. Prominent applications include the United Nations Security Council, the Electoral College of the United States and the International Monetary Fund ([3], [4]). The weights of the players do not always indicate the power the player has in affecting decisions. This has led to a significant literature on voting power in WVG's. The distribution of voting power in the European Union Council of Ministers has received special attention in [5], [6], [7], [8], [9] and [10]. Voting power is also used in joint stock companies where each shareholder gets votes in proportion to the ownership of a stock ([11], [12]).

B. Outline

Section II outlines preliminaries definitions related to voting games, voting power and complexity. Section III covers the background and structure of multiple weighted voting games. In section IV, we analyse combinatorial properties o multiple weighted voting games especially with respect to dictatorship,

[1]A preliminary version of the paper has been presented at ACID2007, Algorithms & Complexity in Durham conference.

veto power, dummy players. Section V outlines algorithmic considerations when computing voting power of players in multiple weighted voting games. We then conclude in section VI.

II. Preliminaries

A. Voting Games

In this section we give definitions and notations of key terms. The set of voters is $N = \{1, ..., n\}$.

Definition II.1. *A simple voting game is a pair* (N, v) *where the valuation function* $v : 2^N \rightarrow \{0, 1\}$ *has the properties that* $v(\emptyset) = 0$*,* $v(N) = 1$ *and* $v(S) \leq v(T)$ *whenever* $S \subseteq T$*. A coalition* $S \subseteq N$ *is winning if* $v(S) = 1$ *and losing if* $v(S) = 0$*. A simple voting game can alternatively be defined as* (N, W) *where* W *is the set of winning coalitions.*

Definition II.2. *The simple voting game* (N, v) *where*
$W = \{X \subseteq N, \sum_{x \in X} w_x \geq q\}$ *is called a WVG. A WVG is denoted by* $[q; w_1, w_2, ..., w_n]$ *where* w_i *is the voting weight of player* i*. Generally* $w_i \geq w_j$ *if* $i < j$*.*

Generally, $q > \frac{1}{2}\sum_{1 \leq i \leq n} w_i$ so that there are no two mutually exclusive winning coalitions at the same time. WVGs with this property are termed *proper*. Proper WVGs are also desirable because they satisfy the criterion of the majority getting preference.

B. Voting Power

Definition II.3. *A player* i *is* critical *in a coalition* S *when* $S \in W$ *and* $S \setminus i \notin W$*. For each* $i \in N$*, we denote the number of coalitions in which* i *is critical in game* v *by* $\eta_i(v)$*. The* Banzhaf Index *of player* i *in weighted voting game* v *is* $\beta_i = \frac{\eta_i(v)}{\sum_{i \in N} \eta_i(v)}$.

Definition II.4. *In a simple game, a player with a Banzhaf index of zero is called a* dummy.

Definition II.5. *A* Dictator *is a player who is present in every winning coalition and absent from every losing coalition.*

The definition implies that the dictator is player 1 with the biggest weight, $\beta_1 = 1$ and $\forall i \neq 1$, $\beta_i = 0$. This is because any other player cannot prove critical in any coalition. Any winning coalition has to include the dictator and the opting out of any other player cannot make the coalition losing. For player 1 to be the dictator, $w_1 \geq q$ and $\sum_{2 \leq i \leq n} w_i < q$. It is obvious that if a dictator exists, he is unique.

Definition II.6. *A player has* veto power *if and only if the player is present in every winning coalition.*

It is evident that a dictator has veto power but a player with veto power is not necessarily a dictator. Moreover there can be two or more players with veto power.

Example II.7. *In the WVG,* $[5; 3, 2, 1]$*, player 1 is not a dictator but it has veto power. Player 2 also has veto power.*

C. Complexity

Definition II.8. *A problem is in complexity class* P *if it can be solved in time which is polynomial in the size of the input. A problem is in the complexity class* NP *if it its solution can be verified in time which is polynomial in the size of the input of the problem. A problem is in the complexity class* NP-HARD *if any problem in* NP *is polynomial time reducible to that problem.* NP-HARD *problems are as hard as the hardest problems in* NP.

III. Multiple Weighted Voting Games

A. Introduction

Definition III.1. *Let $(N_1, v_1), \ldots, (N_m, v_m)$ be simple games. Then a simple game $(N, v) = (N_1, v_1) \times \ldots \times (N_m, v_m)$ is the* product *of these games where $N = \bigcup_{t=1}^{m} N_i$ and $v(S) = 1$ if and only if for all $1 \leq t \leq m$, $v_t(S \cap N_t) = 1$. If all N_is of the games $(N_1, v_1), \ldots, (N_m, v_m)$ coincide then, the product (N, v) of these games is called the* meet *of the games. (N, v) can simply be written as $(N, v_1 \wedge \ldots \wedge v_m)$ and $v(S) = 1$ if and only if for all $1 \leq t \leq m$, $v_t(S) = 1$. A* meet-multiple weighted voting game (meet-MWVG) *is a meet of multiple weighted voting games.*

Taylor and Zwicker [1] refer to the meet of multiple weighted voting games as vector weighted games.

Definition III.2. *Let $(N_1, v_1), \ldots, (N_m, v_m)$ be simple games. Then a simple game $(N, v) = (N_1, v_1) + \ldots + (N_m, v_m)$ is the* sum *of these games where $N = \bigcup_{t=1}^{m} N_i$ and $v(S) = 1$ if and only if there exists a t, $1 \leq i \leq m$ such that $v_t(S \cap N_t) = 1$. If all N_is of the games $(N_1, v_1), \ldots, (N_m, v_m)$ coincide then, the product (N, v) of these games is called the* join *of the games. (N, v) can simply be written as $(N, v_1 \vee \ldots \vee v_m)$ and $v(S) = 1$ if and only if there exists a t, $1 \leq i \leq m$ such that $v_t(S) = 1$. A* join-multiple weighted voting game (join-MWVG) *is a join of multiple weighted voting games.*

Definition III.3. *A coalition S is* blocking *if its complement $\overline{S}$ is losing. Then $G^d = (N, W^d)$ is the* dual *of the game $G = (N, W)$ where W^d is the set of all blocking coalitions.*

Taylor and Zwicker [1] prove the following proposition:

Proposition III.4. *(Taylor & Zwicker) Let $G_1 = (N_1, v_1)$ and $G_2 = (N_2, v_2)$ be simple games. Then $(G_1 + G_2)^d = {G_1}^d \times {G_2}^d$ and $(G_1 \times G_2)^d = {G_1}^d + {G_2}^d$.*

We can get the following corollary from this proposition:

Corollary III.5. *For simple games $G_i = (N_i, v_i)$ for all $i = 1, \ldots, n$, $(\sum G_i)^d = \prod {G_i}^d$*

From now on when we mention, MWVG, we will assume the meet of the respective games unless otherwise stated. MWVGs are utilized in various situations. The treaty of Nice made the overall voting games of the EU countries a triple majority weighed voting game with certain additional constraints. MWVGs are useful in multi-criteria multi-agent systems.

B. Structure

We define S_i as the set of coalitions not including player i. Then S_i can be partitioned into three mutually exclusive sets:

$$S_i = W_i(v) \cup C_i(v) \cup L_i(v)$$

where

- $W_i(v)$ is the set of coalitions not including player i which are winning in the multiple game v
- $L_i(v)$ is the set of coalitions not including player i which are losing in the multiple game v even if player i joins the coalitions.
- $C_i(v)$ is the set of coalitions not including player i which are losing in the multiple game v but winning in v if player i joins the coalitions.

The number of coalitions in which player i is critical in the multiple game v is $\eta_i(v) = |C_i(v)|$. In a 2-game MWVG, i is critical in a coalition c if $(c \in C_i(v_1) \wedge c \in C_i(v_2)) \vee (c \in W_i(v_1) \wedge c \in C_i(v_2)) \vee (c \in C_i(v_1) \wedge c \in W_i(v_2))$. In a MWVG, i is critical in a coalition c if

$$(\forall j : (c \in C_i(v_j) \vee c \in W_i(v_j)) \wedge (\exists j : c \in C_i(v_j))$$

We define $W(v)$ as the set of winning coalitions in v and $W(v_i)$ as the set of winning coalitions in v_i. In that case

$$W(v) = W(v_1) \wedge W(v_2) ... \wedge W(v_m)$$

Similarly if we define $L(v)$ as the set of losing coalitions in v and $L(v_i)$ as the set of losing coalitions in v_i. In that case

$$L(v) = L(v_1) \vee L(v_2) ... \vee L(v_m)$$

C. Trade robustness, Dimension

Definition III.6. *The dimension of* (N, v) *is the least* k *such that there exist WMGs* $(N, v_1), \dots, (N, v_k)$ *such that* $v = (N, v_1) \wedge \dots \wedge (N, v_k)$

Deineko and Woeginger [13] show that it is NP-hard to verify the dimension of multiple-weighted voting games. In [14], it is pointed out that the dimension of a game is at most the number of maximal losing coalitions. This kind of configuration is not very helpful though in estimating the actual dimension of a MWVG.

Taylor and Zwicker [1] defined the trade-robustness of simple games: a simple game (N, v) is *k-trade robust* if no trading among $j \leq k$ winning coalitions $W_1, \dots W_j$ that leads to losing coalitions $L_1, \dots, L_j$ in such a way that $|\{p : i \in W_p\}| = |\{p : i \in L_p\}|$ for each $i \in N$. A simple game is *trade robust* if it is *k-trade robust* for all k. They proved that a simple game is *trade robust* if and only if it is a WVG. However we observe that MWVGs are not even *swap-robust* which is robustness for a more restricted notion of trading in which a one to one player exchange between any two winning coalitions does not render both coalitions losing:

Example III.7. *Let* $(N, v) = (N, v_1 \wedge v_2)$ *where* $v_1 = [20; 18, 5, 0, 5, 5, 2, 5]$ *and* $v_2 = [20; 0, 5, 18, 5, 5, 2, 5]$. *We see that coalitions* $\{1, 3, 6\}$ *and* $\{2, 4, 5, 7\}$ *are winning in* v. *However if we have a trade so that the resultant coalitions are* $\{2, 3, 6\}$ *and* $\{1, 4, 5, 7\}$*, then both coalitions are losing.*

IV. Properties of MWVGs

A. Unity and zero in MWVGs

We define u as the *unanimity* WVG in which a coalition is only winning if it is the grand coalition $N = \{1, 2, ..., n\}$. Every player has veto power in u. We know that in u, all players are critical only in N and therefore have uniform Banzhaf Indices. Similarly we define s as the *singleton* weighted voting game in which every coalition is winning except the empty coalition.

Proposition IV.1. *In the meet of WVGs, the* unanimity *WVG acts as a zero and the* singleton *WVG acts as a unity.*

Proof: For a WVG (N, v) and a unanimity WVG (N, u), we notice that for any coalition c to be winning in $(N, v \wedge u)$, it must be winning in both (N, v) and (N, u). This the grand coalition is the only winning coalition. So $v \wedge u = u$.

For a WVG (N, v) and a singleton WVG (N, s), we notice that for any coalition c to be winning in $(N, v \wedge u)$ it just has to non-empty. So $v \wedge s = v$. ■

So for $v = v_1 \wedge ... \wedge v_m$, if $\exists j : v_j = u$, then $v = u$. This implies that even if player i is a dictator in one game of the MWVG, it does not mean it is a dictator in the MWVG. Moreover, even if a player is a dummy in all the games apart from the unanimity game v_j, then that player will have Banzhaf power of $1/n$.

Example IV.2. $v = v_1 \wedge v_2$ *where* $v_1 = [3; 4, 1, 1]$ *and* $v_2 = [3; 1, 1, 1]$. *Player* 1 *is a dictator in* v_1 *but it is not a dictator in* v.

B. Identifying players with dictator, veto or no powers

Assuming that we are given $\sum w_i = W$, we notice that a dictator in a single WVG is verifiable in $O(1)$ time. This is because we just need to check the two conditions of the player 1 being a dictator: $w_1 \geq q$ and $\sum_{2 \leq i \leq n} w_i < q$. Similarly, a dictator in a multiple m-WVG is verifiable in $O(m)$ time. We need to verify that $\{i\}$ wins in the MWVG and that $N \setminus \{i\}$ loses in the MWVG.

Moreover, it is evident from the definition that in a WVG, a player i has a veto power, if and only if $N \setminus \{i\}$ is a losing coalition. Thus, we can check whether a player has veto power in $O(1)$ time. One can extend this idea to MWVs. We present an algorithm to compute players with veto power in a MWVG. The algorithm has time complexity $O(mn)$.

Algorithm 1 VetoPlayersInMWVG

Input: m multiple weighted voting game (MWVG), $(N, v_1 \wedge ... \wedge v_m)$ where the games (N, v_t) are the weighted voting games $[q^t; w_1^t, ...w_n^t]$ for $1 \leq t \leq m$.
Output: vetoplayerset

vetoplayerset $\leftarrow \{\}$
for $i = 1$ to n **do**
    isvetoplayer $\leftarrow$ **false**
    for $j = 1$ to m **do**
        if $w^j(N) - w_i^j < q^j$ **then**
            isvetoplayer $\leftarrow$ **true**
        end if
    end for
    if isvetoplayer **then**
        vetoplayerset $\leftarrow$ vetoplayerset $\cup \{i\}$
    end if
end for
return vetoplayerset

Unlike, dictators and veto players, it is not easy to identify dummies in MWVGs. In fact, Matsui et al. [15] show that it is even NP-hard to identify players with zero powers or players with same powers in a single WVG.

C. Inherited properties of constituent games

Proposition IV.3. *For MWVG, $v = v_1 \wedge ... \wedge v_m$:*

1) $\forall i$: *player 1 is a dictator in $v_i \implies$ player 1 is a dictator in v*
2) $\forall j$: *player i is a dummy in $v_j \implies$ player i is a dummy in v*
3) $\exists j$: *player i has veto power in $v_j \implies$ player i has veto power in v*
4) $\forall j$: *v_j is proper $\implies v$ is proper.*

Proof:

1) Let player 1 be a dictator in v_i for all $i = 1, \ldots m$. Thus $\forall i$, $1 \leq i \leq m$, $w_1^i \geq q$ and $\sum_{2 \leq j \leq m} w_j < q$. This means that $\{1\}$ is winning in v and $\{2, ..., n\}$ are losing in v
2) We know that $\forall j$, $C_i(v_j) = \{\}$. Then by definition, $C_i(v) = \{\}$.
3) If for some $t = 1, \ldots, m$, $N \setminus \{i\} \notin W(v_t)$, $N \setminus \{i\} \notin W(v)$.
4) Since all WVGs v_ts are proper, $1 \leq t \leq m$, if $v_t(S) = 1$ then $v_t(\overline{S}) = 0$ If $v(S) = 1$, then by definition $v_t(S) = 1$, for $1 \leq t \leq m$. Then $v_i(\overline{S}) = 0$ for all t which implies that $v_t(\overline{S}) = 0$.

■

Counter-Examples IV.4. *The converses for the previous proposition do not hold:*

1) $v = v_1 \wedge v_2$ *where* $v_1 = [4; 5, 1, 1]$ *and* $v_2 = [2; 5, 1, 1]$. *Although player* 1 *is a dictator in* v*, it is not a dictator in* v_2.
Moreover, even if there is WVG v_i *in which player 1 does not have the biggest weight, it can still be the dictator:* $v = v_1 \wedge v_2$ *where* $v_1 = [2; 5, 1]$ *and* $v_2 = [2; 2, 3]$. *Player* 1 *is a dictator in* v.
2) *Let* $v = v_1 \wedge v_2$ *where* $v_1 = [5; 3, 2, 1]$ *and* $v_2 = [5; 3, 2, 2]$. *Player* 3 *is a dummy in* v *but not a dummy in* v_2.
In fact a player can be a dummy in v *even if he is not a dummy in any of the games: Let* $v = v_1 \wedge v_2$ *where* $v_1 = [7; 4, 3, 3, 1]$ *and* $v_2 = [8; 7, 3, 3, 1]$. *Player* 4 *is a not a dummy in* v_1 *and* v_2 *but a dummy in* v.
3) *Let* $v = v_1 \wedge v_2$ *where* $v_1 = [5; 3, 2, 1]$ *and* $v_2 = [6; 5, 2, 1]$. *Player* 2 *has veto power in* v *but does not have veto power in* v_2
4) *Let* $v = v_1 \wedge u$ *where* $v_1 = [5; 4, 3, 2, 2]$ *and* $u = [4; 1, 1, 1, 1]$. *We see that although* v_1 *is not proper,* v *is proper.*

Proposition IV.5. *For MWVG,* $v = v_1 \wedge ... \wedge v_m$*, if* $\exists i$ *: player 1 is a dictator in* v_i*, then player 1 has veto power in* v.

Proof: If player 1 is a dictator in v_i, he is in every winning coalition of v_i. Therefore for any coalition c which is winning in v, if the dictator opts out of c, c loses in v_i and therefore loses in v. ∎

V. Computation of voting power

A. Complexity

Proposition V.1. *The problem of computing the Banzhaf indices of players or even identifying dummies in a multiple weighted voting game is NP-Hard.*

Proof: Matsui and Matsui [15] show that it is NP-Hard to compute the Banzhaf indices of players or identifying dummies in a single weighted voting game. show that it is NP-Hard to compute the Banzhaf indices of players or identifying dummies in a single weighted voting game. Their proof is by a polynomial time reduction of the Partition problem.

Let V be a set of WVGs, m, an integer and let f_m be a function, $f_m : V \mapsto V$ such that $f_m(v) = \bigwedge_m v$. We notice that a player i is critical in a coalition $S \subset N$ for WVG v if and only if it is critical in coalition S for $f_m(v)$. Therefore f_m is a polynomial time function which reduces an instance of a single WVG to an instance of a MWVG. ∎

Klinz and Woeginger [16] devised the fastest exact algorithm to compute Banzhaf indices in a WVG. In the algorithm, they applied a partitioning approach that dates back to Horowitz and Sahni [17]. The complexity of the algorithm is $O(n^2 2^{\frac{n}{2}})$. This partitioning approach is not suitable for MWVGs though. A useful website for voting power analysis is available at [18].

B. GF for MWVGs

The generating function method provides an efficient way of computing Banzhaf indices if the voting weights are integers [15]. Algaba et al. [19] outline a generating function method to find the Banzhaf indices of players in a multiple weighted majority game. Their algorithm m-BanzhafPower computes the Banzhaf index of the players in O(max$(m, n^2 c)$) time where c is the number of terms of $B(x_1, ..., x_m) = \prod_{j=1}^{n}(1 + x_1^{w_j^1} ... x_m^{w_j^m}) = \sum_{k_t=0, 1\leq t\leq m}^{w^t(N)} b_{k_1...k_m} {x_1}^{k_1} ... {x_m}^{k_m}$. The coefficient, $b_{k_1...k_m}$ of each term ${x_1}^{k_1} ... {x_m}^{k_m}$ is the number of coalitions such that $w^t(S) = k_t$ for t ranging from 1 to m.

One can make generating functions, $B_i(x_1, ..., x_m)$ for each player i by excluding its influence from the considered coalitions just like in the single WVG case. Therefore $B_i(x_1, ..., x_m) = B(x_1, ..., x_m)/((1 + x_1^{w_i^1} ... x_m^{w_i^m}))$. These generating functions can be encoded in the form of a coefficient array which gives a clear picture and make the computation of coefficients easier. We present the algorithm due to Algaba et al. with some modifications to avoid extra computations and also to compute the total number of winning coalitions:

Algorithm 2 VotingPowersOfMWVGs

Input:MWVG: $[q^t; w_1^t, ...w_n^t]$ for $1 \le t \le m$.
Output: Number of winning coalitions w and Banzhaf indices: $\{w, (\beta_1, \ldots, \beta_n)\}$.

```
B(x_1, ..., x_m) ← ∏_{j=1}^{n} (1 + x_1^{w_j^1} ... x_m^{w_j^m})
coeff = Coeff(B(x_1, ..., x_m))
For k_t from q^t to w^t(N), 1 ≤ t ≤ m,
w ← Sum(coef[k_1, ..., k_m])
for i = 1 to n do
    if i ≠ 1 and w_i^t = w_{i-1}^t for t = 1, ..., m then
        η_i ← η_{i-1}
    else
        B_i(x_1, ..., x_m) ← B(x_1, ..., x_m) / (1 + x_1^{w_i^1} ... x_m^{w_i^m})
        coeff_i = Coeff(B_i(x_1, ..., x_m))
        For k_t from q^t − w_i^t + 1 to w^t(N \ i) + 1, 1 ≤ t ≤ m,
        s_1^i ← Sum(coef_i[k_1, ..., k_m])
        For k_t from q^t + 1 to w^t(N \ i) + 1, 1 ≤ t ≤ m,
        s_2^i ← Sum(coef_i[k_1, ..., k_m])
        η_i ← s_1^i − s_2^i
    end if
end for
η ← ∑_{i=1}^{n} η_i
for i = 1 to n do
    β_i ← η_i / η
end for
return {w, (β_1, ..., β_n)}
```

For a bi-weighted voting game which is a meet of its respective games, the critical region, C_i for a player i is shaded in Figure 1. The corresponding figure for the join of games is provided in Figure 2.

We give an example of how Algaba et al. have utilized the generating functions to compute Banzhaf indices in MWVGs.

Example V.2. *Let* $v = v_1 \wedge v_2$ *where* $v_1 = [7; 5, 2, 1, 1]$ *and* $v_2 = [5; 3, 2, 1, 1]$. *The generating function of the overall game is* $B(x_1, x_2) = (1 + {x_1}^5{x_2}^3)(1 + {x_1}^2{x_2}^2)(1 + x_1x_2)^2$. $B(x_1, x_2)$ *can be encapsulated by a coefficient array (see Figure 3) which gives the coefficients of each term. Similarly the coefficient arrays of the generating function for each player can be computed (See Figures 4, 5 and 6). The shaded region signifies those coalitions in which the player is critical. The sum of the values in the shaded area then gives the Banzhaf value of each player.*

Therefore, number of swings of players 1, 2, 3 *and* 4 *are 5, 3, 1 and 1 respectively. Thus* $\beta_i = 1/2$, $\beta_i = 3/10$, $\beta_i = 1/10$ *and* $\beta_i = 1/10$.

C. Analysis and Improvements

The enumeration algorithm to compute Banzhaf indices of players in a MWVG has an exponential time complexity because of the need to compute and analyse each possible coalition. The generating

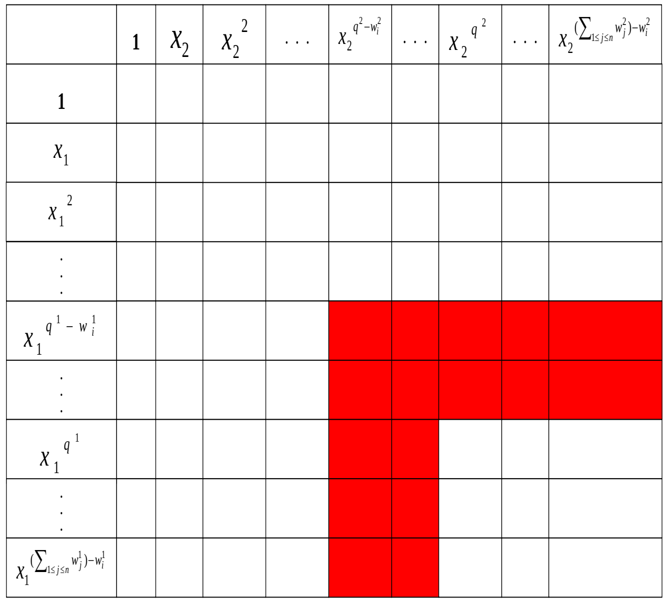

Fig. 1. Analysis of the effect of excluding a player i coalitions in a bi-weighted voting game which is a meet of its respective games.

function method can be more time efficient but involves more storage of data. It requires the computation of $B(x_1,...,x_m)$ and $B_i(x_1,...,x_m)$ for players, for $1 \le i \le n$. The storage requirements increase even more if $B(x_1,...,x_m)$ is encoded in a coefficient array. This makes the storage dependent on the sum of the weights in each component game.

Proposition V.3. *The space complexity of the generating function method to compute Banzhaf indices of players in a MWVG is $c + \sum_{1\le i\le n} c_i + k$ where c is the number of terms of $B(x_1,...,x_m)$ and c_i is the number of terms in $B_i(x_1,...,x_m)$. Moreover*

$$c + \sum_{1\le i\le n} c_i \le c + nc \le (n+1)(\prod_{1\le t\le m}(1+w^t(N)))$$

Proof: The proposition follows from the fact that the generating function method requires computation of $B(x_1,...,x_m)$, the generating function of the over all-game and $B_i(x_1,...,x_m)$, the generating function of each player i. ■

We can utilize the following observation to control the time and space required in computing Banzhaf indices via the generating function method where coefficient arrays are used:

Proposition V.4. *The power indices of players in weighted voting game $v = [q; w_1,...,w_n]$ are the same as the power indices in the weighted voting game $\lambda v = [\lambda q; \lambda w_1,...,\lambda w_n]$*

Proof: The proof is trivial. We notice that the set of coalitions for which player i is critical is the same for both games v and λv. This means that the Banzhaf indices of players in both games are the same. ■

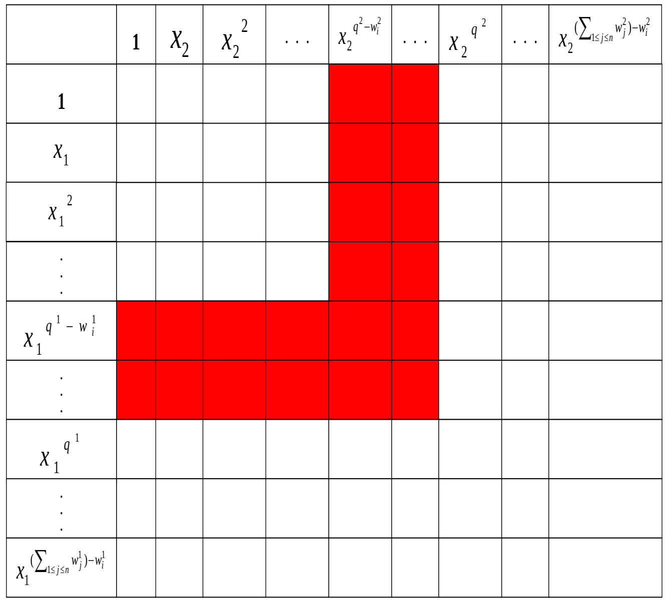

Fig. 2. Analysis of the effect of excluding a player i coalitions in a bi-weighted voting game which is a join of its respective games.

This scaling of the WVGs into WVGs with smaller weights keeps the properties of the WVG invariant. Moreover, we have identified players with same voting weight to avoid re-computation of their generating functions and their underlying coefficient arrays. Whereas the Mathematica programs to compute Banzhaf indices of multiple weighted voting games with 2 or 3 games are available, the appendix gives the Mathematica code to compute Banzhaf indices of an arbitrary number of players. In case the space complexity of the generating function method is high, the generating function and the corresponding coefficient array for each player can be computed, and then cleared after extracting the number of swings of that player.

VI. Conclusion and Future work

In this paper, we have examined the computational and combinatorial aspects of multiple weighted voting games especially with respect to voting power. The computational complexity of computing cooperative game theoretic solutions of weighted voting games has been examined by Elkind [20]. It will be interesting to analyse the algorithms and complexity to compute other game theoretic solutions of multiple weighted voting games.

Acknowledgment

Partial support for this research was provided by DIMAP (the Centre for Discrete Mathematics and its Applications). DIMAP is funded by the UK EPSRC. The first author would also like to thank the Pakistan National ICT R & D Fund for funding his research.

	1	x_2	x_2^2	x_2^3	x_2^4	x_2^5	x_2^6	x_2^7
1	1	0	0	0	0	0	0	0
x_1^1	0	2	0	0	0	0	0	0
x_1^2	0	0	2	0	0	0	0	0
x_1^3	0	0	0	2	0	0	0	0
x_1^4	0	0	0	0	1	0	0	0
x_1^5	0	0	0	1	0	0	0	0
x_1^6	0	0	0	0	2	0	0	0
x_1^7	0	0	0	0	0	2	0	0
x_1^8	0	0	0	0	0	0	2	0
x_1^9	0	0	0	0	0	0	0	1

Fig. 3. Coefficient array of $B(x_1, x_2)$.

REFERENCES

[1] A. Taylor and W. Zwicker, *Simple Games: Desirability Relations, Trading, Pseudoweightings*, first edition ed. New Jersey: Princeton University Press, 1999.

[2] L. Nordmann and H. Pham, "Weighted voting systems," *IEEE Transactions on Reliability*, vol. 48, no. 1, pp. 42–49, Mar 1999.

[3] D. Leech, "Voting power in the governance of the international monetary fund," *Annals of Operations Research*, vol. 109, no. 1, pp. 375–397, 2002.

[4] J. Alonso-Meijide, "Generating functions for coalition power indices: An application to the *IMF*," *Annals of Operations Research*, vol. 137, pp. 21–44, 2005.

[5] E. Algaba, J. M. Bilbao, and J. Fernandez, "The distribution of power in the *European Constitution*," *European Journal of Operational Research*, vol. 176, no. 3, pp. 1752–1755, 2007.

[6] J. Bilbao, J. Fernandez, N. Jimenez, and J. Lopez, "Voting power in the *European Union Enlargement*," *European Journal of Operational Research*, vol. 143, no. 1, pp. 181–196, 2002.

[7] A. Laruelle and M. Widgren, "Is the allocation of voting power among *EU* states fair?" *Public Choice*, vol. 94, no. 3-4, pp. 317–39, March 1998, available at http://ideas.repec.org/a/kap/pubcho/v94y1998i3-4p317-39.html.

[8] J.-E. Lane and R. Maeland, "Constitutional analysis: The power index approach," *European Journal of Political Research*, vol. 37, pp. 31–56, 2000.

[9] D. Leech, "Designing the voting system for the council of the european union volume," *Public Choice*, vol. 113, no. 3, pp. 437–464, 1962.

[10] D. S. Felsenthal and M. Machover, "Analysis of *QM* rules in the draft constitution for *Europe* proposed by the *European Convention*, 2003," *Social Choice and Welfare*, vol. 23, no. 1, pp. 1–20, 08 2004.

[11] G. Arcaini and G. Gambarelli, "Algorithm for automatic computation of the power variations in share tradings," *Calcolo*, vol. 23, no. 1, pp. 13–19, January 1986.

[12] G. Gambarelli, "Power indices for political and financial decision making: A review," *Annals of Operations Research*, vol. 51, pp. 1572–9338, 1994.

[13] V. G. Deineko and G. J. Woeginger, "On the dimension of simple monotonic games." *European Journal of Operational Research*, vol. 170, no. 1, pp. 315–318, 2006.

[14] J. Freixas and M. A. Puente, "A note about games-composition dimension," *Discrete Appl. Math.*, vol. 113, no. 2-3, pp. 265–273, 2001.

	1	x_2	x_2^2	x_2^3	x_2^4
1	1	0	0	0	0
x_1^1	0	2	0	0	0
x_1^2	0	0	2	0	0
x_1^3	0	0	0	2	0
x_1^4	0	0	0	0	1

Fig. 4. Coefficient array of $B_1(x_1, x_2)$.

[15] T. Matsui and Y. Matsui, "A survey of algorithms for calculating power indices of weighted majority games," *Journal of the Operations Research Society of Japan*, vol. 43, no. 7186, 2000, available at http://citeseer.ist.psu.edu/matsui00survey.html.

[16] B. Klinz and G. J. Woeginger, "Faster algorithms for computing power indices in weighted voting games," *Mathematical Social Sciences*, vol. 49, no. 1, pp. 111–116, January 2005, available at http://ideas.repec.org/a/eee/matsoc/v49y2005i1p111-116.html.

[17] E. Horowitz and S. Sahni, "Computing partitions with applications to the knapsack problem," *J. ACM*, vol. 21, no. 2, pp. 277–292, 1974.

[18] D. Leech, "Voting power algorithms website," *http://www.warwick.ac.uk/∼ecaae/*, 2007.

[19] E. Algaba, J. M. Bilbao, J. R. Fernandez Garcia, and J. J. Lopez, "Computing power indices in weighted multiple majority games," *Mathematical Social Sciences*, vol. 46, no. 1, pp. 63–80, 2003.

[20] E. Elkind, L. Goldberg, P. Goldberg, and M. Wooldbridge, "Computational complexity of weighted threshold games," *AAAI-07 (Twenty-Second National Conference on Artificial Intelligence)*, 2007.

	1	x_2	x_2^2	x_2^3	x_2^4	x_2^5
1	1	0	0	0	0	0
x_1^1	0	2	0	0	0	0
x_1^2	0	0	1	0	0	0
x_1^3	0	0	1	0	0	0
x_1^4	0	0	0	0	0	0
x_1^5	0	0	0	1	0	0
x_1^6	0	0	0	0	2	0
x_1^7	0	0	0	0	0	1

Fig. 5. Coefficient array of $B_2(x_1, x_2)$.

	1	x_2	x_2^2	x_2^3	x_2^4	x_2^5	x_2^6
1	1	0	0	0	0	0	0
x_1^1	0	1	0	0	0	0	0
x_1^2	0	0	1	0	0	0	0
x_1^3	0	0	0	1	0	0	0
x_1^4	0	0	0	0	0	0	0
x_1^5	0	0	0	1	0	0	0
x_1^6	0	0	0	0	1	0	0
x_1^7	0	0	0	0	0	1	0
x_1^8	0	0	0	0	0	0	1

Fig. 6. Coefficient array of $B_3(x_1, x_2)$ and $B_4(x_1, x_2)$.

Appendix

Mathematica Code to compute Banzhaf Indices of players in a Multiple Weighted Voting Game

In[44]:=
```
(*:Mathematica Version:5.2, Package Version:1.10 *)
(*:Name:Compute_Banzhaf _Indices _of _MWVG *)
(*:Authors:Haris Aziz (haris.aziz@warwick.ac.uk) *)
(*:Summary: The program takes as input a multiple
    weighted voting game with integer weights and quotas. It uses the
    generating functions to compute the Banzhaf index of every player*)
(*:References: Computing power indices in weighted multiple majority games
    by E.Algaba, Mathematical Social Sciences 46 (2003) pages 63-80.*)

w = {{5, 2, 1, 1}, {3, 2, 1, 1}}; q = {{7}, {5}};
Print["weights: ", MatrixForm[w]]; Print["quotas: ", MatrixForm[q]];
m = Part[Dimensions[w], 1]; Print["There are ", m, " weighted voting games"]
n = Part[Dimensions[w], 2]; Print["There are ", n, " players"];

Array[symmwithprevious, n]; symmwithprevious[1] = False;
For[i = 2, i < n + 1, i ++, symmwithprevious[i] = True;];
For[i = 2, i < n + 1, i ++ ,
  For[j = 1, j < m + 1, j ++, If[w[[j, i]] != w[[j, i - 1]], symmwithprevious[i] = False;,]]];
For[i = 1, i < n + 1, i ++, If[ symmwithprevious[i],
   Print["Player ", i, " has same weights as player ", i - 1],
   Print["Player ", i, " does not have same weights as player ", i - 1] ]];

Bfunction = Product[1 + Product[x[i] ^w[[i, j]], {i, 1, m}], {j, 1, n}];
longBfunction = Expand[Bfunction]; Print["Bfunction = ", Bfunction];
Print[Array[x, m]]; maincoefmatrix = CoefficientList[longBfunction, Array[x, m]];
Print["CoefficientMatrix for the main GF is ", MatrixForm[maincoefmatrix]];

For[j = 1, j < n + 1, j ++ , b[j] = Bfunction / (1 + Product[x[i] ^w[[i, j]], {i, 1, m}]);
 longb[j] = Expand[b[j]]; Print["Generating Function of player ", j, "=", b[j]] ]

kk = {}; For[j = 1, j < m + 1, j ++, kk = Append[kk, {q[[j, 1]] + 1, Total[w[[j]]] + 1}]];
winningmatrix = Take[maincoefmatrix, Part[kk, 1], Part[kk, 2]];
numofwinningcoalitons = Total[winningmatrix, m];
Array[x, m]; Array[coefmatrix, m];

coefmatrix[1] = CoefficientList[longb[1], Array[x, m]];
Print["Coefficient Matrix of player", 1, " =", MatrixForm[coefmatrix[1]]];
For[j = 2, j < n + 1, j ++ , If[symmwithprevious[j], coefmatrix[j] = coefmatrix[j - 1],
   coefmatrix[j] = CoefficientList[longb[j], Array[x, m]];];
  Print["Coefficient Matrix of player", j, " =", MatrixForm[coefmatrix[j]]]];

d = Table[0, {m}, { n}];
For[t = 1, t < m + 1, t ++, For[i = 1, i < n + 1, i ++, d[[t, i]] = q[[t, 1]] - w[[t, i]]]];
e = Table[0, {m}, {n}];
For[t = 1, t < m + 1, t ++,
  For[i = 1, i < n + 1, i ++, e[[t, i]] = Total[w[[t]]] - w[[t, i]]]];

For[i = 1, i < n + 1, i ++, ll[i] = {};];
For[i = 1, i < n + 1, i ++, For[t = 1, t < m + 1, t ++,
  ll[i] = Append[ll[i], {d[[t, i]] + 1, Part[Dimensions[coefmatrix[i]], t]}];]]
```

```
Print ("Computing Small1 matrices");
small1[1] = Take[coefmatrix[1], Part[ll[1], 1], Part[ll[1], 2] ];
Print["small1[", 1, "] = ", MatrixForm[small1[1]]]
For[i = 2,  i < n + 1, i ++, If[symmwithprevious[i], small1[i] = small1[i - 1],
   small1[i] = Take[coefmatrix[i], Part[ll[i], 1], Part[ll[i], 2] ]];
  Print["small1[", i, "] = ", MatrixForm[small1[i]]]];

For[i = 1, i < n + 1, i ++,
  sum1[i] = Total[small1[i], Infinity]; Print["sum1[", i, "] = ",  sum1[i]]];
g = Table[0, {m}, {n}];
For[t = 1, t < m + 1, t ++,
  For[i = 1, i < n + 1, i ++, g[[t, i]] = Total[w[[t]]] - w[[t, i]] + 1;]];

mm[1] = {}; For[i = 1, i < n + 1, i ++,  mm[i] = {};] ;

Array[errorcheck, n];
For[z = 1, z < n + 1, z ++, errorcheck[z] = 0;];
For[i = 1, i < n + 1, i ++,
  For[t = 1,  t < m + 1, t ++,  mm[i] = Append[mm[i], {q[[t, 1]] + 1, g[[t, i]]}];
   If[q[[t, 1]] + 1 > g[[t, i]], errorcheck[i] = 1;,];] ];

For[i = 1,  i < n + 1, i ++, If[errorcheck[i] == 1, small2[i] = {},
   small2[i] = Take[coefmatrix[i], Part[mm[i], 1], Part[mm[i], 2] ]];
  Print["small2[", i, "] = ", MatrixForm[small2[i]]]];
For[i = 1, i < n + 1, i ++, If[small2[i] == {}, sum2[i] = 0,
   sum2[i] = Total[small2[i], Infinity]]; Print["sum2[", i, "] = ",  sum2[i]]];

totalswings = 0;
For[i = 1, i < n + 1, i ++, swings[i] = sum1[i] - sum2[i];
  totalswings = totalswings + swings[i]; Print["swings[", i, "] = ", swings[i]]];
For[i = 1, i < n + 1, i ++, banzhafindex[i] = swings[i] / totalswings;
  Print["Banzhaf Index of player", i, " is ",  banzhafindex[i]]];

vetoplayerlist = {};
For[i = 1, i < n + 1, i ++, isvetoplayer = False; For[j = 1, j < m + 1,
  j ++, If[(Total[w[[j]]] - w[[j, i]]) < q[[j, 1]], isvetoplayer = True;, ] ];
 If[isvetoplayer, vetoplayerlist = Append[vetoplayerlist, i];
  Print[i, " has veto powers"], Print[i, " does not have veto powers"]]]

Print["Number of winning coalitions = ", numofwinningcoalitons];
```

weights: $\begin{pmatrix} 5 & 2 & 1 & 1 \\ 3 & 2 & 1 & 1 \end{pmatrix}$

quotas: $\begin{pmatrix} 7 \\ 5 \end{pmatrix}$

There are 2 weighted voting games

There are 4 players

Player 1 does not have same weights as player 0

Player 2 does not have same weights as player 1

Player 3 does not have same weights as player 2

Player 4 has same weights as player 3

Bfunction = $(1+x[1]\,x[2])^2\,(1+x[1]^2\,x[2]^2)\,(1+x[1]^5\,x[2]^3)$

{x[1], x[2]}

CoefficientMatrix for the main GF is $\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

Generating Function of player 1= $(1+x[1]\,x[2])^2\,(1+x[1]^2\,x[2]^2)$

Generating Function of player 2= $(1+x[1]\,x[2])^2\,(1+x[1]^5\,x[2]^3)$

Generating Function of player 3= $(1+x[1]\,x[2])\,(1+x[1]^2\,x[2]^2)\,(1+x[1]^5\,x[2]^3)$

Generating Function of player 4= $(1+x[1]\,x[2])\,(1+x[1]^2\,x[2]^2)\,(1+x[1]^5\,x[2]^3)$

Coefficient Matrix of player1 = $\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

Coefficient Matrix of player2 = $\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

Coefficient Matrix of player3 = $\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

Coefficient Matrix of player4 = $\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

small1[1] = $\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

small1[2] = $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

small1[3] = $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

small1[4] = $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

sum1[1] = 5

sum1[2] = 4

sum1[3] = 3

sum1[4] = 3

small2[1] = {}

small2[2] = (1)

small2[3] = $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

small2[4] = $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

sum2[1] = 0

sum2[2] = 1

sum2[3] = 2

sum2[4] = 2

swings[1] = 5

swings[2] = 3

swings[3] = 1

swings[4] = 1

Banzhaf Index of player1 is $\frac{1}{2}$

Banzhaf Index of player2 is $\frac{3}{10}$

Banzhaf Index of player3 is $\frac{1}{10}$

Banzhaf Index of player4 is $\frac{1}{10}$

1 has veto powers

2 does not have veto powers

3 does not have veto powers

```
4 does not have veto powers

Number of winning coalitions = 5
```