Annals of Operations Research 109, 143–158, 2002

Generating Functions for Computing the Myerson Value

J.R. FERNÁNDEZ, E. ALGABA, J.M. BILBAO *, A. JIMÉNEZ, N. JIMÉNEZ and J.J. LÓPEZ mbilbao@us.es
Matemática Aplicada II, Escuela Superior de Ingenieros, Camino de los Descubrimientos s/n, 41092 Sevilla, Spain

Abstract. The complexity of a computational problem is the order of computational resources which are necessary and sufficient to solve the problem. The algorithm complexity is the cost of a particular algorithm. We say that a problem has polynomial complexity if its computational complexity is a polynomial in the measure of input size. We introduce polynomial time algorithms based in generating functions for computing the Myerson value in weighted voting games restricted by a tree. Moreover, we apply the new generating algorithm for computing the Myerson value in the Council of Ministers of the European Union restricted by a communication structure.

Keywords: complexity, Myerson value, weighted voting games

AMS subject classification: 91A12

1. The Myerson value for voting games

In general, it is difficult to define the idea of power, but for the special case of voting power there are mathematical power indices that have been used. The first such power index was proposed by Shapley and Shubik [15] who apply the Shapley value [14] to the case of simple games. Another concept for measuring voting power was introduced by Banzhaf [1], a lawyer, whose work has appeared mainly in law journals, and whose index has been used in arguments in various legal proceedings.

A *simple game* is a function $v : 2^N \to \{0, 1\}$, such that $v(N) = 1$ and v is nondecreasing, i.e., $v(S) \leqslant v(T)$ whenever $S \subseteq T \subseteq N$. A coalition is *winning* if $v(S) = 1$, and *losing* if $v(S) = 0$. The collection of all winning coalitions is denoted by $\mathcal{W}$. We introduce a class of games called *weighted voting games*. The symbol $[q;\; w_1, \ldots, w_n]$ will be used, where the quota q and the weights $w_1, \ldots, w_n$ are positive integers with $0 < q \leqslant \sum_{i=1}^{n} w_i$. Here there are n players, w_i is the number of votes of player i, and q is the quota needed for a coalition to win. Then, the above symbol represents the simple game $v : 2^N \to \{0, 1\}$ defined for all $S \subseteq N$ by

$$v(S) = \begin{cases} 1 & \text{if } w(S) \geqslant q, \\ 0 & \text{if } w(S) < q, \end{cases}$$

where $w(S) = \sum_{i \in S} w_i$.

* http://www.esi2.us.es/~mbilbao/.

Definition 1.1. The Shapley–Shubik index for the simple game (N, v) is the vector $\Phi(N, v) = (\Phi_1(N, v), \dots, \Phi_n(N, v))$, given by

$$\begin{aligned}\Phi_i(N, v) &= \sum_{\{S \subseteq N:\ i \in S\}} \frac{(s-1)!(n-s)!}{n!} \big(v(S) - v\big(S \setminus \{i\}\big)\big) \\ &= \sum_{\{S \in \mathcal{W}:\ S \setminus \{i\} \notin \mathcal{W}\}} \frac{(s-1)!(n-s)!}{n!}.\end{aligned}$$

We introduce games in which the cooperation among the players is partial. Several models of partial cooperation have been proposed, among which are those derived from *communication situations* as introduced by Myerson [12] and analyzed by Owen [13]. Given a graph $G = (N, E)$, let us consider the collection

$$\mathcal{F} = \big\{S \subseteq N:\ (S, E(S)) \text{ is a connected subgraph of } G\big\}.$$

The sets belonging to $\mathcal{F}$ are called *feasible coalitions*. For any $S \subseteq N$, maximal feasible subsets of S are called *components* of S. The *graph-restricted game* $(N, v^{\mathcal{F}})$ is defined, for all $S \subseteq N$, by

$$v^{\mathcal{F}}(S) = \sum_{T \in \Pi(S)} v(T),$$

where $\Pi(S)$ is the collection of the components of S.

Definition 1.2. The Myerson value for player i is given by $\Phi_i(N, v^{\mathcal{F}})$, for all $i \in N$.

A *n-tree* $G = (N, E)$ is a connected, acyclic graph on $|N| = n$ vertices. A *j-subtree* S of G is a set of j vertices of G which induce a connected subgraph of G. A subgraph B of a graph G is a *block* of G if either B is a bridge or else it is a maximal 2-connected subgraph of G. A graph G is a *block graph* if every block is a complete graph. If G is an acyclic graph or a tree, then G is a block graph.

Convex geometries are a combinatorial abstraction of convex sets introduced by Edelman and Jamison [5].

Definition 1.3. The set system $\mathcal{F} \subseteq 2^N$ is a convex geometry over N if it satisfies the properties:

(C1) $\emptyset \in \mathcal{F}$,

(C2) $\mathcal{F}$ is closed under intersection,

(C3) If $S \in \mathcal{F}$ and $S \neq N$, then there exists $i \in N \setminus S$ such that $S \cup \{i\} \in \mathcal{F}$.

Note that property (C3) implies $N \in \mathcal{F}$. Edelman and Jamison [5] showed that the graph $G = (N, E)$ is a connected block graph if and only if the collection $\mathcal{F}$ of subsets of N which induce connected subgraphs is a convex geometry. An element i

of a feasible set $S \in \mathcal{F}$ is an *extreme point* of S if $S \setminus \{i\} \in \mathcal{F}$. The set of extreme points of S is denoted by $\mathrm{ex}(S)$. If $\mathcal{F}$ is a convex geometry and $S \in \mathcal{F}$, then the interval $[S^-, S] = \{C \in \mathcal{F}\colon\ S^- \subseteq C \subseteq S\}$ is a Boolean algebra, where $S^- = S \setminus \mathrm{ex}(S)$. Thus $[S^-, S]$ is isomorphic to $2^{\mathrm{ex}(S)}$. We also consider the interval $[T, T^+]$, where $T \in \mathcal{F}$ and the set $T^+ = \{i \in N\colon\ T \cup \{i\} \in \mathcal{F}\}$. Bilbao [2] proved the following formulas for the Myerson value.

Theorem 1.4. Let (N, v) be a game and let $G = (N, E)$ be a connected block graph. If $\mathcal{F}$ is the collection of subsets of N which induce connected subgraphs, then the Myerson value for any player i is given by

$$\begin{aligned}\Phi_i\big(N, v^{\mathcal{F}}\big) = &\sum_{T \in \mathcal{F}_i^+} \frac{(t-1)!(t^+ - t)!}{t^+!}\big(v(T) - v\big(T \setminus \{i\}\big)\big)\\ &+ \sum_{T \in \mathcal{F}_i \setminus \mathcal{F}_i^+} \frac{(t-1)!(t^+ - t)!}{t^+!} v(T) - \sum_{T \in \mathcal{F}_i^\star} \frac{(t)!(t^+ - t - 1)!}{t^+!} v(T),\end{aligned}$$

where $t = |T|,\ t^+ = |T^+|$ and

$$\begin{aligned}&\mathcal{F}_i = \{T \in \mathcal{F}\colon\ i \in T\}, \qquad \mathcal{F}_i^+ = \big\{T \in \mathcal{F}\colon\ i \in \mathrm{ex}(T),\ \big(T \setminus \{i\}\big)^+ = T^+\big\},\\ &\mathcal{F}_i^\star = \big\{T \in \mathcal{F}\colon\ i \notin T,\ T \cup \{i\} \in \mathcal{F},\ T^+ \neq \big(T \cup \{i\}\big)^+\big\}.\end{aligned}$$

Theorem 1.5. Let (N, v) be a game such that $v(i) = 0$ for all $i \in N$. Then the Myerson value for any player $i \in \mathrm{ex}(N)$ satisfy

$$\Phi_i\big(N, v^{\mathcal{F}}\big) = \sum_{T \in \mathcal{F}} \frac{(t-1)!(t^+ - t)!}{t^+!}\big(v(T) - v\big(T \setminus \{i\}\big)\big),$$

where $t = |T|$ and $t^+ = |T^+|$.

We can now obtain direct formulas for the Myerson value of weighted voting games restricted by stars.

Theorem 1.6. Let $[q; w_1, \ldots, w_n]$ be a weighted voting game with $w_i < q$ for all $i \in N$, and let $K_{1,n-1}$ be a star on n vertices. If $i \in \mathrm{ex}(N)$ then

$$\Phi_i\big(N, v^{\mathcal{F}}\big) = \sum_{j=2}^{n} \frac{(j-1)!(n-j)!}{n!} \left(\sum_{k=q}^{q-1+w_i} c_{kj}^i\right),$$

where c_{kj}^i is the number of j-subtrees T of $K_{1,n-1}$ which contain i and satisfy $w(T) = k$.

Proof. We may suppose that player 1 is the center of star $K_{1,n-1}$. Then $\mathrm{ex}(N) = \{2, \dots, n\}$, and, for all $T \in \mathcal{F}$ such that $|T| \geqslant 2$, we infer that $1 \in T$ and $T^+ = N$. By Theorem 1.5,

$$\Phi_i\big(N, v^{\mathcal{F}}\big) = \sum_{\{T\in\mathcal{F}:\ |T|\geqslant 2\}} \frac{(t-1)!(t^+-t)!}{t^+!}\big(v(T) - v\big(T\setminus\{i\}\big)\big)$$
$$= \sum_{\{T\in\mathcal{F}:\ |T|\geqslant 2\}} \frac{(t-1)!(n-t)!}{n!}\big(v(T) - v\big(T\setminus\{i\}\big)\big),$$

for all $i \in \mathrm{ex}(N)$. Since $v(T) - v(T \setminus \{i\}) = 1$ if and only if $w(T) \in [q, q + w_i - 1]$, we have

$$\Phi_i(N, v^{\mathcal{F}}) = \sum_{j=2}^{n} \frac{(j-1)!(n-j)!}{n!} \left(\sum_{k=q}^{q-1+w_i} c_{kj}^i \right),$$

where c_{kj}^i is the number of j-subtrees T of $K_{1,n-1}$ which contain i and satisfy $w(T) = k$. □

Theorem 1.7. Let $[q; w_1, \dots, w_n]$ be a weighted voting game with $w_i < q$ for all $i \in N$, and let $K_{1,n-1}$ be a star on n vertices. If player 1 is the center of $K_{1,n-1}$ then

$$\Phi_1\big(N, v^{\mathcal{F}}\big) = \sum_{j=2}^{n} \frac{(j-1)!(n-j)!}{n!} \left(\sum_{k=q}^{w(N)} c_{kj}^1 \right),$$

where c_{kj}^1 is the number of j-subtrees T of $K_{1,n-1}$ which contain 1 and satisfy $w(S) = k$.

Proof. Let player 1 be the center of star $K_{1,n-1}$. Since $v(i) = 0$ for all $i \in N$, it follows from Theorem 1.4 that

$$\Phi_1\big(N, v^{\mathcal{F}}\big) = \sum_{\{T\in\mathcal{F}_1^+:\ |T|\geqslant 2\}} \frac{(t-1)!(t^+-t)!}{t^+!}\big(v(T) - v\big(T\setminus\{1\}\big)\big)$$
$$+ \sum_{\{T\in\mathcal{F}_1\setminus\mathcal{F}_1^+:\ |T|\geqslant 2\}} \frac{(t-1)!(t^+-t)!}{t^+!} v(T) - \sum_{\{T\in\mathcal{F}_1^\star:\ |T|\geqslant 2\}} \frac{(t)!(t^+-t-1)!}{t^+!} v(T).$$

By definition

$$\big\{T \in \mathcal{F}_1^+:\ |T| \geqslant 2\big\} = \big\{T \in \mathcal{F}:\ 1 \in \mathrm{ex}(T),\ \big(T\setminus\{1\}\big)^+ = T^+,\ |T| \geqslant 2\big\}$$
$$= \big\{T \in \mathcal{F}:\ \big(T\setminus\{1\}\big)^+ = T^+,\ |T| = 2\big\}.$$

Since $T = (T \setminus \{1\})^+ \neq T^+ = N$ for every edge T of the star on $n \geqslant 3$, the above set is empty. Assume that

$$\big\{T \in \mathcal{F}:\ 1 \notin T,\ T \cup \{1\} \in \mathcal{F},\ T^+ \neq \big(T \cup \{1\}\big)^+,\ |T| \geqslant 2\big\} \neq \emptyset.$$

Since any $T \in \mathcal{F}$ with $|T| \geqslant 2$ contains the center 1, we obtain a contradiction. Therefore, we obtain

$$\begin{aligned}\Phi_1\big(N, v^{\mathcal{F}}\big) &= \sum_{\{T\in\mathcal{F}_1:\ |T|\geqslant 2\}} \frac{(t-1)!(n-t)!}{n!} v(T) \\ &= \sum_{\{T\in\mathcal{F}:\ |T|\geqslant 2\}} \frac{(t-1)!(n-t)!}{n!} v(T) \\ &= \sum_{j=2}^{n} \frac{(j-1)!(n-j)!}{n!} \left(\sum_{k=q}^{w(N)} c_{kj}^{1} \right),\end{aligned}$$

where c_{kj}^1 is the number of j-subtrees T of $K_{1,n-1}$ which contain 1 and satisfy $w(T) = k$. □

2. Algorithm complexity

The *time complexity* function $f : \mathbb{Z}_+ \to \mathbb{Z}_+$ of an algorithm is the maximal number $f(n)$ of iterations of a universal Turing machine makes before halting, taken over all inputs of size n. We say that an algorithm has *space complexity* at most $f(n)$ if it can be computed by a Turing machine with space demand (cells and tapes) at most $f(n)$.

Let f and g be functions from $\mathbb{Z}_+$ to $\mathbb{Z}_+$. We write $f(n) = \mathrm{O}(g(n))$, in words f *is of the order of* g, if there are positive integers c and n_0 such that $f(n) \leqslant cg(n)$ for all $n \geqslant n_0$. We write $f(n) = \Omega(g(n))$ if the opposite happens, that is, $g(n) = \mathrm{O}(f(n))$. If f and g have exactly the same rate of growth, then we write $f(n) = \Theta(g(n))$. For instance, if $p(n)$ is a polynomial of degree d, then $p(n) = \Theta(n^d)$. The above $\mathrm{O}\Omega\Theta$-notation was proposed by Knuth [10]. For a more detailed exposition, see the books of Bilbao [3, chapter 4], and Gács and Lovász [6].

The classical procedures for computing the power indices are based in the enumeration of all coalitions. Thus, if the input size of the problem is n, then the function which measures the worst case running time for computing the Shapley–Shubik index is in $\mathrm{O}(n2^n)$. Moreover, to obtain the restricted game $v^{\mathcal{F}}$ we need to compute the set of the components $\Pi(S)$ of every subset $S \subseteq N$. Then it is necessary to consider all the feasible subsets of S and hence the time complexity is $\mathrm{O}(t)$ where

$$t = \sum_{s=0}^{n} \binom{n}{s} 2^s = 3^n.$$

The time complexity of the formulas given by theorems 1.4 and 1.5 is polynomial in the cardinality $|\mathcal{F}|$ and so it is $\mathrm{O}(2^{n-1})$ for games restricted by stars. In the next section, we use generating functions to obtain *pseudo polynomial* algorithms, i.e., polynomial in n and c, for computing the Myerson value. In general, we cannot hope for a polynomial time complexity for the generating functions algorithms, but in many problems we obtain polynomial time whenever the number of coefficients and the maximum of the weights are polynomial in n.

The programs of our language contain only *assignments* and a **for**-*loop* construct. We use the symbol $\leftarrow$ for assignments, for example, $g(x) \leftarrow 1$ denotes setting the value of $g(x)$ to 1. A **for**-*loop* for calculate $\sum_{i\in I} a_i$, can be defined by

$h \leftarrow 0$

for $i \in I$ **do**

  $h \leftarrow h + a_i$

endfor

3. Generating functions for the Whitney numbers

Let $G = (N, E)$ be a tree on $|N| = n$ vertices. The number $A_j(G)$ of j-subtrees is the j-th *Whitney number* of the lattice of subtrees (see Stanley [17]). For the path P and star $K_{1,n-1}$ on n vertices, the Whitney numbers satisfy, for any tree G, the following property

$$A_j(P) = n - j + 1 \leqslant A_j(G) \leqslant \binom{n-1}{j-1} = A_j(K_{1,n-1}).$$

Jamison [8,9] introduced the generating function

$$\Phi_G(z) = \sum_{j=1}^{n} A_j(G) z^j,$$

for the Whitney numbers of a tree. First, we consider a local or rooted version. If p is any vertex of G, let

$$\varphi_G(p; z) = \sum_{j=1}^{n} \alpha_j(G; p) z^j,$$

where $\alpha_j(G; p)$ is the number of j-subtrees of G which contain p. Let G_p be the tree G rooted at p and let

$$D(i; G_p) = \{k \in N\colon\ i \text{ lies on the path from } k \text{ to } p\},$$

be the set of descendents of the vertex i in G_p. To simplify the notation, the function $\varphi_G(i \mid p; z)$ will denote $\varphi_D(i; z)$, where $D = D(i; G_p)$.

Theorem 3.1. For any vertex p in a tree G, we have:

(a) $\varphi(p; z) = z \prod(1+\varphi(i|p; z))$, where the product runs over all neighbors i of p in G.

(b) $\Phi_G(z) = \sum \varphi(i|p; z)$, where the sum runs over all vertices i of G.

Proof. Theorem 2.1 by Jamison [8]. □

Example 3.2. Let us consider the tree $G = (N, E)$, where

$$N = \{1, 2, 3, 4, 5\} \quad \text{and} \quad E = \big\{\{1, 2\}, \{1, 3\}, \{2, 4\}, \{2, 5\}\big\}.$$

The algorithm for computing the Whitney numbers of G, is showed in figure 1. First, root G at vertex 1 and label all pendant vertices, except 1, with z. Next, we recursively obtain

$$\begin{aligned} \varphi(3|1; z) &= \varphi(4|1; z) = \varphi(5|1; z) = z, \\ \varphi(2|1; z) &= z\big(1 + \varphi(4|1; z)\big)\big(1 + \varphi(5|1; z)\big) = z(1 + z)^2, \\ \varphi(1; z) &= z\big(1 + \varphi(3|1; z)\big)\big(1 + \varphi(2|1; z)\big) \\ &= z(1 + z)\big(1 + z(1 + z)^2\big). \end{aligned}$$

Finally, theorem 3.1(b) implies that

$$\begin{aligned} \Phi_G(z) &= \sum_{i=1}^{5} \varphi(i|1; z) = z(1 + z)\big(1 + z(1 + z)^2\big) + z(1 + z)^2 + 3z \\ &= 5z + 4z^2 + 4z^3 + 3z^4 + z^5. \end{aligned}$$

The collection $\mathcal{F}$ of subsets of N which induce subtrees of G is

$$\begin{aligned} \mathcal{F} = \big\{&\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{1, 2\}, \{1, 3\}, \{2, 4\}, \{2, 5\}, \{1, 2, 3\}, \\ &\{1, 2, 4\}, \{1, 2, 5\}, \{2, 4, 5\}, \{1, 2, 3, 4\}, \{1, 2, 3, 5\}, \{1, 2, 4, 5\}, N\big\}. \end{aligned}$$

Note that the Whitney numbers are

$$A_1(G) = 5, \quad A_2(G) = 4, \quad A_3(G) = 4, \quad A_4(G) = 3, \quad A_5(G) = 1.$$

We implement the algorithm *Jamison*(p) of theorem 3.1 as follows.

Input: G rooted at p

$\varphi(p; z) \leftarrow z$

for i neighbor of p **do**

$\varphi(p; z) \leftarrow \varphi(p; z)\big(1 + \mathit{Jamison}(i)\big)$

endfor

Output: $\varphi(p; z)$

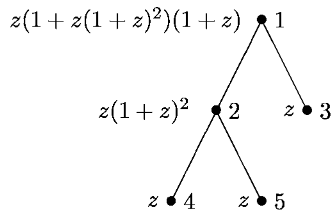

Figure 1.

Theorem 3.3. For any vertex p in a tree G, we have:

(a) Computing the generating function $\varphi(p; z)$ by the Jamison algorithm, requires a time $\mathrm{O}(r^h)$.

(b) Computing the coefficients of $\varphi(p; z)$ requires a time $\mathrm{O}(r^h)$, where r is the maximum degree of G and h is the length of the longest path between two vertices (diameter) of G.

Proof. (a) Let G_p be the tree G rooted at p. Suppose that the number of neighbors (degree) for any vertex is r. The execution time of *Jamison* for a vertex p with diameter h is

$$t(h) = 2 + 2r + rt(h-1),$$

where $t(h-1)$ is the execution time of *Jamison* for the neighbors of p. This recursive formula implies that

$$\begin{aligned} t(h) &= 2 + 2r + rt(h-1) \\ &= 2 + 2r + r\bigl(2 + 2r + rt(h-2)\bigr) \\ &= 2 + 4r + 2r^2 + r^2\bigl(2 + 2r + rt(h-3)\bigr) \\ &= 2 + 4r + 4r^2 + 2r^3 + r^3 t(h-3) \\ &\;\vdots \\ &= 2 + 4r + 4r^2 + 4r^3 + \cdots + 2r^k + r^k t(h-k). \end{aligned}$$

Since $t(0) = 2$, we have

$$\begin{aligned} t(h) &= 2 + 4r + 4r^2 + 4r^3 + \cdots + 2r^h + r^h t(0) \\ &= 2 + 4r + 4r^2 + 4r^3 + \cdots + 4r^h. \end{aligned}$$

Therefore, $\mathrm{O}(t(h)) = \mathrm{O}(r^h)$.

(b) We analyze the time complexity of the expansion for

$$\varphi(p; z) = z \prod \bigl(1 + \varphi(i|p; z)\bigr).$$

We may suppose that the degree of any vertex is r and the length of the path from p to any pendant vertex is h. Choose a vertex l such that its neighbors l_j are the pendant vertices of G. Thus,

$$\varphi(l; z) = z \prod_{j=1}^{r} \bigl(1 + \varphi(l_j; z)\bigr) = z(1+z)^r.$$

The complexity of the expansion for $(1+z)^r$ is $\mathrm{O}(t(1)) = \mathrm{O}(r^2)$. We can apply this reasoning to the vertices of G and get

$$t(h) = \begin{cases} r^2 & \text{if } h = 1, \\ rt(h-1) & \text{if } h \geqslant 2. \end{cases}$$

Therefore $t(h) = rt(h-1) = r^2 t(h-2) = \cdots = r^k t(h-k)$. Since $t(1) = r^2$, we have $t(h) = r^{h-1} t(1) = r^{h+1}$. We conclude that $\mathrm{O}(t(h)) = \mathrm{O}(r^h)$. □

4. Generating functions for weighted voting games

We now present generating functions for computing the Myerson value in weighted voting games $[q; w_1, \ldots, w_n]$, restricted by a tree $G = (N, E)$. The generating function of the numbers $A_{kj}(G)$ of feasible coalitions S of j players and weight $w(S) = k$, is given by

$$\Psi_G(x, z) = \sum_{k \geqslant 0} \sum_{1 \leqslant j \leqslant n} A_{kj}(G) x^k z^j.$$

If p is any vertex of G, the rooted version is given by

$$\psi_G(p; x, z) = \sum_{k \geqslant 0} \sum_{1 \leqslant j \leqslant n} c_{kj}(G; p) x^k z^j,$$

where $c_{kj}(G; p)$ is the number of j-subtrees of G which contain p such that $w(S) = k$.

Theorem 4.1. Let $[q; w_1, \ldots, w_n]$ be a weighted voting game. For any vertex p in a tree G, we have:

(a) $\psi(p; x, z) = x^{w_p} z \prod (1 + \psi(i|p; x, z))$, where the product runs over all neighbors i of p in G.

(b) $\Psi_G(x, z) = \sum \psi(i|p; x, z)$, where the sum runs over all vertices i of G.

Proof. (a) We can use theorem 3.1(a) to obtain

$$\begin{aligned}\psi(p; x, z) &= x^{w_p} z \prod \bigl(1 + \psi(i|p; x, z)\bigr) \\ &= \sum_{\{S \in \mathcal{F}:\ p \in S\}} \prod_{i \in S} x^{w_i} z \\ &= \sum_{\{S \in \mathcal{F}:\ p \in S\}} x^{w(S)} z^{|S|} \\ &= \sum_{k \geqslant 0} \sum_{1 \leqslant j \leqslant n} c_{kj}(G; p) x^k z^j,\end{aligned}$$

where $\mathcal{F}$ is the collection of j-subtrees of G.

(b) This follows from theorem 3.1(b). □

Theorem 4.2. Let $[q; w_1, \ldots, w_n]$ be a weighted voting game. For any vertex p in a tree G, we have:

(a) Computing the generating function $\psi(p; x, z)$ by the algorithm given in theorem 4.1, requires a time $\mathrm{O}(r^h)$.

(b) Computing the coefficients of $\psi(p; x, z)$ requires a time $O(r^h)$, where r is the maximum degree of G and h is the length of the longest path between two vertices (diameter) of G.

Proof. The statements follow from theorem 3.3. □

We implement the algorithm *MyersonStar*$(q, \textit{weights})$ of theorem 4.1 as follows.

Input: $\{q; w_1, \ldots, w_n\}$ and G

for $l \in \{2, \ldots, n\}$ **do**

(assig 1) $\psi(l; x, z) \leftarrow x^{w_l} z \prod \big(1 + \psi(i|l; x, z)\big) \Big\{ \psi(l; x, z) = \sum_{k \geqslant 0} \sum_{1 \leqslant j \leqslant n} c_{kj}^l x^k z^j \Big\}$

(assig 2) $\Phi_l\big(N, v^{\mathcal{F}}\big) \leftarrow \sum_{j=2}^{n} \frac{(j-1)!(n-j)!}{n!} \left(\sum_{k=q}^{q-1+w_l} c_{kj}^l \right)$

endfor

(assig 3) $\psi(1; x, z) \leftarrow x^{w_1} z \prod_{j=2}^{n} \big(1 + x^{w_j} z\big) \Big\{ \psi(1; x, z) = \sum_{k \geqslant 0} \sum_{1 \leqslant j \leqslant n} c_{kj}^1 x^k z^j \Big\}$

(assig 4) $\Phi_1\big(N, v^{\mathcal{F}}\big) \leftarrow \sum_{j=2}^{n} \frac{(j-1)!(n-j)!}{n!} \left(\sum_{k=q}^{w(N)} c_{kj}^1 \right)$

Output: $\big\{\Phi_1\big(N, v^{\mathcal{F}}\big), \ldots, \Phi_n\big(N, v^{\mathcal{F}}\big)\big\}$

Theorem 4.3. Let $[q; w_1, \ldots, w_n]$ be a weighted voting game and let G be a tree. The algorithm of the generating function $\Psi_G(x, z)$ computes the Myerson value of the n players in time $O(n^2 c)$, where c is the number of nonzero coefficients of $\Psi_G(x, z)$.

Proof. The execution time $f(n)$ of the *MyersonStar* algorithm satisfies

$$O\big(f(n)\big) = O\big(t(\textit{loop}) + t(\textit{assignment}\ 3) + t(\textit{assignment}\ 4)\big),$$
$$O\big(t(\textit{loop})\big) = O\big((n-1)\big(t(\textit{assignment}\ 1) + t(\textit{assignment}\ 2)\big)\big).$$

Using theorem 4.2, we infer that $O(t(\textit{assignment}\ 1)) = O(r^h)$, and since $r = n - 1$ and $h = 2$, this implies that $O(t(\textit{assignment}\ 1)) = O(n^2)$. Moreover,

$$O\big(t(\textit{assignment}\ 2)\big) = O\big((n-1)\big(t(\textit{coef}) + t\big(\textit{sum}\ 1\big)\big)\big).$$

We observe that

$$O\big(t(\textit{coef})\big) = O(n) \quad \text{and} \quad O\big(t(\textit{sum}\ 1)\big) = O(c),$$

and since $c \geqslant n$, we have

$$\mathrm{O}\big(t\,(assignment\,2)\big) = \mathrm{O}\big(n^2 + nc\big) = \mathrm{O}(nc).$$

We use the above properties to estimate the execution time

$$\begin{aligned}\mathrm{O}\big(t\,(loop)\big) &= \mathrm{O}\big((n-1)t\,(assignment\,1) + nt(assignment\,2)\big)\\ &= \mathrm{O}\big(n^3 + n^2c\big) = \mathrm{O}\big(n^2c\big).\end{aligned}$$

Similarly, $\mathrm{O}(t\,(assignment\,3)) = \mathrm{O}(r^h) = \mathrm{O}(n)$. Furthermore,

$$\begin{aligned}\mathrm{O}\big(t\,(assignment\,4)\big) &= \mathrm{O}\big(n\big(t\,(coef) + t\,(sum\,2)\big)\big)\\ &= \mathrm{O}\big(n^2 + nc\big) = \mathrm{O}(nc).\end{aligned}$$

We deduce that $\mathrm{O}(f(n)) = \mathrm{O}(\max(n^2c, n, nc)) = \mathrm{O}(n^2c)$. □

5. The Myerson value in the European Union

The package *DiscreteMath′Combinatorica′* extends the computer system Mathematica to Combinatorics and Graph Theory. The best guide for this package is the book by Skiena [16]. The package *Cooperat* included in Carter [4] presents tools for solving cooperative games, including solution concepts as the core, the Shapley value and the nucleolus. Herne and Nurmi [7], Widgrén [18] and Lane and Maeland [11] have studied the power indices (Shapley–Shubik and Banzhaf) in the European Union Council. The game of the countries in the EU Council is defined by

$$\begin{aligned}N &= \{\text{GE, UK, FR, IT, SP, NE, GR, BE, PO, SW, AU, DE, FI, IR, LU}\},\\ v &= [q;\ 10, 10, 10, 10, 8, 5, 5, 5, 5, 4, 4, 3, 3, 3, 2],\end{aligned}$$

where $q = 62$ or $q = 65$. We present a notebook of the system Mathematica by Wolfram [19] that will be used for computing the Myerson value of the voting power in the Council of Ministers of the European Union. Moreover, this game is restricted by a communication structure given by the star $K_{1,14}$ such that the degree of Germany is 14 and the other countries are degree 1. Then the collection of connected coalitions is

$$\mathcal{F} = \{S \subseteq N\colon \text{ Germany} \in S \text{ or } |S| = 1\}.$$

In[1]:=

```
<<DiscreteMath'Combinatorica'
```

In[2]:=

```
<<Cooperat'Cooperat'
```

In[3]:=

```
Clear[n,graph,weights,edges,q,f];
```

In[4]:=

```
n=15; graph=Star[n];
countries={''UK'',''FR'',''IT'',''SP'',''NE'',''GR'',
      ''BE'',''PO'',''SW'',''AU'',''DE'',''FI'',''IR'',
      ''LU'',''GE''};
```

In[5]:=

```
ShowLabeledGraph[graph,countries];
```

Out[5]=

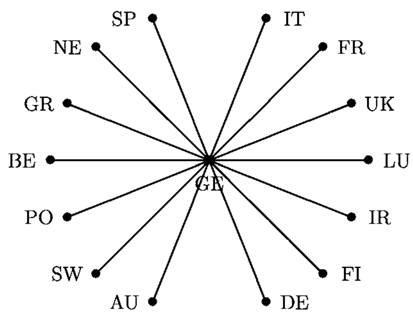

Figure 2. The European Union star.

Notice that player 15 is the center of the star.

In[6]:=

```
weights={10,10,10,8,5,5,5,5,4,4,3,3,3,2,10};
```

In[7]:=

```
edges=Flatten[Prepend[{n},#]]&/@Partition[Range[n-1],1];
```

We define the generating function f of the *weighted subtrees* of the star, by using an extension of the algorithm implemented by Jamison [8,9].

In[8]:=

```
f[{i_},weights_List,edges_List]:=
Module[{arcs,A,adjacents,factor,value},
arcs=edges;
A=Select[arcs,(MemberQ[#,i])&];
adjacents=Complement[Flatten[A],{i}];
arcs=Complement[arcs,A];
factor=(1+f[#,weights,arcs])& /@ Partition[adjacents,1];
value=z*x^(weights[[i]])*Times @@ factor;
factor=. & /@ Partition[adjacents,1];
Return[value]];
```

From theorems 1.6, 1.7 and the generating function f, we obtain the following formulas of the Myerson value for voting games restricted by stars.

In[9]:=

```
MyersonStarPlayer[i_,weights_List,q_]:=
Module[{g,coeff,m,gg},
g=Expand[f[{i},weights,edges]];
coeff=CoefficientList[g,x];
m=Exponent[g,x]+1;
gg=Apply[Plus,coeff[[Range[q+1,Min[m,q+weights[[i]]]]]]];
Sum[Coefficient[gg,z^j]*(j-1)!*(n-j)!,{j,2,n}]/n!]
```

In[10]:=

```
MyersonStarCenter[weights_List,q_]:=
Module[{g,coeff,m,gg},
g=Expand[f[{n},weights,edges]];
coeff=CoefficientList[g,x];
m=Exponent[g,x]+1;
gg=Apply[Plus,coeff[[Range[q+1,m]]]];
Sum[Coefficient[gg,z^j]*(j-1)!*(n-j)!,{j,2,n}]/n!]
```

In[11]:=

```
MyersonStarExtreme[weights_List,q_]:=Module[{value},
value=Table[MyersonStarPlayer[i,weights,q],{i,n-1}];
Return[value]]
```

In[12]:=

```
MyersonStar[weights_List,q_]:=
Prepend[MyersonStarExtreme[weights,q],
MyersonStarCenter[weights,q]]
```

In[13]:=

```
Timing[MyersonStar[weights,62]]//N
```

Out[13]=

```
{0.94 Second,{0.345987,0.0882839,0.0882839,0.0882839,
 0.0708236,0.0406288,0.0406288,0.0406288,0.0406288,
 0.0251637,0.0251637,0.0251637,0.0328172,0.0328172,
 0.0146964}}
```

In[14]:=

```
MyersonStar[weights,65]//N
```

Out[14]=

```
{0.308997,0.0964258,0.0964258,0.0964258,0.0739816,
 0.0439616,0.0439616,0.0439616,0.0439616,0.0306582,
 0.0306582,0.025197,0.025197,0.025197,0.0149906}
```

In[15]:=

```
MyersonStar[weights,70]//N
```

Out[15]=

```
{0.248601,0.105744,0.105744,0.105744,0.084016,0.0476024,
 0.0476024,0.0476024,0.0476024,0.0326174,0.0326174,
 0.0270729,0.0270729,0.0270729,0.0132867}
```

Table 1 and figure 3 contain the population, votes, and Myerson values for the voting game in the European Union Council restricted by the German-star. Moreover, the majority voting rule is assumed to be 62, 65 and 70 out of 87 votes in the European Union Council.

6. Concluding remarks

In the present paper we have considered weighted voting games restricted by star trees and generating functions for computing the Myerson value of these games. We have shown that there exist polynomial time algorithms based in generating functions for computing the Myerson value. We have also obtained that the complexity bound for the new algorithm is the number c of nonzero coefficients of the generating function

Table 1

Country	Population	Votes	Myerson 62	Myerson 65	Myerson 70
Germany	0.219	0.115	0.346	0.309	0.249
U. Kingdom	0.157	0.115	0.088	0.096	0.106
France	0.156	0.115	0.088	0.096	0.106
Italy	0.155	0.115	0.088	0.096	0.106
Spain	0.106	0.092	0.071	0.074	0.084
Netherlands	0.041	0.058	0.041	0.044	0.048
Greece	0.028	0.058	0.041	0.044	0.048
Belgium	0.027	0.058	0.041	0.044	0.048
Portugal	0.027	0.058	0.041	0.044	0.048
Sweden	0.024	0.046	0.033	0.031	0.033
Austria	0.022	0.046	0.033	0.031	0.033
Denmark	0.014	0.035	0.025	0.025	0.027
Finland	0.014	0.035	0.025	0.025	0.027
Ireland	0.010	0.035	0.025	0.025	0.027
Luxembourg	0.001	0.023	0.015	0.015	0.013

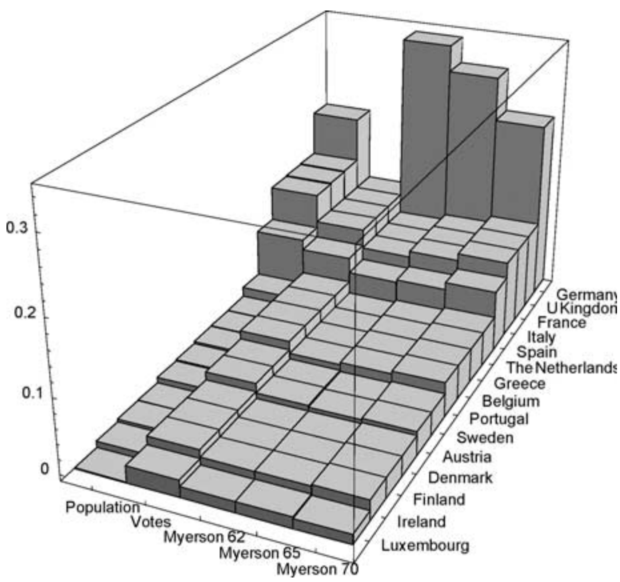

Figure 3. The EU star game.

obtained by the Jamison procedure. Finally, the next table shows the *time in seconds* for the new generating algorithm and the direct formulas (theorems 1.4 and 1.5) applied to the European Union star game.

Generating algorithm	Direct formulas
0.94	4339.27

References

[1] J.F. Banzhaf, Weighted voting doesn't work: a mathematical analysis, Rutgers Law Review 19 (1965) 317–343.

[2] J.M. Bilbao, Values and potential of games with cooperation structure, International Journal of Game Theory 27 (1998) 131–145.

[3] J.M. Bilbao, *Cooperative Games on Combinatorial Structures* (Kluwer Academic, Boston, 2000).

[4] M. Carter, Cooperative games, in: *Economic and Financial Modeling with Mathematica*, ed. H.R. Varian (Springer, Berlin, 1993) pp. 167–191.

[5] P.H. Edelman and R.E. Jamison, The theory of convex geometries, Geometriae Dedicata 19 (1985) 247–270.

[6] P. Gács and L. Lovász, Complexity of algorithms, Lecture Notes, Yale University (1999). Available at http://www.esi2.us.es/~mbilbao/pdffiles/complex.pdf.

[7] K. Herne and H. Nurmi, The distribution of a priori voting power in the EC Council of Ministers and the European Parliament, Scandinavian Political Studies 16 (1993) 269–284.

[8] R.E. Jamison, Alternating Whitney sums and matchings in trees, Part I, Discrete Mathematics 67 (1987) 177–189.

[9] R.E. Jamison, Alternating Whitney sums and matchings in trees, Part II, Discrete Mathematics 79 (1989/90) 177–189.

[10] D.E. Knuth, Big omicron and big omega and big theta, ACM SIGACT News 8 (1976) 18–24.

[11] J.E. Lane and R. Maeland, Voting power under the EU Constitution, Journal of Theoretical Politics 7 (1995) 223–230.

[12] R.B. Myerson, Graphs and cooperation in games, Mathematics of Operations Research 2 (1977) 225–229.

[13] G. Owen, Values of graph-restricted games, SIAM J. Algebraic and Discrete Methods 7 (1986) 210–220.

[14] L.S. Shapley, A value for n-person games, Annals of Mathematical Studies 28 (1953) 307–317.

[15] L.S. Shapley and M. Shubik, A method for evaluating the distribution of power in a committee system, American Political Science Review 48 (1954) 787–792.

[16] S. Skiena, *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica* (Addison-Wesley, Reading, MA, 1990).

[17] R.P. Stanley, *Enumerative Combinatorics*, Vol. I (Wadsworth, Monterey, CA, 1986).

[18] M. Widgrén, Voting power in the EC decision making and the consequences of two different enlargements, European Economic Review 38 (1994) 1153–1170.

[19] S. Wolfram, *Mathematica: A System for Doing Mathematics by Computer* (Addison-Wesley, Reading, MA, 1991).