Theoretical Computer Science 440–441 (2012) 52–59

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Choosing the best among peers

Jurek Czyzowicz [a,*], Leszek Gąsieniec [b], Andrzej Pelc [a]

[a] *Département d'informatique, Université du Québec en Outaouais, Gatineau, Québec J8X 3X7, Canada*
[b] *Department of Computer Science, University of Liverpool, Liverpool L69 3BX, UK*

ARTICLE INFO

Article history:
Received 30 March 2011
Received in revised form 11 November 2011
Accepted 3 April 2012
Communicated by R. Klasing

Keywords:
Weighted voting
Voting system
Vote manipulation
Directed graph
Ranking
Value function

ABSTRACT

A group of n peers, e.g., computer scientists, has to choose the best, i.e., the most competent among them. Each member of the group may vote for one other member, or abstain. Self-voting is not allowed. A voting graph is a directed graph in which an arc (u, v) means that u votes for v. While opinions may be subjective, resulting in various voting graphs, it is natural to assume that more competent peers are also, in general, more competent in evaluating competence of others. We capture this by proposing a voting system in which each member is assigned a positive integer *value* satisfying the following *strict support monotonicity* property: the value of x is larger than the value of y if and only if the sum of values of members voting for x is larger than the sum of values of members voting for y. Then we choose the member with the highest value, or if there are several such members, another election mechanism, e.g., using randomness, chooses one of them.

We show that for every voting graph there is a value function satisfying the strict support monotonicity property and that such a function can be computed in linear time. However, it turns out that this method of choosing the best among peers is vulnerable to vote manipulation: even one voter of very low value may change his/her vote so as to get the highest value. This is due to the possibility of loops (directed cycles) in the voting graph. Hence we slightly modify voting graphs by erasing all arcs that belong to some cycle. This modification results in a *pruned voting graph* which is always a rooted forest. We show that for all pruned voting graphs there are value functions giving a guarantee against manipulation. More precisely, we show a value function guaranteeing that no coalition of k members all of whose values are lower than those of $(1 - 1/(k + 1))n$ other members can manipulate their votes so that one of them gets the largest value. In particular, no single member from the lower half of the group is able to manipulate his/her vote to become elected. We also show that no better guarantee can be given for any value function satisfying the strict support monotonicity property.

1. Introduction

1.1. Preliminaries and statement of the problem

A group of n peers (e.g., computer scientists) has to choose the best (most competent) among them. Each member of the group may vote for one other member (self-voting is not allowed), or abstain. While opinions may be subjective, resulting in various voting graphs (directed graphs in which an arc (u, v) means that u votes for v), it is natural to assume that more

* Corresponding author. Tel.: +1 819 777 2278.
E-mail addresses: jurek@uqo.ca (J. Czyzowicz), L.A.Gasieniec@liverpool.ac.uk (L. Gąsieniec), pelc@uqo.ca (A. Pelc).

0304-3975/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2012.04.005

competent peers are also, in general, more competent in evaluating competence of others. Hence they should be given more voting weight. This is in sharp contrast to most political elections respecting the principle "one person – one vote". Weighted voting is commonly adopted, e.g., in the corporate context, where in the assembly of shareholders each member is given a voting weight proportional to the number of owned shares. In our situation, however, there are no such clear objective external indicators (peers do rarely agree on a common objective indicator of competence) and differences in voting weight must arise exclusively from the voting graph itself.

We capture this by proposing a voting system in which each member is assigned a positive integer *value* satisfying the following *strict support monotonicity* property: the value of x is larger than the value of y if and only if the sum of values of members voting for x is larger than the sum of values of members voting for y. Then we choose the member with the highest value, or if there are several such members, another election mechanism (e.g., random) chooses one of them.

More formally, we define a *voting graph* to be any directed graph, all of whose nodes have out-degree at most 1. If there exists a directed edge (arc) (x, y) in the graph, we say that x is a *predecessor* of y and that y is a *successor* of x. By $\Gamma^-(x)$ we denote the set of predecessors of x. Nodes represent members of the group and we will use terms "node" and "member" as synonyms. (Out-degree 0 means that the member abstained from voting.) Any such graph consists of connected components, each of which has the following form: it is either a directed rooted tree in which each arc goes from child to parent or a collection of such trees whose roots are on a common directed cycle (some trees may consist only of the root).

Given a voting graph G, a value function V is defined on the set of nodes of G and has positive integer values. For any node x define the *support* of x as $S(x) = \Sigma_{y\in\Gamma^-(x)}V(y)$. We assume that all value functions must satisfy the following *strict support monotonicity* property:

$$\text{(SSM)} \qquad V(x) < V(y) \text{ if and only if } S(x) < S(y).$$

This simple property captures both the assumption that competence value is attributed on the basis of peer support in a strictly monotonic way (more support is equivalent to a larger value) and that the weight of the vote of any node is its competence value (support is computed as the sum of values of nodes voting for a given node). Notice that strict support monotonicity implies that $V(x) = V(y)$ if and only if $S(x) = S(y)$.

Choosing the best (most competent) member of the group as the one with the largest value obviously depends on the particular value function and may differ for various such functions.

Example 1.1. Consider the voting graph with nodes a, b, c, d, e, f in which a, b, c, d vote for e which votes for f, and f abstains. Then both the value function V_1 such that $V_1(a) = V_1(b) = V_1(c) = V_1(d) = 1, V_1(e) = 3$ and $V_1(f) = 2$ and the value function V_2 such that $V_2(a) = V_2(b) = V_2(c) = V_2(d) = 1, V_1(e) = 5$ and $V_1(f) = 6$ satisfy SSM. However, with value function V_1, node e has the largest value and with value function V_2, node f has the largest value.

The non-uniqueness of value functions satisfying the SSM property, occurring for a given graph, leaves room for various ways of fine tuning that can result in different choices of the best among peers. This phenomenon, illustrated in the above example, holds for many voting graphs. Of course, in applications, the value functions, for any possible graph, have to be fixed before starting the voting process.

In order to use value functions to choose the best among peers, it is important to answer the following questions:

- Does there exist, for every voting graph, a value function satisfying the SSM property?
- If so, how difficult is it to compute?

We will show that the answer to the first question is positive and that such value functions can always be computed efficiently. However, an important drawback of the above designed voting system is its significant vulnerability to vote manipulation. By manipulation we mean such changes of votes of some members from a subset (coalition) of the group, all having low values (changes with respect to their conviction of who is the best) that give one or more of the coalition members the highest value, strictly larger than that of any member outside the coalition. This vulnerability allows a manipulation of votes by a coalition of weak members, resulting in the election of one of them. This is well illustrated in the following example.

Example 1.2. Consider a voting graph which is a directed line $(a_1, \ldots, a_n)$, where a_i votes for a_{i+1}, if $i < n$, and a_n abstains. It is easy to see that every value function satisfying the SSM property on such a line is strictly increasing. Consequently, node a_2 has the second lowest value. However, if this node manipulates its vote by changing it to a vote for a_1, the voting graph changes to a graph with two components: the directed line $(a_3, \ldots, a_n)$ and the loop (a_1, a_2) with these nodes pointing to one another. We now argue that for *any* value function V on the new graph, satisfying the SSM property, the values of a_1 and a_2 must be equal and strictly larger than the value of any other node. The first assertion is obvious by symmetry and the second can be justified as follows. Let $V(a_1) = V(a_2) = x$. By definition, $x \geq 1$. x must be larger than $V(a_3)$ because $S(a_3) = 0$ while $S(a_1) = x$. Suppose that $x = V(a_i)$ for some $i > 3$. Then $S(a_i) = V(a_{i-1}) < x$ and $S(a_2) = V(a_1) = x$, contradicting SSM. Finally, if $V(a_i) < x < V(a_{i+1})$ then $S(a_{i+1}) = V(a_i) < x = S(a_2)$, while $V(a_{i+1}) > x = V(a_2)$, again contradicting SSM. This shows that x must be larger than values of all nodes outside the loop and hence the coalition $\{a_1, a_2\}$

of the weakest members of the group can manipulate votes (actually only a_2 has to change her vote,) so that one of the members of this coalition gets elected.

Hence, although our election method using value functions can be applied to any voting graph, it is vulnerable in the presence of dishonest members of the group willing to manipulate their votes in order to promote a weak member as the best. This yields our third question:

- How to design voting systems robust against vote manipulations?

We have seen in the example that the manipulation involved creating a directed cycle of votes (which was not present in the original voting graph). Hence we propose the following adjustment of our voting system. Take any graph G of votes and define the *pruned graph* G^* obtained from G by deleting all edges that belong to any directed cycle. (Recall that there exists at most one such cycle in every connected component.) Any pruned graph is a directed forest in which each arc goes from a child to a parent. Now the value function is applied to the pruned graph and the rest of the election process remains the same as before. We will show that this simple adjustment makes our method much more robust: although some manipulation is still possible, very weak members cannot get elected.

In order to state our results we need a formal notion of guarantee. We will use the following definitions.

Consider a pruned graph and a value function V on it satisfying the SSM property. Let $0 < \alpha < 1$. A node x is *outside of the top α fraction* of all nodes if there are at least αn nodes with values larger than $V(x)$.

A *valuation* $\mathscr{V}$ is a function that to any pruned voting graph G assigns a value function $V = \mathscr{V}(G)$ defined on it, satisfying the SSM property. Fix a valuation $\mathscr{V}$. For a pruned graph G, a coalition C (which can be an arbitrary subset of nodes of G) is *threatening*, if some change of votes or abstention of some members of the coalition results in a graph $M(G)$, such that for the value function $V = \mathscr{V}(M(G)^*)$, at least one of the elements of C gets a value larger than the value of any member outside the coalition. Intuitively, a coalition is threatening, for a given valuation and given voting graph, if it can manipulate their votes so that, after pruning, one of the members of the coalition becomes elected, using the given valuation for the resulting pruned graph.

Let $0 < \alpha < 1$ and let k be a positive integer. A valuation $\mathscr{V}$ gives *guarantee* α against k-manipulation, if, for every pruned graph G, no coalition of k nodes containing only elements outside of the top α fraction is threatening. For example, guarantee 1/2 against 1-manipulation means that no single member having value in the lower half of the group can change her vote to be elected.

1.2. Our results

We first show that for any voting graph there exists a value function satisfying the SSM property. We also show that such a function can be computed in linear time for any voting graph. Then we turn our attention to the problem of robustness against vote manipulation by considering pruned voting graphs. For any positive integer k we show a valuation that gives guarantee $\alpha = 1 - \frac{1}{k+1}$ against k-manipulation. We also show that this is best possible: no valuation gives guarantee β against k-manipulation, for any $\beta < \alpha$.

1.3. Related work

The voting system proposed in this paper is close to the idea of the Pagerank Algorithm [6] used by Google to rank usefulness of web pages. Web pages play the role of peers and similarly as in our case, the voting weight of a page is its value. There are, however, important differences. First, in our case each node must have out-degree at most 1, while in the Pagerank Algorithm there is no such restriction. Second, in the Pagerank Algorithm, the sum of values of pages voting for a given one is divided by the number of outgoing links and an additional summand is added to account for pages with out-degree 0. Third and most important, the value obtained by the Pagerank Algorithm does not usually satisfy our SSM property. The algorithm is designed to calculate the probability that a given page is reached in a random walk starting arbitrarily and using at every page an outgoing link uniformly at random. While the Pagerank Algorithm is a concrete computing procedure, it has been proved in [1] that this page ranking is the only one satisfying a set of natural axioms. It should be stressed that these axioms are not related to our SSM property, which is not surprising, in view of the fact that the Pagerank Algorithm itself often produces values not satisfying this property.

The study and comparison of voting systems, both from the mathematical and the social perspective, has a long history going back to Borda and Condorcet in the 18th century (see, e.g., the recent book [9]). Most of this theory concentrates on situations when voting alternatives (the objects for which votes are cast) are distinct from voters. Hence voting weights, even when they are different, do not depend on decisions of other voters. Nevertheless, weighted voting systems with voting weights determined externally, have been thoroughly studied. The main focus in this domain is the study of voting power (see, e.g., [3]), defined as the ability of a group to change the outcome of a vote. This research on weighted voting goes back to classical studies [2,8].

Vote manipulation, defined as not reporting the voter's real preferences, has been also a subject of intense research (see, e.g., [10]). The classic Gibbard–Satterthwaite Theorem [4,7] says that any voting system which is not dictatorial and has at least three alternatives must sometimes give incentive to manipulation. Thus, some amount of manipulation is usually unavoidable.

2. Computing a valuation

In this section, we give a construction showing our main positive result for arbitrary voting graphs. First we define a value function satisfying the SSM property for the class of forests in whose component trees children are predecessors of parents. Let Φ_t be the following function.

$$\Phi_t(x) = t + \sum_{p \in \Gamma^-(x)} \Phi_t(p)$$

where t is any positive integer. Observe that $\Phi_t(x) = t \cdot \tau(x)$, where $\tau(x)$ is the number of nodes in the subtree rooted at x. The following fact is straightforward.

Lemma 2.1. *The value function Φ_t defined on a voting graph being a forest satisfies the SSM property.*

Now we show that the value function Φ_t may be extended to any voting graph (for appropriately chosen t), so that the SSM property is satisfied.

Theorem 2.2. *For any voting graph there exists a value function satisfying the SSM property. Such a function can be computed in time $O(n)$ for n-node voting graphs.*

Proof. Let G be a voting graph. As already observed, since each node of G has out-degree of at most one, every connected component of G is either a directed tree (with children being predecessors of their parent), or it is a directed cycle, such that each of its nodes may also have as predecessors roots of such directed trees. We construct the value function Φ, defined separately for each connected component of G, and then we prove that Φ satisfies the SSM property for G.

We assume that there are s connected components which are numbered $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_s$. Take any such component $\mathcal{C}_q$ and denote by $c_0, c_1, \ldots, c_{r-1}$ the consecutive nodes of the directed cycle of this component (this cycle is empty if the component is just a directed tree). Consider a node c_i having in-degree at least 2. Then c_i has at least one predecessor outside the cycle, which must be the root of some directed tree T. Define the value function Φ on every node x of T, so that $\Phi(x) = \Phi_{n^2}(x)$. We proceed similarly for all other trees, whose roots are predecessors of nodes $c_0, c_1, \ldots, c_{r-1}$.

To extend the value function Φ on the nodes of the cycle we first assign to each node c_i of the cycle an ordered triplet (q, f_i, ρ_i) defined in the following way. The value of q is the index of the component containing c_i. The value of f_i equals

$$f_i = \sum_{p \in \Gamma^-(x) \setminus c_j} \Phi(p)$$

where c_j is the predecessor of c_i in the cycle (i.e. $j = (i-1) \mod r$). In order to define ρ_i we denote by $f_i^{(r)}$ the r-tuple

$$f_i^{(r)} = (f_i, f_{(i-1) \mod r}, \ldots, f_{(i+1) \mod r})$$

and take the lexicographic ordering of $f_0^{(r)}, f_1^{(r)}, \ldots, f_{r-1}^{(r)}$. We set ρ_i to the rank of $f_i^{(r)}$ in this ordering (ranking starting at 0) assuming that the same r-tuples get the same rank (i.e. $f_i^{(r)} = f_j^{(r)} \implies \rho_i = \rho_j$); see Fig. 1. We now extend the value function to the nodes of the cycle as follows:

$$\Phi(c_i) = n^3 q + f_i/n + \rho_i.$$

Notice that, since each number f_i is a multiple of n^2, values of Φ are positive integers. It remains to prove that Φ satisfies the SSM property.

By Lemma 2.1 the SSM property is satisfied if both x and y do not belong to cycles. Suppose that exactly one of x, y, say $x = c_i$ belongs to a cycle of some component $\mathcal{C}_q$. For any pair of such nodes we have

$$\Phi(x) = n^3 q + f_i/n + \rho_i \geq n^3 > \Phi_{n^2}(y) = \Phi(y),$$

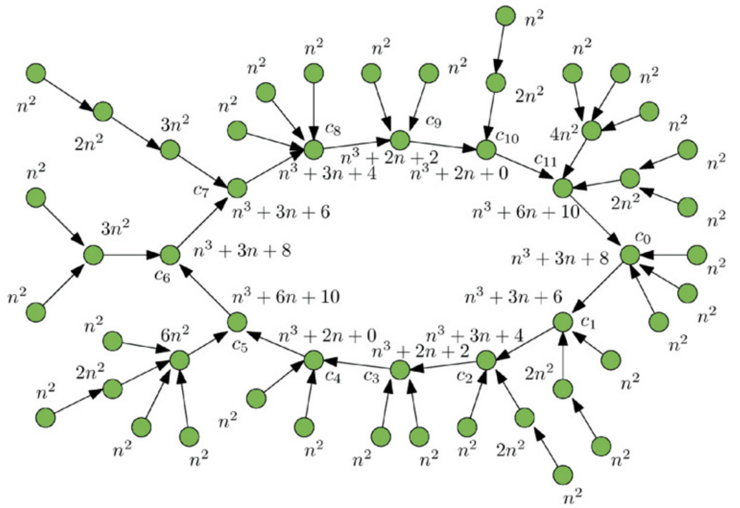
since a tree rooted at y contains at most $n-1$ nodes, so $\Phi_{n^2}(y) \leq n^2(n-1)$. This implies the SSM property for such pairs x, y, because $S(x) \geq n^3$ and $S(y) < n^3$, using the same argument.

It remains to prove that the SSM property is satisfied when both nodes x, y belong to cycles. Denote the triples assigned to nodes x and y by (q_x, f', ρ') and (q_y, f'', ρ''), respectively. First suppose that the nodes x, y belong, respectively, to cycles from different components $\mathcal{C}_{q_x}$ and $\mathcal{C}_{q_y}$ and assume, by symmetry, that $q_x > q_y$. Then

$$\begin{aligned}\Phi(x) - \Phi(y) &= (n^3 q_x + f'/n + \rho') - (n^3 q_y + f''/n + \rho'') \\ &= n^3(q_x - q_y) + (f' - f'')/n + (\rho' - \rho'') \geq n^3 - (n-1)n - (n-1) > 0,\end{aligned}$$

since $q_x \geq q_y - 1, f'' \leq (n-1)n^2$ and $\rho'' \geq n - 1$. On the other hand,

$$S(x) \geq n^3 q_x \geq n^3 q_y + n^3,$$



Node	Value	Support
c_0	n^3+3n+8	$n^3+3n^2+6n+10$
c_1	n^3+3n+6	n^3+3n^2+3n+8
c_2	n^3+3n+4	n^3+3n^2+3n+6
c_3	n^3+2n+2	n^3+2n^2+3n+4
c_4	n^3+2n+0	n^3+2n^2+2n+2
c_5	$n^3+6n+10$	n^3+6n^2+2n
c_6	n^3+3n+8	$n^3+3n^2+6n+10$
c_7	n^3+3n+6	n^3+3n^2+3n+8
c_8	n^3+3n+4	n^3+3n^2+3n+6
c_9	n^3+2n+2	n^3+2n^2+3n+4
c_{10}	n^3+2n+0	n^3+2n^2+2n+2
c_{11}	$n^3+6n+10$	n^3+6n^2+2n

Fig. 1. An example of a voting graph and its value function satisfying the SSM property. The values of all nodes are marked. The three terms of the cycle nodes' values correspond to the triples attached to them. We have $n = 51$.

since the cycle predecessor of x also belongs to the component $\mathcal{C}_{q_x}$. However, the support of y comes from its predecessor in the cycle having value $n^3q_y + n'n^2 + \rho$ and from all other predecessors (roots of trees), summing up to the value of $n''n^2$. Since $n' + n'' \le n - 1$ and $\rho \le n - 1$, we have

$$S(y) \le n^3q_y + (n' + n'')n^2 + (n-1) \le n^3q_y + n^2(n-1) + (n-1) < n^3q_y + n^3.$$

Therefore $S(x) > S(y)$.

Hence in the remainder of the proof we can assume that $q_x = q_y$, i.e. x and y are from the same cycle. First suppose the case $f' \neq f''$, i.e., by symmetry, we assume $f' > f''$. Then, since each f' and f'' is a multiple of n^2 we have $f' - f'' > n^2$. Therefore,

$$\begin{aligned}\Phi(x) - \Phi(y) &= (n^3q_x + f'/n + \rho') - (n^3q_y + f''/n + \rho'') \\ &= n^3(q_x - q_x) + (f' - f'')/n + (\rho' - \rho'')) \ge n(n-1) - (n-1) > 0.\end{aligned}$$

On the other hand, if the values of cycle predecessors of x and y are, respectively, $n^3q_x + f'_p + \rho'_p$ and $n^3q_y + f''_p + \rho''_p$, and $q_x = q_y$ we have

$$\begin{aligned}S(x) - S(y) &= (f' + n^3q_x + f'_p/n + \rho'_p) - (f'' + n^3q_y + f''_p/n + \rho''_p)) \\ &= (f' - f'') + (f'_p - f''_p)/n + (\rho'_p - \rho''_p) \ge n^2 - (n-2)n^2/n + n - 1 > 0\end{aligned}$$

since at most $(n-2)$ nodes (x and y excluded) may contribute to the value of f'. Again $S(x) > S(y)$.

To complete the proof we need to show that the SSM property is satisfied in the case when $q_x = q_y$ and $f' = f''$. First suppose that $\Phi(x) > \Phi(y)$. This implies that $\rho' > \rho''$, which means that for $x = c_i$ and $y = c_j$ the r-tuple $(f_i, f_{(i-1) \bmod r}, \ldots, f_{(i+1) \bmod r})$ is lexicographically larger than $(f_j, f_{(j-1) \bmod r}, \ldots, f_{(j+1) \bmod r})$. Let (q_x, f'_p, ρ'_p) and (q_x, f''_p, ρ''_p) be the triples assigned, respectively, to the nodes $c_{(i-1) \bmod r}, c_{(j-1) \bmod r}$, the respective predecessors of x, y in the cycle. Since $f_i = f' = f'' = f_j$ and $\rho' > \rho''$ we must have $\rho'_p > \rho''_p$ and $f'_p \ge f''_p$. Hence

$$\begin{aligned}S(x) - S(y) &= (f' + n^3q_x + f'_p/n + \rho'_p) - (f'' + n^3q_x + f''_p/n + \rho'_p) \\ &= (f'_p - f''_p) + (\rho'_p - \rho''_p) > 0.\end{aligned}$$

Now suppose that $S(x) > S(y)$. Since $f' = f''$, for the cycle predecessors of x, y we must have $\Phi(c_{(i-1) \bmod r}) > \Phi(c_{(j-1) \bmod r})$. On the other hand,

$$\begin{aligned}\Phi(c_{(i-1) \bmod r}) - \Phi(c_{(j-1) \bmod r}) &= (n^3q_x + f'_p/n + \rho'_p) - (n^3q_x + f''_p/n + \rho''_p) \\ &= (f'_p - f''_p)/n + (\rho'_p - \rho''_p).\end{aligned}$$

Since the first term of the last equation is a multiple of n and the second term is smaller than n this sum is positive when either $f'_p = f_{(i-1) \bmod r} > f_{(j-1) \bmod r} = f''_p$ or $f_{(i-1) \bmod r} = f_{(j-1) \bmod r}$ and $\rho'_p > \rho''_p$. In either case this implies that $\rho' > \rho''$. However, since $q_x = q_y$ and $f' = f''$ we have

$$\Phi(x) - \Phi(y) = (n^3q_x + f'/n + \rho') - (n^3q_x + f''/n + \rho') = \rho' - \rho'' > 0.$$

This concludes the proof that the value function Φ satisfies the SSM property.

We now observe that the computation of the value function for a given voting graph may be performed in linear time. The connected components, their cycle nodes as well as leaf nodes may be identified in $O(n)$ time by standard techniques. The values of all tree nodes may be computed in a bottom-up way. Then we compute the triples (q, f_i, ρ_i) assigned to each cycle node. The values of q and f_i are computed easily. In order to compute the value ρ_i it is sufficient to find the ordering of all rotations of a given sequence $(f_i, f_{(i-1) \mod r}, \dots, f_{(i+1) \mod r})$, which may be done in linear time using the approach from [5]. □

An example of a voting graph and its value function is given in Fig. 1. The values of all cycle nodes are sums of three terms, computed from the triples attached to the nodes. There is only one component, so $q = 1$ is the coefficient of all terms involving n^3. Observe that the order of values of the nodes corresponds to the order of their supports, implying the SSM property.

3. Robustness against vote manipulation

In this section, we study the problem of robustness against vote manipulation by considering pruned voting graphs. Our main result shows a valuation which gives optimal guarantee against k-manipulation, for any positive integer k.

Theorem 3.1. *Let k be a positive integer and let $\alpha = 1 - \frac{1}{k+1}$.*

1. *There exists a valuation that gives guarantee α against k-manipulation.*
2. *No valuation gives guarantee β against k-manipulation, for any $\beta < \alpha$.*

Proof. Recall that a coalition C is threatening if after a manipulation at least one of the members of C receives a value strictly larger than the value of any node outside of the coalition. Recall also that the pruned voting graph is a rooted forest that we name here T^*.

In order to improve the readability of the proof, we present it first for the specific case when $k = 1$, and only later for arbitrary values of k.

1. Consider a value function ϕ defined on rooted trees, where the value of each node v is the number of nodes in the tree $T(v)$ rooted in v. In other words, for any leaf u the value $\phi(u) = |T(u)| = 1$ and for any internal node w with m children $e_1, \dots, e_m$ we have $\phi(w) = |T(w)| = \phi(e_1) + \cdots + \phi(e_m) + 1$. We will refer later to ϕ as the (tree) *volume function*. Notice that $\phi(x) = \Phi_1(x)$, where Φ_t was defined in the previous section.

   1-manipulation. First consider a coalition C with a single member c. In this case C can be threatening only if node c can be elected after it manipulates its vote. Suppose that node c is outside of the top $\frac{1}{2}$-fraction of all nodes. Let $T^* = \{T_0, T_1, \dots, T_l\}$ be the forest before manipulation, where T_0 contains c. T_0 can be partitioned into two trees A and B, such that, A is the subtree of T_0 rooted in c, i.e., $A = T(c)$, and $B = T_0 \setminus A$ contains all the remaining nodes of T_0. The only manipulation operations available to c are:
   - *removing* support for (arc towards) the parent of c in T_0,
   - *granting* support to some node different from c in either of $A, B, T_1, \dots, T_{l-1}$ or T_l.

   Note that according to the definition of the volume function neither of these operations can increase the value of $\phi(c)$, since $T(c)$ can only be reduced in size.

   First assume that there is a tree T_i, for some $i \in \{1, \dots, l\}$, s.t., $|T_i| \geq |A|$. In other words, for the root r of T_i we have $\phi(r) \geq \phi(c)$. Since during the vote manipulation of c the value of $\phi(r)$ cannot decrease, the coalition $\{c\}$ is not threatening.

   Now assume that all trees $T_1, \dots, T_l$ contain only nodes with values strictly smaller than $\phi(c)$, i.e., $|A| > |T_i|$, for all $i = 1, \dots, l$. Since we assumed that the node c is outside of the top $\frac{1}{2}$ fraction, we obtain $|A| + |T_1| + \cdots + |T_l| \leq \frac{n}{2}$ and hence $|B| \geq \frac{n}{2}$. Thus in this case, independently of the vote manipulation by c, the root of B will have a value not smaller than c.

   k-manipulation. Now consider a larger coalition $C = \{c_1, \dots, c_k\}$. We show that the volume function gives guarantee $\alpha = 1 - \frac{1}{k+1}$ against k-manipulation.

   Let this time $T^* = \{T_0^1, \dots, T_0^{k'}, T_1, \dots, T_l\}$, for some $0 \leq k' \leq k$, be the structure of the forest before manipulation, where trees T_0^1 through $T_0^{k'}$, contain nodes from C. Let K be the number of nodes in all subtrees rooted at nodes from C. Observe that according to the definition of the volume function none of the nodes in C can get a value greater than K after a vote manipulation by nodes from C, and the value K can be met only if the sets of descendants of all nodes in C are disjoint.

   First assume that there is a tree T_i, for some $i \in \{1, \dots, l\}$, s.t., $|T_i| > K$. Since after any vote manipulation by nodes in C, all of these nodes have values of at most K, we conclude that the coalition C is not threatening in this case.

   Otherwise, let $\phi_C = \max_{c_i \in C} \phi(c_i)$. Note that $K \leq k\phi_C$. Suppose that all nodes in C are outside of the top α fraction of all nodes. We partition the set of nodes into two sets U and L with U containing all nodes with values greater than ϕ_C. Since $\alpha = 1 - \frac{1}{k+1}$, we get $|U| \geq n(1 - \frac{1}{k+1})$ and $|L| \leq n\frac{1}{k+1}$.

   Consider trees $T_1, \dots, T_l$. Let $(l - x)$ be the number of trees with at most ϕ_C nodes and let t_C be the total number of nodes in those trees. Note that all of these nodes must be in the set L. In each of the remaining x trees there are more than ϕ_C and at most K nodes. This means that in each of these trees at least ϕ_C nodes are in L and at most $K - \phi_C$ nodes are in U.

Using this observation we conclude that in trees $T_0^1, \ldots, T_0^{k'}$ there are at least $|U| - x(K - \phi_C)$ nodes from U. We also argue that $K \leq |L| - x \cdot \phi_C - t_C$, since at least $x \cdot \phi_C + t_C$ nodes from L reside in trees $T_1, \ldots, T_l$.

Now we show that at least one of the trees $T_0^1, \ldots, T_0^{k'}$ must still contain more than K nodes after any vote manipulation by the coalition C. Indeed, since the number of nodes from U in trees $T_0^1, \ldots, T_0^{k'}$ is at least $|U| - x(K - \phi_C)$ and the number of trees is k', using the pigeon hole principle we conclude that one of the trees $T_0^1, \ldots, T_0^{k'}$ has at least $\frac{|U|-x(K-\phi_C)}{k'}$ nodes from U. Since $|U| \geq n(1 - \frac{1}{k+1})$ and $k' \leq k$ we get

$$\frac{|U| - x(K - \phi_C)}{k'} \geq \frac{n(1 - \frac{1}{k+1}) - x(K - \phi_C)}{k}.$$

Recalling that $K \leq k\phi_C$ we get

$$\frac{n(1 - \frac{1}{k+1}) - x(K - \phi_C)}{k} \geq \frac{n(1 - \frac{1}{k+1}) - (k-1)x\phi_C}{k}$$

and further

$$\frac{n(1 - \frac{1}{k+1}) - (k-1)x\phi_C}{k} = \frac{n}{k+1} - \frac{k-1}{k}x\phi_C \geq \frac{n}{k+1} - x\phi_C \geq \frac{n}{k+1} - x\phi_C - t_C \geq K.$$

Thus after any vote manipulation by nodes from C there always exists a tree of size at least K, i.e., with the root having value at least K.

2. Now we show that no valuation gives guarantee β against k-manipulation, for any $\beta < \alpha$. Consider a voting graph G_L composed of a number of directed lines $L_0, \ldots, L_m$, for some non-negative integer m. Each line $L_i = \langle p_0^i, p_1^i, \ldots, p_{n_i-1}^i \rangle$, for $i = 0, \ldots, m$, is formed of n_i nodes $p_0^i, p_1^i, \ldots, p_{n_i-1}^i$, s.t., the node p_0^i has no predecessor and for each $j = 1, \ldots, n_i-1$ there is a directed edge connecting p_{j-1}^i with its successor p_j^i. We start with the following observation.

Fact 3.2. For a voting graph G_L and any value function V on this graph we have the following.

(a) For any $L_i = \langle p_0^i, p_1^i, \ldots, p_{n_i-1}^i \rangle \in G_L$ and a positive $j \leq n_i - 1$,

$$V(p_{j-1}^i) < V(p_j^i),$$

(b) For any $L_i, L_{i'} \in G_L$ and a positive $j \leq \min(n_i - 1, n_{i'} - 1)$,

$$V(p_j^i) = V(p_j^{i'}).$$

Proof. We prove the inequality (a) by induction. The support $S(p_0^i)$ is null; let $V(p_0^i) = d$, for some positive integer d. Since $V(p_0^i) = S(p_1^i)$ we get $S(p_0^i) < S(p_1^i)$ and hence $V(p_0^i) < V(p_1^i)$. Also for all $j = 2, \ldots, n_i - 1$, we have $V(p_{j-1}^i) = S(p_j^i)$. Thus by the inductive hypothesis, we have $V(p_{j-1}^i) < V(p_j^i)$, which implies $S(p_j^i) < S(p_{j+1}^i)$ and hence $V(p_j^i) < V(p_{j+1}^i)$.

We also prove the equality (b) by induction. Note that, for all $i = 1, \ldots, m$, $S(p_0^i) = 0$ implying that $V(p_0^i) = c$, for some positive integer c. Assume inductively that for $i, i' \in \{0, \ldots, m\}$ we have $V(p_{j'}^i) = V(p_{j'}^{i'})$ for all $j' < j$. Thus $V(p_{j-1}^i) = V(p_{j-1}^{i'})$ and hence $S(p_j^i) = S(p_j^{i'})$, implying $V(p_j^i) = V(p_j^{i'})$. □

1-manipulation. Assume that $\beta < \alpha = \frac{1}{2}$. Consider a voting graph G_L containing a single line $L = \langle p_0, p_1, \ldots, p_{2r} \rangle$, s.t., $\frac{r}{2r+1} > \beta$. Note that such r must exist since β is strictly smaller than $\frac{1}{2}$ and the fraction $\frac{r}{2r+1}$ converges to $\frac{1}{2}$. Let the node $c = p_r$ form a singleton coalition. Since $\frac{r}{2r+1} > \beta$, the node c is outside of the top β fraction. Consider the vote manipulation in which c removes support for its successor p_{r+1}, and abstains. Consequently, we obtain two lines $L_L = \langle p_0, p_1, \ldots, p_r \rangle$ and $L_U = \langle p_{r+1}, p_1, \ldots, p_{2r} \rangle$. According to Fact 3.2 $V(p_j) = V(p_{j+r+1})$, for all $j = 0, \ldots, r-1$ and $V(p_r) > V(p_{r-1}) = V(p_{2r})$. Thus, using the above manipulation, c becomes elected while being outside of the top β fraction before the manipulation.

k-manipulation. Assume this time that $\beta < \alpha = 1 - \frac{1}{k+1}$. Consider a voting graph G_L consisting of k lines $L_0, \ldots, L_{k-1}$, where each line $L_i = \langle p_0^i, \ldots, p_{(k+1)r}^i \rangle$ is of length $(k+1)r + 1$, s.t., $\frac{kr}{(k+1)r+1} > \beta$. Note that such r must exist since β is strictly smaller than $\frac{k}{k+1}$ and the fraction $\frac{kr}{(k+1)r+1}$ converges to $\frac{k}{k+1}$, as r grows to infinity. Select all nodes p_r^i, for $i = 0, \ldots, k-1$, for the coalition C of size k. Note that in each line L_i there are kr nodes with values larger than $V(p_r^i)$. Due to Fact 3.2 and since $\frac{kr}{(k+1)r+1} > \beta$ all nodes in the coalition C are outside of the top β fraction. Consider the vote manipulation in which each node in C removes its support for its successor, and instead all nodes in C vote to form a single line $\langle p_0^0, \ldots, p_r^0, p_0^1, \ldots, p_r^1, \ldots, p_0^{k-1}, \ldots, p_r^{k-1} \rangle$ of length $k(r+1)$. Since after this manipulation the remainders of lines $L_0, \ldots, L_{k-1}$ have the same length kr, and $k(r+1) > kr$, for any $k, r \geq 1$, the node $p_r^{k-1} \in C$ becomes elected, while being outside of the top β fraction before the manipulation. □

4. Conclusion

We proposed a voting system to elect the most competent member of a group of peers. Each member of the group can vote for one other member, or abstain. A natural generalization of the proposed system would be to allow each member to vote for k other members, where k is larger than 1, or even for an arbitrary number of other members. In the latter case, a vote for some member could be naturally interpreted as approval and the lack of vote as disapproval. Another enhancement to the voting system would be to give z votes to each member of the group, with the possibility of distributing the votes among any subset of other members. In this case, support given to a member a should be computed as the sum of $x_i v_i$, over all supporters i of a, where x_i is the number of votes given for a by its supporter i and v_i is the value of this supporter. In each of these scenarios one could seek value functions satisfying the strict support monotonicity property. It would be interesting to see if such value functions exist for arbitrary voting graphs in these enhanced scenarios, and to study issues of vulnerability to vote manipulations.

Acknowledgments

The first author was partially supported by NSERC discovery grant. The second author's work was done during his visit at the Université du Québec en Outaouais. The third author was partially supported by NSERC discovery grant and by the Research Chair in Distributed Computing at the Université du Québec en Outaouais.

References

[1] A. Altman, M. Tennenholtz, Ranking systems: the PageRank axioms, in: Proc. ACM Conference on Electronic Commerce 2005, pp. 1–8.
[2] J.F. Banzhaf, Weighted voting doesn't work: a mathematical analysis, Rutgers Law Review 19 (1965) 317–343.
[3] A. Gelman, J.N. Katz, F. Tuerlinckx, The mathematics and statistics of voting power, Statistical Science 17 (2002) 420–435.
[4] A. Gibbard, Manipulation of voting schemes: a general result, Econometrica 41 (1973) 587–601.
[5] D. Okanohara, K. Sadakane, A linear-time Burrows-Wheeler transform using induced sorting, in: 16th International Symposium on String Processing and Information Retrieval, SPIRE, in: LNCS, vol. 5721, 2009, pp. 90–101.
[6] L. Page, S. Brin, R. Motwani, T. Winograd, The pagerank citation ranking: bringing order to the web, Technical Report, Stanford University, 1998.
[7] M.A. Satterthwaite, Strategy-proofness and Arrow's conditions: existence and correspondence theorems for voting procedures and social welfare functions, Journal of Economic Theory 10 (1975) 187–217.
[8] L.S. Shapley, M. Shubik, A method for evaluating the distribution of power in a committee system, American Political Science Review 48 (1954) 787–792.
[9] G.G. Szpiro, Numbers Rule: The Vexing Mathematics of Democracy, from Plato to the Present, Princeton University Press, 2010.
[10] A.D. Taylor, Social Choice and the Mathematics of Manipulation, Cambridge University Press, 2005.