Computation and Bribery of Voting Power in Delegative Simple Games

Gianlorenzo D'Angelo
Gran Sasso Science Institute
L'Aquila, Italy
gianlorenzo.dangelo@gssi.it

Esmaeil Delfaraz
Gran Sasso Science Institute
L'Aquila, Italy
esmaiel.delfaraz@gssi.it

Hugo Gilbert
Université Paris-Dauphine, Université PSL, CNRS, LAMSADE
75016 Paris, France
hugo.gilbert@dauphine.psl.eu

ABSTRACT

Following Zhang and Grossi (AAAI 2021), we study in more depth a variant of weighted voting games in which agents' weights are induced by a transitive support structure. This class of simple games is notably well suited to study the relative importance of agents in the liquid democracy framework. We first propose a pseudo-polynomial time algorithm to compute the Banzhaf and Shapley-Shubik indices for this class of game. Then, we study a bribery problem, in which one tries to maximize/minimize the voting power/weight of a given agent by changing the support structure under a budget constraint. We show that these problems are computationally hard and provide several parameterized complexity results.

KEYWORDS

Liquid Democracy; Voting Power Measurement; Manipulation Problems

ACM Reference Format:
Gianlorenzo D'Angelo, Esmaeil Delfaraz, and Hugo Gilbert. 2022. Computation and Bribery of Voting Power in Delegative Simple Games. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), online, May 9–13, 2022*, IFAAMAS, 9 pages.

1 INTRODUCTION

Weighted Voting Games (WVG) form a simple scheme to model situations in which voters must make a yes/no decision about accepting a given proposal [11]. Each voter has a corresponding weight and the proposal is accepted if the sum of weights of agents supporting the proposal exceeds a fixed threshold called the quota. In a WVG, weights of voters can represent an amount of resource and the quota represents the quantity of this resource which should be gathered to enforce the proposal. For instance, agents may be political parties and weights could be derived from the relative importance of each party in terms of number of votes received. There is a large literature to measure the relative importance of each agent in such a situation [11, 17]. Notably, the computational and axiomatic properties of two such measures, the Banzhaf measure of voting power and the Shapley-Shubik index [4, 34], have been extensively studied. The computational properties investigated obviously include the computational complexity of computing these measures for a given agent [14] but they also involve several manipulation problems [2, 17, 36, 37].

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, online.

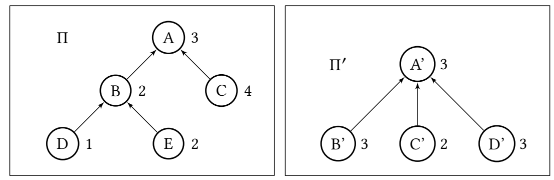

Figure 1: Two political parties Π and Π′ with two different inner structures.

One possible limitation of WVGs is that they consider each agent as an indivisible entity and may not be able to represent agents who are composed of a complex structure. However, if agents are corporations or political parties, then they do have some inner structure which may have an impact on their relative strength.

Example 1. *Let us consider two political parties* Π *and* Π′ *represented in Figure 1. Each party has a political leader (agents A and A′ respectively) and different inner political trends with subleaders (agents B, C, D, E in* Π *and agents B′, C′, and D′ in* Π′*). Each leader and subleader have their own supporters which provide them some voting weights, written next to each agent in Figure 1. In each party, the different agents form a directed tree structure of support, where each arc represents the fact that an agent supports another agent. For instance, in party* Π*, agent D endorses agent B which herself endorses agent A. In this way, agent D implicitly supports agent A and puts at her disposal her voting weight. Note however that it may not be possible for agent D to directly endorse agent A. Indeed, if agent D represents the most left-wing sensibility of the party whereas agent A represents a more consensual political trend, it may be difficult for agent D to publicly support agent A without losing credibility in the eye of her supporters. By delegating to A through B, agent D indicates that her support to A is conditioned to the presence of agent B.*

The total weight accumulated by voters A and A′ are respectively worth 12 and 11. Hence, the total weight gathered by A is greater than the one of A′. Could this indicate that agent A is at least as powerful as agent A′. The inner structures of parties Π *and* Π′ *suggest otherwise. Indeed, agent A receives a greater total weight and is endorsed (directly and indirectly) by more agents. However, agent A′ receives direct support from all other subleaders of her party which is not the case of agent A. As a result, if agent B decides to secede and create her own party, then agent A would lose the support of agents D and E conceding a total weight lose of 5. Conversely, the most important weight lose that agent A′ can suffer from the secession of another agent is worth 3. Hence, the inner structure of party* Π′ *which supports candidate A′ seems more robust than the one of candidate A.*

Example 1 suggests to study a more complex model than the one of WVGs where agents are backed up by an internal support structure with transitive supports. Studying this kind of model has been recently initiated by Zhang and Grossi [35]. Indeed, the authors investigate how to measure the relative importance of voters in the framework of *Liquid Democracy* (LD) [6, 18]. LD is a collective decision paradigm in which agents can vote themselves or delegate their vote to another agent. One important feature, is that an agent who receives delegations can in turn delegate her vote and the ones that she has received to another agent which is exactly the kind of transitive support discussed in Example 1. In their work, Zhang and Grossi [35] define *Delegative Simple Games* (DSG), a variant of WVGs in which the capacity of a subset of agents to reach the quota does not only depend on their voting weights but also on their delegations. DSGs make it possible to take into account the support (delegation) structure underlying the game and to favor agents who receive more direct supports, compared to agents receiving more distant chains of support. The authors notably study axiomatically the Banzhaf measure of voting power applied to this kind of game which they term the *delegative Banzhaf measure of voting power.*[1]

Our contribution. We obtain several results related to DSGs. We first study several properties, notably computational properties, of the delegative Banzhaf and Shapley-Shubik measures of power. For instance, while the computation of these measures is computationally hard, we show that they can be calculated by a pseudo-polynomial dynamic programming algorithm similar to the one for Banzhaf and Shapley-Shubik measures in WVGs. We then investigate a bribery problem where, given a delegation graph, the goal is to maximize/minimize the voting power/weight of an agent by changing at most a fixed number of delegations. We show that the problems related to bribing voting power are hard, and that the maximization problems are hard to approximate even when the social network is a tree. We then move to the conceptually simple bribery problems related to voting weight. On these problems, we obtain both hardness and tractability results by investigating the approximation and the parameterized complexity viewpoints. All missing or incomplete proofs can be found in a long version of the paper [13].

2 RELATED WORK

WVGs originated in the domain of cooperative game theory [10, 11] and are used to study the a-priori voting power of voters in an election [17]. Two well-known solutions to measure the importance of an agent in a WVG are the Shapley-Shubik index and the Banzhaf measure of voting power [4, 34]. Computing these measures is #P-Complete [14, 32]. However, Matsui and Matsui [27] designed pseudopolynomial algorithms that can compute the Shapley-Shubik and Banzhaf measures in time $O(n^3 w_{\max})$ and $O(n^2 w_{\max})$ respectively, where n is the number of agents and $w_{\max}$ is the maximum weight of an agent. Other works have been dedicated to their computation, either to compute them exactly [9, 12, 21, 22], or approximately [3, 16, 25, 26, 29]. Moreover, several manipulation problems involving voting power measures have been investigated, as computing the quota maximizing or minimizing the importance of a specific voter [36, 37] or determining the impact of splitting an agent in two [2, 17, 24].

Several works have studied the measurement of voters' importance in an LD election [7, 8, 23, 35]. Boldi et al. [7, 8] proposed a way to measure the relative importance of each voter in a social network using a power index similar to PageRank. In their model, each voter can recommend one of her neighbor in the network. The recommendations of the voters are then transitively delegated but are attenuated by using a multiplicative damping factor. The authors term the resulting system a viscous democracy election as the voting power does not flow completely but rather meets some resistance. In this way, the model by Boldi et al. [7, 8] favors the voters who receive direct supports instead of more distant ones. Kling et al. [23] studied the behavior of voters using the Liquid-Feedback software in the German Pirate Party. The authors studied the number and types of interactions with the software, the distribution in terms of number of delegations received per voter as well as the behavior of "super-voters" which receive a large number of delegations. The authors then applied several power measures (e.g., Shapley-Shubik and Banzhaf) to analyze the power distribution in LD elections. This analysis led the authors to propose modifications to the existing power measures to fit better to the data they gathered. More precisely, the authors designed generalizations of theses measures which allow to model non-uniform distributions of approval rates. Lastly, as detailed in the introduction, Zhang and Grossi [35] have recently proposed a formal way to measure the influence of voters in an LD election by introducing a variant of WVGs. This variant as well as the resulting power measures will be presented in Section 3.

3 PRELIMINARIES

3.1 Weighted voting games

A simple game is a tuple $\mathcal{G} = \langle V, v \rangle$, where $V = [n]$ is a set of n agents and $v : 2^n \rightarrow \{0, 1\}$ is a characteristic function which only takes values 0 and 1. The notation $[i]$ and $[i]_0$ will denote the sets $\{1, \dots i\}$ and $\{0, 1, \dots, i\}$ respectively. A subset $C \subseteq V$ will also be called a coalition. For any coalition $C \subseteq V$, C is said to be a winning (resp. losing) coalition if $v(C) = 1$ (resp. 0). An agent i is said to be a swing agent for coalition C if $\delta_i(C) := v(C \cup \{i\}) - v(C)$ equals 1.

In WVGs, there exists a quota q and each agent (also called voter) i is associated with a weight w_i. The characteristic function v is then defined by $v(C) = 1$ iff $\sum_{i \in C} w_i \geq q$. Stated otherwise, a coalition is winning if the sum of weights of agents in the coalition exceeds the quota. Several ways of measuring the importance of an agent in WVGs have been studied. We recall two of the most well known:

Definition 1. *The Banzhaf measure $B_i(\mathcal{G})$ and Shapley-Shubik index $Sh_i(\mathcal{G})$ of a voter i in a simple game $\mathcal{G}$ are defined as*

$$B_i(\mathcal{G}) := \sum_{C \subseteq V \setminus \{i\}} \frac{1}{2^{n-1}} \delta_i(C),$$

$$Sh_i(\mathcal{G}) := \sum_{C \subseteq V \setminus \{i\}} \frac{1}{n} \frac{(n - |C| - 1)!|C|!}{(n-1)!} \delta_i(C).$$

[1] This measure was in fact termed, the delegative Banzhaf index. However, following Felsenthal and Machover [17], we reserve the term index for measures whose values sum up to one when considering all agents.

Hence, the Banzhaf measure and the Shapley-Shubik index provide two ways to measure the importance of a voter by quantifying her ability to be a swing agent. While both measures are worth investigation, they are quite different in nature as explained by Felsenthal and Machover [17]. Indeed, while the Shapley-Shubik index is better explained as the expected share that an agent should earn from the election, seen as a game, the Banzhaf measure computes (based on a probabilistic model) the extent to which an agent is able to control the outcome of the election. Note that the first (resp. second) kind of measure is referred to as a notion of P-Power (resp. I-Power), where P stands for Prize (resp. I stands for Influence). As both measure have been extensively studied, we will study both in this paper.

3.2 A model of liquid democracy

In the sequel, while our introduction suggests that DSGs can be used in a broader setting, we will follow the LD paradigm which showcases an interesting application where transitive support structures play a key role. This will notably be convenient to use the notations from Zhang and Grossi [35].

A liquid democracy election. A finite set of agents $V = [n]$ will vote on a proposal. There is a weight function $\omega : V \longrightarrow \mathbb{N}_{>0}$, assigning a positive weight $\omega(i) = w_i$ to each voter i. A special case of interest is the one where all voters have weight one. The rule used is a super-majority rule with quota $q \in (\frac{\sum_{i\in V} w_i}{2}, \sum_{i\in V} w_i] \cap \mathbb{N}_{>0}$.[2] Stated otherwise, the proposal is accepted if the total voting weight in favor of it is at least worth q.

We assume the election to follow the LD paradigm. Notably, voters are vertices of a Social Network (SN) modeled as a directed graph $D = (V, A)$. Each node in the SN corresponds to a voter $i \in V$ and a directed edge $(i, j) \in A$ corresponds to a social relation between i and j: it specifies that agent i would accept to delegate her vote to j or more generally speaking to endorse j. The set of out-neighbors of voter i is denoted by $\mathsf{Nb}_{out}(i) = \{j \in V|(i, j) \in A\}$. Each agent i has two possible choices: either she can vote directly, or she can delegate her vote to one of her neighbors in $\mathsf{Nb}_{out}(i)$. The information about delegation choices is formalized by a delegation function d, where $d(i) = j$ if voter i delegates to voter $j \in \mathsf{Nb}_{out}(i)$, and $d(i) = i$ if voter i votes directly.

The delegation digraph $H_d = (V, E)$ resulting from d is the subgraph of D, where $(i, j) \in E$ iff $i \neq j$ and $d(i) = j$. We assume that this digraph is acyclic. More precisely, we assume H_d is a spanning forest of in-trees where all vertices have out-degree 1 except the roots which have out-degree 0. For any digraph D, $\Delta(D)$ denotes the set of all acyclic delegation graphs H_d that can be induced by delegation functions d on D. In H_d, we denote by $c_d(i, j)$ the delegation chain which starts with voter i and ends with voter j. Put another way, $c_d(i, j)$ is a sequence $((v_1, v_2), \ldots, (v_{k-1}, v_k))$ of arcs such that $v_1 = i$, $v_k = j$ and $\forall l \in [k-1], d(v_l) = v_{l+1}$. By abuse of notation, we may also use this notation to denote the set $\{v_i, i \in [k]\}$ of voters in the chain of delegations from i to j. Moreover, we denote by $T_d(i)$ the directed subtree rooted in i in delegation graph H_d, where i has out-degree 0 and all other vertices have out-degree 1.

Delegations are transitive, meaning that if voter i delegates to voter j, and voter j delegates to voter k, then voter i indirectly delegates to voter k. If an agent votes, she is called the *guru* of the people she represents and has an *accumulated voting weight* equal to the total weight of people who directly or indirectly delegated to her. We denote by d^*_i the guru of voter i and by $Gu_d = \{i \in V|d(i) = i\}$ the set of gurus induced by d, i.e., the roots of the in-trees in H_d. If an agent delegates, she is called a *follower* and has an accumulated voting weight worth 0. Hence, the delegation function d induces an *accumulated weight function* α_d, such that $\alpha_d(i) = \sum_{j\in T_d(i)} w_j$ if $i \in Gu_d$ and 0 otherwise.

To define DSGs, Zhang and Grossi defined another weight function. Given a set $C \subseteq V$, let us denote by $T_d(i, C)$ the directed subtree rooted in i in the delegation graph $H_d[C]$. The subtree $T_d(i, C)$ contains each voter h such that $d^*_h = i$ and $c_d(h, i)$ only contains elements from C. Given a set $C \subseteq V$, we define the weight function $\gamma_{d,C}$, such that $\gamma_{d,C}(i) = \sum_{j\in T_d(i,C)} w_j$ if $i \in Gu_d$ and 0 otherwise. Moreover, we denote by $\gamma_d(C) = \sum_{i\in C} \gamma_{d,C}(i)$. The value $\gamma_d(C)$ represents the sum of weights of voters that have a guru in C and such that the chain of delegations leading to this guru is contained in C.

3.3 Delegative Simple Games

We call a Liquid Democracy Election (LDE) a tuple $\mathcal{E} = \langle D, \omega, d, q\rangle$ or $\mathcal{E} = \langle D, \omega, H_d, q\rangle$ (we may use H_d in place of d when convenient). DSGs are motivated by LDEs. In the DSG $\mathcal{G}_\mathcal{E}$ induced by an LDE $\mathcal{E} = \langle D, \omega, d, q\rangle$, $v_\mathcal{E}(C) = 1$ iff $\gamma_d(C) \geq q$, i.e., a coalition C is winning whenever the sum of weights accumulated by gurus in C from agents in C meets the quota. This defines a new class of simple games which have a compact structure but different from the one of WVGs. In our case, the weights are not given in advance, but rather derived from a graph structure. The different power indices defined for simple games can of course be applied to this new class. It is worth noticing that in DSGs the power of each voter i does not only depend on the amount of delegations she receives through delegations, but also on the structure of the subtree rooted in her in the delegation graph. Notably, the more direct the supports of i are, the more important she is in the delegation graph (see Property 3 for a formal statement).

Definition 2 (Active Agent)**.** *Consider a delegation function d and a coalition $C \subseteq V$. We say that voter $i \in C$ is* active *in C if $d^*(i) \in C$ and $c_d(i, d^*(i)) \subseteq C$; otherwise, i is called an* inactive *agent.*

Notice that according to the definition of DSGs, an active (resp. inactive) agent $i \in C$ contributes weight w_i (resp. 0) to the coalition C (see Example 2). Given a DSG $\mathcal{G}_\mathcal{E}$, let $DB_i(\mathcal{E})$ and $DS_i(\mathcal{E})$ denote the delegative Banzhaf and Shapley-Shubik values of voter i in $\mathcal{G}_\mathcal{E}$, respectively. When speaking about these indices, we drop parameter $\mathcal{E}$ (or $\mathcal{G}_\mathcal{E}$) when it is clear from the context. Let us give an illustrative example to understand all aspects of our model.

Example 2. *Consider an LDE $\mathcal{E} = \langle D = (V, A), \omega, d, q\rangle$, where $V = [8]$ is a set of 8 voters delegating through a SN with $d(1) = d(2) = d(3) = 3$, $d(4) = d(6) = 7$, $d(5) = 6$ and $d(7) = d(8) = 8$ as illustrated in Fig. 2 and $q = 3$. Each voter i has weight $\omega(i) = 1$.*

[2] In this work, we restrict the values of weights w_i and q to $\mathbb{N}_{>0}$ This restriction can be motivated by a result by Muroga [30].

The set of gurus is $Gu_d = \{3, 8\}$, so, $\alpha_d(3) = \sum_{j \in T_d(3)} w_j = 3$, $\alpha_d(8) = \sum_{j \in T_d(8)} w_j = 5$, and for any $i \in V \setminus \{3, 8\}$, $\alpha_d(i) = 0$. Consider the set $C = \{3, 5, 7, 8\}$. $T_d(8, C)$ (resp. $T_d(3, C)$) is the subtree rooted in 8 (resp. 3) and composed of voters $\{7, 8\}$ (resp. $\{3\}$). Thus, $\gamma_{d,C}(3) = \sum_{j \in T_d(3,C)} w_j = 1$, $\gamma_{d,C}(8) = \sum_{j \in T_d(8,C)} w_j = 2$ and $\gamma_{d,C}(i) = 0$ for $i \in \{5, 7\}$, and then $\gamma_d(C) = \sum_{i \in C} \gamma_{d,C}(i) = 3$. Also, the set of active agents is $\{3, 7, 8\}$. Note that 5 is an inactive agent in C as $c_d(5, 8) \nsubseteq C$. Now we intend to compute DB_8 and DB_6.

We first compute DB_8 by counting the coalitions $C \subseteq V \setminus \{8\}$ for which 8 is a swing agent. Note that if $\{1, 2, 3\} \subseteq C$, then 8 cannot be a swing agent for C. Second, if $\{1, 2, 3\} \cap C \in \{\{1, 3\}, \{2, 3\}\}$, then 8 will always be a swing agent for C. There are 32 such coalitions. Third, if $\{1, 2, 3\} \cap C = \{3\}$, then 8 will be a swing agent for C iff $7 \in C$. There are 8 such coalitions. Last, if $3 \notin (\{1, 2, 3\} \cap C)$, then 8 will be a swing agent for C iff $|C \cap \{4, 6, 7\}| \geq 2$. There are 24 such coalitions. Hence $DB_8(V) = \frac{1}{2^7} \times 64 = \frac{1}{2}$.

Now we compute DB_6. Note that in order for 6 to be a swing agent in any coalition C, it is necessary that $\{7, 8\} \subseteq C$. Hence, the coalitions C for which 6 is a swing agent are of the form $C = \{7, 8\} \cup S$ where $S \subseteq \{1, 2, 5\}$. The number of such coalitions is 8. Hence, $DB_6 = \frac{1}{2^7} \times 8 = \frac{1}{16}$.

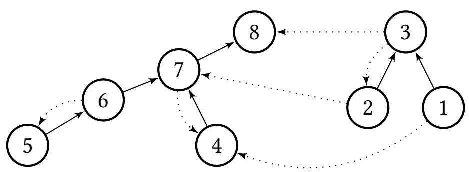

Figure 2: The delegation graph H_d (dotted arcs are the other arcs of $D = (V, A)$) with the set of gurus $Gu_d = \{3, 8\}$.

4 PROPERTIES OF DB AND DS

This section investigates properties of the delegative Banzhaf measure and the delegative Shapley-Shubik index. First we investigate if, these values can be computed efficiently.

Evaluating the "standard" Banzhaf and Shapley-Shubik measures in WVGs is #P-complete [14, 32] and several decision problems related to their computation are NP-complete or coNP-complete [10, 28, 32]. As DSGs include WVGs as a subcase (consider the restriction when each voter votes herself), these hardness results also hold for DSGs. However, these measures can be computed by a pseudo-polynomial algorithm [28] in WVGs, that is an algorithm running in time $poly(n, w_{\max})$, where $w_{\max} := \max_{i \in V} \omega(i)$ is the maximal weight of an agent. We can extend this dynamic programming approach to compute the delegative Banzhaf measure and the delegative Shapley-Shubik index in pseudo-polynomial time.

Theorem 1. *Given a delegative simple game $\mathcal{G}_\mathcal{E}$ induced by an LDE $\mathcal{E} = \langle D = (V, A), \omega, d, q \rangle$ and a voter $i \in V$, DS_i and DB_i can be computed in $O(n^4 w_{\max}^2)$ and $O(n^2 w_{\max}^2)$, respectively.*

The description of the algorithm is similar to the one used in WVGs but it carefully takes into account the delegation graph provided in the input.

We now investigate how these measures evolve when the delegation structure of the LDE is slightly modified. The purpose for studying these properties is that they are notably relevant when studying bribery problems where one tries to modify the structure of the instance to increase (or decrease) the relative importance of a given agent, as will be done in Section 5. Possible modifications of the delegation structure can consist in adding a direct or indirect delegation from one agent to another.

We now present three properties. Let $v_\mathcal{E}$ be the characteristic function of a DSG $\mathcal{G}_\mathcal{E}$ induced by an LDE $\mathcal{E} = \langle D = (V, A), \omega, d, q \rangle$. In the following f denotes a generic delegative power measure and f_i is the delegative power measure value of agent i.

The following property states that if some voters delegate to a voter i directly, i should get more powerful.

Property 1 (Monotonicity w.r.t. Direct Delegations (**MDD**)). *Consider two LDEs $\mathcal{E} = \langle D = (V, A), \omega, d, q \rangle$ and $\mathcal{E}' = \langle D = (V, A), \omega, d', q \rangle$ and two agents $i, j \in V$ such that $j \notin c_d(i, d_i^*)$, $d'(j) = i$ and $d(l) = d'(l)$ for $l \neq j$. Then, $f_i(\mathcal{E}') \geq f_i(\mathcal{E})$.*

Next, we introduce another property which says that if a voter receives some delegation (directly or indirectly), she should become more powerful.

Property 2 (Monotonicity w.r.t. Delegations (**MD**)). *Consider two LDEs $\mathcal{E} = \langle D = (V, A), \omega, d, q \rangle$ and $\mathcal{E}' = \langle D = (V, A), \omega, d', q \rangle$ and three agents $i, j, k \in V$ s.t. $j \notin c_d(k, d_k^*)$, $i \notin c_d(j, d_j^*)$, $i \in c_d(k, d_k^*)$, $d'(j) = k$ and $d(l) = d'(l)$ for $l \neq j$. Then, $f_i(\mathcal{E}') \geq f_i(\mathcal{E})$.*

Consider a delegation graph H_d and two voters $i, j \in V$ such that $i \in c_d(j, d_j^*)$. We now introduce the last property stating that in a new delegation graph d' obtained from d in which all voters have the same delegation strategies as d except that j delegates to a voter $l \in c_d(j, d_j^*) \setminus c_d(i, d_i^*)$, i gets more powerful. In other words, this property says that if the voters who support a specific voter i through some intermediaries, support i through a smaller set of intermediaries, i gets more powerful.

Property 3 (Monotonicity w.r.t. Intermediaries (**MI**)). *Consider two LDEs $\mathcal{E} = \langle D = (V, A), \omega, d, q \rangle$ and $\mathcal{E}' = \langle D = (V, A), \omega, d', q \rangle$ and three agents $i, j, k \in V$ such that $k \in c_d(j, d_j^*)$, $i \in c_d(k, d_k^*)$, $\forall l \in V \setminus \{j\}, d(l) = d'(l)$ and $d'(j) = k$. Then, $f_i(\mathcal{E}') \geq f_i(\mathcal{E})$.*

We now investigate which of these three properties are satisfied by the delegative Banzhaf and Shapley-Shubik measures. Counterintuitively, we first show that while these indices satisfy properties **MDD** and **MI**, it is not the case for property **MD**.

Proposition 1. *The delegative Banzhaf measure and the delegative Shapley-Shubik index satisfy property **MDD**.*

Proof. Consider two LDEs $\mathcal{E} = \langle D = (V, A), \omega, d, q \rangle$ and $\mathcal{E}' = \langle D', \omega', d', q' \rangle$ and two agents $i, j \in V$ such that $j \notin c_d(i, d_i^*)$, $D = D'$, $\omega = \omega'$, $q = q'$, $d'(j) = i$ and $d(l) = d'(l)$ for $l \neq j$. Let us consider a coalition $C \subseteq V \setminus \{i\}$ such that $v_\mathcal{E}(C \cup \{i\}) - v_\mathcal{E}(C) = 1$. This implies that $c_d(i, d_i^*) \subseteq C \cup \{i\}$. From $v_\mathcal{E}(C \cup \{i\}) = 1$, it is straightforward that $v_{\mathcal{E}'}(C \cup \{i\}) = 1$ as $d(l) = d'(l)$ for $l \neq j$, $d'(j) = i$ and $c_d(i, d_i^*) \subseteq C \cup \{i\}$. Moreover, from $v_\mathcal{E}(C) = 0$, it is also easy to see that $v_{\mathcal{E}'}(C) = 0$ as $d(l) = d'(l)$ for $l \neq j$, $d'(j) = i$ and $i \notin C$. Because this reasoning holds for any coalition for which i is a swing agent, we obtain that $DS_i(\mathcal{E}') \geq DS_i(\mathcal{E})$ and $DB_i(\mathcal{E}') \geq DB_i(\mathcal{E})$. □

Proposition 2. *The delegative Banzhaf measure and the delegative Shapley-Shubik index do not satisfy property **MD**.*

Proof. Consider an LDE $\mathcal{E} = \langle D = (V, A), \omega, d, q\rangle$, where $V = \{1, 2, 3\}$ is a set of 3 agents delegating through a complete SN D (A contain all possible arcs) with delegations $d(2) = d(1) = 1$ and $d(3) = 3$, all weights equal to one, and $q = 2$. In $\mathcal{E}$ voter 1 is a swing agent for sets in $\{\{2\}, \{3\}, \{2, 3\}\}$. Now consider the LDE $\mathcal{E}'$ which is identical to $\mathcal{E}$ except that $d(3) = 2$. In $\mathcal{E}'$ voter 1 is a swing agent for sets in $\{\{2\}, \{2, 3\}\}$. Hence, $DS_1(\mathcal{E}') < DS_1(\mathcal{E})$ and $DB_1(\mathcal{E}') < DB_1(\mathcal{E})$. □

Hence, Proposition 2 highlights a paradox for measures DB and DS: a voter who receives more voting power can become less powerful because of the delegation structure underlying the DSG. Hence, if one studies a bribery problem in which she wants to change some delegations to make an agent i^* more powerful, then she is on the safe side if she decides to add direct delegations to i^* but adding indirect delegations may be counterproductive.

Proposition 3. *The delegative Banzhaf measure and the delegative Shapley-Shubik index satisfy property **MI**.*

Proof. One may use the same argument as that of Proposition 1 to prove the proposition. □

5 BRIBERY BY DELEGATION MODIFICATIONS

5.1 Power index modification by bribery

The support structure in DSGs induces the following natural question: which voters should one influence under a budget constraint to maximize/minimize the voting power of a given voter? This question leads to the following computational bribery problems.

> *Problems*: **BMinP**, **SMinP**, **BMaxP** and **SMaxP**
> *Input*: An LDE $\mathcal{E} = \langle D = (V, A), \omega, d, q\rangle$, a voter $v^* \in V$, a budget $k \in \mathbb{N}$, and a threshold $\tau \in \mathbb{Q}_+$.
> *Feasible solution*: A delegation function $d' \in \Delta(D)$ s.t. $|\{i \in V : d(i) \neq d'(i)\}| \leq k$ leading to an LDE $\mathcal{E}' = \langle D, \omega, d', q\rangle$.
> *Question*: Can we find a feasible solution d' such that:
> **BMinP**: $DB_{v^*}(\mathcal{E}') \leq \tau$? **BMaxP**: $DB_{v^*}(\mathcal{E}') \geq \tau$?
> **SMinP**: $DS_{v^*}(\mathcal{E}') \leq \tau$? **SMaxP**: $DS_{v^*}(\mathcal{E}') \geq \tau$?

Stated otherwise, in the Banzhaf Minimization (resp. Maximization) Problem, **BMinP** (resp. **BMaxP**) for short, we wish to determine if we can make the Banzhaf measure of voter v^* lower (resp. greater) than or equal to a given threshold, by only modifying k delegations. This cardinality constraint can be justified by the fact that influencing each voter is costly. **SMinP** and **SMaxP** are similar problems corresponding to the Shapley-Shubik index. While **BMaxP** and **SMaxP** correspond to the constructive variant of the bribery problem, **BMinP** and **SMinP** correspond to its destructive variant. These bribery problems are natural in the setting of LD, where one voter could for instance try to get the delegations of several other voters to increase her influence on the election. Moreover, we believe these bribery problems are also relevant in more traditional elections where several politicians or political parties could seek which alliances to foster as to increase their centrality or to make an opponent powerless.

We first show several hardness and hardness of approximation results on these four problems.

Theorem 2. *The restriction of **BMinP** and **SMinP** to the case where all voters have weight 1, i.e., when $\forall i \in V, \omega(i) = 1$, is NP-complete. Moreover, under the same restriction, the minimization versions of problems **BMinP** and **SMinP** cannot be approximated within any factor in polynomial time if* P ≠ NP.

Sketch of proof. We use a reduction from the NP-complete Hamiltonian path problem [19] where the goal is to determine if there exists a path in an undirected graph that visits each vertex exactly once. From an instance of the Hamiltonian path problem with n vertices we create an instance of **BMinP** (or **SMinP**). The quota of this **BMinP** (or **SMinP**) instance is set such that a specific voter will have power index value 0 iff it is at the end of a delegation path of length n. The idea is that such a path necessarily consists in a Hamiltonian path in the original instance. Hence, in the **BMinP** (or **SMinP**) instance a successful solution for the briber will consist in creating a delegation path along a Hamiltonian path (if one exists). The hardness of approximation result is then obtained from the fact that a polynomial-time algorithm with some multiplicative approximation guarantee would be able to distinguish between instances where we can make the agent dummy and the ones where we cannot. □

Theorem 3. *Problems **BMinP** and **SMinP** are coNP-hard even if the SN is a tree.*

We move to problems **BMaxP** and **SMaxP**.

Theorem 4. *Problems **BMaxP** and **SMaxP** with voters' weights and the quota given in unary are NP-complete.*

Theorem 5. *The maximization versions of problems **BMaxP** and **SMaxP** cannot be approximated within any factor in polynomial time if* P ≠ NP *even if the SN is a tree.*

Proof. Consider the subset sum problem with positive integers. An instance J of this problem is composed of a set of positive integers $S = \{a_1, \ldots, a_n\}$ and a target sum M: $J = \langle S, M\rangle$ is a Yes-instance iff there exists a subset $L \subseteq S$ such that $\sum_{a_i \in L} a_i = M$. We transform an instance $J = \langle S, M\rangle$ of the subset sum problem to an instance $I = \langle \mathcal{E} = \langle D = (V, A), \omega, d, q\rangle, v^*, k, \tau\rangle$ of **BMaxP** (resp. **SMaxP**). We create a tree D where:

- $V = V_1 \cup \{v, v', v^*\}$ with $V_1 = \{u_i : a_i \in S\}$.
- $A = \{(v_i, v) : v_i \in V \setminus \{v\}\}$.

Let $R = \sum_{a_i \in S} a_i$ and $q = R + 2M + 3$. The weight function ω is set as follows: $\omega(v') = R + M + 2$, $\omega(v^*) = 1$, $\omega(v) = M + 1$ and for any $u_i \in V_1$, $\omega(u_i) = a_i$. The initial delegation function d is set as follows: $d(v') = v'$, $d(v^*) = v^*$, $d(v) = v$ and for any $u_i \in V_1$, $d(u_i) = v$. As v^* is not swing in any coalition C for $\mathcal{E}$, $DB_{v^*}(\mathcal{E}) = 0$ (resp. $DS_{v^*}(\mathcal{E}) = 0$). First notice that all successful coalitions should contain voter v' and that v^* cannot be a swing agent in a coalition that contains both v and v'. In fact, to make v^* a swing for some coalition, one should select a subset U of voters from V_1 such that $\sum_{v \in U} \omega(v) = M$ and remove their delegations from v. Indeed, v^* is then a swing agent for the coalition $U \cup \{v'\}$.

Now, assume, for a contradiction, that there is a (not-necessarily constant) factor β ($0 < \beta \leq 1$) polynomial time approximation algorithm, $\mathcal{A}$, for **BMaxP** and **SMaxP**. Imagine using $\mathcal{A}$ several times with $k = 1$ to $k = n$, where $n = |S|$ in the subset sum instance. Thus, if J is a Yes-instance, then there exists a set $L \subseteq S$ such that $\sum_{a_i \in L} a_i = M$. Consider, one of minimal size k_{min}. Then, for $k = k_{min}$, $\mathcal{A}$ will necessarily output a delegation function to **BMaxP** (resp. **SMaxP**) for which $DB_{v^*}(\mathcal{E}') > 0$ (resp. $DS_{v^*}(\mathcal{E}') > 0$), and in which the set of delegations changed affects the voters $\{u_i : i \in L\}$. By investigating the solution, one can check in polynomial time if it indeed corresponds to a valid certificate for the subset sum problem. This concludes the proof. □

More positively, when the input is a complete graph, we can provide more positive results on the approximation viewpoint. We detail an algorithm, called GAMW, standing for Greedy Algorithm with Maximum Weight, which works as follows.

If v^* is a guru. It iteratively picks a guru g (different from v^*) with maximum accumulated weight (if there are several gurus with the same accumulated weight, it selects one arbitrarily), set $d(g) = v^*$ and $k = k - 1$. This process continues until $k = 0$ or no guru remains to delegate to v^*.

If v^* is a follower. Let $Del_d(j) = c_d(j, d^*(j)) \setminus \{j\}$ be the set of voters to which j delegates to directly or indirectly. GAMW distinguishes two subcases: i) $k \geq 2$, it assumes that v^* is a guru and sets $d(v^*) = v^*$ and $k = k - 1$. It then proceeds as when v^* is a guru, ii) $k = 1$, GAMW checks if $q - \sum_{i \in Del_{d_1}(v^*)} \omega(i) > 0$ (i.e., otherwise she is a dummy player.), then it finds a voter among the gurus $g \neq d^*_{v^*}$ and the voters who delegate directly to $Del_d(j)$ with the highest accumulated weight and make her delegate to v^*, otherwise (i.e., $q - \sum_{i \in Del_{d_1}(v^*)} \omega(i) \leq 0$) it sets $d(v^*) = v^*$.

Theorem 6. *GAMW is a factor $\frac{1}{2^{n-1}}$ (resp. $\frac{1}{n!}$) approximation algorithm for **BMaxP** (resp. **SMaxP**) on complete graphs.*

Sketch of proof. The intuition is that, as GAMW assigns a set of subtrees with the highest weight to v^* among all algorithms respecting the budget constraint, if there is no coalition C' for which v^* is swing in after applying GAMW, no algorithm can result in a better solution. In particular, suppose that we are given a delegation graph H_d and a guru $g_{max} \in Gu_d$ with the highest accumulated weight. Consider any losing coalition $C \subseteq V \setminus T_d(g_{max})$ in H_d. If the coalition $C' = C \cup T_d(g_{max})$ is not a winning coalition, then no guru $g \in Gu_d$ can make C a winning coalition as $\alpha_d(g_{max}) \geq \alpha_d(g)$. In case $k = 1$ and v^* is a follower, we consider any losing coalition $C \setminus T_d(v^*)$, where $Del_d(j) \subseteq C$ and use a similar argument. □

To conclude this subsection, we note that, as shown by Theorems 2, 3, 4, and 5, problems **BMinP**, **SMinP**, **BMaxP** and **SMaxP** are hard. We also believe that these problems are complex in the sense that the power measures they rely on can be hard to grasp for people not used to solution concepts from cooperative game theory. In the next subsection, instead of maximizing a power measure, we study a problem with a conceptually simpler objective as surrogate.

5.2 Voting weight modification by bribery

In this subsection, we investigate if we can modify at most k delegation choices to make the accumulated weight of a given voter i^* greater than or equal to a given threshold τ. We term this optimization problem **WMaxP** for Weight Maximization Problem. It is clear that problem **WMaxP** is related to problems **BMaxP** and **SMaxP** in the sense that a greater voting weight may result in a greater power measure value. However, it is well known from the literature on WVGs that this relation is limited as voters with sensibly different weights may have the same relative importance in the election. Less intuitively, it is even possible that if the given voter receives too much weight, we may end up in a situation where the voter's delegative power gets decreased (see Proposition 2).

We now formally introduce **WMaxP**. As this problem does not require to know the quota, we define a Partial LDE (PLDE) as a tuple $\mathcal{E} = \langle D = (V, A), \omega, d \rangle$, i.e., an LDE without a quota value.

Problems: **WMaxP**
Input: A PLDE $\mathcal{E} = \langle D = (V, A), \omega, d \rangle$, a voter $i^* \in V$, a budget $k \in \mathbb{N}$, and a threshold $\tau \in \mathbb{N}$.
Feasible Solution: A delegation function $d' \in \Delta(D)$ s.t. $|\{i \in V : d(i) \neq d'(i)\}| \leq k$ leading to a PLDE $\mathcal{E}' = \langle D, \omega, d' \rangle$.
Question: Can we find a solution d' such that $\alpha_{d'}(i^*) \geq \tau$.

We first provide a hardness result for **WMaxP** and an inapproximability result for the optimization variant of **WMaxP**, denoted by **OWMaxP**.

Theorem 7. ***WMaxP** is NP-complete and **OWMaxP** cannot be approximated with an approximation ratio better than $1 - 1/e$ if $P \neq NP$, even when all voters have weight one.*

To obtain more positive results, we consider both the approximation and the parameterized complexity viewpoints.

An approximation algorithm point of view. Interestingly, a variant of **OWMaxP**, called **DTO** (Directed Tree Orienteering), has been investigated by Ghuge and Nagarajan [20]. In **DTO**, we are given a directed graph $D = (V, A)$ with edge costs $c : A \to \mathbb{Z}^+$, a root vertex $r^* \in V$, a budget $B \in \mathbb{Z}^+$, and a weight function $p : V \to \mathbb{Z}^+$. For any subgraph G' of a given (directed or undirected) graph G, let $V(G')$ and $E(G')$ represent the set of nodes and edges in G'. The goal is to find an out-directed arborescence T^* rooted at r^* maximizing $p(V(T^*)) = \sum_{v \in V(T^*)} p(v)$ such that $\sum_{e \in E(T^*)} c(e) \leq B$. Ghuge and Nagarajan [20] provided a quasi-polynomial time $O(\frac{\log n'}{\log \log n'})$-approximation algorithm, where n' is the number of vertices in an optimal solution. The authors mentioned that this factor is tight for **DTO** in quasi-polynomial time. It is worth mentioning that Paul et al. [31] proposed a 2-approximation algorithm for the undirected version of **DTO**.

Here we show that any approximation algorithm for a variant of **DTO** can also be used with **OWMaxP**, preserving the approximation factor. In particular, consider instances $I' = \langle D' = (V', A'), r^*, c, p, B \rangle$, where $D' = (V', A')$ is a directed graph with edge costs $c : A' \to \{0, 1\}$, a root vertex $r^* \in V'$, a budget $B \in \mathbb{Z}^+$, and a weight function $p : V' \to \mathbb{Z}^+$. For any node $v \in V'$, there exists at most one incoming edge e of cost $c(e) = 0$. More importantly, there is no cycle C in D' with the total cost $\sum_{e \in E(C)} c(e) = 0$, i.e., there exists at least one edge $e \in C$ with $c(e) = 1$. We call **RDTO** this variant. Note that **RDTO** is not a special case of **DTO** investigated by Ghuge and Nagarajan [20] as in their case the costs on edges are at least one. **RDTO** coincides with **DTO** when each edge costs 1 in both problems.

Theorem 8. *Consider a parameter β, where $0 < \beta < 1$ (not-necessarily constant). The following statements are equivalent:*

(i) *There is an β-approximation algorithm for **RDTO**.*
(ii) *There is an β-approximation algorithm for **OWMaxP**.*

Proof. To prove that (i) implies (ii), we proceed as follows. Let $I = \langle D = (V, A), \omega, d, i^*, k\rangle$ be an instance of **OWMaxP**. We suppose that $d_{i^*} = i^*$; otherwise we define a new delegation graph d' such that $d'(i^*) = i^*$, $d'(i) = d(i)$ for any $i \in V \setminus \{i^*\}$ and $k = k - 1$. Indeed, i^* should be a guru to have a non-zero accumulated weight. Let $I' = \langle D' = (V', A'), r^*, c, p, B\rangle$ be an instance of **RDTO** obtained from I as follows. We set $V' = V$, $r^* = i^*$ and $A' = \{(i, j) : (j, i) \in A\}$, i.e., we reverse the edges. For any $e = (i, j) \in A'$ in I', if $d(j) = i$ in I, $c(e) = 0$; $c(e) = 1$ otherwise. Lastly, $B = k$ and $p(v) = \omega(v)$ for any $v \in V'$. Consider a solution T^* to **RDTO** on I'. We can simply reverse the edges in T^* and obtain a subtree of D rooted in $i^* = r^*$ with cost $c(T^*) \leq B = k$ that induces a delegation graph d'. We conclude by noticing that any edge $e = (i, j) \in E(T^*)$ either costs 0 if $d(j) = i$ or 1 otherwise and $\alpha_{d'}(i^*) = p(V(T^*))$. This concludes the first direction.

Now we show that (ii) implies (i). Let $I' = \langle D' = (V', A'), r^*, c, p, B\rangle$ be an instance of **RDTO**. Let $I = \langle D = (V, A), \omega, d, i^*, k\rangle$ be an instance of **OWMaxP** obtained from I' as follows. We set $V' = V$, $r^* = i^*$ and $A = \{(i, j) : (j, i) \in A'\}$, i.e., reversing edges. For any voter $i \in V \setminus \{i^*\}$ if there exists an incoming edge $e = (j, i) \in A'$ with cost $c(e) = 0$, we set $d(i) = j$; $d(i) = i$ otherwise. We set $d(i^*) = i^*$. For any $v \in V$, let $\omega(v) = p(v)$. Lastly, we set $k = B$. As there exists no cycle $C \in D'$ with $\sum_{e\in C} c(e) = 0$ and for any $i \in V'$ there is at most one incoming edge e with cost $c(e) = 0$, the resulting delegation graph H_d is feasible. Now consider another delegation graph $H_{d'}$ such that $|\{i \in V : d(i) \neq d'(i)\}| \leq k$. By reversing the edges in subtree rooted at i^* in $H_{d'}$ we get an arborescence T^* in D' that is rooted at r^* with $p(V(T^*)) = \alpha_{d'}(i^*)$. □

We now present a polynomial-time approximation algorithm for **OWMaxP** to achieve a trade-off between the violation of budget constraint and the approximation factor. Given an undirected graph $G = (V(G), E(G))$, a distinguished vertex $r \in V(G)$ and a budget B, where each vertex $v \in V(G)$ is assigned with a prize $p'(v)$ and a cost $c'(v)$. A graph G is called B-proper for the vertex r if the cost of reaching any vertex from r is at most B. Consider a subtree $T = (V(T), E(T))$ of G, where $V(T) \subseteq V(G)$ and $E(T) \subseteq E(G)$. Let $c'(T) = \sum_{v\in V(T)} c'(v)$ and $p'(T) = \sum_{v\in V(T)} p'(v)$. Let $\gamma = \frac{p'(T)}{c'(T)}$ be the prize-to-cost ratio of T. Bateni, Hajiaghayi and Liaghat [5] proposed a trimming process that leads to the following.

Lemma 1 (Lemma 3 in [5]). *Let T be a subtree rooted at r with the prize-to-cost ratio γ. Suppose the underlying graph is B-proper for r and for $\epsilon \in (0, 1]$ the cost of the tree is at least $\frac{\epsilon B}{2}$. One can find a tree T^* containing r with the prize-to-cost ratio at least $\frac{\epsilon\gamma}{4}$ such that $\epsilon B/2 \leq c'(T^*) \leq (1 + \epsilon)B$.*

We show that Lemma 1 can be applied to our case. Given an instance $I = \langle D = (V, A), \omega, d, i^* \in V, k\rangle$ of **OWMaxP**. We create an edge-cost directed graph $D_d = (V_d, A_d)$ respecting the delegation function d as follows: $V_d = V$, $A_d = A$, each vertex $v \in V_d$ is associated with a weight $\omega(v)$ and each edge $e = (i, j) \in A_d$, is associated with a cost $c(e) = 0$ if $d(i) = j$, $c(e) = 1$ otherwise. D_d is called the *d-edge-cost* graph of D. Let V' be all vertices in V_d such that the cost of reaching from any node $v' \in V'$ to i^* is at most k. We call subgraph $D' = (V', A')$ of D_d *k-appropriate* for i^* where $A' = V' \times V' \cap A_d$ (we make this definition to avoid confusions between the undirected and directed cases). Consider a subtree T of D'. Let $\omega(T) = \sum_{v\in V'(T)} \omega(v)$ and $c(T) = \sum_{e\in A'(T)} c(e)$. Let $\gamma = \frac{\omega(T)}{c(T)}$ be the weight-to-cost ratio of T.

Lemma 2. *Given an instance $I = \langle D = (V, A), \omega, d, i^*, k\rangle$ of **OWMaxP**. Consider the d-edge-cost graph D_d which is k-appropriate for i^*. Let T be a subtree of D_d rooted at i^* with the weight-to-cost ratio γ. Suppose that for $\epsilon \in (0, 1]$ $c(T) \geq \frac{\epsilon B}{2}$. One can find a tree T^* containing i^* with the weight-to-cost ratio at least $\frac{\epsilon\gamma}{4}$ such that $\epsilon B/2 \leq c'(T^*) \leq (1 + \epsilon)B$.*

Proof. Let $V(T)$ and $A(T)$ be the set of vertices and edges of T. Now we create another subtree $T' = (V'(T), A'(T))$ as follows:

- $V(T') = V(T) \cup V_1$ with $V_1 = \{v_e : e \in A(T)\}$.
- $A(T') = \{(i, v_e), (v_e, j) : e = (i, j) \in A(T)\}$.

Each vertex $v \in V(T') \cap V(T)$ (resp. $v_e \in V(T') \cap V_1$) is assigned with a prize $p'(v) = \omega(v)$ (resp. $p'(v_e) = 0$) and a cost $c'(v) = 0$ (resp. $c'(v_e) = c(e)$). Lemma 1 can be applied to the subtree $T' = (V(T'), A(T'))$, as the trimming process by Bateni, Hajiaghayi and Liaghat [5] only removes some subtrees of T' to reach the guarentees mentioned in Lemma 1. This completes the proof. □

Now we are ready to propose our approximation algorithm for **OWMaxP**, called VBAMW. Given an instance of **OWMaxP** $I = \langle D = (V, A), \omega, d, i^*, k\rangle$, VBAMW first creates the d-edge-cost graph $D_d = (V_d, A_d)$ which is also maximal inclusion-wise k-appropriate graph for i^*. Now VBAMW finds a spanning arborescence $T = (V(T), A(T))$ of D_d with minimum cost $c(T)$, using Edmonds' algorithm [15]. If $c(T) \leq (1 + \epsilon)k$, we are done. Suppose it is not the case. Let $\gamma = \frac{\omega(T)}{c(T)}$ be the weight-to-cost ratio of tree T. By Lemma 2, from tree T, VBAMW finds another subtree $T^* \subseteq T$ of the cost at most $(1 + \epsilon)k$ and the weight-to-cost ratio $\frac{\epsilon\gamma}{4}$.

Theorem 9. *VBAMW is a $\frac{\epsilon^2 k}{8n}$ approximation algorithm with the cost at most $(1 + \epsilon)k$ for **OWMaxP**.*

Proof. Let T be the spanning arborescence returned by Edmonds' algorithm [15] with weight-to-cost ratio γ. It is clear that $\omega(T) \geq OPT$, where OPT is the optimum weight to **OWMaxP**. By Lemma 2, VBAMW will find another subtree T^* of cost $\epsilon k/2 \leq c(T^*) \leq (1 + \epsilon)k$ and weight-to-cost ratio:

$$\frac{w(T^*)}{c(T^*)} \geq \frac{\epsilon\gamma}{4} \geq \frac{\epsilon\omega(T)}{4c(T)} \geq \frac{\epsilon}{4c(T)} OPT \geq \frac{\epsilon}{4n} OPT.$$

As $c(T^*) \geq \epsilon k/2$, we have $\omega(T^*) \geq \frac{\epsilon^2 k}{8n} OPT$, concluding the proof. □

A parameterized complexity point of view. We now define the two following parameters:

- We denote by $\bar{\texttt{req}} = \sum_{i\in V} \omega(i) - \tau$ the amount of voting weight that i^* does not need to reach the threshold τ;
- We denote by $\texttt{req} = \tau - \alpha_d(i^*)$ the amount of additional voting weight that i^* needs to reach the threshold τ.

We study the parameterized complexity of **WMaxP** w.r.t. these two parameters. It can indeed be expected that the problem becomes easier if one of them is small. If $\mathtt{r\bar{e}q}$ is small, then the combinations of voters that may not delegate to i^* in a solution d, such that $\alpha_d(i^*) \geq \tau$, are probably limited. Conversely, if $\mathtt{req}$ is small, then the number of voters that i^* needs an additional support of to reach τ is small. These intuitions indeed yield positive results (Theorems 10 and 12). These two parameters seem to be opposite from one another. Indeed a small value for parameter $\mathtt{r\bar{e}q}$ (resp. $\mathtt{req}$) indicates that reaching the threshold τ is probably hard (resp. easy). Parameter $\mathtt{r\bar{e}q}$ could for instance be small if $\tau = q$ and the election is conservative (i.e., q is close to $\sum_{i \in V} \omega(i)$). Meanwhile, parameter $\mathtt{req}$ can be small if i^* has already a large voting power.

We start with parameter $\mathtt{r\bar{e}q}$.

Theorem 10. ***WMaxP*** *is in XP with respect to parameter* $\mathtt{r\bar{e}q}$.

To prove this theorem, we need the following.

Lemma 3. ***WMaxP*** *can be solved in polynomial time if* $\mathtt{r\bar{e}q} = 0$.

Using Lemma 3, we prove Theorem 10.

Proof of Theorem 10. As voters' weights are positive integers, the maximum number of voters that i^* does not necessarily need the support of to reach the threshold is bounded by $\mathtt{r\bar{e}q}$. One can hence guess the set C of voters that are not required with $|C| \leq \mathtt{r\bar{e}q}$. Indeed, the number of possible guesses is bounded by $\frac{|V|^{\mathtt{r\bar{e}q}+1} - |V|}{|V|-1}$. Let $X \subset V$ be one such guess. Once these voters are removed from the instance, we obtain another instance of **WMaxP** in which (if the guess is correct) i^* should obtain the support of all other voters. This amounts to solving an instance of **WMaxP** where $\mathtt{r\bar{e}q} = 0$. Hence, it can be solved in polynomial time by Lemma 3. □

Hence, interestingly **WMaxP** can be solved in polynomial time if $\mathtt{r\bar{e}q}$ is bounded by a constant. Unfortunately, **WMaxP** is W[1]-hard w.r.t. $\mathtt{r\bar{e}q}$ and hence is unlikely to be FPT for this parameter.

Theorem 11. ***WMaxP*** *is W[1]-hard with respect to* $\mathtt{r\bar{e}q}$*, even when all voters have weight one.*

Proof. We design a parameterized reduction from the independent set problem. In the independent set problem, we are given a graph $G = (V, E)$ and an integer k and we are asked if there exists an independent set of size k. The independent set problem is W[1]-hard parameterized by k. From an instance $I = (G = (V, E), k)$ of the independence set problem, we create the following **WMaxP** instance. We create a digraph $D = (\bar{V}, \bar{A})$ where:

- $\bar{V} = U \cup W \cup \{i^*\}$ with $U = \{u_v : v \in V\}$ and $W = \{w_e, w_e^1, \dots, w_e^k : e \in E\}$.
- $\bar{A} = \{(u_v, i^*) : v \in V\} \cup \{(w_e, u_v) : e \in E, v \in V, v \in e\} \cup \{(w_e^1, w_e), \dots, (w_e^k, w_e) : e \in E\}$.

All voters have weight one. The initial delegation function is such that $d(x) = x$ for $x \in U \cup \{i^*\} \cup \{w_e : e \in E\}$ and $d(w_e^j) = w_e$ for each $j \in [k]$ and $e \in E$. The budget $\bar{k} = |E| + |V| - k$ and τ is set to $(k+1)|E| + |V| - k + 1$. Hence, $\mathtt{r\bar{e}q} = k$. We show that the instance of the independent set problem is a yes instance iff the instance of the **WMaxP** problem is a yes instance. To reach the threshold of τ, i^* necessarily needs the delegations of all voters w_e. This requires spending a budget of $|E|$ to make all voters w_e delegate to some voters in U (which should then delegate to i^*). Then, there only remains a budget $|V| - k$ to make these voters in U delegate to i^*. Hence, we can reach the threshold τ iff we can make all voters in $\{w_e : e \in E\}$ delegate to less than $|V| - k$ voters in U. This is possible iff I is a yes instance. □

Interestingly, **WMaxP** is FPT with respect to $\mathtt{req}$.

Theorem 12. ***WMaxP*** *is FPT with respect to* $\mathtt{req}$.

Proof sketch. Let $I = (\langle D = (V, A), \omega, d \rangle, i^*, k, \tau)$ be an instance of **WMaxP**. As voters' weights are positive integers, the maximum number of additional voters that i^* needs the support of to reach the threshold is bounded by $\mathtt{req}$. We first note that one can collapse the tree $T_d(i^*)$ in one vertex with weight $\alpha_d(i^*)$. Let us consider a delegation function d' such that $|\{i : d(i) \neq d'(i)\}| \leq k$, and $\alpha_{d'}(i^*) \geq \tau$ (assuming such a solution exists). A subtree of $T_{d'}(i^*)$ rooted in i^* with at most $\mathtt{req} + 1$ voters accumulates a voting weight greater than or equal to τ. Our FPT algorithm guesses the shape of this tree and then looks for this tree in D by adapting the color coding technique [1]. The idea is to color the graph randomly with $\mathtt{req} + 1$ colors. If the tree that we are looking for is present in graph D, it will be colored with the $\mathtt{req} + 1$ colors (i.e., one color per vertex) with some probability only dependent of $\mathtt{req}$. We say that such a tree is colorful. One can then resort to dynamic programming to find the best colorful tree rooted in i^* in D and which contains at most k arcs not in H_d. This algorithm can then be derandomized using families of perfect hash functions [1, 33]. □

6 CONCLUSION

Following a recent work by Zhang and Grossi [35], we investigated delegative simple games, a variant of weighted voting games in which agents' weights are derived from a transitive support structure. We proposed a pseudo-polynomial time algorithm to compute the Banzhaf and Shapley-Shubik measures for this class of cooperative games and investigated several of their properties highlighting that they could lead to manipulations, e.g., by changing the delegation structure underlying the game. From this observation, we investigated a bribery problem in which we aim to maximize/minimize the power/weight of a given voter. We showed that these problems are NP-hard to solve and provided some more positive results (from the algorithmic viewpoint) by resorting to approximation algorithms and parameterized complexity.

Several directions of future work are conceivable. First, for both destructive and constructive bribery problems, designing some algorithms with tighter approximation guarantees under some conditions is one direction. Second, it would be interesting to study bribery problems related to alternative, maybe finer power measures. For instance, it is known that the Banzhaf index can be decomposed into two parts (the Coleman measures), one that measures the ability to initiate action, and one other to prevent it [17].

ACKNOWLEDGMENTS

This work was partially supported by the Italian MIUR PRIN 2017 Project "ALGADIMAR" Algorithms, Games, and Digital Markets.

REFERENCES

[1] Noga Alon, Raphael Yuster, and Uri Zwick. 1995. Color-coding. *Journal of the ACM (JACM)* 42, 4 (1995), 844–856.

[2] Haris Aziz, Yoram Bachrach, Edith Elkind, and Mike Paterson. 2011. False-name manipulations in weighted voting games. *Journal of Artificial Intelligence Research* 40 (2011), 57–93.

[3] Yoram Bachrach, Evangelos Markakis, Ariel D. Procaccia, Jeffrey S. Rosenschein, and Amin Saberi. 2008. Approximating power indices. In *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal, May 12-16, 2008, Volume 2*, Lin Padgham, David C. Parkes, Jörg P. Müller, and Simon Parsons (Eds.). IFAAMAS, Estoril, 943–950.

[4] John F Banzhaf III. 1964. Weighted voting doesn't work: A mathematical analysis. *Rutgers L. Rev.* 19 (1964), 317.

[5] Mohammad Hossein Bateni, Mohammad Taghi Hajiaghayi, and Vahid Liaghat. 2018. Improved Approximation Algorithms for (Budgeted) Node-weighted Steiner Problems. *SIAM J. Comput.* 47, 4 (2018), 1275–1293.

[6] J. Behrens, A. Kistner, A. Nitsche, and B. Swierczek. 2014. *The principles of LiquidFeedback.* Interacktive Demokratie, Berlin.

[7] Paolo Boldi, Francesco Bonchi, Carlos Castillo, and Sebastiano Vigna. 2009. Voting in social networks. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, November 2-6, 2009*, David Wai-Lok Cheung, Il-Yeol Song, Wesley W. Chu, Xiaohua Hu, and Jimmy Lin (Eds.). ACM, Hong Kong, 777–786.

[8] Paolo Boldi, Francesco Bonchi, Carlos Castillo, and Sebastiano Vigna. 2011. Viscous democracy for social networks. *Commun. ACM* 54, 6 (2011), 129–137.

[9] Steven J Brams and Paul J Affuso. 1976. Power and size: A new paradox. *Theory and Decision* 7, 1-2 (1976), 29–56.

[10] Georgios Chalkiadakis, Edith Elkind, and Michael J. Wooldridge. 2011. *Computational Aspects of Cooperative Game Theory.* Morgan & Claypool Publishers.

[11] Georgios Chalkiadakis and Michael J. Wooldridge. 2016. Weighted Voting Games. In *Handbook of Computational Social Choice*, Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia (Eds.). Cambridge University Press, Cambridge, 377–396.

[12] Vincent Conitzer and Tuomas Sandholm. 2004. Computing Shapley Values, Manipulating Value Division Schemes, and Checking Core Membership in Multi-Issue Domains. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence, July 25-29, 2004, San Jose, California, USA*, Deborah L. McGuinness and George Ferguson (Eds.). AAAI Press / The MIT Press, California, 219–225.

[13] Gianlorenzo D'Angelo, Esmaeil Delfaraz, and Hugo Gilbert. 2022. Computation and Bribery of Voting Power in Delegative Simple Games. arXiv:2104.03692

[14] Xiaotie Deng and Christos H Papadimitriou. 1994. On the complexity of cooperative solution concepts. *Mathematics of operations research* 19, 2 (1994), 257–266.

[15] Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the national Bureau of Standards B* 71, 4 (1967), 233–240.

[16] Shaheen S Fatima, Michael Wooldridge, and Nicholas R Jennings. 2010. An approximation method for power indices for voting games. In *Innovations in Agent-Based Complex Automated Negotiations.* Springer, Berlin, 179–194.

[17] Dan S. Felsenthal and Moshé Machover. 1998. *The Measurement of Voting Power.* Number 1489 in Books. Edward Elgar Publishing, Bodmin.

[18] Bryan Ford. 2020. A Liquid Perspective on Democratic Choice. arXiv:2003.12393 [cs.CY]

[19] M. R. Garey and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, United States.

[20] Rohan Ghuge and Viswanath Nagarajan. 2020. Quasi-Polynomial Algorithms for Submodular Tree Orienteering and Other Directed Network Design Problems. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, Shuchi Chawla (Ed.). SIAM, Salt Lake City, 1039–1048.

[21] Samuel Ieong and Yoav Shoham. 2005. Marginal contribution nets: a compact representation scheme for coalitional games. In *Proceedings 6th ACM Conference on Electronic Commerce (EC-2005), Vancouver, BC, Canada, June 5-8, 2005*, John Riedl, Michael J. Kearns, and Michael K. Reiter (Eds.). ACM, Vancouver, 193–202.

[22] Samuel Ieong and Yoav Shoham. 2006. Multi-attribute coalitional games. In *Proceedings 7th ACM Conference on Electronic Commerce (EC-2006), Ann Arbor, Michigan, USA, June 11-15, 2006*, Joan Feigenbaum, John C.-I. Chuang, and David M. Pennock (Eds.). ACM, Michigan, 170–179.

[23] Christoph Carl Kling, Jérôme Kunegis, Heinrich Hartmann, Markus Strohmaier, and Steffen Staab. 2015. Voting Behaviour and Power in Online Democracy: A Study of LiquidFeedback in Germany's Pirate Party. In *Proceedings of the Ninth International Conference on Web and Social Media, ICWSM 2015, University of Oxford, Oxford, UK, May 26-29, 2015*, Meeyoung Cha, Cecilia Mascolo, and Christian Sandvig (Eds.). AAAI Press, Oxford, 208–217.

[24] Annick Laruelle and Federico Valenciano. 2005. A critical reappraisal of some voting power paradoxes. *Public Choice* 125, 1-2 (2005), 17–41.

[25] Dennis Leech. 2003. Computing power indices for large voting games. *Management Science* 49, 6 (2003), 831–837.

[26] Irwin Mann and Lloyd S. Shapley. 1960. *Values of Large Games, IV: Evaluating the Electoral College by Montecarlo Techniques.* RAND Corporation, Santa Monica, CA.

[27] Tomomi Matsui and Yasuko Matsui. 2000. A survey of algorithms for calculating power indices of weighted majority games. *Journal of the Operations Research Society of Japan* 43, 1 (2000), 71–86.

[28] Yasuko Matsui and Tomomi Matsui. 2001. NP-completeness for calculating power indices of weighted majority games. *Theor. Comput. Sci.* 263, 1-2 (2001), 305–310.

[29] Samuel Merrill III. 1982. Approximations to the Banzhaf index of voting power. *The American Mathematical Monthly* 89, 2 (1982), 108–110.

[30] S. Muroga. 1971. *Threshold Logic and Its Applications.* Wiley-Interscience. https://books.google.it/books?id=wvtQAAAAMAAJ

[31] Alice Paul, Daniel Freund, Aaron M. Ferber, David B. Shmoys, and David P. Williamson. 2020. Budgeted Prize-Collecting Traveling Salesman and Minimum Spanning Tree Problems. *Math. Oper. Res.* 45, 2 (2020), 576–590.

[32] Kislaya Prasad and Jerry S Kelly. 1990. NP-completeness of some problems concerning voting games. *International Journal of Game Theory* 19, 1 (1990), 1–9.

[33] Jeanette P Schmidt and Alan Siegel. 1990. The spatial complexity of oblivious k-probe hash functions. *SIAM J. Comput.* 19, 5 (1990), 775–786.

[34] Lloyd S Shapley. 1953. A value for n-person games. *Contributions to the Theory of Games* 2, 28 (1953), 307–317.

[35] Yuzhe Zhang and Davide Grossi. 2021. Power in Liquid Democracy. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021.* AAAI Press, 5822–5830.

[36] Yair Zick, Alexander Skopalik, and Edith Elkind. 2011. The Shapley Value as a Function of the Quota in Weighted Voting Games. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, Toby Walsh (Ed.). IJCAI/AAAI, Barcelona, 490–496.

[37] Michael Zuckerman, Piotr Faliszewski, Yoram Bachrach, and Edith Elkind. 2012. Manipulating the quota in weighted voting games. *Artificial Intelligence* 180 (2012), 1–19.