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 $c(T^*) \le B = k$ 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 $|\{i \in V : d(i) \ne d'(i)\}| \le k$. 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 $\epsilon B/2 \le c'(T^*) \le (1 + \epsilon)B$.*

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 $\epsilon \in (0, 1]$ $c(T) \ge \frac{\epsilon B}{2}$. One can find a tree T^* containing i^* with the weight-to-cost ratio at least $\frac{\epsilon\gamma}{4}$ such that $\epsilon B/2 \le c'(T^*) \le (1 + \epsilon)B$.*

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 $c(T) \le (1 + \epsilon)k$, 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 $\omega(T) \ge OPT$, where OPT is the optimum weight to **OWMaxP**. By Lemma 2, VBAMW will find another subtree T^* of cost $\epsilon k/2 \le c(T^*) \le (1 + \epsilon)k$ and weight-to-cost ratio:

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

As $c(T^*) \ge \epsilon k/2$, we have $\omega(T^*) \ge \frac{\epsilon^2 k}{8n}OPT$, 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 $W = \{w_e, w_e^1, \ldots, w_e^k : e \in E\}$.
- $\bar{A} = \{(u_v, i^*) : v \in V\} \cup \{(w_e, u_v) : e \in E, v \in V, v \in e\} \cup \{(w_e^1, w_e), \ldots, (w_e^k, w_e) : e \in E\}$.

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.