Divide and Conquer: False-Name Manipulations in Weighted Voting Games

Yoram Bachrach and Edith Elkind

Abstract

Weighted voting is a well-known model of cooperation among agents in decision-making domains. In such games, each player has a weight, and a coalition of players wins if its total weight meets or exceeds a given quota. Usually, the agents' power in such games is measured by a *power index*, such as, e.g., Shapley–Shubik index. In this paper, we study how an agent can manipulate its voting power (as measured by Shapley–Shubik index) by distributing his weight among several false identities. We show that such manipulations can indeed increase an agent's power and provide upper and lower bounds on the effects of such manipulations. We then study this issue from the computational perspective, and show that checking whether a beneficial split exists is NP-hard. We also discuss efficient algorithms for restricted cases of this problem, as well as randomized algorithms for the general case.

1 Introduction

Collaboration and cooperative decision-making are important issues in many types of interactions among self-interested agents. In many situations, agents must take a joint decision leading to a certain outcome, which may have a different impact on each of the agents. A standard and well-studied way of doing so is by means of voting, and in recent years, there has been a lot of research on applications of voting to multiagent systems as well as on computational aspects of various voting procedures. One of the key issues in this domain is how to measure the *power* of each voter, i.e., his impact on the final outcome. In particular, this question becomes important when the agents have to decide how to distribute the payoffs resulting from their joint action: a natural approach would be to pay each agent according to his contribution, i.e., his voting power.

This issue is traditionally studied within the framework of *weighted voting games*, which provide a model of decision-making in many political and legislative bodies, and have also been applied in the context of multiagent systems. In such a game, each of the agents has a weight, and a coalition of agents wins the game if the sum of the weights of its participants exceeds a certain quota. Having a larger weight makes it easier for an agent to affect the outcome; however, the agent's power is not always proportional to his weight. For example, if the quota is so high that the only winning coalition is the one that includes all agents, intuitively, all agents have equal power, irrespective of their weight. This idea is formalized using the concept of a *power index*, which is a systematic way of measuring a player's influence in a weighted voting game. There are several ways to define power indices. One of the most popular approaches relies on the fact that weighted voting games form a subclass of *coalitional games*, and therefore one can use the terminology and solution concepts that have been developed in the context of general coalitional games. In particular, an important notion in coalitional games is that of the Shapley value [12], which is a classical way to distribute the gains of the grand coalition in general coalitional games. In the context of weighted voting, the Shapley value (also known as the Shapley–Shubik power index [13]) provides a convenient measure of an agent's power and has been widely studied from both a normative and a computational perspective.

As suggested above, power indices measure the agents' power and can be used to determine their payoffs. However, to be applicable in real-world scenarios, this approach has to be resistant to dishonest behavior, or manipulation, by the participating agents. In this paper we study the effects of one form of manipulation in weighted voting games, namely, false-name voting. Under this manipulation, an agent splits his weight between himself and a "fake" agent who enters the game. While the total weight of all identities of the cheating agent remains the same, his power (as measured by the Shapley–Shubik power index) may change. In open anonymous environments, such as the internet, this behavior is virtually impossible to detect, and therefore it presents a challenge to the designers of multiagent systems that rely on weighted voting. The goal of this paper is to measure the effects of false-name voting and analyze its computational feasibility. Our main results here are as follows:

- We precisely quantify the worst-case effect of false-name voting on agents' payoffs. Namely, we show that in an n-player game, false-name voting can increase an agent's payoff by a factor of $2n/(n+1)$, and this bound is tight. On the other hand, we show that false-name voting can decrease an agent's payoff by a factor of $(n+1)/2$, and this bound is also tight.
- We demonstrate that finding a successful manipulation is not a trivial task by proving that it is NP-hard to verify if a beneficial split exists. However, we show that if all weights are polynomially bounded, the problem can be solved in polynomial time, and discuss efficient randomized algorithms for this problem.

We also study several variants of the problem, such as splitting into more than two identities, as well as the dual problem of manipulation by merging, where several agents pretend to be one.

2 Related Work

In his seminal paper, Shapley [12] considered coalitional games and the question of fair allocation of the utility gained by the grand coalition. The solution concept introduced in this paper became known as the *Shapley value* of the game. The subsequent paper [13] studies the Shapley value in the context of simple coalitional games, where it is usually referred to as the *Shapley–Shubik power index*. Another measure of a player's influence in voting games is the Banzhaf power index [1].

Both of these power indices have been well studied. Straffin [14] shows that each index reflects certain conditions in a voting body. Paper [5] describes certain axioms that characterize these two indices, as well as several others. These indices were used to analyze the voting structures of the European Union Council of Ministers and the IMF [7, 6].

The applicability of the power indices to measuring political power in various domains has raised the question of finding tractable ways to compute them. However, this problem appears to be computationally hard. Indeed, the naive algorithm for calculating the Shapley value (or the Shapley–Shubik power index) considers all permutations of the players and hence runs in exponential time. Moreover, paper [3] shows that computing the Shapley value in weighted voting games is #P-complete.

Despite this hardness result, several works show how to compute these power indices in some *restricted domains*, or discuss ways to *approximate* them [9, 11, 8, 4]. A good survey of algorithms for calculating power indices in weighted voting games is [10]. Many of these approaches work well in practice, which justifies the use of these indices as payoff distribution schemes in multiagent domains.

False-name manipulation has been studied in the context of non-cooperative games such as auctions [16, 17], and, more recently, also in cooperative games [2, 15]. However, to the best of our knowledge, ours is the first paper to systematically study this type of behavior in weighted voting.

3 Preliminaries and Notation

Coalitional Games A *coalitional game* $G = (I, v)$ is given by a set of agents $I = \{a_1, \ldots, a_n\}$, $|I| = n$, and a function $v : 2^I \to \mathbb{R}$ that maps any subset (coalition) of the agents to a real value. This value is the total utility these agents can guarantee to themselves when working together. To simplify notation, we will sometimes write i instead of a_i.

A coalitional game is *simple* if v can only take values 0 and 1, i.e., $v : 2^I \to \{0, 1\}$. In such games, we say that a coalition $C \subseteq I$ *wins* if $v(C) = 1$, and *loses* if $v(C) = 0$. An agent i is *critical*, or *pivotal*, to a winning coalition C if the agent's removal from that coalition would make it a losing coalition: $v(C) = 1$, $v(C \setminus \{i\}) = 0$.

Weighted Voting Games A *weighted voting game* G is a simple game that is described by a vector of players' *weights* $\mathbf{w} = (w_1, \ldots, w_n)$ and a *quota* q. We write $G = [w_1, \ldots, w_n; q]$, or $G = [\mathbf{w}; q]$. In these games, a coalition is winning if its total weight meets or exceeds the quota. Formally, for any $J \subseteq I$ we have $v(J) = 1$ if $\sum_{i \in J} w_i \geq q$ and $v(J) = 0$ otherwise. We will often write $w(J)$ to denote the total weight of a coalition J, i.e., $w(J) = \sum_{i \in J} w_i$. Also, we set $w_{\max} = \max_{i=1,\ldots,n} w_i$.

Shapley Value Intuitively, the Shapley value of an agent is determined by his marginal contribution to possible coalitions. Let Π_n be the set of all possible permutations (orderings) of n agents. Each $\pi \in \Pi_n$ is a one-to-one mapping from $\{1, \ldots, n\}$ to $\{1, \ldots, n\}$. Denote by $S_\pi(i)$ the predecessors of agent i in π, i.e., $S_\pi(i) = \{j \mid \pi(j) < \pi(i)\}$. The Shapley value of the ith agent in a game $G = (I, v)$ is denoted by $\varphi_G(i)$ and is given by the following expression:

$$\varphi_G(i) = \frac{1}{n!} \sum_{\pi \in \Pi_n} [v(S_\pi(i) \cup \{i\}) - v(S_\pi(i))]. \tag{1}$$

We will occasionally abuse notation and say that an agent i is pivotal for a permutation π if it is pivotal for the coalition $S_\pi(i) \cup \{i\}$.

The Shapley-Shubik power index is simply the Shapley value in a simple coalitional game (and therefore in the rest of the paper we will use these terms interchangeably). In such games the value of a coalition is either 0 or 1, so the formula (1) simply counts the fraction of all orderings of the agents in which agent i is critical for the coalition formed by his predecessors and himself. The Shapley-Shubik power index thus reflects the assumption that when forming a coalition, any *ordering* of the agents entering the coalition has an equal probability of occurring, and expresses the probability that agent i is critical.

While there exist several other approaches to determining the players' influence in a game, the Shapley value has many useful properties that make it very convenient to work with. We will make use of two of these properties, namely, the normalization property and the dummy player property. The former simply states that the sum of Shapley values of all players is equal to 1. The latter claims that the value of a dummy player is 0, where a player i is called a *dummy* if he contributes nothing to any coalition, i.e., for any $C \subseteq I$ we have $v(C \cup \{i\}) = v(C)$. It is easy to verify from the definitions that Shapley value has both of these properties.

4 False-Name Manipulations

As discussed in the introduction, it might be possible for a player to change his total payoff by splitting his weight between several identities. We will start by providing a few examples of such scenarios.

Example 1. *Consider a voting game $G = [8, 8, 1, 2; 11]$, i.e., a game with a quota of $q = 11$, and four agents $a_1, \dots, a_4$, where a_1 has weight $w_1 = 8$, a_2 has weight $w_2 = 8$, a_3 has weight $w_3 = 1$, and agent a_4 has weight $w_4 = 2$.*

Using formula (1) to compute the Shapley value of a_4, we get $\varphi_G(a_4) = 4/24$. Now, suppose agent a_4 splits his weight equally between two new identities a'_4 and a''_4, resulting in a new game $G' = [8, 8, 1, 1, 1; 11]$. Calculating the Shapley value of a'_4 and a''_4 in this game, we get $\varphi_{G'}(a'_4) = \varphi_{G'}(a''_4) = 12/120$. Hence, although the total weight *of all identities of player a_4 is the same as in the original game, the total* power *held by the agent and his false-name "accomplice" has increased, since $\varphi_{G'}(a'_4) + \varphi_{G'}(a''_4) = 24/120 > \varphi_G(a_4) = 4/24$.*

Our next example shows that false-name manipulations are not necessarily beneficial.

Example 2. *Consider a game $G = [3, 3, 2; 4]$. Agent a_3 with the weight of $w_3 = 2$ has Shapley value of $\varphi_G(a_3) = 2/6$. Splitting his weight between two identities a'_3 and a''_3 results in a game $G' = [3, 3, 1, 1; 4]$. The Shapley values of the two identities a'_3 and a''_3 of the agent a_3 in the new game are $\varphi_{G'}(a'_3) = \varphi_{G'}(a''_3) = 4/24$, so we have $\varphi_G(a_3) = \varphi_{G'}(a'_3) + \varphi_{G'}(a''_3)$. Consequently, a_3 neither gained nor lost power by splitting.*

Moreover, weight-splitting can be risky for the manipulator, as illustrated by the following example.

Example 3. *Consider a game $G = [2, 2, 2; 5]$. Agent a_3 with the weight of $w_3 = 2$ has Shapley value of $\varphi_G(a_3) = 2/6$. By splitting into two agents, each with a weight of 1, this agent can get the following game $G' = [2, 2, 1, 1; 5]$. The Shapley values of the two agents a'_3 and a''_3 in the new game are $\varphi_{G'}(a'_3) = \varphi_{G'}(a''_3) = 2/24$, so we have $\varphi_G(a_3) > \varphi_{G'}(a'_3) + \varphi_{G'}(a''_3)$. Hence, the splitting agent has lost power by splitting, i.e., it was harmful for him to split.*

4.1 Effects of Manipulation: Upper and Lower Bounds

We have seen that an agent can both increase and decrease his total payoff by splitting his weight. In this subsection, we provide upper and lower bounds on how much he can change his payoff by doing so. We restrict our attention to the case of splitting into two identities; the general case is briefly discussed in Section 7.

To simplify notation, in the rest of this section we assume that in the original game $G = [w_1, \dots, w_n; q]$ the manipulator is agent a_n, and he splits into two new identities a'_n and a''_n, resulting in a new game G'.

Theorem 4. *For any game $G = [w_1, \dots, w_n; q]$ and any split of a_n's into a'_n and a''_n, we have $\varphi_{G'}(a'_n) + \varphi_{G'}(a''_n) \le \frac{2n}{n+1}\varphi_G(a_n)$, i.e., the manipulator cannot gain more than a factor of $2n/(n+1) < 2$ by splitting his weight between two identities. Moreover, this bound is tight, i.e., there exists a game in which agent a_n increases his payoff by a factor of $2n/(n+1)$ by splitting into two identities.*

Proof. Fix a split of a_n into a'_n and a''_n. Let Π_{n-1} be the set of all permutations of the first $n-1$ agents. Consider any $\pi \in \Pi_{n-1}$. Let $P(\pi)$ be the set of all permutations of the agents in G' that can be obtained by inserting a'_n and a''_n into π. Let Π^*_{n+1} be the set of all permutations π^* of agents in G' such that a'_n or a''_n is pivotal for π^*. Finally, Let $P^*(\pi, k)$

be the subset of $P(\pi) \cap \Pi^*_{n+1}$ that consists of all permutations $\pi' \in P(\pi)$ in which at least one of the a'_n and a''_n appears between the kth and the $(k+1)$st element of π' and is pivotal for π'. Every permutation in Π^*_{n+1} appears in one of the sets $P^*(\pi, k)$ for some π, k, so we have

$$\varphi_{G'}(a'_n) + \varphi_{G'}(a''_n) = \frac{|\Pi^*_{n+1}|}{(n+1)!} \leq \frac{1}{(n+1)!} \sum_{\pi,k} |P^*(\pi, k)|.$$

On the other hand, it is not hard to see that $|P^*(\pi, k)| \leq 2n$ for any π, k: there are two ways to place a'_n and a''_n between the kth and the $(k+1)$st element of π, $n-1$ permutations in $P^*(\pi, k)$ in which a'_n appears after the kth element of π, but a''_n is not adjacent to it, and $n-1$ permutations in $P^*(\pi, k)$ in which a''_n appears after the kth element of π, but a'_n is not adjacent to it. Moreover, if $P^*(\pi, k)$ is not empty, then a_n is pivotal for the permutation $f(\pi, k)$ obtained from π by inserting a_n after the kth element of π. Moreover, if $(\pi_1, k_1) \neq (\pi_2, k_2)$ then $f(\pi_1, k_1) \neq f(\pi_2, k_2)$. Hence,

$$\varphi_G(a_n) \geq \frac{1}{n!} \sum_{\pi,k: P^*(\pi,k) \neq \emptyset} 1 \geq \frac{1}{n! \cdot 2n} \sum_{\pi,k} |P^*(\pi, k)| \geq \frac{n+1}{2n} (\varphi_{G'}(a'_n) + \varphi_{G'}(a''_n)).$$

We conclude that the manipulator cannot gain more than a factor of $2n/(n+1) < 2$ by splitting his weight between two identities.

To see that this bound is tight, consider the game $G = [2, 2, \ldots, 2; 2n]$ and suppose that one of the agents (say, a_n) decides to split into two identities a'_n and a''_n resulting in the game $G' = [2, \ldots, 2, 1, 1; 2n]$. Clearly, in both games the only winning coalition consists of all agents, so we have $\varphi_G(a_n) = 1/n$, $\varphi_{G'}(a'_n) = \varphi_{G'}(a''_n) = 1/(n+1)$, i.e., $\varphi_{G'}(a'_n) + \varphi_{G'}(a''_n) = \frac{2n}{n+1} \varphi_G(a_n)$. □

We have seen that no agent can increase his payoff by more than a factor of 2 by splitting his weight between two identities. In contrast, we will now show that an agent can decrease his payoff by a factor of $\Theta(n)$ by doing so. This shows that a would-be manipulator has to be careful when deciding whether to split his weight, and motivates the algorithmic questions studied in the next two sections.

Theorem 5. *In any weighted voting game, no agent can lower his payoff by more than a factor of $(n+1)/2$ by splitting his weight between two identities. Moreover, there exists a weighted voting game in which splitting into two identities decreases the manipulator's payoff by a factor of $(n+1)/2$.*

Proof. To prove the first part of the theorem, fix a split of a_n into a'_n and a''_n and consider any permutation π of agents in G such that a_n is pivotal for π. It is easy to see that at least one of a'_n and a''_n is pivotal for the permutation $f(\pi)$ obtained from π by replacing a_n with a'_n and a''_n (in this order). Similarly, at least one of a'_n and a''_n is pivotal for the permutation $g(\pi)$ obtained from π by replacing a_n with a''_n and a'_n (in this order). Moreover, all permutations of agents in G' obtained in this manner are distinct, i.e., for any π, π' we have $g(\pi) \neq f(\pi')$, and $\pi \neq \pi'$ implies $f(\pi) \neq f(\pi')$, $g(\pi) \neq g(\pi')$. Consequently, if Π^*_n is the set of all permutations π of the agents in G such that a_n is pivotal for π, and Π^*_{n+1} is the set of all permutations π of the agents in G' such that a'_n or a''_n is pivotal for π, we have $|\Pi^*_{n+1}| \geq 2|\Pi^*_n|$ and

$$\varphi_{G'}(a'_n) + \varphi_{G'}(a''_n) = \frac{|\Pi^*_{n+1}|}{(n+1)!} \geq \frac{2|\Pi^*_n|}{(n+1)!} = \frac{2}{n+1} \varphi_G(a_n).$$

To see that this bound is tight, consider the game $G = [2, 2, \ldots, 2; 2n-1]$ and suppose that one of the agents (say, a_n) decides to split into two identities a'_n and a''_n resulting in the game $G' = [2, \dots, 2, 1, 1; 2n-1]$. In the original game G, the only winning coalition consists of all agents, so we have $\varphi_G(a_n) = 1/n$. Now, consider any permutation π of the players in G'. We claim that a'_n is pivotal for π if and only if it appears in the nth position of π, followed by a''_n. Indeed, if $\pi(a'_n) = n$, $\pi(a''_n) = n+1$, then all players in the first $n-1$ positions have weight 2, so $w(S_\pi(a'_n)) = 2n-2$, $w(S_\pi(a'_n) \cup \{a'_n\}) = 2n-1$. Conversely, if $\pi(a'_n) = n+1$, we have $w(S_\pi(a'_n)) = 2n-1 = q$, and if $\pi(a'_n) \leq n-1$, we have $w(S_\pi(a'_n) \cup \{a'_n\}) \leq 2(n-1)$. Finally, if $\pi(a'_n) = n$, but $\pi(a''_n) \neq n+1$, we have $w(S_\pi(a'_n) \cup \{a'_n\}) = 2n-2 < q$. Consequently, a'_n is pivotal for $(n-1)!$ permutations, and, by the same argument, a''_n is also pivotal for (a disjoint set of) $(n-1)!$ permutations. Hence, we have $\varphi_{G'}(a'_n) + \varphi_{G'}(a''_n) = \frac{2(n-1)!}{(n+1)!} = \frac{2}{n+1}\varphi_G(a_n)$. □

5 Complexity of Finding a Beneficial Manipulation

We now examine the problem of finding a beneficial weight split in weighted voting games from the computational perspective. Ideally, the manipulator would like to find a payoff-maximizing split, i.e., a way to split his weight between two or more identities that results in the maximal total payoff. A less ambitious goal is to decide whether there exists a manipulation that increases the manipulator's payoff. However, it turns out that even this problem is computationally hard: in the rest of the section, we will show that checking whether there exists a payoff-increasing split is NP-hard, even if the player is only allowed to use two identities. To formally define the computational problem, we assume that all weights and the quota are integer numbers given in binary.

Definition 6. *An instance of* Beneficial Split *is given by a weighted voting game $G = [w_1, \dots, w_n; q]$ and a certain target agent a_i. We are asked if there is a way for a_i to split his weight w_i between several new agents $a_i^{(1)}, \dots, a_i^{(k)}$ so that the sum of their Shapley values is greater than the Shapley value of a_i. Formally, an instance (G, a_i) is a "yes"-instance if there exists a $k \geq 2$ and weights $w_i^{(1)}, \dots, w_i^{(k)}$ such that $\sum_{j=1,\dots,k} w_i^{(j)} = w_i$ and in the new game*

$$G' = [w_1, \dots, w_{i-1}, w_i^{(1)}, \dots, w_i^{(k)}, w_{i+1}, \dots, w_n; q]$$

we have $\varphi_{G'}(a_i^{(1)}) + \dots + \varphi_{G'}(a_i^{(k)}) > \varphi_G(a_i)$.

Remark 7. *Note that we are looking for a* strictly *beneficial manipulation, i.e., one that* increases *the total Shapley value of the manipulator. Indeed, if we were just interested in a split that is not harmful, the problem would always have a trivial solution: by the dummy axiom, assigning a weight of 0 to the new agent does not affect the Shapley value of all agents and therefore is not harmful.*

Our hardness proof is by a reduction from Partition, which is a classical NP-complete problem. An instance of Partition is given by a set of n weights $T = \{t_1, \dots, t_n\}$. It is a "yes"-instance if it is possible to split T into two subsets $P_1 \subseteq T$, $P_2 \subseteq T$ so that $P_1 \cap P_2 = \emptyset$, $P_1 \cup P_2 = T$, and $\sum_{t_i \in P_1} t_i = \sum_{t_i \in P_2} t_i$, and a "no"-instance if no such partition exists.

The high-level idea of the reduction is as follows. Given an instance of Partition $T = \{t_1, \dots, t_n\}$, we create a weighted voting game G with $n+2$ agents $a_1, \dots, a_n, a_x, a_y$, weights $\mathbf{w} = (8t_1, ..., 8t_n, 1, 2)$, and a quota of $q = 4\sum_i t_i + 3$. The weights are chosen so that a_y is a dummy if the original instance of Partition is a "no"-instance, but has some power if it is a "yes"-instance. Moreover, when a partition exists, agent a_y can gain power by splitting into two agents of weight 1 each. It follows that T is a "yes"-instance of Partition if and only if (G, a_y) a "yes"-instance of Beneficial Split. For the rest of the proof, we write $A = \{a_1, \dots, a_n\}$ and for any $A' \subseteq A$ we set $w(A') = \sum_{i: a_i \in A'} w_i$.

Lemma 8. *If T is a "no"-instance of* PARTITION*, then agent a_y is a dummy player.*

Proof. Suppose that T is a "no"-instance of PARTITON. Consider any $A' \subseteq A$. The set A can be partitioned into two equal-weight subsets if and only if T can, so either $w(A') < w(A)/2$, or $w(A') > w(A)/2$. We will show that in either case a_y cannot be critical to either $A' \cup \{a_y\}$ or $A' \cup \{a_x, a_y\}$, i.e., a_y is dummy player.

The weights of all agents in A are multiples of 8, so $w(A)/2$ is a multiple of 4. Similarly, the weight of A' is a multiple of 8. Hence, if $w(A') < w(A)/2$, it follows that $w(A') \leq w(A)/2 - 4$ and $w(A' \cup \{a_x, a_y\}) \leq w(A)/2 - 4 + 3 < q$. Therefore, $A' \cup \{a_x, a_y\}$ (and *a forteriori* $A' \cup \{a_y\}$) is not a winning coalition, so a_y cannot be pivotal for it.

Similarly, if $w(A') > w(A)/2$, then $w(A') \geq w(A)/2 + 4 > q$, so A' is a winning coalition. Therefore a_y cannot be critical for $A' \cup \{a_y\}$ or $A' \cup \{a_x, a_y\}$. We conclude that by the dummy axiom the Shapley value of a_y is 0. □

Corollary 9. If T is a "no"-instance of PARTITION, then (G, a_y) is a "no"-instance of BENEFICIAL SPLIT.

Proof. By Lemma 8, if T is a "no"-instance of PARTITION, the agent a_y is a dummy in G. Now, take any possible split of a_y into several agents $a_y^{(1)}, \ldots, a_y^{(k)}$ and consider any permutation π of the agents in the new game. If there is a $j \leq k$ such that $a_y^{(j)}$ is pivotal for π, then a_y is pivotal for the permutation obtained from π by deleting all agents $a_y^{(l)}$ with $l \neq j$ and replacing $a_y^{(j)}$, with a_y, a contradiction. We conclude that all $a_y^{(j)}$, $j = 1, \ldots, k$ are dummy players and hence their total Shapley value is 0. Therefore, a_y gains no power by splitting and (G, a_y) is a "no"-instance of BENEFICIAL SPLIT. □

Lemma 10. *If T is a "yes"-instance of* PARTITION*, then a_y can increase his power by splitting into two agents, i.e., (G, a_y) is a "yes"-instance of* BENEFICIAL SPLIT.

Proof. Let T be a "yes"-instance of PARTITION. Let $\langle P_1, P_2 \rangle$ be a partition of T, so $w(P_1) = w(P_2)$. It corresponds to a partition $\langle A_1, A_2 \rangle$ of A, where $a_i \in A_1$ if and only if $t_i \in P_1$; obviously, we have $w(A_1) = w(A_2)$. We denote $|A_1| = s$, so $|A_2| = n - s$.

It is easy to see that a_y is critical for $A_1 \cup \{a_x, a_y\}$ as well as for $A_2 \cup \{a_x, a_y\}$. There are $(s+1)!(n-s)!$ permutations of $a_1, \ldots, a_n, a_x, a_y$ that put a_y directly after some permutation of $A_1 \cup \{a_x\}$. Similarly, there are $s!(n-s+1)!$ permutations putting a_y directly after some permutation of $A_2 \cup \{a_x\}$. Thus, for each partition $X_i = \langle P_1^i, P_2^i \rangle$, where $|P_1^i| = s$, we have at least $(s+1)!(n-s)! + s!(n-s+1)!$ distinct permutations where a_y is critical. On the other hand, as shown in Lemma 8, if A' is a subset of A such that $w(A') \neq w(A)/2$, then a_y is not critical for $A' \cup \{a_y\}$ or $A' \cup \{a_y, a_x\}$, since either $w(A') \leq w(A)/2 - 4 < q - 3$ or $w(A') \geq w(A)/2 + 4 > q$.

Let $\mathcal{P}$ be the set of all partitions of T, where each partition is counted only once, i.e., $\mathcal{P}$ contains exactly one of the $\langle P_1, P_2 \rangle$ and $\langle P_2, P_1 \rangle$. For each $P_i = \langle P_1^i, P_2^i \rangle \in \mathcal{P}$, we denote $|P_1^i| = s_i$. There is a total of $(n+2)!$ permutations of players in G. Thus, the Shapley value of a_y in G is

$$\varphi_G(a_y) = \sum_{P_i \in \mathcal{P}} \frac{(s_i+1)!(n-s_i)! + s_i!(n-s_i+1)!}{(n+2)!} = \sum_{P_i \in \mathcal{P}} \frac{s_i!(n-s_i)!(s_i+1+n-s_i+1)}{(n+2)!} =$$

$$\sum_{P_i \in \mathcal{P}} \frac{s_i!(n-s_i)!(n+2)}{(n+2)!} = \sum_{P_i \in \mathcal{P}} \frac{s_i!(n-s_i)!}{(n+1)!}.$$

We now consider what happens when a_y splits into two agents, a'_y and a''_y, $w(a'_y) = w(a'_y) = 1$, resulting in a game $G' = [8t_1, \ldots, 8t_n, 1, 1, 1; \sum_{i=1}^n t_i + 3]$.

Again, let $\langle P_1, P_2 \rangle$, $|P_1| = s_i$, $|P_2| = n - s_i$, be a partition of T, so $w(P_1) = w(P_2)$, and let $\langle A_1, A_2 \rangle$ be the corresponding partition of A. Consider any permutation π which places a''_y directly after some permutation of $A_1 \cup \{a_x, a'_y\}$; clearly, a''_y is critical for π. Similarly,

a''_y is critical for any permutation π' which places a''_y directly after some permutation of $A_2 \cup \{a_x, a'_y\}$. There are $(s_i+2)!(n-s_i)!$ permutations putting a''_y directly after some permutation of $A_1 \cup \{a_x, a'_y\}$ and $s_i!(n-s_i+2)!$ permutations putting a''_y directly after some permutation of $A_2 \cup \{a_x, a'_y\}$.

Thus, for each partition $P_i =< P_1^i, P_2^i >$, where $|P_1^i| = s_i$, we have $(s_i+2)!(n-s_i)! + s_i!(n-s_i+2)!$ permutations where a''_y is critical. Switching the roles of a'_y and a''_y, both of which have the same weight and are thus equivalent, we also get that there are $(s_i+2)!(n-s_i)!+s_i!(n-s_i+2)!$ permutations where a'_y is critical. There are $n+3$ agents in G', so there is a total of $(n+3)!$ permutations of the agents. Thus each partition $P_i = (P_1^i, P_2^i)$, $|P_i| = s_i$, contributes $\frac{s_i!(n-s_i)!}{(n+1)!}$ to the Shapley value of a_y in G, and $2\frac{(s_i+2)!(n-s_i)!+s_i!(n-s_i+2)!}{(n+3)!}$ to the sum of the Shapley values of a'_y or a''_y in G'. We will now show that for any partition P_i

$$2\frac{(s_i+2)!(n-s_i)! + s_i!(n-s_i+2)!}{(n+3)!} > \frac{s_i!(n-s_i)!}{(n+1)!}. \tag{2}$$

Summing these inequalities over all partitions P_i will imply $\varphi_{G'}(a'_y) + \varphi_{G'}(a''_y) > \varphi_G(a_y)$, as desired. Inequality (2) can be simplified to

$$2\frac{(s+1)(s+2) + (n-s+1)(n-s+2)}{(n+2)(n+3)} > 1,$$

where we use s instead of s_i to simplify notation, or, equivalently, $2(s+1)(s+2) + 2(n-s+1)(n-s+2) - (n+2)(n+3) > 0$. Now, observe that $2(s+1)(s+2) + 2(n-s+1)(n-s+2) - (n+2)(n+3) = (n-2s)^2 + n + 2 > 0$ for any n. This proves inequality (2) for any s, n. It follows that if there is a partition, agent a_y always gains by splitting into two agents of weight 1. Thus, if T is a "yes"-instance of PARTITION, then (G, a_y) is a "yes"-instance of BENEFICIAL SPLIT. □

We summarize our results in the following theorem.

Theorem 11. BENEFICIAL SPLIT *is NP-hard, even if the only allowed split is into two identities with equal weights.*

Proof. We transform an instance T of PARTITION into an instance (G, a_y) of BENEFICIAL SPLIT as explained above. We combine Corollary 9 and Lemma 10, and see that T is a "yes" instance of PARTITION if and only if the (G, a_y) is a "yes"-instance of BENEFICIAL SPLIT. This completes the reduction. □

Remark 12. *Note that we have not shown that* BENEFICIAL SPLIT *is in NP, so we have not proved that it is NP-complete. There are two reasons for this. First, if we allow splits into an arbitrary number of identities, some of the candidate solutions may have exponentially many new agents (e.g., an agent with weight w_i can split into w_i agents of weight 1), or agents whose weights are rational numbers with superpolynomially many digits in their binary representation. Second, even if we circumvent this issue by only considering splits into two identities with integer weights, it is not clear how to verify in polynomial time whether a particular split is beneficial. In fact, since computing the Shapley value in weighted voting games is #P-complete, it is quite possible that* BENEFICIAL SPLIT *is not in NP.*

6 Finding Beneficial Splits

In Section 5, we have shown that it is hard even to test if any beneficial split exists, let alone to find the optimal split. This can be seen as a positive result, since complexity of finding beneficial splits serves as a barrier for this kind of manipulative behavior. However, it turns out that in many cases manipulators can overcome this difficulty. Indeed, recall that our hardness reduction is from PARTITION. While this problem is NP-hard, its hardness—and hence the hardness of our problem—crucially relies on the fact that the weights of the elements are represented in binary. Indeed, if the weights are given in unary, there is a dynamic programming-based algorithm for PARTITION that runs is time polynomial in size of the input (such algorithms are usually referred to as *pseudopolynomial*). In particular, if all weights are polynomial in n, the running time of this algorithm is polynomial in n. In many natural voting domains the weights of all agents are not too large, so this scenario is quite realistic. It is therefore natural to ask if there exists a pseudopolynomial algorithm for the problem of finding a beneficial split.

It turns out that the answer to this question is indeed positive as long as there is a constant upper bound K on the number of identities that the manipulator can use and all weights are required to be integer. To see this, recall that there is a pseudopolynomial algorithm for computing the Shapley value of any player in a weighted voting game [10]. This algorithm is based on dynamic programming: for any weight W and any $1 \leq k \leq n$, it calculates the number of coalitions of size k that have weight W. One can use the algorithm of [10] to find a beneficial split for a player a_i with weight w_i in a game G as follows. Consider all possible splits $w_i = w_i^{(1)} + \cdots + w_i^{(K)}$, where $w_i^{(j)} \in \mathbb{Z}$, $w_i^{(j)} \geq 0$ for $j = 1, \ldots, K$. Clearly, the number of such splits is at most $(w_i)^K$, which is polynomial in n for constant K. Evaluate the Shapley values of all new agents in any such split and return "yes" if and only if any of these splits results in an increased total payoff. Let $A(G)$ be the running time of the algorithm of paper [10] on instance G. The running time of our algorithm is $O((w_i)^K K \cdot A(G))$, which is clearly pseudopolynomial.

We will now consider a more general setting, where only the weight of the manipulator is polynomially bounded, while the weights of other players can be large. To simplify the presentation, we limit ourselves to the case of two-way splits; however, our approach applies to splits into any constant number of identities. We can use the same high-level approach as in the previous case, i.e., considering all possible splits (because of the weight restriction, there is only polynomially many of them), and computing the Shapley values of both new agents for each split. However, if we were to implement the latter step exactly, it would take exponential time. Therefore, in this version of our algorithm, we replace the algorithm of [10] with an approximation algorithm for computing the Shapley value. Several such algorithms are known: see, e.g., [8, 4]. We will use these algorithms in a black-box fashion. Namely, we assume that we are given a procedure $\texttt{Shapley}(G, a_i, \delta, \epsilon)$ that for any given values of $\epsilon > 0$ and $\delta > 0$ outputs a number v that with probability $1 - \delta$ satisfies $|v - \varphi_G(a_i)| \leq \epsilon$ and runs in time $\mathrm{poly}(n \log w_{\max}, 1/\epsilon, 1/\delta)$. We show how to use this procedure to design an algorithm for finding a beneficial split and relate the performance of our algorithm to that of $\texttt{Shapley}(G, a_i, \delta, \epsilon)$.

Our algorithm is described in Figure 1. It takes a pair of parameters (δ, ϵ) as an input, and uses the procedure $\texttt{Shapley}(G, a_i, \delta, \epsilon)$ as a subroutine. The algorithm outputs "yes" if it finds a split whose total estimated payoff exceeds the payoff of the manipulator in the original game by at least 3ϵ. It can easily be modified to output the optimal split.

Proposition 13. *With probability $1-3\delta$, the output of our algorithm satisfies the following: (i) If the algorithm outputs "yes", then (G, a_i) admits a beneficial integer split; (ii) Conversely, if there is an integer split that increases the payoff to the manipulator by more than 6ϵ, our algorithm outputs "yes". Moreover, the running time of our algorithm is polynomial in nw_i, $1/\epsilon$, and $1/\delta$.*

Proof. Suppose that the algorithm outputs "yes". We have $Prob[v^* < \varphi_G(a_i) - \epsilon] < \delta$, $Prob[v' > \varphi_{G'}(a_i') + \epsilon] < \delta$, $Prob[v'' > \varphi_{G'}(a_i'') + \epsilon] < \delta$. Hence, with probability at least

```
FindSplit(G = [w; q], a_i, δ, ε);
v* =Shapley(G, a_i, δ, ε);
for j = 0, ..., w_i
    w'_i = j, w''_i = w_i − j;
    G' = [w_1, ..., w_{i−1}, w'_i, w''_i, w_{i+1}, ..., w_n; q];
    v' = Shapley(G', a'_i, δ, ε), v'' =Shapley(G', a''_i, δ, ε);
    v = v' + v'';
    if v > v* + 3ε then return yes;
return no;
```

Figure 1: Algorithm $\texttt{FindSplit}(G = [\mathbf{w}; q], a_i, \delta, \epsilon)$

$1 - 3\delta$, if $v' + v'' > v^* + 3\epsilon$, then $\varphi_{G'}(a'_i) + \varphi_{G'}(a''_i) + 2\epsilon > \varphi_G(a_i) - \epsilon + 3\epsilon$, or, equivalently, $\varphi_{G'}(a'_i) + \varphi_{G'}(a''_i) > \varphi_G(a_i)$.

Conversely, suppose that there is a beneficial split of the form (w'_i, w''_i) that improves player a_i's payoff by at least 6ϵ. As before, with probability at least $1 - 3\delta$ we have that $v^* \leq \varphi_G(a_i) + \epsilon$ and at the step $j = w'_i$ it holds that $v' \geq \varphi_{G'}(a'_i) - \epsilon$, $v'' \geq \varphi_{G'}(a''_i) - \epsilon$. Then $v = v' + v'' \geq \varphi_{G'}(a'_i) + \varphi_{G'}(a''_i) - 2\epsilon > \varphi_G(a_i) + 6\epsilon - 2\epsilon \geq v^* + 3\epsilon$, so the algorithm will output "yes". □

While our algorithm does not *guarantee* finding a successful manipulation, it is possible to control the approximation quality (at the cost of increasing the running time), so that a successful manipulation is found with high probability.

Thus we can see that manipulators have several ways to overcome the computational difficulty of finding the optimal manipulation. Thus, other measures are required to avoid such manipulations.

7 Extensions

In this section, we consider some variants of the model studied in the paper. Our results here are rather preliminary and provide several interesting directions for future research.

Splitting into more than two identities So far, we have mostly discussed the gain (or loss) that an agent can achieve by splitting into two identities. However, it is also possible for an agent to use three or more false names. Potentially, the number of identities an agent can use can be as large as his weight (and if the weights are not required to be integer, it can even be infinite). It would be interesting to see which of our results hold in this more general setting. For example, while our computational hardness result holds for splits into any number of identities, the algorithmic results of the previous section only apply to splits into a constant number of new identities. An obvious open problem here is to design a pseudopolynomial algorithm for finding a beneficial integer split into any number of identities, or to prove that this problem is NP-hard even for small weights (i.e., weights that are polynomial in n). Another question of interest here is to extend the upper and lower bounds of Section 4.1 for this setting.

Manipulation by merging Each situation in which splitting is harmful for an agent directly corresponds to a situation where it is beneficial for several agents to merge, i.e., pretend that they are a single agent whose weight is equal to the total weight of the manipulators.

Some of the results presented in the paper can easily be translated to this domain. In particular, it is not hard to see that the proof of Theorem 11 can be adapted to show that it is NP-hard to check whether there exists a beneficial merge, and the results of Section 4.1 can be interpreted in terms of merging rather than splitting. However, this problem is very different from the game-theoretic perspective, as it involves coordinated actions by several would-be manipulators who then have to decide how to split the (increased) total payoff. We propose it as a direction for future work.

8 Conclusions

We have considered false-name manipulations in weighted voting games. We have shown that these manipulations can both increase and decrease the manipulator's payoffs, and provided tight upper and lower bounds on the effects of false-name voting. We have also shown that testing whether a beneficial manipulation exists is NP-hard. One may ask why we view this hardness result as an adequate barrier to manipulation, while using Shapley value (which itself is #P-hard to compute) as a payoff division scheme and therefore assuming that it can be computed. To resolve this apparent contradiction, note that the Shapley value corresponds to the voting power, and the players may try to increase their voting power by weight-splitting manipulation even if they cannot compute it. Also, when the Shapley value is used to compute payments, the center, which performs this computation, may have more computational power than individual agents. Furthermore, a payoff division scheme that is based on approximate computation of Shapley value may still be acceptable to the agents, whereas the manipulator may want to know for sure that attempted manipulation will not hurt him (and we have seen that in some cases weight-splitting can considerably decrease the agent's payoffs), or provide him with sufficient benefits to offset the costs of splitting. While the approximation algorithm discussed in the previous section can be used for this purpose, it only works if the manipulator's weight is small. Generalizing it to large weights (i.e., showing that if a beneficial split exists, it can be found by testing a polynomial number of splits) is an interesting open question.

In this paper, we presented results on false-name voting for the case when the payoffs are distributed according to the Shapley value. An obvious research direction is to see if one can derive similar results for other power indices, such as Banzhaf index, as well as other solution concepts used in co-operative games such as, e.g., the nucleolus. More generally, it would be interesting to design a payoff distribution scheme that is resistant to this type of manipulation, or prove that it does not exist.

The study of weighted voting has many applications both in political science and in multiagent systems. There are several possible interpretations for identity-splitting in these contexts, such as obtaining a higher share of the grand coalition's gains when these are distributed according to the Shapley value, or obtaining more political power by splitting a political party into several parties with similar political platforms. In the first case, a false-name manipulation is hard to detect in open anonymous environments, and can thus be very effective. In the second case, the manipulation is done using legitimate tools of political conduct. Therefore, we conjecture that false-name manipulation is widespread in the real world and may become a serious issue in multiagent systems. It is therefore important to develop a better understanding of the effects of this behavior and/or design methods of preventing it.

References

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

[2] V. Conitzer and T. Sandholm. Computing Shapley values, manipulating value division schemes, and checking core membership in multi-issue domains. In *Proc. AAAI'04*, pp. 219–225, 2004.

[3] X. Deng and C. H. Papadimitriou. On the complexity of cooperative solution concepts. *Math. Oper. Res.*, 19(2):257–266, 1994.

[4] S. S. Fatima, M. Wooldridge, and N. R. Jennings. A randomized method for the Shapley value for the voting game. In *Proc. AAMAS'07*, pp. 955–962, 2007.

[5] A. Laruelle. On the choice of a power index. Papers 99-10, Valencia — Instituto de Investigaciones Economicas, 1999.

[6] D. Leech. Voting power in the governance of the international monetary fund. *Annals of Operations Research*, 109(1-4):375–397, 2002.

[7] M. Machover and D. S. Felsenthal. The treaty of Nice and qualified majority voting. *Social Choice and Welfare*, 18(3):431–464, 2001.

[8] I. Mann and L. S. Shapley. Values of large games, IV: Evaluating the electoral college by Monte-Carlo techniques. Technical report, The Rand Corporation, Santa Monica, CA, 1960.

[9] I. Mann and L. S. Shapley. Values of large games, VI: Evaluating the electoral college exactly. Technical report, The Rand Corporation, Santa Monica, CA, 1962.

[10] Y. Matsui and T. Matsui. A survey of algorithms for calculating power indices of weighted majority games. *Journal of the Operations Research Society of Japan*, 43, 2000.

[11] G. Owen. Multilinear extensions and the Banzhaf Value. *Naval Research Logistics Quarterly*, 22(4):741–750, 1975.

[12] L. S. Shapley. A value for n-person games. *Contributions to the Theory of Games*, pp. 31–40, 1953.

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

[14] P. Straffin. Homogeneity, independence and power indices. *Public Choice*, 30:107–118, 1977.

[15] M. Yokoo, V. Conitzer, T. Sandholm, N. Ohta, A. Iwasaki. Coalitional games in open anonymous environments. In *Proc. AAAI'05*, pp. 509–515, 2005.

[16] M. Yokoo, Y. Sakurai, S. Matsubara. Robust combinatorial auction protocol against false-name bids. *Artificial Intelligence*, 130(2):167.181, 2001.

[17] M. Yokoo, Y. Sakurai, S. Matsubara. The effect of false-name bids in combinatorial auctions: New fraud in Internet auctions. *Games and Economic Behavior*, 46(1):174.188, 2004.

Yoram Bachrach
School of Engineering and Computer Science
Hebrew University
Jerusalem, Israel
Email: `yori@cs.huji.ac.il`

Edith Elkind
School of Electronics and Computer Science
University of Southampton
Southampton, United Kingdom
Email: `ee@ecs.soton.ac.uk`