Reliability Weighted Voting Games

Yoram Bachrach[1] and Nisarg Shah[2]

[1] Microsoft Research Cambridge, UK
yobach@microsoft.com
[2] Carnegie Mellon University, USA
nkshah@cs.cmu.edu

Abstract. We examine agent failures in weighted voting games. In our cooperative game model, R-WVG, each agent has a weight and a survival probability, and the value of an agent coalition is the probability that its surviving members would have a total weight exceeding a threshold. We propose algorithms for computing the value of a coalition, finding stable payoff allocations, and estimating the power of agents. We provide simulation results showing that on average the stability level of a game increases as the failure probabilities of the agents increase. This conforms to several recent results showing that failures increase stability in cooperative games.

Keywords: Cooperative game theory, Weighted voting game, Reliability extension, Agent failures, Stability

1 Introduction

Consider several firms collaborating to complete a joint project. The project requires a threshold amount of a certain resource to be completed successfully, and each firm owns a different amount of the resource that it can contribute to the project. If enough firms commit their resources so that the total contributed amount is at least the threshold amount, the project is completed and generates a certain revenue. One important question is how this revenue should be distributed among the participating firms. Traditionally, such domains were modelled as Weighted Voting Games (WVGs), and various game theoretic solution concepts were used for revenue distribution (see [16]).

However, in the real world, a firm may promise to deliver resources but fail to do so afterwards, or its delivered resources may fail during the execution of the project, due to reasons beyond the firm's control. In this case, the project can only be completed if the total amount of resources that did *not* fail exceeds the threshold. One might suggest to model this as a WVG among the firms which successfully delivered working resources, and distribute the revenue only among these firms. However, this might deter some firms from participating in the first place, since due to such failures they may not get paid even after exerting effort to deliver resources. One way to circumvent this is using an ex-ante contract to divide the revenue (generated only if the project finishes) that is independent of which firms eventually failed and which did not.

Another similar domain is the case of *lobbying agents* in parliamentary settings [26, 9], where the agents exert lobbying efforts to convince parties with different weights (e.g., the number of seats) to vote for a new legislation, but may fail to do so with a certain probability. Again, the agents might prefer an ex-ante contract for payoff division to avoid not being paid ex-post for their persuasion efforts in case they fail.

Clearly, such domains require explicit modeling of agent failures. Although failures were widely studied in *non cooperative game theory* [7, 23, 22], such analysis has surprisingly ignored the prominent WVGs model from cooperative game theory.

We study the effect of agent failures on the solutions of WVGs using the recently proposed reliability extension model [3]. The heart of a cooperative game is the *characteristic function* which maps every agent subset to the utility the agents achieve together. Under specified agent survival probabilities, and assuming such failures are independent, the reliability extension modifies the characteristic function and maps every agent subset to its *expected* value. We examine the reliability extension of WVGs, which we denote "R-WVGs" (Reliability Weighted Voting Games). We analyze how the reliability extension changes the outcome in WVGs, as captured by solutions such as the Shapley value [24] and the core [19], providing both theoretical and empirical results.

Our Contribution: The contribution of this paper is threefold. First, we contrast the computational hardness of various solution concepts in WVGs with that in R-WVGs. While the problems of computing the value of a coalition, testing emptiness of the core, and checking if a given imputation is in the core are in $\mathcal{P}$ for WVGs, we prove they are $\#\mathcal{P}$-hard and co$\mathcal{NP}$-hard for R-WVGs. For computing the value of a coalition, we provide an exact dynamic programming algorithm, as well as a polynomial time additive approximation method. We show that the latter two problems remain hard even if only one agent may fail. We develop an algorithm to compute a core imputation for R-WVGs with constantly many unreliable agents and small weights. Second, the hardness of computing power indices (the Shapley value and the Banzhaf index) in R-WVGs follows from the hardness in WVGs. We develop dynamic programming algorithms for computing these indices in restricted R-WVGs. Third, we provide simulation results for R-WVGs which indicate that, *on average*, lower reliability levels of agents lead to higher stability of the game, as measured by the probability of having a non-empty core, the least-core value, and the Cost of Stability.

1.1 Related Work

Computational aspects of cooperative game theory have recently received much attention. The problems of testing emptiness of the core and finding a core imputation have been investigated for many cooperative games, ranging from network games [4, 1] through combinatorial games [14, 10] to general representation languages [13]. The core is easy to analyze in simple games (including WVGs), as it is closely related to the existence of veto players (see [12]). However, R-WVGs are not simple games, and as our analysis shows, questions regarding the core are computationally harder in R-WVGs. While emptiness/non-emptiness of the core is a qualitative measure of stability, the least-core value [25] and the Cost of Stability (CoS) [5] serve as its quantitative generalizations; we use all three of them as stability measures. As weighted voting games (WVGs) also model decision making bodies [16, 12], computing power indices (the Shapley value [24] and the Banzhaf index [8]) is a central question. Due to the hardness of computing them in WVGs [21], approximations have been proposed [17, 2].

Failures were investigated in non-cooperative game theory, in congestion games [23, 22], reliable network formation [7], Nash networks [11] and sensor networks [20], but received less attention in *cooperative* games. We used the recent reliability extension model of [3], and applied it to WVGs to obtain the class of R-WVGs.

Another recent paper [6] examines the core in the reliability extension of totally-balanced games, showing that in such games agent failures only help stability in terms of non-emptiness of the core. While the general theme is in accordance with our results, their analysis is irrelevant for R-WVGs as R-WVGs are not totally-balanced.

2 Preliminaries

A transferable utility cooperative game $G = (N, v)$ is composed of a set of agents $N = \{1, 2, \ldots, n\}$ and a characteristic function $v : 2^N \to \mathbb{R}$ mapping any coalition (agent subset) to the utility these agents achieve together. By convention, $v(\emptyset) = 0$. For an agent $i \in N$ and a coalition $S \subseteq N$, we denote $S \cup \{i\}$ by $S + i$ and $S \setminus \{i\}$ by $S - i$. A game is called *simple* if the characteristic function only takes values of 0 or 1, so $v : 2^N \to \{0, 1\}$. In simple games, a coalition $C \subseteq N$ is called winning if $v(C) = 1$, and losing otherwise.

Weighted Voting Games (WVGs): A WVG is a game where each agent $i \in N$ has a weight $w_i > 0$, and a coalition $C \subseteq N$ is winning iff its total weight exceeds a given threshold t: if $\sum_{i \in C} w_i \geq t$ then $v(C) = 1$, else $v(C) = 0$.

The Core: The characteristic function defines the value that a coalition achieves, but not how it should be *distributed* among its members. A payment vector $\mathbf{p} = (p_1, \ldots, p_n)$ is an *imputation* if $\sum_{i=1}^{n} p_i = v(N)$ (efficiency) and $p_i \geq v(\{i\})$ for every $i \in N$ (individual rationality). Here, p_i is the payoff to agent i, and the payoff to a coalition C is $p(C) = \sum_{i \in C} p_i$. The *core requirement* is that the payoff to every coalition is at least as much as it can gain on its own, so no coalition can gain by defecting from the grand coalition of all the agents. The *core* [19] is defined as the set of all imputations $\mathbf{p}$ such that $p(N) = v(N)$ and $p(S) \geq v(S)$ for all $S \subseteq N$.

The ϵ-core: The definition of the core is quite demanding; many games of interest have empty core. One popular relaxation to circumvent this is the ϵ-core [25]. For any ϵ, the ϵ-core is the set of all payoff vectors $\mathbf{p}$ such that $p(N) = v(N)$ and $p(S) \geq v(S) - \epsilon$ for all $S \subseteq N$. One way to interpret this is that if a coalition incurs a cost of ϵ for deviating from the grand coalition, then the imputation is stable. Higher deviation cost makes it easier to find a stable imputation. For any game, the set $\{\epsilon \mid$ the ϵ-core is non-empty$\}$ has a minimum element $\epsilon_{\min}$, known as the *least core value* (LCV). The LCV is the minimal deviation cost admitting a stable enough payoff allocation. Higher LCV implies that the game is unstable.

The Cost of Stability: In games with an empty core, it is impossible to distribute the gains of the grand coalition N in a stable way. An external party can induce agent cooperation by offering a *supplemental payment* if the grand coalition is formed. Bachrach et. al. [5] formalized this as follows. Given a game $G = (N, v)$ and a supplemental payment $\Delta \in \mathbb{R}$, the *adjusted game* $G(\Delta) = (N, v')$, where the characteristic function is defined by: $v'(N) = v(N) + \Delta$ and $v'(S) = v(S)$ for all $S \neq N$. The *Cost of Stability* (CoS) of a game G, denoted $CoS(G)$, is the minimum supplement Δ^* for which the core of the adjusted game $G(\Delta^*)$ is non-empty. The CoS quantifies the extent of instability by measuring the subsidy required to overcome agents' resistance to cooperation. A higher CoS therefore indicates that the game is more unstable.

The Shapley Value: Power indices analyze the contributions of the agents to different coalitions, proposing ways to divide the gains based on fairness criteria. The *marginal contribution* of an agent $i \in N$ to a coalition $S \subseteq N - i$ is $v(S+i) - v(S)$. The *Shapley value* is uniquely characterized by four important fairness axioms [15]. For any permutation π of agents (i.e., $\pi : \{1, 2, \dots, n\} \to \{1, 2, \dots, n\}$ and π is onto), let $\Gamma_i^\pi = \{j | \pi(j) < \pi(i)\}$ be the set of agents before i in π. The Shapley value is the payoff vector $(\varphi_1, \dots, \varphi_n)$, where φ_i is the Shapley value of agent i given by: $\varphi(i) = \frac{1}{n!} \sum_{\pi \in S_n} (v(\Gamma_i^\pi + i) - v(\Gamma_i^\pi))$. For any coalition $S \subseteq N - i$, the number of permutations $\pi \in S_n$ where $\Gamma_i^\pi = S$ is exactly $(|S|)! \cdot (n - |S| - 1)!$. Thus, summing over coalitions, we get: $\varphi_i = \frac{1}{n!} \sum_{S \subseteq N-i} \Big[(|S|)! \cdot (n - |S| - 1)! \cdot (v(S+i) - v(S))\Big]$. The Banzhaf index is another prominent power index.

Reliability Games: A model for agent failures in cooperative games was proposed in [3]. A *reliability game* $G = (N, v, \mathbf{r})$ consists of the set of agents $N = \{1, 2, \dots, n\}$, the *base characteristic function* $v : 2^N \to \mathbb{R}$ describing values in the absence of failures, and the reliability vector $\mathbf{r}$, where r_i is the probability of agent i surviving (i.e., not failing). The characteristic function $v^{\mathbf{r}}$ of the reliability game with failures now considers the expected value of the survivors:

$$v^{\mathbf{r}}(S) = \sum_{S' \subseteq S} \Pr(S'|S) \cdot v(S') = \sum_{S' \subseteq S} \left(\prod_{i \in S'} r_i \cdot \prod_{j \in S \setminus S'} (1 - r_j) \right) \cdot v(S'). \qquad (1)$$

Here, $\Pr[S'|S]$ is the probability that every agent in S' survives and every agent in $S \setminus S'$ fails. For the *base game* $G = (N, v)$, the game $G^{\mathbf{r}} = (N, v, \mathbf{r})$ is called the *reliability extension* of G with the reliability vector $\mathbf{r}$. An agent is called fully reliable (or reliable) if its reliability is 1, and unreliable otherwise.

3 Reliability Weighted Voting Games

In this paper, we examine the reliability extension of weighted voting games (R-WVGs). Formally, an R-WVG $G^{\mathbf{r}} = (N, \mathbf{w}, t, \mathbf{r})$, where $N = \{1, \dots, n\}$ is the set of agents, $\mathbf{w} = (w_1, \dots, w_n)$ is the vector of agent weights, t is the threshold (weight quota), and $\mathbf{r}$ is the vector of agent survival probabilities. The characteristic function $v^{\mathbf{r}}$ is given by Equation (1), where the base characteristic function follows $v(C) = 1$ if $\sum_{i \in C} w_i \geq t$ and $v(C) = 0$ otherwise.

We now discuss the complexity of computing various solutions in R-WVGs. We first emphasize that R-WVGs are a generalization of WVGs: WVGs are recovered when all agents are fully reliable, i.e., $\mathbf{r} = (1, \dots, 1)$. Thus, solving R-WVGs is more demanding than solving WVGs — any problem that is computationally hard for WVGs remains hard in R-WVGs. For example, computing the Shapley value or the Banzhaf index in WVGs is known to be $\#\mathcal{P}$-hard [21, 16], and thus remains $\#\mathcal{P}$-hard even in R-WVGs. However, certain prominent problems are easy for WVGs. For example, computing the value of a coalition in a WVG is simple, as it only requires summing the weights of the members and testing whether the sum exceeds the threshold. Testing if the core is empty and checking if a given imputation is in the core are other examples of problems that are in $\mathcal{P}$ for WVGs. We show that all these problems become hard in R-WVGs.

Theorem 1. *Finding value of the grand coalition is $\#\mathcal{P}$-hard in R-WVGs.*

Proof. We use a reduction from $\#SUBSET\text{-}SUM$, the counting version of the subset sum problem. $\#SUBSET\text{-}SUM$, which is known to be $\#\mathcal{P}$-hard, requires counting the number of subsets of a given set S of positive integers that sum to another positive integer t. Take an instance (S,t) of $\#SUBSET\text{-}SUM$. Let $|S| = n$. Create an R-WVG G_1 with n agents having elements of S as the weights, threshold t, and reliability vector $\mathbf{r} = (1/2, 1/2, \ldots, 1/2)$. Create another R-WVG G_2, which is identical to G_1 except the threshold in G_2 is $t+1$. Let $v(G_1)$ and $v(G_2)$ denote the values of grand coalitions of G_1 and G_2 respectively.

Note that with the reliability vector $\mathbf{r} = (1/2, \ldots, 1/2)$, the value of the grand coalition is the fraction of coalitions having total weight at least as much as the threshold. Formally, let $\#^t S$ denote the number of subsets of S with total weight at least t. Then, $v(G_1) = (\#^t S)/2^n$ and $v(G_2) = (\#^{t+1} S)/2^n$. So if we can compute the value of the grand coalition in R-WVGs, we can compute $v(G_1)$ and $v(G_2)$, and obtain $\#^t S - \#^{t+1} S = 2^n \cdot (v(G_1) - v(G_2))$, which is the number of subsets of S that sum to exactly t, i.e., the answer to the $\#SUBSET\text{-}SUM$ instance (S,t). □

Though it is hard to compute the value of a coalition, we can approximate it additively. Consider an R-WVG $G^{\mathbf{r}} = (N, \mathbf{w}, t, \mathbf{r})$ and any coalition $S \subseteq N$. To approximate $v^{\mathbf{r}}(S)$, run k iterations such that in each iteration, every agent $i \in S$ survives with probability r_i. Let C_t be the surviving sub-coalition in iteration t. Then, $v^{\mathbf{r}}(S) \approx \hat{V} = \frac{1}{k} \cdot \sum_{t=1}^{k} v(C_t)$. Using Hoeffding's inequality and Equation (1), it can be shown that $k = \frac{1}{2\cdot\epsilon^2} \cdot \log(2/\delta)$ is sufficient to achieve $\Pr[|v^{\mathbf{r}}(S) - \hat{V}| > \epsilon] \leq \delta$.[3] Further, if the agent weights and the threshold are integers, we can use a dynamic programming approach to calculate the value exactly. For simplicity, consider the grand coalition.[4] Let $T(j,q)$ denote the value of the coalition $\{1, \ldots, j\}$ in the R-WVG where the threshold is changed to q. Then, $T(j,q) = r_j \cdot T(j-1, q-w_j) + (1-r_j) \cdot T(j-1, q)$, where $T(0,q) = 0$ if $q > 0$ and for all j, $T(j,q) = 1$ if $q \leq 0$. Now $v(N) = T(n,t)$, which can be computed in time $O(t \cdot n)$.

4 The Core of R-WVGs

Next, we examine the complexity of finding core-related solutions. Checking if a given imputation is in the core, testing emptiness of the core, and finding a core imputation are computationally easy (in $\mathcal{P}$) for WVGs. We show that all of them are computationally hard for R-WVGs.

[3] This method works for the reliability extension of any cooperative game in general.

[4] The value of any coalition can be obtained in the same way by examining the restricted game where the other agents are removed.

4.1 Checking if a given imputation is in the core

Theorem 2. *Checking if a given imputation is in the core is co$\mathcal{NP}$-hard for R-WVGs, even with a single unreliable agent.*

Proof. We reduce $SUBSET\text{-}SUM$ to the complement of our problem, i.e., checking if an imputation is *not* in the core. Given an instance (S, t) of $SUBSET\text{-}SUM$ where $S = \{w_1, \ldots, w_n\}$, the question is to check if there is a subset of S whose elements sum to t. Define $W = \sum_{i=1}^{n} w_i$. If $W \leq t$, the reduction is trivial: reduce the case of $W < t$ to any NO instance, and the case of $W = t$ to any YES instance.

If $W > t$, construct an R-WVG G with $n + 1$ agents, where first n agents have reliability 1 and weights $w_1, \ldots, w_n$, agent $n+1$ has reliability $(t+1)/W$ and weight $W - t$, and threshold is W. Consider the payments $\mathbf{p} = \{p_1, \ldots, p_{n+1}\}$ where $p_i = w_i/W$ for $1 \leq i \leq n$ and $p_{n+1} = 0$. We show that $\mathbf{p}$ is not in the core of G iff the answer to the $SUBSET\text{-}SUM$ instance is YES. First, $\mathbf{p}$ is an imputation since the value of every single agent is 0 and the sum of payoffs is 1 (the value of the grand coalition).

Next, $\mathbf{p}$ is not in the core iff there is a coalition with total payoff less than its value. It can be checked that any coalition not containing agent $n + 1$ or containing first n agents has value either 0 or 1, and receives total payoff no less than its value. Thus, a violating coalition must contain agent $n + 1$ and not all of the first n agents. Such a coalition has value $r_{n+1} = (t + 1)/W$ if the total weight of agents other than agent $n+1$ in the coalition is at least t, and 0 otherwise. If this total weight is at least $t+1$, the coalition receives at least $(t + 1)/W$, which is its value. Thus, a violating coalition exists iff there is a subset of the first n agents whose weights sum to exactly t. □

4.2 Testing emptiness of the core

Theorem 3. *Testing emptiness of the core in R-WVGs with a single unreliable agent (*SUCORE*) and testing emptiness of the ϵ-core in WVGs (*EPSCORE*) are polynomial-time reducible to each other.*

Proof. First, take an instance (G, ϵ) of EPSCORE where WVG $G = (N, \mathbf{w}, t)$ has n agents, weight vector $\mathbf{w}$ and threshold t, and $\epsilon \geq 0$. The task is to check if the ϵ-core of G is empty. Define $W = \sum_{i=1}^{n} w_i$. If $W \leq t$, the reduction is trivial: If $W < t$, the grand coalition has value 0 and a payoff of 0 to every agent is in the ϵ-core. If $W = t$, the grand coalition has value 1 but every other coalition has value 0, so a payoff of 1 to any single agent and 0 to the rest is in the ϵ-core. In both cases, the ϵ-core of G is not empty, so we reduce to any NO instance of SUCORE. If $W \geq t$, form an R-WVG $G' = (N', \mathbf{w}', t', \mathbf{r}')$ (instance of SUCORE) with $n + 1$ agents, weight vector $\mathbf{w}' = \{w_1, \ldots, w_n, W - t\}$, threshold $t' = W$, and reliability vector $\mathbf{r}' = \{1, \ldots, 1, 1 - \epsilon\}$. We show that the ϵ-core of G is empty iff the core of G' is empty.

Let v' be the characteristic function of G'. Since $v'(\{1, \ldots, n\}) = 1 = v'(N')$, agent $n+1$ must receive zero payoff in any core imputation of G'. Thus, the core of G' is non-empty iff there exists a payoff vector $\mathbf{p} = \{p_1, \ldots, p_n, 0\}$ such that $\sum_{i=1}^{n} p_i = 1$ and the payoff to every coalition is at least its value. Any coalition not containing agent $n + 1$ or containing all of first n agents receives at least its value by construction. Any coalition containing agent $n + 1$ but not all of first n agents has value $r_{n+1} = 1 - \epsilon$

if the total weight of the reliable agents (among first n agents) in it is at least t, and 0 otherwise. Thus, G' has non-empty core iff there is a solution to: $\sum_{i=1}^{n} p_i = 1$ and $p(S) \geq 1 - \epsilon$ whenever $w(S) \geq t$. But this is the LP for checking emptiness of ϵ-core for G, so the core of G' is empty iff the ϵ-core of G is non-empty.

We show a reduction in the other direction. Take any instance $H = (N, \mathbf{w}, t, \mathbf{r})$ of SUCORE with n agents, weight vector $\mathbf{w}$, threshold t, reliability vector $\mathbf{r}$, and characteristic function v. Without loss of generality, let agent n be the unreliable agent with reliability $r_n = x$. Now, $v(\{1, 2, \ldots, n-1\}) \in \{0, 1\}$. If $v(\{1, 2, \ldots, n-1\}) = 0$, paying $v(N)$ to agent n and 0 to other agents is a core imputation. Hence, the core is not empty, and we reduce this to any NO instance of EPSCORE. If $v(\{1, 2, \ldots, n-1\}) = 1$, then agent n has zero payoff in any core imputation of H. Hence, the core of H is non-empty iff there exists a payoff vector $\mathbf{p} = \{p_1, \ldots, p_{n-1}, 0\}$ such that the payoff to any coalition is at least its value. Any coalition containing all of first $n-1$ agents or not containing agent n receives at least as much as its value. Any coalition containing agent n but not all of first $n-1$ agents has value x if the total weight of the reliable agents (among first $n-1$ agents) in the coalition is at least $t - w_n$, and 0 otherwise. That is, the core of H is non-empty iff there is a solution to: $\sum_{i=1}^{n-1} p_i = 1$ and $p(S) \geq x$ whenever $w(S) \geq t - w_n$. However, this is exactly the LP for checking emptiness of ϵ-core for the WVG $H' = (N', \mathbf{w}', t - w_n)$ with the set of agents $N' = \{1, \ldots, n-1\}$, weight vector $\mathbf{w}' = \{w_1, \ldots, w_{n-1}\}$ and threshold $t - w_n$, with $\epsilon = 1 - x$. □

Elkind et. al. [16] proved that testing emptiness of ϵ-core of WVGs is co$\mathcal{NP}$-hard. Further, they gave an algorithm to compute an ϵ-core imputation of a WVG using a separation oracle that runs in time pseudo-polynomial in agent weights. Theorem 3 and its constructive proof allow us to translate these results to the domain of R-WVGs.

Corollary 1. *Testing emptiness of the core in R-WVGs is co$\mathcal{NP}$-hard, even with a single unreliable agent.*

Corollary 2. *If all weights are represented in unary, finding a core imputation of an R-WVG with a single unreliable agent is in $\mathcal{P}$.*

4.3 Finding a core imputation

Finding a core imputation, if one exists, is computationally easy (in $\mathcal{P}$) for WVGs (see [12]). Theorem 1 shows that even computing the value of the grand coalition is $\#\mathcal{P}$-hard for R-WVGs. Since total payoff in any core imputation equals the value of the grand coalition, finding a core imputation in R-WVGs is clearly $\#\mathcal{P}$-hard as well.

Corollary 3. *Finding a core imputation is $\#\mathcal{P}$-hard in R-WVGs.*

Corollary 2 gives us a pseudo-polynomial time algorithm to find a core imputation of an R-WVG with a single unreliable agent. We now extend this result to the case of a few (more than one) unreliable agents. The algorithm of [16] to find an ϵ-core imputation of a WVG works using a separation oracle (that runs in time pseudo-polynomial in weights) to solve the exponential sized LP of ϵ-core. It uses an important subroutine that finds, given any x, the minimum total payoff to any coalition with total weight at least x. We denote it MINPAY. So, $\text{MINPAY}(\mathbf{p}, x) = \min_{S \subseteq N, w(S) \geq x} p(S)$.

Consider an R-WVG $G^{\mathbf{r}} = (N, \mathbf{w}, t, \mathbf{r})$, where $|N| = n$. Without loss of generality, let UR be the set of unreliable agents and R be the set of reliable agents. Let $\mathbf{p} = \{p_1, \ldots, p_n\}$ denote a payoff vector. For any coalition S, $p(S) = \sum_{i \in S} p_i$ and $w(S) = \sum_{i \in S} w_i$. We aim to find a separation oracle for the LP of the core: $\sum_{i=1}^{n} p_i = v^{\mathbf{r}}(N)$, and $p(S) \geq v^{\mathbf{r}}(S)$ for all $S \subseteq N$.

Divide S into reliable and unreliable parts: $S = S_1 \cup S_2$, where $S_1 \subseteq UR$ and $S_2 \subseteq R$. Note that $p(S) = p(S_1 \cup S_2) = p(S_1) + p(S_2)$. We examine cases for $v^{\mathbf{r}}(S) = v^{\mathbf{r}}(S_1 \cup S_2)$. Consider the power set $\mathcal{P}(S_1) = \{T_1, \ldots, T_{2^{|S_1|}}\}$. Let $y_i = w(T_i)$, $q_i = \Pr[T_i | S_1]$ (the probability that exactly the agents in T_i survive out of agents in S_1), and $Q_i = \sum_{j \geq i} q_j$. Without loss of generality, let $y_i \leq y_{i+1}$ for all i. Note that $y_1 = 0$ as $T_1 = \emptyset$, and $Q_1 = 1$ as there must be a unique surviving sub-coalition. Now,

$$v^{\mathbf{r}}(S) = v^{\mathbf{r}}(S_1 \cup S_2) = \begin{cases} Q_1 = 1 & \text{if } w(S_2) \geq t - y_1 = t, \\ Q_i & \text{if } w(S_2) \in [t - y_i, t - y_{i-1}) \text{ where } i \in [2, 2^{|S_1|}], \\ 0 & \text{if } w(S_2) < t - y_{2^{|S_1|}}. \end{cases}$$

Using these observations, we can simplify the LP to:

$$\sum_{i=1}^{n} p_i = v^{\mathbf{r}}(N) \tag{2}$$
$$p(S_2) \geq Q_i - p(S_1), \forall (S_1 \subseteq UR, i \in [1, 2^{|S_1|}], S_2 \subseteq R) \text{ s.t. } w(S_2) \geq t - y_i.$$

Note that we have introduced additional constraints, but it is easy to check that they are redundant, and thus do not change the LP.[5] However, they enable us to use the subroutine MINPAY, for which we have a dynamic programming formulation.

Algorithm CORE-FEW-UNREL: Solve LP (2) using the following separation oracle.

ALGORITHM: SEPARATIONORACLE

Data: R-WVG $G^{\mathbf{r}} = (N, \mathbf{w}, t, \mathbf{r})$, payoff vector $\mathbf{p}$

Result: Either returns a violated constraint of LP (2) or returns `SATISFIED`

1. Compute $v^{\mathbf{r}}(N)$ from Equation (1): $v^{\mathbf{r}}(N) = \sum_{C \subseteq UR} \Pr[C|UR] \cdot v(C \cup R)$ (since agents in R always survive).
2. Check if $\sum_{i=1}^{n} p_i = v^{\mathbf{r}}(N)$. If not, then return this violated constraint.
3. For every $S_1 \subseteq UR$, compute y_i, q_i, and thus Q_i, for $i \in [1, 2^{|S_1|}]$. For all i, check if MINPAY$(t - y_i) \geq Q_i - p(S_1)$. If not, return the violated constraint.
4. If no violated constraints are found above, then return `SATISFIED`.

Running Time: The time required to compute $v^{\mathbf{r}}(N)$ is $O(2^{|UR|} \cdot n)$. For any $S_1 \subseteq UR$, the time required to compute y_i, q_i and Q_i is $O(2^{|S_1|} \cdot |S_1|)$. We make $O(2^{|S_1|})$ calls to MINPAY, each of which takes $O(n \cdot W)$ time, where $W = \sum_{i=1}^{n} w_i$. Thus, the total time required to check the constraints for any S_1 is $O(2^{|S_1|} \cdot n \cdot W)$. Summing over all $S_1 \subseteq UR$, the total running time is $O\left(\sum_{S_1 \subseteq UR} 2^{|S_1|} \cdot n \cdot W\right) =$

[5] The constraint $p(S_2) \geq Q_i - p(S_1)$ is required only when $w(S_2) \in [t - y_i, t - y_{i-1}]$, but we added it for all S_2 where $w(S_2) \geq t - y_i$. If $w(S_2) \in [t - y_j, t - y_{j-1}]$ for $j < i$ (or $w(S_2) \geq t - y_1$), then the constraint $p(S_2) \geq Q_j - p(S_1)$ (resp. $p(S_2) \geq Q_1 - p(S_1)$) strictly dominates the additional constraints added.

$O\left(\sum_{k=1}^{|UR|}\binom{|UR|}{k}\cdot 2^k\cdot n\cdot W\right) = O\left(3^{|UR|}\cdot n\cdot W\right)$. The last equation follows using binomial expansion. Thus, we have:

Theorem 4. *If all weights are represented in unary, finding a core imputation in an R-WVG with constantly many unreliable agents is in $\mathcal{P}$.*

Despite significant effort, we could not settle the question of existence of a pseudo-polynomial time algorithm for R-WVGs with arbitrarily many unreliable agents.

5 Power Indices

We now examine fair payoff divisions (power indices) in R-WVGs. Two prominent indices, the Shapley value and the Banzhaf index, are known to be $\#\mathcal{P}$-hard even in WVGs [21, 16], and thus also in R-WVGs. Bachrach et. al. [3] gave an algorithm to additively approximate the Shapley value, which can easily be adapted for the Banzhaf index. The algorithm works for reliability extensions of any cooperative game,[6] thus also for R-WVGs. Additionally, dynamic programming algorithms are known for computing both indices in WVGs [18, 21] when the weights and the threshold are integers. We give non-trivial extensions of these algorithms for computing both indices in R-WVGs with identical agent reliabilities (uniform reliability case), and integer weights and threshold. We only give a sketch for the Shapley value due to lack of space. The details appear in the full version of the paper.[7]

Consider an R-WVG $G^{\mathbf{r}} = (N, \mathbf{w}, t, \mathbf{r})$ where $r_j = p$ for all j (uniform reliability). Bachrach et. al. [3] showed that the Shapley value of agent i in the reliability extension of any cooperative game satisfies

$$\varphi_i = \frac{r_i}{|N|!}\cdot\sum_{\pi\in S_n}\left[\sum_{S\subseteq\Gamma_i^\pi} m_i(S)\cdot\Pr[S|\Gamma_i^\pi]\right],$$

where $m_i(S) = v(S+i) - v(S)$ is the marginal contribution of agent i to coalition S in the base game. Changing the order of summations, we get:

$$\varphi_i = \frac{r_i}{|N|!}\cdot\sum_{S\subseteq N-i} m_i(S)\left[\sum_{\Gamma\subseteq N-i\ s.t.\ S\subseteq\Gamma}\left(\Pr[S|\Gamma]\cdot\sum_{\pi\in S_n,\Gamma_i^\pi=\Gamma} 1\right)\right].$$

Now, $\Pr[S|\Gamma] = p^{|S|}\cdot(1-p)^{|\Gamma|-|S|}$, and $\sum_{\pi\in S_n,\Gamma_i^\pi=\Gamma} 1 = |\Gamma|!\cdot(n-|\Gamma|-1)!$. Thus, all the required quantities except $m_i(S)$ depend only on $|S|$ and $|\Gamma|$. We break the summations of S and Γ further over the sizes of the coalitions, and show that the expression can be computed in pseudo-polynomial time. The overall running time of our methods[8] is $O(t\cdot n^2)$, where t is the threshold and n is the number of agents. We

[6] Bachrach et. al. [3] consider network games, but their method works for any cooperative game.
[7] See `http://www.cs.cmu.edu/~nkshah/papers.html` for the full version.
[8] The running time is for both the Shapley value and the Banzhaf index.

remark that this is identical to the running time of the known dynamic programming algorithms for these indices in WVGs. This is surprising, given our results of Sections 3 and 4 that moving from WVGs to R-WVGs raises the computational complexity of many questions significantly.

6 The Relation Between Reliability And Stability

We examine the relation between agent reliability and stability of the game in our R-WVG model. We randomly construct many R-WVGs using a *generation model* depending on a *reliability parameter*, quantify the degree of stability in each generated game according to some *stability measures*, and examine the *expected* degree of stability for each reliability parameter. We use three metrics as measures of stability. On the qualitative level, a game is *completely stable* if its core is non-empty, as there exists a fully stable payoff division. On the quantitative level, we use the least core value (LCV) and the Cost of Stability (CoS). The LCV is the minimal deviation cost that admits a stable imputation, so a low LCV indicates high stability. The CoS is the external subsidy required to make the grand coalition stable, so a low CoS also indicates high stability. These three measures are related: by definition, the core is non-empty iff the game has a non-positive LCV and iff the game has a non-positive CoS.

Bachrach et. al. [3] initiated the study of the relation between agent failures and stability. They showed that when starting with a simple game with *zero failure probabilities*, increasing failure probabilities can only *increase* stability of the game in terms of non-emptiness of the core, and thus under all our measures.[9] Later, it was demonstrated [6] that in simple games, increasing failure probabilities starting from *non-zero values* may not preserve non-emptiness of the core, i.e., might *reduce* stability under all our measures. These results apply to WVGs as they are simple games. However, it was proved [6] that non-emptiness of the core is *always* preserved when failure probabilities are increased, starting from possibly non-zero values, if the game is *totally balanced* (i.e., if every subgame has non-empty core). This discussion indicates that although there is evidence that failures help stability in other classes of cooperative games, the relation is not so clear-cut in WVGs; in some R-WVGs increasing failure probabilities increases stability, while in others it decreases stability. We empirically show that even in R-WVGs, *on average* increasing failure probabilities increases stability.

First, we analyze R-WVGs where reliabilities of all the agents are *equal*. For 100 values of uniform reliability from 0.01 to 1, we generated 10^6 R-WVGs with the number of agents drawn uniformly at random between 5 and 10. We had few agents since computing the stability measures is computationally hard, and we solve many games to compute the average stability level. Weights were sampled from various distributions: Gaussian, Uniform, Poisson and Exponential. Figures 1 (for Gaussian) and 2 (for Uniform) show that the average LCV, the average CoS, and the probability of having an empty core (measures of instability) increase with the uniform reliability. Thus, stability increases as the uniform failure probability increases, according to all our measures. The plots for Poisson and Exponential are omitted as they are very similar.

[9] Recall the quantitative and qualitative measure are linked.

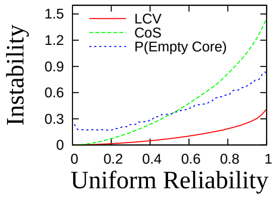

Fig. 1. Gaussian Weights

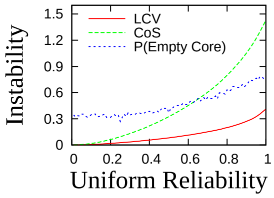

Fig. 2. Uniform Weights

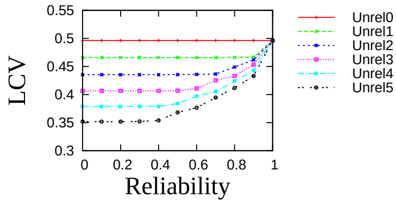

Fig. 3. Few Unreliable Agents

Next, we analyze games where only a few agents are unreliable. One such a domain is a decision making body where most decision makers are known to either support or object a legislation, and lobbying agents may convince the others to vote for it, but may fail with a certain probability. We built 10^4 WVGs with 30 agents, and weights uniformly chosen from 1 to 10.[10] We then made up to 5 agents unreliable *one by one*, changing their reliability to each of 10 values from 0.1 to 1, and measured the LCV using Algorithm CORE-FEW-UNREL.[11] The results, shown in Figure 3, indicate that *instability* (as measured by the average LCV) increases as agents become more reliable, so again increasing failure probabilities tends to increase stability on average. Further, we can see that the more agents we have that may fail, the more stable the game is.

All the above results reflect a similar pattern: Although there exist specific examples where making agents less reliable makes the game less stable, *on average* increasing failure probabilities in an R-WVG makes the game *more* stable. That is, failures help stabilize the game on average, which conforms to the results of [6].

7 Conclusion

We examined the impact of possible agents failures on the solutions to weighted voting games using the reliability extension model [3], which resulted in the class of R-WVGs. We contrasted the computational ease of calculating the value of a coalition and several core related questions in WVGs with hardness results for R-WVGs. We developed tractable tools for computing various solution concepts (core related or power indices) approximately, or exactly in restricted games. Using these tools, we explored the relation between agent reliability and stability, and empirically showed that on average higher failure probabilities make the game more stable.

Many questions are left open for future research. Could better computational tools be developed to solve R-WVGs, allowing us to handle larger games? Are there specific WVG domains that exhibit a different relation between agent reliabilities and stability? Does the general trend where introducing more failures causes the game to become more stable hold in other classes of cooperative games? Finally, how do failures affect other cooperative game solutions, such as coalition structures or the nucleolus?

---

[10] Our algorithms are pseudo-polynomial in the weights, so low weights are required.

[11] The algorithm finds a core imputation, but can easily be extended to compute the LCV, the CoS, and to test emptiness of the core.

References

1. Aziz, H., Lachish, O., Paterson, M., Savani, R.: Power indices in spanning connectivity games. In: AAIM. (2009) 55–67
2. Bachrach, Y., Markakis, E., Resnick, E., Procaccia, A., Rosenschein, J., Saberi, A.: Approximating power indices: theoretical and empirical analysis. In: AAMAS. (2010) 105–122
3. Bachrach, Y., Meir, R., Feldman, M., Tennenholtz, M.: Solving cooperative reliability games. In: UAI. (2011) 27–34
4. Bachrach, Y., Rosenschein, J.: Power in threshold network flow games. In: AAMAS. (2009) 106–132
5. Bachrach, Y., Elkind, E., Meir, R., Pasechnik, D., Zuckerman, M., Rothe, J., Rosenschein, J.S.: The cost of stability in coalitional games. In: SAGT. (2009) 122–134
6. Bachrach, Y., Kash, I., Shah, N.: Agent failures in totally balanced games and convex games. In: WINE. (2012) 15–29
7. Bala, V., Goyal, S.: A strategic analysis of network reliability. Review of Economic Design **5**(3) (2000) 205–228
8. Banzhaf, J.F.: Weighted voting doesn't work: A mathematical analysis. Rutgers Law Review **19** (1964) 317
9. Baye, M.R., Kovenock, D., De Vries, C.G.: Rigging the lobbying process: an application of the all-pay auction. The American Economic Review **83**(1) (1993) 289–294
10. Bilbao, J.M.: Cooperative games on combinatorial structures. Kluwer Academic Boston (2000)
11. Billand, P., Bravard, C., Sarangi, S.: Nash networks with imperfect reliability and heterogeous players. IGTR **13**(2) (2011) 181–194
12. Chalkiadakis, G., Elkind, E., Wooldridge, M.: Computational aspects of cooperative game theory. Synth. Lect. on Artif. Intell. and Machine Learning **5**(6) (2011) 1–168
13. Conitzer, V., Sandholm, T.: Computing shapley values, manipulating value division schemes, and checking core membership in multi-issue domains. In: AAAI. (2004) 219–225
14. Deng, X., Papadimitriou, C.H.: On the complexity of cooperative solution concepts. MOR **19**(2) (1994) 257–266
15. Dubey, P.: On the uniqueness of the shapley value. IJGT **4**(3) (1975) 131–139
16. Elkind, E., Goldberg, L., Goldberg, P., Wooldridge, M.: On the computational complexity of weighted voting games. Ann. Math. Artif. Intell. **56**(2) (2009) 109–131
17. Fatima, S.S., Wooldridge, M., Jennings, N.R.: A randomized method for the Shapley value for the voting game. In: AAMAS. (2007) 955–962
18. Garey, M.R., Johnson, D.S.: Computers and Intractability: a Guide to the Theory of NP-Completeness. W. H. Freeman and Company (1979)
19. Gillies, D.B.: Some theorems on n-person games. PhD thesis, Princeton University (1953)
20. Kannan, R., Sarangi, S., Iyengar, S.: A simple model for reliable query reporting in sensor networks. In: Information Fusion. (2002) 1070–1075
21. Matsui, Y., Matsui, T.: A survey of algorithms for calculating power indices of weighted majority games. Journal of the Operations Research Society of Japan **43** (2000) 71–86
22. Meir, R., Tennenholtz, M., Bachrach, Y., Key, P.: Congestion games with agent failures. In: AAAI. (2012) 1401–1407
23. Penn, M., Polukarov, M., Tennenholtz, M.: Congestion games with failures. In: EC. (2005) 259–268
24. Shapley, L.S.: A value for n-person games. Contrib. to the Theory of Games (1953) 31–40
25. Shapley, L., Shubik, M.: Quasi-cores in a monetary economy with nonconvex preferences. Econometrica: Journal of the Econometric Society (1966) 805–827
26. Young, H.: The allocation of funds in lobbying and campaigning. Behavioral Science **23**(1) (1978) 21–31

APPENDIX

The Shapley Value

Theorem 5. *If all the weights are represented in unary, and all the reliabilities are equal, then the problem of computing the Shapley value of an agent in an R-WVG is in $\mathcal{P}$.*

Proof. Consider an R-WVG $G^{\mathbf{r}} = (N, \mathbf{w}, t, \mathbf{r})$ where $r_j = p$ for all j (uniform reliability) and $|N| = n$. Bachrach et. al. [3] showed that the Shapley value of an agent i in $G^{\mathbf{r}}$ satisfies[12]

$$\varphi_i = \frac{r_i}{n!} \cdot \sum_{\pi \in S_n} \left[\sum_{S \subseteq \Gamma_i^\pi} m_i(S) \cdot \Pr[S|\Gamma_i^\pi] \right], \tag{3}$$

where $m_i(S) = v(S+i) - v(S)$ is the marginal contribution of agent i to coalition S in the base game, and Γ_i^π is the set of agents before agent i in permutation π (as defined in Section 2). Changing the order of summations, and substituting $r_i = p$, we get:

$$\begin{aligned}
\varphi_i &= \frac{p}{n!} \sum_{S \subseteq N-i} m_i(S) \cdot \left[\sum_{\pi \in S_n \ s.t.\ S \subseteq \Gamma_i^\pi} \Pr[S|\Gamma_i^\pi] \right] \\
&= \frac{p}{n!} \sum_{S \subseteq N-i} m_i(S) \cdot \left[\sum_{\Gamma \subseteq N-i \ s.t.\ S \subseteq \Gamma} \left(\Pr[S|\Gamma] \cdot \sum_{\pi \in S_n \ s.t.\ \Gamma_i^\pi = \Gamma} 1 \right) \right] \\
&= \frac{p}{n!} \sum_{S \subseteq N-i} m_i(S) \cdot \left[\sum_{\Gamma \subseteq N-i \ s.t.\ S \subseteq \Gamma} p^{|S|}(1-p)^{|\Gamma|-|S|} |\Gamma|!(n-|\Gamma|-1)! \right].
\end{aligned}$$

Here, the second equality follows by taking cases over the possible values of Γ_i^π (and grouping all permutations π that yield the same Γ_i^π). The third equality follows by observing that $\Pr[S|\Gamma] = p^{|S|} \cdot (1-p)^{|\Gamma|-|S|}$ since $|S|$ agents need to succeed and $|\Gamma| - |S|$ agents need to fail, and $\sum_{\pi \in S_n \ s.t.\ \Gamma_i^\pi = \Gamma} 1 = |\Gamma|! \cdot (n - |\Gamma| - 1)!$ since agents of Γ can appear in any of $|\Gamma|!$ orders before agent i, and the rest (except agent i) can appear in any of $(n - |\Gamma| - 1)!$ orders after agent i. Observe that the quantity inside the square brackets only depends on $|S|$ and $|\Gamma|$. Let $q = |S|$ and $k = |\Gamma|$. Thus,

$$\begin{aligned}
\varphi_i &= \frac{p}{n!} \sum_{S \subseteq N-i} m_i(S) \cdot \left[\sum_{\Gamma \subseteq N-i \ s.t.\ S \subseteq \Gamma} p^{|S|} \cdot (1-p)^{|\Gamma|-|S|} \cdot |\Gamma|! \cdot (n-|\Gamma|-1)! \right] \\
&= \frac{p}{n!} \sum_{q=1}^{n-1} \left(\sum_{\substack{S \in N-i \\ s.t.\ |S|=q}} m_i(S) \right) \left[\sum_{k=q}^{n-1} \left(\sum_{\substack{\Gamma \subseteq N-i \ s.t. \\ S \subseteq \Gamma, |\Gamma|=k}} 1 \right) \cdot p^q (1-p)^{k-q} k! (n-k-1)! \right].
\end{aligned}$$

[12] Their formula works for the reliability extension of any cooperative game.

First, the number of $\Gamma \subseteq N - i$ such that $S \subseteq \Gamma$ and $|\Gamma| = k$ is $\binom{n-q}{k-q}$, since the q agents of S need to be selected and we have freedom to choose any $k - q$ agents out of the remaining $n - q$ agents. Further, note that $m_i(S) \in \{0, 1\}$, since in our case the base game is a WVG, and thus a simple game. Hence, we want to count the number of $S \subseteq N - i$ with $|S| = q$ such that $m_i(S) = 1$. Let $c_i(q)$ denote this quantity. Then,

$$\varphi_i = \frac{p}{n!} \cdot \sum_{q=1}^{n-1} c_i(q) \cdot \left[\sum_{k=q}^{n-1} \binom{n-q}{k-q} \cdot p^q \cdot (1-p)^{k-q} \cdot k! \cdot (n-k-1)! \right]. \tag{4}$$

From the above equation, it is clear that the only computational hurdle is $c_i(q)$. It is easy to check that for the base WVG, $m_i(S) = 1$ if and only if $w(S) \in [t - w_i, t)$. Let $c_i(w, q)$ denote the number of subsets of $N - i$ whose weight is w and size is q. Then, $c_i(q) = \sum_{w=t-w_i}^{t-1} c_i(w, q)$. Further, let $c_i(w, q, j)$ be the number of subsets of $\{1, \ldots, j\} - i$ that have weight w and size q. Then, we have

$$c_i(w,q,j) = \begin{cases} c_i(w,q,j-1) + c_i(w-w_j, q-1, j-1) & \text{if } j \neq i, \\ c_i(w,q,j-1) & \text{if } j = i, \end{cases}$$

and $c_i(w, q) = c_i(w, q, n)$. Thus, we can compute $c_i(w, q, j)$ for $1 \leq w \leq t$, $1 \leq q < n$, and $1 \leq j \leq n$ in total time $O(t \cdot n^2)$ using dynamic programming. This easily yields us $c_i(q)$ for $1 \leq q < n$ required in Equation (4). The rest of the computation of Equation (4) can be performed in $O(n^2)$ time. Note that the inner loop requires $O(n)$ iterations, but the quantity inside the brackets for an iteration can be computed in constant time from that quantity in the previous iteration. Using this, the inner loop can be executed in $O(n)$ time. Since the outer loop has $O(n)$ such iterations, we get that the total time is $O(n^2)$. Thus, the overall running time of this two-phase computation is $O(t \cdot n^2) + O(n^2) = O(t \cdot n^2)$. □

Note that our $c_i(w, q, j)$ is very similar to the $c_i(w, q, x)$ defined by Matsui et. al. [21]. Using $c_i(w, q, x)$ is more efficient in some games, but we demonstrate our results using $c_i(w, q, j)$ since it is easier to work with and sufficient to derive the results. Computing $c_i(w, q, x)$ is essentially the main component in the pseudo-polynomial time algorithms of Matsui et. al. [21] for the two indices in the base WVGs. Our algorithms build this, and use additional ideas inherited from [3]. Thus, our algorithms are generalizations of those by Matsui et. al. [21], but have identical asymptotic running times since computing $c_i(w, q, j)$ is still the time-critical step in our algorithms.

The Banzhaf Index

The Banzhaf index of agent i, denoted β_i, is its average marginal contribution to all coalitions that do not contain it [8]. For a cooperative game $G = (N, v)$, we have

$$\beta_i = \frac{1}{2^{n-1}} \sum_{C \subseteq N - i} (v(C+i) - v(C)). \tag{5}$$

Now, we prove a useful result regarding the Banzhaf index in the reliability extension model, which is similar to Equation (3) for the Shapley value.

Lemma 1. *Let $G^{\mathbf{r}} = (N, v, \mathbf{r})$ be any reliability game with n agents, and $i \in N$ be any agent. Let $m_i(S) = v(S+i) - v(S)$ is the marginal contribution of agent i to coalition S in the base game. Then, the Banzhaf index of agent i in $G^{\mathbf{r}}$ satisfies*

$$\beta_i = \frac{r_i}{2^{n-1}} \cdot \sum_{C \subseteq N-i} \left[\sum_{S \subseteq C} m_i(S) \cdot \Pr[S|C] \right],$$

Proof. The proof is essentially works by starting from the definition of the Banzhaf index, and applying several algebraic transformations that are similar to those applied in [3] for the Shapley value. Let $v^{\mathbf{r}}$ be the characteristic function of $G^{\mathbf{r}}$. Then, by the definition of the Banzhaf index, we have

$$\beta_i = \frac{1}{2^{n-1}} \cdot \sum_{C \subseteq N-i} v^{\mathbf{r}}(C+i) - v^{\mathbf{r}}(C). \tag{6}$$

Next, we expand $v^{\mathbf{r}}(C+i)$ and $v^{\mathbf{r}}(C)$.

$$\begin{aligned} v^{\mathbf{r}}(C+i) &= \sum_{S \subseteq C+i} \Pr[S|C+i] \cdot v(S) \\ &= \sum_{S \subseteq C+i, i \in S} \Pr[S|C+i] \cdot v(S) + \sum_{S \subseteq C+i, i \notin S} \Pr[S|C+i] \cdot v(S) \\ &= \sum_{S \subseteq C} \Pr[S+i|C+i] \cdot v(S+i) + \sum_{S \subseteq C} \Pr[S|C+i] \cdot v(S) \\ &= \sum_{S \subseteq C} \Pr[S|C] \cdot r_i \cdot v(S+i) + \sum_{S \subseteq C} \Pr[S|C] \cdot (1-r_i) \cdot v(S), \end{aligned} \tag{7}$$

where the second equality follows by breaking the summation into two cases — $i \in S$ and $i \notin S$, the third equality follows by a change of variable, and the final equality follows by explicitly observing and separating the survival or failure of agent i. Also,

$$v^{\mathbf{r}}(C) = \sum_{S \subseteq C} \Pr[S|C] \cdot v(S). \tag{8}$$

Substituting Equations (7) and (8) in Equation (6), we get

$$\begin{aligned} \beta_i &= \frac{1}{2^{n-1}} \cdot \sum_{C \subseteq N-i} \left[\sum_{S \subseteq C} \Pr[S|C] \cdot r_i \cdot v(S+i) + \sum_{S \subseteq C} \Pr[S|C] \cdot (1-r_i) \cdot v(S) \right. \\ &\qquad \left. - \sum_{S \subseteq C} \Pr[S|C] \cdot v(S) \right] \\ &= \frac{1}{2^{n-1}} \cdot \sum_{C \subseteq N-i} \left[r_i \cdot \sum_{S \subseteq C} \Pr[S|C] \cdot v(S+i) - r_i \cdot \sum_{S \subseteq C} \Pr[S|C] \cdot v(S) \right] \end{aligned}$$

$$= \frac{r_i}{2^{n-1}} \cdot \sum_{C \subseteq N-i} \left[\sum_{S \subseteq C} (v(S+i) - v(S)) \cdot \Pr[S|C] \right]$$
$$= \frac{r_i}{2^{n-1}} \cdot \sum_{C \subseteq N-i} \left[\sum_{S \subseteq C} m_i(S) \cdot \Pr[S|C] \right],$$

as required. □

Theorem 6. *If all the weights are represented in unary, and all the reliabilities are equal, then the problem of computing the Banzhaf index of an agent in an R-WVG is in $\mathcal{P}$.*

Proof. This proof is along the line of the proof of Theorem 5. Consider an R-WVG $G^{\mathbf{r}} = (N, \mathbf{w}, t, \mathbf{r})$ where the uniform reliability is p. Using Lemma 1, we have the following result regarding the Banzhaf index of agent i in $G^{\mathbf{r}}$, which is similar to Equation (3) for the Shapley value.

$$\beta_i = \frac{r_i}{2^{n-1}} \cdot \sum_{C \subseteq N-i} \left[\sum_{S \subseteq C} m_i(S) \cdot \Pr[S|C] \right],$$

where $m_i(S) = v(S+i) - v(S)$ is the marginal contribution of agent i to coalition S in the base game. Now, following the exact steps of the proof of Theorem 5,

$$\begin{aligned}
\beta_i &= \frac{r_i}{2^{n-1}} \cdot \sum_{C \subseteq N-i} \left[\sum_{S \subseteq C} m_i(S) \cdot \Pr[S|C] \right] \\
&= \frac{p}{2^{n-1}} \cdot \sum_{S \subseteq N-i} m_i(S) \cdot \left[\sum_{\substack{C \subseteq N-i \\ s.t.\ S \subseteq C}} \Pr[S|C] \right] \\
&= \frac{p}{2^{n-1}} \cdot \sum_{S \subseteq N-i} m_i(S) \cdot \left[\sum_{\substack{C \subseteq N-i \\ s.t.\ S \subseteq C}} p^{|S|} \cdot (1-p)^{|C|-|S|} \right] \\
&= \frac{p}{2^{n-1}} \cdot \sum_{q=1}^{n-1} \left(\sum_{\substack{S \subseteq N-i \\ s.t.\ |S|=q}} m_i(S) \right) \cdot \left[\sum_{k=q}^{n-1} \binom{n-q}{k-q} p^{|S|} \cdot (1-p)^{|C|-|S|} \right] \\
&= \frac{p}{2^{n-1}} \cdot \sum_{q=1}^{n-1} c_i(q) \cdot \left[\sum_{k=q}^{n-1} \binom{n-q}{k-q} p^{|S|} \cdot (1-p)^{|C|-|S|} \right],
\end{aligned} \tag{9}$$

where the second equality follows by changing the order of the summations and substituting $r_i = p$, the third equality follows since $\Pr[S|C]$ is the probability that $|S|$ agents survive and $|C|-|S|$ agents fail, the fourth equality follows by breaking the summations over $|S|$ and $|C|$, and the final equality follows by the definition of $c_i(q)$ given in the proof of Theorem 5. Note that Equation (9) is almost identical to Equation (4), except that we have $p/2^{n-1}$ in place of $p/n!$ and the term $k! \cdot (n-k-1)!$ from Equation (4) disappears. Using the dynamic programming method to compute $c_i(q)$ and other implementation details given in the proof of Theorem 5, we can conclude that the Banzhaf index can also be computed in $O(t \cdot n^2)$ time. □