

Deterministic algorithms for inhomogeneous Bernoulli trials: Shapley value of network devices

Jesse D Wei¹, Guo Wei²

¹ Department of Computer Science, University of North Carolina, Chapel Hill, NC, 27599, USA

² Department of Mathematics & Computer Science, University of North Carolina
at Pembroke, Pembroke, North Carolina 28372, U.S.A.

Abstract. Suppose that n computer devices are to be connected to a network via inhomogeneous Bernoulli trials. The Shapley value of a device quantifies how much the network's value increases due to the participation of that device. Characteristic functions of such games are naturally taken as the belief function (containment function) and Choquet capacity (hitting probability) of a random set (random network of devices).

Traditionally, the Shapley value is either calculated as the expected marginal contribution over all possible coalitions (subnetworks), which results in exponential computational complexity, or approximated by the Monte Carlo sampling technique, where the performance is highly dependent on the stochastic sampling process.

The purpose of this study is to design deterministic algorithms for games formulated via inhomogeneous Bernoulli trials that approximate the Shapley value in linear or quadratic time, with rigorous error analysis (Sections 3 and 4). Additionally, we provide a review of relevant literature on existing calculation methods in Remark 3.1 and Appendix I.

A further goal is to supplement Shapley's original proof by deriving the Shapley value formula using a rigorous approach based on definite integrals and combinatorial analysis. This method explicitly highlights the roles of the Binomial Theorem and the Beta function in the proof, addressing a gap in Shapley's work (Appendix II).

Key words: Inhomogeneous Bernoulli trials, Random network, Choquet capacity, Shapley value, Deterministic approximation algorithm

MSC: 60D05, 68Q87

ACM: G.3, I.2.4.

1. KNOWN APPROXIMATION METHODS TO THE SHAPLEY VALUE AND THE EXISTING RESEARCH GAP

Weighted Shapley values were introduced in Shapley's original Ph.D. dissertation (1953a, 1953b). Owen (1968, 1972) studied weighted Shapley values through probabilistic approaches. Axiomatizations of nonsymmetric values were investigated by Weber (Chapter 7 this volume), Shapley (1981), Kalai and Samet (1987), and Hart and Mas-Colell (1987). In Appendix I, a brief review of various studies of the Shapley value is provided.

Moreover, Murofushi and Soneda (Murofushi and Soneda 1993) introduced the interaction index for a pair of players. In 1997, Grabisch (Grabisch, 1997) generalized the index to any coalition S . However, interaction index of coalitions will not be covered in this work.

In the applications, one major gap that has not been satisfactorily resolved is the exponential computational time. While there are some approximation methods with polynomial time, these methods depend on the stochastic samples, whose performance is highly dependent on the samples. The main goal of this study is, for inhomogeneous Bernoulli trials where devices independently join the network, to design a deterministic algorithm that approximates the Shapley value in linear time

² Corresponding author. tel.: (910) 521 6582.

E-mail addresses: jesse@cs.unc.edu (J. Wei), guo.wei@uncp.edu (G. Wei).

(with respect to the number of devices) and with an adjustable calculation accuracy (Sections 2 through 5). Our approach is to incorporate a random set model and sub-players technique.

In this paper, devices of a network are understood as payers of a game, and subnetworks are identical with coalitions, etc. Section 2 presents the random set representation of games. In Sections 3 through 6, for the Choquet capacity associating with the random set X that is formed through n inhomogeneous Bernoulli trials, we will design deterministic algorithms, linear or quadratic time in the number of players, to approximate their Shapley values. Moreover, error analysis will be performed to examine the effectiveness of the algorithms, along with examples for examining the accuracy of the algorithm. Specifically, In Section 3, a deterministic algorithm is designed to approximate the Shapley value in linear time, along with a rigorous analysis of errors. Section 4 provides illustrative examples, including a network application. Section 5 discusses the computational complexity of the proposed algorithm, with the comparison to the calculation using the exact Shapley value formula. Section 6 develops another algorithm, with an emphasis on the error analysis methods through the elementary symmetric sum and Taylor approximation. In Section 7 (Appendix I), historic remarks are provided to review the various approaches to interpret the Shapley value and reduce the computation. In Section 8 (Appendix II), we supplement Shapley's original proof by deriving the Shapley value formula using a rigorous approach based on definite integrals and combinatorial analysis. This method explicitly highlights the roles of the Binomial Theorem and the Beta function in the proof, addressing a gap in Shapley's work.

2. RANDOM SET REPRESENTATION OF CHARACTERISTIC FUNCTIONS

Let $E = \{1, \dots, n\}$ be the set of n players (i.e. devices). The characteristic function of a game of these n players is a real-valued set function defined on the power set of E , i.e. $v : 2^E \rightarrow [0, 1]$ satisfying

- i) $v(\emptyset) = 0, v(E) = 1$. (Normalization)
- ii) If $A, B \in 2^E, A \subseteq B$, then $v(A) \leq v(B)$. (Monotonicity)

Definition 2.1 (Möbius Coefficients). The Möbius transform of v is given by

$$m_v(S) = \sum_{T \subseteq S} (-1)^{|S|-|T|} v(T).$$

Definition 2.2 (Random Set). A random set X is a random element taking values in 2^E . Its Choquet capacity (covering function) is

$$v(S) = \mathbb{P}(X \cap S \neq \emptyset), \quad S \subseteq E.$$

Theorem 2.3 (See Shafer, 1976). *If v satisfies $v(E) = 1$ and $m_v(S) \geq 0$ for all $S \subseteq E$, then there exists a (unique) random set X such that*

$$v(S) = \mathbb{P}(X \cap S \neq \emptyset) \quad \forall S \subseteq E.$$

This is a well known result. Its proof proceeds in three steps:

- (1) **Möbius positivity implies belief measure:** A function v with non-negative Möbius coefficients is a belief function in Dempster-Shafer theory.
- (2) **Constructing the random set:** Define the probability distribution:

$$\mathbb{P}(X = T) = m_v(T) \quad \forall T \subseteq E$$

This is valid since $\sum_{T \subseteq E} m_v(T) = v(E) = 1$.

- (3) **Verification:** The hitting probability satisfies:

$$\mathbb{P}(X \cap S \neq \emptyset) = \sum_{\substack{T \subseteq E \\ T \cap S \neq \emptyset}} m_v(T)$$

By Möbius inversion:

$$v(S) = \sum_{T \subseteq S} m_v(T) = \sum_{\substack{T \subseteq E \\ T \cap S^c \neq \emptyset}} m_v(T)$$

where the last equality holds because $m_v(T) = 0$ when $T \not\subseteq S$.

Therefore, any totally positive characteristic function with $v(E) = 1$ can be represented as the hitting probability of a random set.

Remark 2.4. Given two capacities v and u on E satisfying (i) $v(E) = u(E) = 1$, and (ii) u is the conjugacy of v , i.e., for every $S \subseteq E$, it holds that $u(S) = 1 - v(S^c)$. Then the Shapley values of two capacities are equal. In fact, we have

$$\begin{aligned} \phi_i(u) &= \sum_{S \subseteq E \setminus \{i\}} \frac{s!(n-1-s)!}{n!} [u(S \cup \{i\}) - u(S)] \\ &= \sum_{S \subseteq E \setminus \{i\}} \frac{s!(n-1-s)!}{n!} [(1 - v((S \cup \{i\})^c)) - (1 - v(S^c))] \\ &= \sum_{S \subseteq E \setminus \{i\}} \frac{s!(n-1-s)!}{n!} [v(S^c) - v((S \cup \{i\})^c)]. \end{aligned}$$

Let $T = S^c$. Then $s = n - t$. Hence we have

$$\phi_i(u) = \sum_{i \in T} \frac{(n-t)!(t-1)!}{n!} [v(T) - v(T \setminus \{i\})].$$

Now, let $T' = T \setminus \{i\}$, so $t = 1 + t'$. Then we get

$$\phi_i(u) = \sum_{T' \subseteq E \setminus \{i\}} \frac{(n-1-t')!t'!}{n!} [v(T' \cup \{i\}) - v(T')],$$

where the right hand side is $\phi_i(v)$.

3. LOWERING THE COMPUTATIONAL COMPLEXITY OF SHAPLEY VALUE

Let $E = \{1, 2, \dots, n\}$, with discrete topology, and $\mathcal{F} = \{F \subseteq E\}$. Let $\mathcal{F}_S = \{F \in \mathcal{F} | F \cap S \neq \emptyset\}$, the set of all subsets of E that intersect with S . Suppose that X is a random set defined on E , and T is the Choquet capacity (covering function) of X , i.e. $T(S) = \mathbb{P}(X \cap S \neq \emptyset)$, the probability that X intersects with S . Also assume that $\mathbf{P}$ is the associated probability measure defined on the hyperspace space $\mathcal{F}$ with the Fell topology (actually discrete as E is finite), satisfying $\mathbf{P}(\mathcal{F}_S) = T(S)$.

Also, define $\mathcal{F}^S = \{F \subseteq E | F \cap S = \emptyset\}$. The events $(X \in \mathcal{F}^S)$ and $(X \in \mathcal{F}^{\{i\}})$, where $i \notin S$, are not independent? In fact, independence holds only if the random set X treats the exclusion of i and the exclusion of S as independent events. This is true for random sets with independent element inclusions, e.g., inhomogeneous Bernoulli trials where each element i of E is included in X with probability p_i , independently of all other elements. Independence fails for most other cases, including fixed-size random sets (e.g., uniform k -subsets), sets with correlated element inclusions etc.

When the elements of the random set X are independent, its Choquet capacity T takes a particularly simple multiplicative form. In the next, let us derive the concrete form of the Choquet capacity T for the inhomogeneous Bernoulli trials.

3.1. Key assumption: Independence.

- Each element $i \in E$ is included in X independently with probability p_i :

$$\mathbb{P}(i \in X) = p_i, \quad \mathbb{P}(i \notin X) = 1 - p_i.$$

- The probability that X does not intersect S (i.e., $X \cap S = \emptyset$) is the product of the probabilities that each $j \in S$ is not in X :

$$\mathbb{P}(X \cap S = \emptyset) = \prod_{j \in S} (1 - p_j).$$

Choquet Capacity $T(S)$. By definition, $T(S)$ is the probability that X intersects S :

$$T(S) = \mathbb{P}(X \cap S \neq \emptyset) = 1 - \mathbb{P}(X \cap S = \emptyset).$$

Substituting the independent case gives

$$T(S) = 1 - \prod_{j \in S} (1 - p_j).$$

3.2. Methods to reduce complexity.

3.2.1. 1. *Assume Independence.* Suppose that the elements of X are independent. Recall that $p_i = \mathbb{P}(X \ni i)$, which we often write as $\mathbb{P}(i \in X)$. Then

$$(3.1) \quad T(S) = 1 - \prod_{j \in S} (1 - p_j), \quad T(S \cup \{i\}) = 1 - (1 - p_i) \prod_{j \in S} (1 - p_j).$$

Hence, the marginal contributions can be written as

$$(3.2) \quad \mathbb{P}(i \in X \text{ and } X \cap S = \emptyset) = T(S \cup \{i\}) - T(S) = p_i \prod_{j \in S} (1 - p_j).$$

The total capacity in this setting is $T(E) = 1 - \prod_{j \in E} (1 - p_j) < 1$.

Subsequently, the Shapley value for the i th player is

$$(3.3) \quad \phi_i(T) = \sum_{S \subseteq E \setminus \{i\}} \frac{s!(n-1-s)!}{n!} \cdot p_i \cdot \prod_{j \in S} (1 - p_j),$$

which seems unsimplifiable in general, as analyzed below.

The Shapley value is the expected marginal contribution of a player and thus it can be written in the expectation form (Refer to (3.1) and (3.2)):

$$\phi_i(T) = \mathbb{E}_{S \subseteq E \setminus \{i\}} [p_i \prod_{j \in S} (1 - p_j)],$$

where the probability of each subset $S \subseteq E \setminus \{i\}$ is $\frac{s!(n-1-s)!}{n!}$.

For the network application, the Shapley value of a device quantifies its contribution to the whole network, taking into account the probabilistic nature of participation by all devices.

Generally, this computation complexity is an obstinate challenge, but in some special cases, it might be significantly reduced.

- **Exponential complexity in general:**

- If X is arbitrary, computing $p_i \prod_{j \in S} (1 - p_j) (= T(S \cup \{i\}) - T(S))$ may require summing over all subsets $S \subseteq E \setminus \{i\}$, which is $O(2^{n-1})$, and infeasible for large n .

- **Special cases where complexity can be reduced:**

- When X has structure (e.g., independence, sparsity, or a known distribution).
- When $T(S)$ can be computed efficiently (e.g., via inclusion-exclusion, Möbius inversion, or approximation methods).

The homogeneous case, where $p_j = p, 1 \leq i \leq n$, is perfectly resolvable. In this case, $T(S) = 1 - (1 - p)^{|S|}$, and the Shapley value for the i th player is simplified below.

$$\phi_i(T) = \sum_{S \subseteq E \setminus \{i\}} \frac{s!(n-1-s)!}{n!} \cdot p \cdot \prod_{j \in S} (1-p) = \sum_{S \subseteq E \setminus \{i\}} \frac{1}{n \binom{n-1}{s}} \cdot p \cdot (1-p)^s = \frac{p}{n} \sum_{S \subseteq E \setminus \{i\}} \frac{(1-p)^s}{\binom{n-1}{s}}.$$

Then we group subsets by size. The sum over all subsets $S \subseteq E \setminus \{i\}$ can be rewritten as a sum over possible subset sizes $s = 0$ to $s = n-1$, with $\binom{n-1}{s}$ subsets of size s :

$$(3.4) \quad \phi_i(T) = \frac{p}{n} \sum_{s=0}^{n-1} \binom{n-1}{s} \frac{(1-p)^s}{\binom{n-1}{s}} = \frac{p}{n} \sum_{s=0}^{n-1} (1-p)^s = \frac{p}{n} \frac{1 - (1-p)^n}{1 - (1-p)} = \frac{1}{n} [1 - (1-p)^n] = \frac{1}{n} \times T(E).$$

which gives a closed form of the Shapley value, with the computational complexity of $O(n)$.

The general case, i.e. inhomogeneous case, is that not all p_i 's are equal.

We will develop methods, in the remainder of Section 3 through Section 6 to effectively estimate the Shapley value (3.3) according to two cases: 1) p_i 's are not all equal, but they are rational numbers, and 2) p_i 's are arbitrary real numbers.

Case 1: p_i 's are not all equal, but they are (positive) rational numbers.

Assume all p_i are positive rational numbers. We first write each probability p_i in a fractional form:

$$p_i = \frac{r_i}{s_i}, \quad \text{with } \gcd(r_i, s_i) = 1, \quad 1 \leq i \leq n.$$

Then define $l = \text{lcm}(s_1, s_2, \dots, s_n)$. and express each p_i as a fraction with the same denominator l :

$$p_i = \frac{r_i}{s_i} = \frac{m_i}{l}, \quad \text{where } m_i = r_i \cdot \frac{l}{s_i} \in \mathbb{N}.$$

It is clear that m_i 's are positive integers. The number of operations needed to find the LCD is dominated by the prime factorization, roughly proportional to $O(n \cdot \sqrt{M})$ where M is the largest denominator of the fractions p_j 's. If the Euclidean algorithm is employed, it can avoid entire factorization, leveraging modular arithmetic for logarithmic-time convergence, i.e. $O(n \cdot \log(M))$.

First we construct a finite probability space, by defining a random set (RACS) Y on

$$M = \bigcup_{i=1}^n \{k_1^{(i)}, \dots, k_{m_i}^{(i)}\}$$

with

$$\mathbb{P}(k_j^{(i)} \in Y) = \frac{1}{l}, \quad k_j^{(i)} = i,$$

and having the following key properties:

Independence: All $k_j^{(i)}, 1 \leq j \leq m_i, 1 \leq i \leq n$, are independent.

Inclusion probability:

$$(3.5) \quad \mathbb{P}(i \in X) = p_i = \frac{m_i}{l} \approx 1 - \left(1 - \frac{1}{l}\right)^{m_i} = \mathbb{P}(i \in Y). \quad (\text{for large } l)$$

Note that $\mathbb{P}(i \in Y) = \mathbb{P}\left((k_1^{(i)} \in Y) \text{ or } \dots \text{ or } (k_{m_i}^{(i)} \in Y)\right)$. Let $m = \sum_{i=1}^n m_i$.

Recall (3.3)

$$\phi_i(T) = \sum_{S \subseteq E \setminus \{i\}} \frac{s!(n-1-s)!}{n!} \cdot p_i \cdot \prod_{j \in S} (1-p_j).$$

Let μ be the Choquet capacity of Y . It now follows from the result (3.4) of the homogeneous case, and the approximation (3.5) that

$$(3.6) \quad \phi_i(T) \approx \phi_i(\mu) = m_i \times \phi_{k_1^{(i)}}(\mu) = m_i \times \frac{1}{m} \left(1 - \left(1 - \frac{1}{l} \right)^m \right).$$

Remark 3.1. Traditional approaches to the independence case. In the literature, for large n there are some methods to approximate the sum (3.3). For example, if further all p_j 's are small (say $p_j = O(1/n)$), one can use the Poisson approximation or mean-field approximation (Lasry and Lions, 2007, Dughmi, 2017). In such a setting, the number of included elements $|S|$ is approximately Poisson-distributed with mean $\lambda = \sum_{j \neq i} (1 - p_j)$ (independent inclusions with probability $1 - p_j$). The product term approximates as

$$\prod_{j \in S} (1 - p_j) \approx e^{-\sum_{j \in S} p_j}.$$

The sum is dominated by subsets S with $|S| \approx \lambda$ due to two competing effects: The number of subsets of size k , $\binom{n-1}{k}$, is maximized at $k \approx \frac{n-1}{2}$. However, $\frac{1}{\binom{n-1}{k}}$ decreases as k moves away from 0 or $n-1$. Specifically, 1) The binomial coefficient favors mid-range k , but exponential decay shifts balance toward smaller k , and 2) The trade-off peaks at $k \approx \lambda$, where the Poisson approximation is most accurate.

Other methods include the mean-field approximation (Lasry and Lions, 2007, Dughmi, 2017) and the saddle-point or Laplace approximation (Daniels, 1954, Neyman, 2002, Kerman, Wootters, 2011).

The sum can be rewritten as

$$\sum_{k=0}^{n-1} \frac{1}{\binom{n-1}{k}} \sum_{\substack{S \subseteq E \setminus \{i\} \\ |S|=k}} \prod_{j \in S} (1 - p_j).$$

This resembles an inclusion-exclusion, weighted by the reciprocal of binomial coefficient. Surprisingly, the above sum can be approximated by a Riemann sum, by simplifying to (See Neyman, 2002, Feller, 1968)

$$\frac{1}{n} \sum_{k=1}^n \prod_{\substack{j=1 \\ j \neq i}}^n \left(1 - \frac{k}{n} p_j \right)$$

Remark 3.2. The mean-field approximation. The capacity of E is $T(E) = 1 - \prod_{i \in E} (1 - p_i)$. In the following, we will relate our approximation method to the mean-field approximation, where the latter is known in the literature. The sum of our approximated Shapley values $\phi_i(\mu)$ is

$$(3.7) \quad \sum_{i=1}^n \phi_i(\mu) = \sum_{i=1}^n m_i \frac{1}{m} \left(1 - \left(1 - \frac{1}{l} \right)^m \right) = 1 - \left(1 - \frac{1}{l} \right)^m.$$

Let $\tilde{p} = \frac{m}{l}$. Then

$$(3.8) \quad \phi_i(\mu) = \frac{p_i}{\tilde{p}} \left(1 - \left(1 - \frac{1}{l} \right)^m \right).$$

For large l , apply $\left(1 - \frac{1}{l} \right)^m \approx e^{-\frac{m}{l}} = e^{-\tilde{p}}$, and we get

$$(3.9) \quad \phi_i(\mu) = \frac{p_i}{\tilde{p}} (1 - e^{-\tilde{p}}).$$

This is essentially a mean-field approximation: treat all players as if their probabilities were equal to the average $\tilde{p}$. More specifically, the approximation $\left(1 - \frac{1}{l} \right)^m \approx e^{-\frac{m}{l}}$ is acceptable if m is large

but $m \leq l^2$. However, for $m > l^2$, this approximation cannot be used. The computation complexity is dominated by the calculation of l , which requires $O(n \times \log M)$ operations.

Remark 3.3. The weighted binomial-type sum. Let

$$\bar{q} = \frac{1}{n-1} \sum_{j \neq i} (1 - p_j).$$

For large n , the law of large numbers suggests that the average $\bar{q}$ dominates the behavior of the product, making the approximation accurate. Hence, $\phi_i(T)$ can be approximated as follows:

$$E_{|S|=k} \left[\prod_{j \in S} (1 - p_j) \right] \approx \left(\frac{1}{n-1} \sum_{j \neq i} (1 - p_j) \right)^k = \bar{q}^k,$$

$$\phi_i(T) \approx p_i \cdot \sum_{k=0}^{n-1} \frac{k!(n-1-k)!}{n!} \cdot \bar{q}^k.$$

This turns the Shapley value into a weighted binomial-type sum. Its computational complexity can be analyzed in two steps:

Step 1. Mean-field approximation of $E_{|S|=k} \left[\prod_{j \in S} (1 - p_j) \right]$

Instead of computing the product over all subsets S of size k , we approximate it using the average $\bar{q}$.

Computing $\bar{q}$ requires a single sum over $n-1$ players $\rightarrow \mathcal{O}(n)$.

Step 2. Shapley value approximation

$$\phi_i(T) \approx p_i \cdot \sum_{k=0}^{n-1} \frac{k!(n-1-k)!}{n!} \cdot \bar{q}^k$$

The sum runs over $k = 0$ to $n-1$, so there are n terms.

Each term involves:

- A combinatorial weight $\frac{k!(n-1-k)!}{n!}$, which can be rewritten as $\frac{1}{n \binom{n-1}{k}}$
- A power $\bar{q}^k$, which can be computed iteratively to avoid recomputation

Complexity:

- If computed naively (calculating each power $\bar{q}^k$ separately), it would be $\mathcal{O}(n^2)$
- If optimized (using $\bar{q}^k = \bar{q}^{k-1} \cdot \bar{q}$), it reduces to $\mathcal{O}(n)$

Overall Complexity:

- $\mathcal{O}(n)$ (due to the mean-field approximation and iterative computation)

The approximation $E_{|S|=k} \left[\prod_{j \in S} (1 - p_j) \right] \approx \bar{q}^k$ is valid for small k 's because sampling without replacement introduces stronger dependence. It deteriorates when k is close to $n-1$ unless p_j 's are nearly identical.

Before performing accuracy, we first analyze the computational complexity.

3.3. Complexity Analysis.

- LCM Computation: $O(n \log M)$ using Euclidean algorithm
- Approximation Formula:

$$\phi_i(\mu) = \frac{m_i}{m} \left(1 - \left(1 - \frac{1}{l} \right)^m \right)$$

- Operations: $O(n)$ after computing m and l

3.4. Accuracy analysis of our binomial approximation method via an example. Consider the following rational probabilities

$$P = [0.2, 0.5, 0.7, 0.3, 0.1, 0.9, 0.4]$$

with $l = 10$ and $m = [2, 5, 7, 3, 1, 9, 4]$.

TABLE 1. Shapley Value Comparison ($l = 10$)

Player i	Exact ϕ_i	Approx ϕ_i	Error (%)	m_i
1	0.0621	0.0632	+1.77%	2
2	0.1763	0.1580	-10.38%	5
3	0.2118	0.2212	+4.44%	7
4	0.0967	0.0948	-1.96%	3
5	0.0308	0.0316	+2.60%	1
6	0.2954	0.2844	-3.72%	9
7	0.1222	0.1264	+3.44%	4

In this example, errors come from the following:

(1) Independence Assumption:

Ignores correlations between $k_j^{(i)}$ points

(2) Finite-precision Effects:

$$\text{approximation } \left(1 - \frac{1}{l}\right)^m \approx e^{-m/l} \text{ holds (requiring large } l)$$

(3) Discretization Error:

$$\max_i \left| p_i - \frac{m_i}{l} \right| = 0 \text{ in this case}$$

The relative error is theoretically bounded by

$$\frac{|\phi_i - \phi_i(\mu)|}{\phi_i} \leq \frac{1}{2} \left(\frac{m}{l} - p_i \right) + O\left(\frac{m^2}{l^2}\right)$$

For this example, the conclusion is:

- Achieves $< 5\%$ error for most players when $l \geq 10$
- Outperforms for small p_i (e.g., Player 5)
- Computational cost remains $O(n \log M)$ for exact rational inputs

3.5. Rigorous error analysis of the approximation. Given the exact expression (3.3) for $\phi_i(T)$ and the approximation (3.6), we now turn to the error analysis.

3.5.1. Error decomposition. The total error $\mathcal{E}_i$ consists of

- (i) The probability space approximation error $\mathcal{E}_i^{(1)}$ that quantifies the discrepancy between:
- The true probability p_i (e.g., player i 's contribution probability), and
 - Its approximation $1 - \left(1 - \frac{1}{l}\right)^{m_i}$, where l and m_i are parameters (e.g., sample sizes or combinatorial counts).

By applying the second order Taylor approximation, we have

$$\mathcal{E}_i^{(1)} = \left| p_i - \left(1 - \left(1 - \frac{1}{l} \right)^{m_i} \right) \right|$$

(ii) The Shapley value estimation error:

$$\mathcal{E}_i^{(2)} = \left| \frac{1 - (1 - p_i)(1 - \bar{p})^{n-1}}{n} - \frac{m_i}{m} \left(1 - \left(1 - \frac{1}{l} \right)^m \right) \right|$$

Here, $\bar{p} = \frac{1}{n-1} \sum_{j \neq i} p_j$. The first term in the expression of $\mathcal{E}_i^{(2)}$ represents the expected marginal contribution of player i in a simplified setting (assuming independent trials and all other p_j 's ($j \neq i$) are equal). The second term is our empirical estimation. In fact, if p_j 's ($j \neq i$) are all equal, let $\bar{p}$ denote this common probability. Then the exact expression of the Shapley value for the i th player is

$$\phi_i(T) = \frac{1 - (1 - p_i)(1 - \bar{p})^{n-1}}{n}.$$

This can be derived from the formula for the Shapley value in this specific context.

The term $1 - (1 - p_i)(1 - \bar{p})^{n-1}$ represents the probability that player i is pivotal in some sense, e.g., the probability that at least one player including i contributes, minus the probability that others contribute without i .

Remark 3.4. On one hand, we say that p_i is the probability that player i contributes to a coalition, and on the other hand we know $p_i = \mathbb{P}(X \ni i)$. There is no difference between these two descriptions. p_i is the marginal probability that player i is part of a randomly formed coalition. For example, if coalitions form randomly, p_i could represent how likely i is to participate in any given group: $p_i = \mathbb{P}(X \ni i)$ represents the probability that a random coalition X contains i . This is a formal measure-theoretic way of stating the same idea. They both describe the same quantity, but the first is intuitive, while the second is rigorous.

Regarding Coalition Formation Model: If coalitions form independently (each player i joins with probability p_i), then we have $\mathbb{P}(X \ni i) = p_i$, $\mathbb{P}(X \supseteq S) = \prod_{j \in S} p_j$. This matches the first description. The term $(1 - p_i)(1 - \bar{p})^{n-1}$ in the exact expression arises because:

- 1) $(1 - p_i)$ = Probability i does not contribute.
- 2) $(1 - \bar{p})^{n-1} \approx$ Probability no other player contributes (assuming symmetry).

If coalition formation is not independent (e.g., players join/leave based on others), then $\mathbb{P}(X \ni i)$ could depend on correlations. In our setting, the descriptions are equivalent because independence is implicitly assumed.

3.5.2. Main error bound. In this section, we analyze the approximation error under the assumption that the p_j 's ($j \neq i$) are relatively small compared to p_i . For the case where all p_j 's (including p_i) are approximately equal, we refer to the previous result (3.4).

Theorem 3.5. *Suppose p_j 's ($j \neq i$) are relatively small compared to p_i . Then the approximation error satisfies:*

$$|\phi_i(T) - \phi_i(\mu)| \leq \underbrace{\frac{m_i m}{2l^2}}_{\text{Binomial approximation}} + \underbrace{\left| \frac{m_i}{m} - \frac{w_i}{W} \right|}_{\text{Weight discrepancy, vanishing}} + O\left(\frac{m_i m^2}{l^3}\right)$$

where

- $w_i = m_i/l = p_i$, $W = \sum_{i=1}^n w_i$, and $m = \sum_{i=1}^n m_i$,
- $\phi_i(T)$ is the exact Shapley value for player i ,
- $\phi_i(\mu) = \frac{m_i}{m} \left(1 - \left(1 - \frac{1}{l} \right)^m \right)$ is the binomial approximation.

Proof. We compare $\phi_i(\mu)$ with $\phi_i(T)$ through Taylor expansions and symmetric expansions.

Step 1: Exact Expression for $\phi_i(T)$

Using the weighted Shapley formula with $p_j = m_j/l$:

$$\begin{aligned}
\phi_i(T) &= \sum_{S \subseteq E \setminus \{i\}} \frac{s!(n-1-s)!}{n!} \cdot p_i \cdot \prod_{j \in S} (1-p_j) \\
&= \frac{m_i}{l} \sum_{S \subseteq E \setminus \{i\}} \frac{s!(n-1-s)!}{n!} \left(1 - \sum_{j \in S} \frac{m_j}{l} + \sum_{j,k \in S, j < k} \frac{m_j m_k}{l^2} + \dots \right) \\
&= \frac{m_i}{l} \left(1 - \frac{1}{2} \sum_{j \in E \setminus \{i\}} \frac{m_j}{l} + \frac{1}{n(n-1)(n-2)} \sum_{j,k \in S, j < k} \frac{m_j m_k}{l^2} + \dots \right) \\
&= \frac{m_i}{l} - \frac{m_i}{2l^2} (m - m_i) + \frac{1}{n(n-1)(n-2)} \sum_{j,k \in S, j < k} p_j p_k + \dots
\end{aligned}$$

Step 2: Taylor Expansion of $\phi_i(\mu)$

The empirical estimate expands as:

$$\begin{aligned}
\phi_i(\mu) &= \frac{m_i}{m} \left(1 - \left(1 - \frac{1}{l} \right)^m \right) \\
&= \frac{m_i}{m} \left(\frac{m}{l} - \frac{m(m-1)}{2l^2} + O\left(\frac{m^3}{l^3}\right) \right) \\
&= \frac{m_i}{l} - \frac{m_i(m-1)}{2l^2} + O\left(p_i \left(\sum_{j=1}^n p_j \right)^2 \right)
\end{aligned}$$

Step 3: Error Decomposition

The difference yields:

$$\begin{aligned}
\phi_i(\mu) - \phi_i(T) &= \frac{m_i(1-m_i)}{2l^2} + O\left(p_i \left(\sum_{j=1}^n p_j \right)^2 \right) - \frac{1}{n(n-1)(n-2)} \sum_{j,k \in S, j < k} p_j p_k - \dots \\
&= \frac{1}{2} p_i \left(\frac{1}{l} - p_i \right) + O\left(p_i \left(\sum_{j=1}^n p_j \right)^2 \right) - \frac{1}{n(n-1)(n-2)} \sum_{j,k \in S, j < k} p_j p_k - \dots
\end{aligned}$$

Dominant Error Terms:

- (1) **Weight Discrepancy:** $\left| \frac{m_i}{m} - \frac{w_i}{W} \right|$ from sampling imbalance, vanishing
- (2) **Binomial Approximation:** $\frac{m_i m}{2l^2}$ from second-order Taylor terms and symmetric expansion respectively
- (3) **Residual:** $O(\frac{m_i m^2}{l^3})$ from higher-order expansions

Weight Discrepancy refers to the mismatch between two weighting schemes in a given system. In this study, it specifically measures the difference between the empirical sampling ratio $\frac{m_i}{m}$ (the fraction of times element i appears in the sampled data) and the theoretical weight ratio $\frac{w_i}{W}$ (the expected fraction based on predefined weights w_i and their sum $W = \sum w_i$). The vanishing weight discrepancy between $\phi_i(\mu)$ and $\phi_i(T)$ is easy to check. $\square$

By applying Theorem 3.5 and using its proof techniques, we obtain the following useful results by breaking down to three cases based on $\frac{m}{l}$:

Assume n is large, and therefore so is m . Let $r = \frac{m}{l}$. We compare $\phi_i(\mu)$ and $\phi_i(T)$ in three different regimes:

Situation 1: $r = \frac{m}{l} \ll 1$, i.e. $\sum_{j=1}^n p_j \ll 1$.

- Since $\frac{1}{l}$ is small and m is not large, we approximate $\phi_i(\mu)$ as

$$\left(1 - \frac{1}{l}\right)^m \approx 1 - \frac{m}{l} \Rightarrow \phi_i(\mu) = m_i \times \frac{1}{m} \left(1 - \left(1 - \frac{1}{l}\right)^m\right) \approx \frac{m_i}{l} = p_i.$$

- For $\phi_i(T)$, we approximate the product term to first order to get

$$\phi_i(T) \approx p_i \cdot \left(1 - \frac{m - m_i}{2l}\right) \approx p_i \text{ (as } \frac{m}{l} \ll 1).$$

- **Conclusion:** The estimation is accurate.

Situation 2: $r = \frac{m}{l} \approx 1$, i.e. $\sum_{j=1}^n p_j \approx 1$.

- Then $\left(1 - \frac{1}{l}\right)^m \approx e^{-1}$ (since m is large and $\frac{m}{l} \approx 1 \Rightarrow m \approx l$), so

$$\phi_i(\mu) \approx \frac{m_i}{m} (1 - e^{-1}) \approx \frac{m_i}{l} (1 - e^{-1}) = p_i (1 - e^{-1}) \approx p_i \cdot 0.632.$$

- For $\phi_i(T)$, combinatorial effects in the product $\prod_{j \in S} (1 - p_j)$ become significant:

$$\phi_i(T) \approx p_i \cdot \left(1 - \frac{m - m_i}{2l}\right) \approx p_i \cdot \left(1 - \frac{1}{2} + \frac{m_i}{2l}\right) = p_i \cdot (0.5 + 0.5p_i).$$

- **Relation:** $\phi_i(T) \approx \frac{\phi_i(\mu)}{1 - e^{-1}} \left(0.5 + 0.5 \frac{\phi_i(\mu)}{1 - e^{-1}}\right)$.

- **Conclusion:** The approximation is quantitatively reasonable if p_i lies around $1 - \frac{2}{e} (\approx 0.26)$, but quantitative errors grow when p_i gets smaller or larger. This error in estimation can be reduced by using the above relation.

Situation 3: $r = \frac{m}{l} \gg 1$, i.e. $\sum_{j=1}^n p_j \gg 1$.

- Then $\left(1 - \frac{1}{l}\right)^m \approx e^{-\frac{m}{l}} \rightarrow 0$, so

$$\phi_i(\mu) \approx \frac{m_i}{m}, \text{ which can be much smaller than } p_i = \frac{m_i}{l} \text{ depending on the ratio } \frac{m}{l}.$$

- (Example: $m = 2l$. Then $\phi_i(\mu) \approx \left(1 - \frac{1}{e^2}\right) p_i$)
- For $\phi_i(T)$, the product $\prod_{j \in S} (1 - p_j)$ becomes very small due to congestion, so marginal contributions shrink:

$$\phi_i(T) \ll p_i. \text{ (See Remark 3.6 below)}$$

- **Conclusion:** Approximation tends to *overestimate* the true Shapley value.

Remark 3.6.

Analysis:

- Dominant contributions occur when $\prod_{j \in S} (1 - p_j)$ is not too small:
 - This happens when S includes only players j with $p_j \ll 1$ (so $1 - p_j \approx 1$).
 - If S includes even one j with $p_j \approx 1$, then $\prod_{j \in S} (1 - p_j) \approx 0$, killing the term.
- When $\sum p_i \gg 1$, most coalitions S include at least one high-probability player:
 - For such S , $\prod_{j \in S} (1 - p_j) \approx 0$, so these terms contribute negligibly.
 - The only surviving terms are those where S consists entirely of players with $p_j \ll 1$.
- Let $L = \{j : p_j \approx 1\}$ (the "high-probability players") and $H = \{j : p_j \ll 1\}$ (the "low-probability players"):

- The non-negligible terms are subsets $S \subseteq H \setminus \{i\}$ (i.e., avoiding high-probability players).
- For these S , $\prod_{j \in S} (1 - p_j) \approx 1$ (since $p_j \ll 1$ for $j \in S$).

Approximation:

- If $i \in H$ (i.e., $p_i \ll 1$):

$$\phi_i(T) \approx \sum_{S \subseteq H \setminus \{i\}} \frac{s!(n-1-s)!}{n!} p_i.$$

This simplifies to

$$\phi_i(T) \approx p_i \cdot \sum_{s=0}^{|H|-1} \frac{s!(n-1-s)!}{n!} \binom{|H|-1}{s}.$$

The combinatorial sum evaluates to $\frac{1}{n}$ (since the Shapley weights sum to 1 over all permutations), so

$$\phi_i(T) \approx \frac{p_i}{n}.$$

- If $i \in L$ (i.e., $p_i \approx 1$):
 - The marginal contribution of i is significant only when i is pivotal, which happens with probability ≈ 1 (since most coalitions without i fail).
 - Thus:

$$\phi_i(T) \approx 1 - (\text{probability that others already succeed without } i).$$

- But since $\sum p_i \gg 1$, the probability that others succeed without i is high, so $\phi_i(T)$ could be small. However, a precise approximation requires more care (e.g., Poisson-like analysis).

Estimation:

- For low-probability players ($p_i \ll 1$):

$$\phi_i(T) \approx \frac{p_i}{n}.$$

- For high-probability players ($p_i \approx 1$):
 - Their Shapley value depends on how critical they are to the coalition. If they are redundant (due to other high-probability players), $\phi_i(T)$ could be small.
 - A rough estimate might be $\phi_i(T) \approx \frac{1}{|L|}$ (where $|L|$ is the number of high-probability players), but this depends on the exact distribution of p_j 's.

Hence, if $p_i \ll 1$ (low-probability player), $\phi_i(T) \approx \frac{p_i}{n}$. If $p_i \approx 1$ (high-probability player), $\phi_i(T)$ depends on redundancy but is typically $O(1/|L|)$, where $|L|$ is the number of high-probability players. This aligns with intuition: low-probability players contribute marginally, while high-probability players compete for pivotal roles. These Shapley values need to be proportionally adjusted so that the total Shapley value is equal to $T(E)$.

The above three situations are summarized below:

Case	Regime	$\phi_i(\mu)$ vs $\phi_i(T)$	Notes
1	$m/l \ll 1$	$\phi_i(\mu) \approx \phi_i(T)$	Accurate approximation
2	$m/l \approx 1$	$\phi_i(\mu) \approx \phi_i(T)$ $\phi_i(T) \approx \frac{\phi_i(\mu)}{1-e^{-1}} \left(0.5 + 0.5 \frac{\phi_i(\mu)}{1-e^{-1}}\right)$	Accurate approximation when p_i is near $1 - \frac{2}{e}$, Otherwise apply the stated “Relation”
3	$m/l \gg 1$	$\phi_i(\mu) > \phi_i(T)$	Overestimation due to overlap, Resolution: Remark 3.6

Case 2: p_i 's are arbitrary real numbers. First we choose rational numbers $r_i, 1 \leq i \leq n$ satisfying $|q_i - p_i| < \delta$ where δ will be determined based on the need of the given application. With a straightforward analysis, the difference in the Shapley value $\phi_i(T)$ when each p_i is replaced by q_i is given by

$$|\tilde{\phi}_i(T) - \phi_i(T)| \leq \delta \cdot \frac{n+1}{2}.$$

Clearly, for any $\epsilon > 0$, it holds that $|\tilde{\phi}_i(T) - \phi_i(T)| < \epsilon$ whenever $\delta < \epsilon \times \frac{2}{n+1}$.

Another algorithm is designed for arbitrary real numbers p_i 's, and it is provided in Section 6.

4. EXAMPLES AND APPLICATION

In this section, we examine the validation and applications of the proposed approximation.

4.1. Examples.

Example 1: Symmetric case. Parameters: $n = 6, p_i = \frac{1}{2}$.

- $p_i = \frac{1}{2}$, so $l = 2, m_i = 1$
- $m = \sum_{i=1}^6 m_i = 6$

Approximation:

$$\phi_i(\mu) = \frac{1}{6} \left(1 - \left(1 - \frac{1}{2} \right)^6 \right) = \frac{1}{6} \left(1 - \frac{1}{64} \right) = \frac{63}{384} = \frac{21}{128}$$

Exact Value:

$$\phi_i(T) = \frac{1 - (1/2)^6}{6} = \frac{63}{384} = \frac{21}{128}$$

Another example: Suppose $n = 6, p_i = \frac{3}{5}$, so $l = 5, m_i = 3, m = \sum_{i=1}^6 m_i = 18$. The final answers are

$$\phi_i(T) = \frac{5187}{31250} \approx 0.1660, \quad \phi_i(\mu) = \frac{1248659262963}{7629394531250} \approx 0.1637.$$

Both results are exact and consistent with the theoretical framework. The fractional forms highlight the combinatorial nature of the calculations, which is essential for precision in network reliability analysis.

Example 2: General case (asymmetric, non-uniform). Consider three players with $p_1 = \frac{3}{6}, p_2 = \frac{2}{6}, p_3 = \frac{1}{6}$. Here, The LCD is $l = 6$. The sub-players' information is

- Player 1: $m_1 = 3$ sub-players
- Player 2: $m_2 = 2$ sub-players
- Player 3: $m_3 = 1$ sub-player

The total sub-players is $m = \sum_{i=1}^3 m_i = 6$.

The Shapley value approximation for player i is:

$$\phi_i(\mu) = \frac{m_i}{m} \left(1 - \left(1 - \frac{1}{l} \right)^m \right)$$

Substituting $l = 6$ and $m = 6$ gives

$$\begin{aligned} \phi_i(\mu) &= \frac{m_i}{6} \left(1 - \left(1 - \frac{1}{6} \right)^6 \right) = \frac{m_i}{6} \left(1 - \left(\frac{5}{6} \right)^6 \right) \\ &= \frac{m_i}{6} (1 - 0.3349) = \frac{m_i}{6} \times 0.6651. \end{aligned}$$

This results

- For Player 1 ($m_1 = 3$): $\phi_1(\mu) = \frac{3}{6} \times 0.6651 \approx 0.3326$
- For Player 2 ($m_2 = 2$): $\phi_2(\mu) = \frac{2}{6} \times 0.6651 \approx 0.2217$
- For Player 3 ($m_3 = 1$): $\phi_3(\mu) = \frac{1}{6} \times 0.6651 \approx 0.1108$

To compare, the exact Shapley values (computed numerically) are approximately:

$$\phi_1(T) \approx 0.3275, \phi_2(T) \approx 0.2475, \phi_3(T) \approx 0.1080.$$

The results are summarized in the below table.

Player i	m_i	Approximation $\phi_i(\mu)$	Exact $\phi_i(T)$	Relative Error
1	3	0.3326	0.3275	1.6%
2	2	0.2217	0.2475	10.4%
3	1	0.1108	0.1080	2.6%

4.2. Application. We will use the data from Example 2 to illustrate the use of the proposed approximation method.

1. Problem Setup

Network devices (players):

- Device 1: Web server ($p_1 = \frac{1}{2}$, high risk)
- Device 2: Database server ($p_2 = \frac{1}{3}$, medium risk)
- Device 3: IoT sensor ($p_3 = \frac{1}{6}$, low risk)

Sub-players: Represent vulnerabilities (e.g., $m_1 = 3$ means 3 known common vulnerabilities and exposures (CVEs) on Device 1)

LCD ($l = 6$): Base failure probability unit (e.g., $1/6 = 16.7\%$ risk per vulnerability)

2. Approximation Implementation

Apply our formula for $\phi_i(\mu)$,

$$\phi_i(\mu) = \frac{m_i}{6} \left(1 - \left(\frac{5}{6} \right)^6 \right) \approx \frac{m_i}{6} \times 0.6651$$

- Device 1 (Web Server): $\phi_1 \approx 0.3326$ (33.26% systemic risk impact)
- Device 2 (Database): $\phi_2 \approx 0.2217$ (22.17% impact)
- Device 3 (IoT): $\phi_3 \approx 0.1108$ (11.08% impact)

3-1. Security Actions

Priority patching: Device 1's 3 vulnerabilities contribute 50% more risk than Device 2's 2 vulnerabilities (0.3326 vs 0.2217), justifying immediate patching.

3-2. Access Control:

- Restrict Device 1's inbound connections (highest ϕ_i).
- Allow Device 3 limited access (lowest ϕ_i).

3-3. Verification: Comparing with the exact Shapley values that are also given in the previous table, $\phi_1 = 0.3275, \phi_2 = 0.2475, \phi_3 = 0.1080$, and using the formula below for error calculation,

$$\text{Error} = \left(\frac{\phi_i^{\text{approx}} - \phi_i^{\text{exact}}}{\phi_i^{\text{exact}}} \right) \times 100\%$$

we have

- Device 1: $\left(\frac{0.3326 - 0.3275}{0.3275} \right) \times 100\% \approx +1.56\%$
- Device 2: $\left(\frac{0.2217 - 0.2475}{0.2475} \right) \times 100\% \approx -10.4\%$
- Device 3: $\left(\frac{0.1108 - 0.1080}{0.1080} \right) \times 100\% \approx +2.6\%$

Acceptable for rapid risk assessment.

4. Dynamic response New CVE discovered on Device 2 $\rightarrow m_2$ increases from 2 to 3:

$$\phi_2^{\text{new}} = \frac{3}{6} \times 0.6651 \approx 0.3326 \quad (\text{now equal to Device 1})$$

Triggers automatic security policy update.

5. Benefits

- Efficiency: Computes in $O(1)$ per device vs. $O(2^n)$ exact calculation.
- Interpretability: Risk scores directly proportional to vulnerability counts (m_i).
- Adaptability: Live updates as m_i changes (e.g., new vulnerabilities detected).

6. Limitations & Mitigations

- Assumption: Vulnerabilities are independent.
Mitigation: Introduce correlation factors for shared exploits.
- Approximation error:
Mitigation: Use exact values for critical devices when feasible.

$$\phi_i(\mu) = \frac{m_i}{m} \left(1 - \left(1 - \frac{1}{l} \right)^m \right) \quad (\text{RACS approximation})$$

This demonstrates how our approximation enables real-time, risk-proportional security decisions for asymmetric networks.

Summary and Conclusion

- Intuition: The RACS construction treats each sub-player equally, which works well for symmetric or near-symmetric cases.
- Accuracy: The approximation is exact for symmetric players and accurate for large l .
- Intuition: The RACS construction treats each player as a group of sub-players, preserving marginal contributions.
- Limitation: For small l and highly asymmetric p_i , use the exact formula.
- Limitation: For highly asymmetric p_i , the exact formula is preferred.

The RACS-based approximation is theoretically justified and matches the exact value in symmetric cases. For asymmetric scenarios, it remains a robust heuristic. The approximation is reasonable but less precise for asymmetric cases. Use exact methods for critical applications.

5. COMPUTATIONAL COMPLEXITY COMPARISON: $\phi_i(\mu)$ VS $\phi_i(T)$

We analyze the two formulas for the Shapley value under probabilistic node failures in networks:

1. Formula for $\phi_i(T)$ (Exact Inclusion-Exclusion)

$$\phi_i(T) = \sum_{S \subseteq E \setminus \{i\}} \frac{s!(n-1-s)!}{n!} \cdot p_i \cdot \prod_{j \in S} (1 - p_j)$$

Complexity:

- Subset Enumeration: Requires iterating over all 2^{n-1} subsets of $E \setminus \{i\}$ (exponential in n).
- Per-Subset Operations:
 - Compute $\prod_{j \in S} (1 - p_j)$ (linear in s per subset)
 - Factorial terms $|S|!$ can be precomputed in $O(n)$ time
- Total Time: $O(2^{n-1} \cdot n)$ (intractable for $n \gg 20$)

Space: $O(1)$ (no additional storage beyond input)

2. Formula for $\phi_i(\mu)$ (RACS Approximation)

$$\phi_i(\mu) = m_i \times \frac{1}{m} \left(1 - \left(1 - \frac{1}{l} \right)^m \right)$$

Complexity:

- Precomputation:
 - Compute $m = \sum_{i=1}^n m_i$ (linear in n)
 - Compute $\left(1 - \frac{1}{l} \right)^m$ (using exponentiation by squaring: $O(\log m)$)
- Per-Node Operations:
 - Evaluate the closed-form expression (constant time per node)
- Total Time: $O(n + \log m)$ (efficient even for large n)

Space: $O(1)$ (only stores m , l , and m_i)

TABLE 2. Complexity Comparison of $\phi_i(T)$ and $\phi_i(\mu)$

Metric	$\phi_i(T)$ (Exact)	$\phi_i(\mu)$ (RACS)
Time Complexity	$O(2^{n-1} \cdot n)$	$O(n + \log m)$
Space Complexity	$O(1)$	$O(1)$
Feasibility for n	Intractable ($n > 20$)	Scalable ($n \geq 10^6$)
Accuracy	Exact	Approximate (error $\approx O(1/l)$)
Use Case	Small networks	Large-scale systems

$\phi_i(T)$ sums over all subsets, exploiting exact inclusion-exclusion, while $\phi_i(\mu)$ leverages probabilistic independence in the RACS model, reducing the problem to algebraic operations.

Example with $n = 6$.

- $\phi_i(T)$: Sums $2^5 = 32$ subsets
- $\phi_i(\mu)$: Computes $m = 18$, then evaluates one exponential term

For $n = 100$:

- $\phi_i(T)$: $2^{99} \approx 6.3 \times 10^{29}$ subsets (infeasible)
- $\phi_i(\mu)$: Computes m (e.g., $m = 300$) and $\left(1 - \frac{1}{l} \right)^m$ in microseconds

Hence, it is suggested to use $\phi_i(\mu)$ for large networks where exact computation is prohibitive, and $\phi_i(T)$ for small n when precision is critical. The RACS model's efficiency comes at a minor cost in accuracy, controlled by l (larger l improves precision).

3. Use Möbius inversion (for general RACS)

If X is not independent, but we know its Möbius mass function $m(F) = \mathbb{P}(X = F)$, then:

$$\mathbb{P}(i \in X \text{ and } X \cap S = \emptyset) = \sum_{\substack{F \subseteq E \setminus S \\ i \in F}} m(F).$$

- **Complexity:** Still $O(2^n)$ in the worst case, but:
 - If $m(F)$ is sparse (many $m(F) = 0$), we can skip computations.

- If X is k -sparse (at most k elements), we only need to sum over $\binom{n}{k}$ subsets.

4. Approximate using belief functions

If X is a belief function (containment function of RACS), then:

$$\mathbb{P}(i \in X \text{ and } X \cap S = \emptyset) = \text{Bel}(\{i\} \cup S^c) - \text{Bel}(S^c),$$

where $\text{Bel}(A) = \mathbb{P}(X \subseteq A)$.

- If $\text{Bel}(A)$ can be computed efficiently (e.g., via Dempster-Shafer theory), this helps.

5. Monte Carlo sampling (for large n)

If exact computation is too expensive:

$$\mathbb{P}(i \in X \text{ and } X \cap S = \emptyset) \approx \frac{\text{Count}(i \in X \text{ and } X \cap S = \emptyset)}{\text{Total samples}}.$$

- **Complexity:** $O(\text{number of samples})$, which is tractable.

6. Assume a low-order interaction model

If X has limited dependencies (e.g., pairwise interactions), we can use:

- Graphical models (Markov random fields).
- Log-linear models (for small clique sizes).

Summary Table: Best approaches for estimating $\mathbb{P}(i \in X \text{ and } X \cap S = \emptyset)$

Case	Method	Complexity
Independent X	Direct product formula	$O(S)$
Sparse X	Möbius inversion on small subsets	$O(\binom{n}{k})$
Belief function	Use $\text{Bel}(A)$	Depends on Bel
Large n	Monte Carlo sampling	$O(\text{samples})$
Low-order interactions	Graphical models	Polynomial in n

5.1. Recommendation.

- If X has independent elements: Use $p_i \prod_{j \in S} (1 - p_j)$. (Fastest)
- If X is sparse: Use Möbius inversion with pruning. (Exact but expensive)
- If n is very large: Use Monte Carlo sampling. (Approximate but scalable)
- If X has known structure: Exploit it (e.g., belief functions, graphical models).

6. SECOND ALGORITHM

Consider n probabilities $p_1, p_2, \dots, p_n$. We now assume $p_1 \leq p_2 \leq \dots \leq p_n$ (sorting requires $n \log(n)$ operations). Note that we do not require p_i 's to be rational numbers in this section.

6.1. The algorithm. The algorithm proceeds as:

- (1) Find $p_{\min} = \min\{p_1, \dots, p_n\}$ (actually the first number in the sequence when the sequence is sorted).
- (2) Compute differences $\Delta_i = p_i - p_{\min}$ for all i .
- (3) Remove zeros (i.e., discard $p_{\min}$) and keep $\Delta_i \neq 0$.
- (4) Repeat with remaining Δ_i until empty.

Theorem 6.1. *The process terminates after at most n iterations for any $p_i \in \mathbb{R}$.*

Proof. At each iteration:

- The minimum $p_{\min}$ is permanently removed (as $\Delta = 0$ for it).
- At least one number is eliminated per iteration.
- Worst case: n iterations (when removing one number per step).

Thus, it terminates in $O(n)$ steps for any real inputs. □

Example: $\{1, \sqrt{2}\}$

Iteration 1: $p_{\min} = 1, \quad \Delta = \sqrt{2} - 1, \quad \text{Keep } \sqrt{2} - 1.$

Iteration 2: $p_{\min} = \sqrt{2} - 1, \quad \Delta = 0, \quad \text{Terminate.}$

Complexity Analysis

- **Per Iteration:**
 - Find minimum: $O(k)$ for k remaining numbers.
 - Subtract and drop zeros: $O(k)$.
- **Total Operations:**

$$\sum_{k=n}^1 O(k) = O(n^2).$$

Time complexity: $O(n^2)$ for any $p_i \in \mathbb{R}$, with guaranteed termination.

In our study, p_i 's as probabilities are positive, lying between 0 and 1.

Weight breakdown

Total weight: $\sum_{i=1}^{n_1} p_i^{(1)}$, where $n_1 = n, p_i^{(1)} = p_i$.

First iteration: Let $r_1 = \min\{p_1^{(1)}, p_2^{(1)}, \dots, p_{n_1}^{(1)}\}$. The weight is $n_1 \times r_1$. Subtracting r_1 from each number in the current sequence gives non-zero numbers $p_1^{(2)}, p_2^{(2)}, \dots, p_{n_2}^{(2)}$ ($n_2 < n_1$).

Second iteration: Put $r_2 = \min\{p_1^{(2)}, p_2^{(2)}, \dots, p_{n_2}^{(2)}\}$. The weight is $n_2 \times r_2$. Subtracting r_2 from each number in the current sequence gives non-zero numbers $p_1^{(3)}, p_2^{(3)}, \dots, p_{n_3}^{(3)}$ ($n_3 < n_2$).

Generally, in the k th iteration, the sequence of non-zero numbers is $p_1^{(k)}, p_2^{(k)}, \dots, p_{n_k}^{(k)}$ ($n_k < n_{k-1}$). Set $r_k = \min\{p_1^{(k)}, p_2^{(k)}, \dots, p_{n_k}^{(k)}\}$. The k th iteration weight is $n_k \times r_k$. Subtracting r_k from each number in the current sequence gives non-zero numbers $p_1^{(k+1)}, p_2^{(k+1)}, \dots, p_{n_{k+1}}^{(k+1)}$ ($n_{k+1} < n_k$).

This process iterates at most n times (exactly n times if all the p_i 's are distinct), terminating when the new sequence becomes no non-zero numbers.

In the k th iteration step, it results a homogeneous case $t_1^{(k)}, t_2^{(k)}, \dots, t_{n_k}^{(k)}$, where $t_j^{(k)} = r_k, 1 \leq j \leq n_k$, and the corresponding (accurate) Shapley value is given by the closed form

$$(6.1) \quad \phi_i^{(k)}(T) = \frac{r_k}{n_k} \sum_{s=0}^{n_k-1} \binom{n_k-1}{s} \frac{(1-r_k)^s}{\binom{k-1}{s}} = \frac{r_k}{n_k} \sum_{s=0}^{n_k-1} (1-r_k)^s = \frac{r_k}{n_k} \frac{1 - (1-r_k)^{n_k}}{1 - (1-r_k)} = \frac{1}{n_k} [1 - (1-r_k)^{n_k}].$$

Finally, the approximation of $\phi_i(T)$ is given by

$$(6.2) \quad \phi_i(T) \approx \sum_{k=1}^{m_i} n_k \times r_k \times \phi_i^{(k)}(T)$$

where $m_i = \#\{j | p_j \leq p_i, 1 \leq j \leq n\}$.

This approximation degenerates to the homogeneous formula (3.4) if all the p_i 's are the same.

In the worst-case iteration, the algorithm runs for at most n iterations (when one player is removed per iteration). For each iteration, the work is dominated by $O(n_k^2)$ operations for finding minima, differences, and filtering. The total number of operations is

$$\sum_{k=1}^n O(n_k) \leq O(n^2), \quad \text{since } \sum_{k=1}^n n_k \leq \sum_{k=1}^n (n - k + 1) = \frac{n(n+1)}{2}.$$

The computational complexity of this whole process will be analyzed next.

6.2. Accuracy Analysis of the Approximation Algorithm vs. Exact Shapley Values.

We analyze the approximation quality both theoretically and through examples, comparing to the exact formula:

$$\phi_i(T) = \sum_{S \subseteq E \setminus \{i\}} \frac{s!(n-1-s)!}{n!} \cdot p_i \cdot \prod_{j \in S} (1-p_j).$$

Theoretical Accuracy

Key Observations:

(1) **Exactness for Homogeneous Cases:**

- If all $p_i = p$, the approximation matches the exact Shapley value:

$$\phi_i = \frac{1 - (1-p)^n}{n}$$

(2) **Additive Error for General Cases:**

- The approximation error stems from:
 - Ignoring coalition-specific interactions (exact formula considers 2^{n-1} subsets)
 - Linearization of higher-order terms

(3) **Error Bound:**

- For player i , the worst-case relative error:

$$\frac{|\phi_i^{\text{approx}} - \phi_i^{\text{exact}}|}{\phi_i^{\text{exact}}} \leq \frac{(1-p_i)^{n-1}}{1 - \prod_{j \neq i} (1-p_j)}$$

Example-Based Analysis

Let

$$p_1 = 0.2, \quad p_2 = 0.5, \quad p_3 = 0.7, \quad p_4 = 0.3, \quad p_5 = 0.1, \quad p_6 = 0.9, \quad p_7 = 0.4.$$

The following table shows a direct comparison between the approximation method and the Exact Shapley value formula.

TABLE 3. Comparison of Exact vs Approximate Shapley Values

Player i	Exact ϕ_i	Approx ϕ_i	Relative Error (%) [*]	Dominant Error Source
1 ($p_1 = 0.2$)	0.062074	0.052189	-15.93%	Underestimates small coalitions
2 ($p_2 = 0.5$)	0.176342	0.174526	-1.03%	Robust for mid-range p_i
3 ($p_3 = 0.7$)	0.211763	0.201563	-4.82%	Misses synergies with other p_j ($j \neq i$)
4 ($p_4 = 0.3$)	0.096693	0.099119	+2.51%	Robust for mid-range p_i
5 ($p_5 = 0.1$)	0.030815	0.008894	-71.13%	Severe underestimation for very small p_i
6 ($p_6 = 0.9$)	0.295382	0.273576	-7.38%	Underestimates dominant players
7 ($p_7 = 0.4$)	0.122157	0.140106	+14.70%	Overestimates for mid-range p_i

^{*} Relative Error % calculated as $(\text{Approx} - \text{Exact}) / \text{Exact} \times 100$

Further improvements of the Second Algorithm and the previously established algorithm in Section 3 will be discussed shortly, under “Extreme situations: Mathematical intuition with examples.”

Theoretical Justification. The approximation error for high p_i can be quantified by analyzing the Taylor expansion of the method.

Approximation Form The approximate Shapley value linearizes contributions as:

$$(6.3) \quad \phi_i^{\text{approx}} \approx p_i \cdot \left(1 - \sum_{j \neq i} \frac{p_j}{2} \right) + \mathcal{O}(p_i p_j^2),$$

The approximate Shapley value for player i linearizes contributions (refer to (6.4)).

The sum in question involves terms like $\prod_{j \in S} (1 - p_j)$. Using the inclusion-exclusion principle, this expands to a symmetric sum:

$$(6.4) \quad \prod_{j \in S} (1 - p_j) = \sum_{k=0}^{|S|} (-1)^k S_k = 1 - \sum_{j \in S} p_j + \sum_{\substack{j, k \in S \\ j < k}} p_j p_k - \cdots + (-1)^{|S|} \prod_{j \in S} p_j,$$

where for variables $\{p_j\}_{j \in S}$, the k -th elementary symmetric sum S_k is defined as:

$$S_k = \sum_{\substack{T \subseteq S \\ |T|=k}} \prod_{j \in T} p_j.$$

By plugging in (6.4) in (3.6) gives

$$(6.5) \quad \phi_i^{\text{approx}}(T) \approx p_i \cdot \left(1 - \sum_{j \neq i} \frac{p_j}{2} \right) + \mathcal{O}(p_i \bar{p}_{-i}^2),$$

where $\bar{p}_{-i} = \frac{1}{n-1} \sum_{j \neq i} p_j$ is the average probability of other players. The error scales with p_i multiplied by quadratic terms in others' probabilities, mathematically coherent since $\bar{p}_{-i}$ aggregates all $j \neq i$, i.e. $\mathcal{O}(p_i \bar{p}_{-i}^2) \equiv$ Error from ignoring 2nd-order interactions $\approx$ Terms like $p_i p_j p_k$ where $j, k \neq i$.

Extreme situations: Mathematical intuition with examples

The approximation error ε_i for player i follows:

$$\varepsilon_i = \phi_i^{\text{approx}} - \phi_i \approx \underbrace{-p_i(1 - p_i)^{n-1}}_{\text{Isolation term}} + \underbrace{\sum_{j \neq i} \frac{p_i p_j}{2}}_{\text{Pairwise overcorrection}} - \cdots$$

For $P = [0.05, 0.95, 0.95]$, the above second algorithm gives the following result (Table 4), with a comparison to the exact Shapley value.

TABLE 4. Comparison of Exact and Approximate Shapley Values (using Second Algorithm)

Agent	p_i	Exact $\phi_i(T)$	Approx $\phi_i(T)$	Relative Error (%)
1	0.05	0.0176	0.0013	-92.6
2	0.95	0.4912	0.4993	+1.6
3	0.95	0.4912	0.4993	+1.6

Note: Relative error calculated as $\frac{\phi_i(\mu) - \phi_i(T)}{\phi_i(T)} \times 100\%$

Results from the method developed previously in Section 3.5, i.e. $\phi_i(\mu) = m_i/m(1 - (1 - 1/l)^m)$, are provided in Table 5.

TABLE 5. Comparison of Shapley Value Calculation Methods (using Method in Section 3.5)

Agent	p_i	Exact $\phi_i(T)$	$\phi_i(\mu)$	Relative Error (%)
1	0.05	0.0176	0.0222	+26.1
2	0.95	0.4912	0.4213	-14.2
3	0.95	0.4912	0.4213	-14.2

If the method given in Section 3.5, Remark 3.6, is used, the comparison result is further improved, as shown in Table 6.

6.2.1. *Exact Form.* Whereas the exact Shapley value includes higher-order interaction terms:

$$(6.6) \quad \phi_i^{\text{exact}} = p_i \cdot \left(1 - \sum_{j \neq i} \frac{p_j}{2} + \sum_{\substack{j < k \\ j, k \neq i}} \frac{p_j p_k}{3} - \cdots \right)$$

TABLE 6. Normalized Shapley Value Comparison (using Remark 3.6)

Agent	p_i	Exact $\phi_i(T)$	Approx. $\phi_i(T)$	Relative Error (%)
1	0.05	0.0176	0.0164	-6.8
2	0.95	0.4912	0.4918	+0.1
3	0.95	0.4912	0.4918	+0.1

6.2.2. *Error Analysis.* For large p_i , the missing terms become significant:

$$\begin{aligned} \text{Error} &= \phi_i^{\text{exact}} - \phi_i^{\text{approx}} \\ &\approx p_i \left(\sum_{\substack{j < k \\ j, k \neq i}} \frac{p_j p_k}{3} - \sum_{\substack{j < k < l \\ j, k, l \neq i}} \frac{p_j p_k p_l}{4} + \dots \right) \end{aligned}$$

Dominant Terms. When $p_i \rightarrow 1$ and other $p_j \ll 1$, the leading error term is:

$$(6.7) \quad \text{Error} \approx \frac{p_i}{3} \sum_{\substack{j < k \\ j, k \neq i}} p_j p_k + \mathcal{O}(p_i p_j^3)$$

- The approximation captures only first-order interactions ($\mathcal{O}(p_j)$)
- Exact computation requires all higher-order terms ($\mathcal{O}(p_j^2), \mathcal{O}(p_j^3)$, etc.)

Table 7 compares the approximation method with the exact Shapley formula.

TABLE 7. Method Comparison

Metric	Exact Formula	Approximation
Computational Cost	$\mathcal{O}(2^n)$	$\mathcal{O}(n^2)$
Accuracy	Exact	Error grows with n
Best For	Small n (≤ 15)	Large n with clustered p_i
Worst For	Large n	Extremal p_i or synergistic games

The theoretical limitations focus on the non-additive interactions. Further improvements include:

(1) **Hybrid Approach:**

Use exact computation for extremal p_i and approximate the rest

(2) **Error-Corrected Approximation:**

$$\phi_i^{\text{corrected}} = \phi_i^{\text{approx}} + \sum_{j \neq i} \frac{p_i p_j}{n(n-1)}$$

7. APPENDIX I. DERIVATION METHODS FOR SHAPLEY VALUE FORMULA: HISTORICAL REMARKS

The Shapley value is a solution concept in cooperative game theory that assigns a fair distribution of payoffs to players based on their marginal contributions. Over the years, researchers have developed multiple methods to derive or calculate the Shapley value, each offering unique insights. Below is a summary of the key calculation/derivation methods found in the literature. For a deep dive into Shapley original method, these are some essential references for the original approach (Shapley, 1953a, Maschler, Solan, Zamir, 2013, Osborne, Rubinstein, 1994, Rota, 1964 and 1988, Winter, 2002).

In this section, the player set will be denoted by N and the characteristics function is written as v .

7.1. Original axiomatic derivation (Shapley, 1953). Shapley derived his value using four axioms:

Efficiency: The total payoff is fully distributed.

Symmetry: Players with identical marginal contributions receive equal payoffs.

Dummy Player (Carrier): A player who contributes nothing to any coalition gets nothing.

Additivity: The Shapley value of a sum of games is the sum of the Shapley values of the individual games.

The unique solution satisfying these axioms is:

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{s!(n-s-1)!}{n!} (v(S \cup \{i\}) - v(S))$$

where N is the set of all n players, v is the characteristic function, and s is the size of coalition S .

7.2. Permutation-based approach. The Shapley value can be interpreted as the average marginal contribution of a player across all possible orderings of players:

$$\phi_i(v) = \frac{1}{n!} \sum_{\pi \in \Pi(N)} [v(P_\pi^i \cup \{i\}) - v(P_\pi^i)]$$

where $\Pi(N)$ is the set of all permutations of N , and P_π^i is the set of players preceding i in permutation π .

7.3. Potential function method. (Hart & Mas-Colell, 1989) This method introduces a potential function P such that:

$$\phi_i(v) = P(N, v) - P(N \setminus \{i\}, v)$$

The potential function is recursively defined, ensuring consistency with the Shapley value.

7.4. Harsanyi dividends. (Harsanyi, 1963) The Shapley value can be expressed in terms of dividends for each coalition:

$$\phi_i(v) = \sum_{\substack{S \subseteq N \\ i \in S}} \frac{\Delta_S(v)}{|S|}$$

where $\Delta_S(v)$ is the Harsanyi dividend of coalition S , defined recursively.

7.5. Multi-linear extension method. (Owen, 1972) This approach uses a continuous extension of the game:

Represent the game v as a multilinear function f over $[0, 1]^n$.

The Shapley value is obtained by integrating the partial derivative:

$$\phi_i(v) = \int_0^1 \frac{\partial f}{\partial x_i}(t, t, \dots, t) dt$$

7.6. Probabilistic interpretation. (Weber, 1988) The Shapley value can be seen as the expected marginal contribution when players join a coalition in a random order, where each ordering is equally likely.

7.7. Linear programming formulation. (Charnes et al., 1988) The Shapley value can be derived as the solution to a linear program minimizing a certain imbalance criterion while satisfying efficiency and symmetry.

7.8. Random order value equivalence. (Roth, 1988) The Shapley value is the unique random-order value that satisfies symmetry and efficiency.

7.9. Aumann-Shapley pricing. (Aumann & Shapley, 1974) For non-atomic games (continuum of players), the Shapley value is derived via differentiation of a path integral.

7.10. Approximation and sampling methods. For large games, exact computation is intractable, so methods like:

Monte Carlo sampling (Castro et al., 2009)

Structured sampling (e.g., stratified, antithetic)

Neural network approximations (e.g., DeepSHAP, Lundberg & Lee, 2017)

Finally, we supplement Shapley’s original proof by deriving the Shapley value formula using a rigorous approach based on definite integrals and combinatorial analysis. This method explicitly reveals the role of the Binomial Theorem and the Beta function in the proof, addressing an omitted part of Shapley’s work.

8. APPENDIX II. AN INTEGRAL AND COMBINATORIAL APPROACH TO DERIVE THE SHAPLEY VALUE FORMULA

Let E be a finite set of players, and $v : 2^E \rightarrow \mathbb{R}^+$ be a real-valued function defined on the power set 2^E of E , satisfying the following boundary condition and super-additivity:

- (i) $v(\emptyset) = 0$, and
- (ii) $v(S \cup T) \geq v(S) + v(T)$ whenever $S \cap T = \emptyset$.

For any $R \subseteq E, R \neq \emptyset$, Shapley defined the unanimity game

$$v_R(S) = \begin{cases} 1 & \text{if } S \supseteq R \\ 0 & \text{otherwise.} \end{cases}$$

The Shapley value of player i with respect to such an unanimity game v_R is given by (Shapley, 1953, Lemma 2)

$$\phi_i(v_R) = \begin{cases} \frac{1}{r} & \text{if } i \in R \\ 0 & \text{if } i \in E \setminus R. \end{cases}$$

In the above formula, lower case letter r stands for the number of players in R . This notation will be also used with other letters, as needed.

Analogous to the usual expansion of periodic functions into Fourier series, i.e., linear combinations of trigonometric functions, Shapley discovered that for real-valued set functions, i.e., Characteristic functions of games, they can be expanded into linear combinations of unanimity games (Shapley, 1953, Lemma 3), i.e.,

$$v(S) = \sum_{\emptyset \neq R \subseteq E} c_R(v) v_R(S), S \subseteq E,$$

where

$$c_R(v) = \sum_{T \subseteq R} (-1)^{r-t} v(T) \tag{8}$$

is the Möbius coefficient with respect to the unanimity game R , a name that Shapley did not explicitly call in his 1953 paper. Möbius transform defined on locally finite posets was extensively and intensively studied by Rota (Rota, 1964). In particular, Möbius transform is widely applied to Boolean lattices such as the power set of a given set with inclusion relation between subsets (Grabisch, Labreuche 2005).

It follows from Lemmas 2 and 3 that Shapley obtained his initial representation of Shapley value for player i :

$$\phi_i(v) = \sum_{i \in R \subseteq E} \frac{c_R(v)}{r}, \quad i \in E. \quad (10)$$

Then he claimed that “Inserting (8) and simplifying the result gives us”

$$\phi_i(v) = \sum_{i \in S \subseteq E} \frac{(s-1)!(n-s)!}{n!} v(S) - \sum_{i \notin S \subseteq E} \frac{s!(n-s-1)!}{n!} v(S), \quad i \in E. \quad (11)$$

The two sums are often put together into one as follows:

$$\sum_{i \notin S \subseteq E} \frac{s!(n-s-1)!}{n!} [v(S \cup i) - v(S)], \quad i \in E. \quad (12)$$

However, the omitted simplification can be of great interest for reasons like mathematical rigor, alternating insights, or just curiosity. In the current literature, the derivation of Formula (12) is often through permutations or combinations, $s!$ permutations before the $(s+1)th$ position and $(n-s-1)!$ permutations after this position. Other different methods will be stated later in the Historical Remarks. A direct counting from Formula (10) seems missing. As such, one purpose of this article is to provide a rigorous integral and combinatorial approach to establish Formula (12).

8.1. An integral and combinatorial approach to Shapley value calculation. In this section, for Shapley’s claim, “Inserting (8) and simplifying the result gives us Formula (11)”, we will perform the calculation and simplification through a rigorous definite integral and the Binomial Theorem.

First by plugging Formula (8) into Formula (10), we get

$$\begin{aligned} \phi_i(v) &= \sum_{i \in R \subseteq E} \frac{1}{r} \sum_{T \subseteq R} (-1)^{r-t} v(T) \\ &= \sum_{i \in R \subseteq E} \frac{1}{r} \left[\sum_{i \in T \subseteq R} (-1)^{r-t} v(T) + \sum_{i \notin T \subseteq R} (-1)^{r-t} v(T) \right] \end{aligned}$$

Let $R' = R \setminus \{i\}$ and $T' = T \setminus \{i\}$. Accordingly, it holds that $r' = r - 1$ and $t' = t - 1$. Then we have

$$\phi_i(v) = \sum_{R' \subseteq E \setminus \{i\}} \frac{1}{r' + 1} \sum_{T' \subseteq R'} (-1)^{(r'+1)-(t'+1)} v(T' \cup \{i\}) + \sum_{R' \subseteq E \setminus \{i\}} \frac{1}{r' + 1} \sum_{T' \subseteq R'} (-1)^{(r'+1)-t'} v(T')$$

By changing T' to S in the first sum and T to S in the second sum, we get

$$\begin{aligned} \phi_i(v) &= \sum_{R' \subseteq E \setminus \{i\}} \frac{1}{r' + 1} \sum_{S \subseteq R'} (-1)^{(r'+1)-(s+1)} v(S \cup \{i\}) + \sum_{R' \subseteq E \setminus \{i\}} \frac{1}{r' + 1} \sum_{S \subseteq R'} (-1)^{(r'+1)-s} v(S) \\ &= \sum_{R' \subseteq E \setminus \{i\}} \frac{1}{r' + 1} \sum_{S \subseteq R'} (-1)^{r'-s} [v(S \cup \{i\}) - v(S)] \end{aligned}$$

Next, interchanging the order of the two sums, in the new inner sum first adding the terms that correspond to the coalitions of a same size k to obtain partial sums, and then putting all these partial sums together, we have

$$\begin{aligned}
 \phi_i(v) &= \sum_{S \subseteq E \setminus \{i\}} \sum_{S \subseteq R' \subseteq E \setminus \{i\}} \frac{1}{r'+1} (-1)^{r'-s} [v(S \cup \{i\}) - v(S)] \\
 &= \sum_{S \subseteq E \setminus \{i\}} \sum_{k=s}^{n-1} \sum_{S \subseteq R' \subseteq E \setminus \{i\}, r'=k} \frac{1}{r'+1} (-1)^{r'-s} [v(S \cup \{i\}) - v(S)] \\
 (8.1) \quad &= \sum_{S \subseteq E \setminus \{i\}} \sum_{k=s}^{n-1} \binom{n-1-s}{k-s} \frac{1}{k+1} (-1)^{k-s} [v(S \cup \{i\}) - v(S)].
 \end{aligned}$$

Let $l = k - s$, then

$$(8.2) \quad \sum_{k=s}^{n-1} \binom{n-1-s}{k-s} \frac{1}{k+1} (-1)^{k-s} = \sum_{l=0}^{n-1-s} \binom{n-1-s}{l} \frac{1}{l+s+1} (-1)^l.$$

Denote this combinatorial expression by $S_{n,s}$.

We next evaluate $S_{n,s}$ using a definite integral representation and the Binomial Theorem.

Substituting $\frac{1}{l+s+1} = \int_0^1 x^{l+s} dx$ in $S_{n,s}$ gives

$$S_{n,s} = \sum_{l=0}^{n-1-s} \binom{n-1-s}{l} (-1)^l \int_0^1 x^{l+s} dx.$$

Interchanging the sum and the integral gives

$$S_{n,s} = \int_0^1 x^s \sum_{l=0}^{n-1-s} \binom{n-1-s}{l} (-1)^l x^l dx.$$

The inner sum is the binomial expansion of $(1-x)^{n-1-s}$. Thus,

$$(8.3) \quad S_{n,s} = \int_0^1 x^s (1-x)^{n-1-s} dx,$$

which is the value of the standard Beta function, i.e., $S_{n,s} = \text{Beta}(s+1, n-s)$.

Now by using the Beta-Gamma relation $\text{Beta}(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$, we obtain

$$\text{Beta}(s+1, n-s) = \frac{\Gamma(s+1)\Gamma(n-s)}{\Gamma(n+1)}.$$

Since $\Gamma(s+1) = s!$ and $\Gamma(n+1) = n!$, we get $\text{Beta}(s+1, n-s) = \frac{s!(n-1-s)!}{n!}$.

Therefore, the following is established:

$$\sum_{l=0}^{n-1-s} \binom{n-1-s}{l} \frac{(-1)^l}{l+s+1} = \frac{s!(n-1-s)!}{n!}.$$

This derives Shapley's Formula (12) using the definite integral and combinatorial approach, and hence provides a supplementary calculation to the omitted part in Shapley's original paper.

9. CONCLUSION

The Shapley value has been derived and calculated through multiple methods, such as combinatorial (permutations), algebraic (axiomatic), analytical (multilinear extension), and probabilistic, each offering distinct insights. The choice of method depends on the application (e.g., cooperative games, explainable AI, or cost allocation).

Beyond its classical applications, the Shapley value can also be computed for non-additive measures such as Dempster-Shafer belief functions and Choquet capacities. As a fair solution for distributing payoffs among coalition members, it has become a cornerstone of cooperative game theory and its applied domains. The rapid development of machine learning (ML) has further established the Shapley value as a powerful tool for quantifying feature contributions to ML model behavior and predictions (Rozemberczki et al., 2022).

Practically, this game-theoretic method has been successfully adapted to evaluate the value of devices in computer networks. More broadly, its applications extend to socio-economic and health network analyses, demonstrating its versatility beyond theoretical contexts.

However, the Shapley value faces an inherent challenge: its exponential computational complexity. This issue generally remains unresolved due to the fundamental structure of the Shapley value, though randomized approximation algorithms offer partial solutions.

In this study, we address a special case of random network created via inhomogeneous Bernoulli trials, by developing deterministic algorithms that compute the Shapley value in linear or quadratic time with rigorous discussions of their accuracy. Future work should extend this approach to the more general setting where players join coalitions dependently.

Additionally, we supplement Shapley's original proof by deriving the Shapley's value formula using a rigorous approach based on definite integrals and combinatorial analysis. This method explicitly highlights the roles of the Binomial Theorem and the Beta function, clarifying an omitted aspect of Shapley's work. We hope this contribution provides deeper insights for future research on the Shapley value.

REFERENCES

- Aumann, R. J., and L. S. Shapley. 1974. *Values of Non-Atomic Games*. Princeton: Princeton University Press.
- Castro, J., D. Gomez, and J. Tejada. 2009. Polynomial Calculation of the Shapley Value Based on Sampling. *Computers & Operations Research* 36 (5): 1726–30.
- Charnes, A., W. W. Cooper, B. Golany, and L. Seiford. 1988. Foundations of Data Envelopment Analysis for Pareto-Koopmans Efficient Empirical Production Functions. *Journal of Econometrics* 38 (1–2): 91–107.
- Choquet, G. 1954. Theory of Capacities. *Annales de l'Institut Fourier* 5: 131–295. <https://doi.org/10.5802/aif>.
- Daniels, H. E. 1954. Saddlepoint Approximations in Statistics. *Annals of Mathematical Statistics* 25 (4): 631–50.
- Dughmi, S. 2017. Algorithmic Applications of Approximate Caratheodory Theorem. In *Advances in Neural Information Processing Systems* 30, 3310–18.
- Feller, W. 1968. *An Introduction to Probability Theory and Its Applications*. 3rd ed. Vol. 1. New York: Wiley.
- Grabisch, M. 1997. k-Order Additive Discrete Fuzzy Measures and Their Representation. *Fuzzy Sets and Systems* 92 (2): 167–89.
- Grabisch, M., and C. Labreuche. 2005. Bi-Capacities Part I: Definition, Möbius Transform and Interaction. *Fuzzy Sets and Systems* 151 (2): 211–36.
- Grimmett, G., and D. Stirzaker. 2020. *Probability and Random Processes*. 4th ed. Oxford: Oxford University Press.
- Harsanyi, J. C. 1963. A Simplified Bargaining Model for the n-Person Cooperative Game. *International Economic Review* 4 (2): 194–220.
- Hart, S., and A. Mas-Colell. 1989. Potential, Value, and Consistency. *Econometrica* 57 (3): 589–614.
- Kalai, E., and D. Samet. 1987. On Weighted Shapley Values. *International Journal of Game Theory* 16 (3): 205–22. <https://doi.org/10.1007/BF01756292>.
- Kerman, J., and M. Wootters. 2011. Laplace Approximations for Efficient Equilibria in Large Games. *Proceedings of the 12th ACM Conference on Electronic Commerce*, 319–26. <https://doi.org/10.1145/1993574.1993621>.
- Lasry, J.-M., and P.-L. Lions. 2007. Mean Field Games. *Japanese Journal of Mathematics* 2 (1): 229–60.
- Lundberg, S. M., and S.-I. Lee. 2017. A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems* 30: 4765–74.
- Maschler, M., E. Solan, and S. Zamir. 2013. *Game Theory*. Cambridge: Cambridge University Press.
- Matheron, G. 1975. *Random Sets and Integral Geometry*. New York: John Wiley & Sons.
- Murofushi, T., and S. Soneda. 1993. Techniques for Reading Fuzzy Measures (III): Interaction Index. In *Proceedings of the 9th Fuzzy Systems Symposium*, Sapporo, Japan, 693–96.
- Neyman, A. 2002. Values of Games with Infinitely Many Players. In *Handbook of Game Theory*, vol. 3, 2121–67. [https://doi.org/10.1016/S1574-0005\(02\)03012-4](https://doi.org/10.1016/S1574-0005(02)03012-4).
- Osborne, M. J., and A. Rubinstein. 1994. *A Course in Game Theory*. Cambridge: MIT Press.
- Owen, G. 1968. A Note on the Shapley Value. *Management Science* 14 (11): 731–32.
- Owen, G. 1972. Multilinear Extensions of Games. *Management Science* 18 (5): P 64 to P79.
- Rota, G.-C. 1964. On the Foundations of Combinatorial Theory I: Theory of Möbius Functions. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete* 2 (4): 340–68.
- Rota, G.-C. 1988. The Many Lives of Lattice Theory. *Notices of the American Mathematical Society* 35 (2): 147–53.
- Roth, A. E., ed. 1988. *The Shapley Value: Essays in Honor of Lloyd S. Shapley*. Cambridge: Cambridge University Press.
- Rozemberczki, B., et al. 2022. The Shapley Value in Machine Learning. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence*, 5632–41. IJCAI.
- Shafer, G. 1976. *A Mathematical Theory of Evidence*. Princeton: Princeton University Press.
- Shapley, L. S. 1953a. A Value for n-Person Games. PhD diss., Princeton University.
- Shapley, L. S. 1953b. A Value for n-Person Games. In *Contributions to the Theory of Games II*, edited by H. W. Kuhn and A. W. Tucker, 307–17. Princeton: Princeton University Press.
- Shapley, L. S. 1981. Weighted Values: The Shapley Value and the Banzhaf Value. In *The Shapley Value: Essays in Honor of Lloyd S. Shapley*, edited by A. E. Roth, 155–70. Cambridge: Cambridge University Press.
- Weber, R. J. 1988. Probabilistic Values for Games. In *The Shapley Value: Essays in Honor of Lloyd S. Shapley*, edited by A. E. Roth, 101–20. Cambridge: Cambridge University Press. <https://doi.org/10.1017/CBO9780511528446.008>.
- Winter, E. 2002. The Shapley Value. In *Handbook of Game Theory with Economic Applications*, vol. 3, edited by R. J. Aumann and S. Hart, 2025–54. Amsterdam: Elsevier.