A Linear Approximation Method for the Shapley Value

Shaheen S. Fatima *

Department of Computer Science, Loughborough University, Loughborough LE11 3TU, U.K.

Michael Wooldridge

Department of Computer Science, University of Liverpool, Liverpool L69 7ZF, U.K.

Nicholas R. Jennings

School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K.

Abstract

The Shapley value is a key solution concept for coalitional games in general and voting games in particular. Its main advantage is that it provides a unique and fair solution, but its main drawback is the complexity of computing it (e.g for *voting games* this complexity is #P-complete). However, given the importance of the Shapley value and voting games, a number of approximation methods have been developed to overcome this complexity. Among these, Owen's *multi-linear extension* method is the most time efficient, being linear in the number of players. Now, in addition to speed, the other key criterion for an approximation algorithm is its *approximation error*. On this dimension, the multi-linear extension method is less impressive. Against this background, this paper presents a new approximation algorithm, based on randomization, for computing the Shapley value of voting games. This method has time complexity linear in the number of players, but has an approximation error that is, on average, lower than Owen's. In addition to this comparative study, we empirically evaluate the error for our method and show how the different parameters of the voting game affect it. Specifically, we show the following effects. First, as the number of players in a voting game increases, the average percentage error decreases. Second, as the quota increases, the average percentage error decreases. Third, the error is different for players with different weights; players with weight closer to the mean weight have a lower error than those with weight further away. We then extend our approximation to the more general k-majority voting games and show that, for n players, the method has time complexity $\mathcal{O}(k^2 n)$ and the upper bound on its approximation error is $\mathcal{O}(k^2/\sqrt{n})$.

Key words: Coalitional game theory, Shapley value, Approximation method

1 Introduction

Coalition formation is a key form of interaction in multi-agent systems. It is the process of bringing together two or more agents so as to achieve goals that individuals on their own cannot, or to achieve them more efficiently [2,19,24,23]. Often, in such situations, there is more than one possible coalition and a player's payoff depends on which one he joins. Given this, there are two key problems in this area. First, to ensure that none of the parties in a coalition has any incentive to break away from it and join another coalition. Second, to determine how the players split the gains from cooperation between themselves.

In this context, cooperative game theory deals with the problem of coalition formation and offers a number of solution concepts that possess desirable properties like *stability*, *fair division of joint gains*, and *uniqueness* [28,24]. Cooperative game theory differs from its non-cooperative counterpart, in that, it allows the players to form binding agreements, and so there is often an incentive to work together to receive the largest total payoff. Also, unlike non-cooperative game theory, cooperative games are not specified through a description of the strategic environment (including the order of the players' moves and the set of actions at each move) and the resulting payoffs. Instead, cooperative game theory reduces this collection of data to the coalitional form, where each coalition is represented by a single real number. In short, there are no actions, moves, or individual payoffs. The chief advantage of this approach, at least in multiple agent environments, is its practical usefulness. Specifically, it allows the abstraction of dealing with groups, rather than the individuals, and so much larger problems can be handled.

In more detail, cooperative game theory offers a number of solution concepts (such as the *core*, *kernel*, and *Shapley value* [24]) and a number of multi-agent systems researchers have used and extended these to facilitate automated coalition formation [34,35,32,27]. In so doing, a key challenge, from the multi-agent systems perspective, is to study the computational aspects of the solutions that game theory provides. This is important because many of these solutions are computationally hard to find and so of limited use in building actual systems. For example, computing the core is often NP-complete [10], while computing the Shapley value is often #P-complete [12].

To this end, this paper is concerned with efficiently computing the *Shapley value* [33]. In more detail, a player's Shapley value reflects how much that player *contributes* to a coalition — that is, how much value the agent adds to a coalition. An agent who never adds much has a small Shapley value, while an agent that always

* Corresponding author.
Email addresses: S.S.Fatima@lboro.ac.uk (Shaheen S. Fatima), M.J.Wooldridge@csc.liv.ac.uk (Michael Wooldridge), nrj@ecs.soton.ac.uk (Nicholas R. Jennings).

makes a significant contribution has a high Shapley value. Now, the main advantage of the Shapley value, over solution concepts such as the core and the kernel, is that it provides a solution that is both *unique* and *fair*. The former is desirable because it leaves no ambiguity; there is only one possible solution for a game and so the players know what they will gain from playing it. The latter property relates to how the gains from cooperation are split between coalition members. In this case, a player's Shapley value is proportional to the contribution he makes as a member of a coalition; the greater the contribution, the higher its value. Thus, from a player's perspective, both uniqueness and fairness are desirable properties.

However, while uniqueness and fairness are both desirable properties, the Shapley value has one major drawback: for many coalitional games, it cannot be determined in polynomial time. One of the most common coalitional games is the *voting game* (which is a means for the players to reach a consensus) and for this game, finding the Shapley value is #P-complete [12] (meaning that it is as hard as counting satisfying assignments of propositional logic formulae [26, p442]). Since #P-completeness subsumes NP-completeness, this implies that computing the Shapley value for the voting game will be intractable in general. In other words, it is practically infeasible to try to compute the exact Shapley value. However, the voting game has practical relevance not only in the context of multi-agent systems [29,35], but also in human settings, as it is an important means of reaching consensus between multiple parties.

Against this background, a number of *approximation* methods have been developed in order to overcome the problem of computational hardness of finding the exact Shapley value (see Section 3 for details). These methods vary in terms of their time complexities. Among these, however, Owen's *multi-linear extension* method [25] for a weighted voting game is one of the most time efficient, requiring time linear in the number of players. However, the accuracy with which it approximates the real value can be an issue in some cases (the method works well for those games for which all the players have small weights). To combat this, this paper presents a new approximation algorithm for computing the Shapley value for a weighted voting game. Our method is based on the technique of randomization and has time complexity that is linear in the number of players, but has a lower approximation error than Owen's method. In addition to this comparative study, we empirically evaluate the error for our method in a range of environments and show how the different parameters of the voting game affect the error. We then extend our approximation method (for a weighted voting game) to the more general k-majority voting games. For this, we show that for n players, the time complexity of our extended method is $\mathcal{O}(k^2 n)$ and the upper bound on its approximation error is $\mathcal{O}(k^2/\sqrt{n})$.

By undertaking this work, this paper makes a number of important contributions to the state of the art. First, and most importantly, it presents a new computationally efficient approximation algorithm for the Shapley value for weighted voting games. The proposed method has linear time complexity, and is better than Owen's method in terms of its error of approximation. Second, we extend our approximation method for a weighted voting game to the more general k-majority voting games. This is the first such method for this game. Finally, we provide a comprehensive error analysis of our approximation method. As mentioned earlier, we not only consider the worst case and obtain the upper bound on the error, but we also consider a general case and show how the different parameters of the voting game affect this error. This analysis distinguishes our work from the existing literature on approximation methods in that these have no error analysis [1] (neither for the worst, nor the general case). Nevertheless, we believe such analysis is essential because it enables us to present a complete picture of our method's performance in terms of how far the approximation can be from the exact Shapley value and how the different parameters of the voting game affect it.

The remainder of the paper is organized as follows. Section 2 defines the Shapley value more formally and details voting games studied in this paper. Section 3 discusses related literature. In Section 4, we present our method for finding the approximate Shapley value and analyze its approximation error in Section 5. In Section 6, we experimentally evaluate our method's approximation error and compare it with that of Owen's method. Section 7 concludes. Appendix A gives a summary of notation employed throughout the paper. Appendices B to F provide proofs of theorems. Appendix G and H give details on the results and data used for our experiments.

2 Background

We begin by introducing coalitional games and the Shapley value. We then define a weighted voting game and, its generalized form of a weighted k-majority game.

2.1 Coalitional games and the Shapley value

A coalition game is where groups of players ('coalitions') may enforce cooperative behavior between their members. Hence, the game is a competition between coalitions of players, rather than between individual players (c.f. non-cooperative game theory). Now, depending on how the players measure utility, coalitional game theory is split into two parts. If the players measure utility or the payoff in the same units and there is a means of exchange of utility, such as side payments, we say

[1] Error bounds for approximate coalition structure generation have been studied in [31], but there has been no such study in the context of finding the approximate Shapley value.

the game has *transferable utility*; otherwise it has *non-transferable utility*. More formally, a coalitional game with transferable utility, $\langle N, v\rangle$, consists of [24]:

(1) a finite set, $N = \{1, 2, \ldots, n\}$, of players and
(2) a function, v, that associates with every non-empty subset S of N (i.e., a *coalition*) a real number $v(S)$ that indicates the worth of S.

For each coalition S, $v(S)$ is the total payoff that is available for division among the members of S. Note that, viewed in this abstract way, a coalitional game gives no indication of how a coalition's value might or should be divided amongst coalition members. Coalitional games with non-transferable payoffs differ from those with transferable payoffs in that they associate with each coalition, a *set* of payoff vectors that is not necessarily the set of all possible divisions of some fixed amount. The focus of this paper is on weighted voting games (described in Section 2.2) which have transferable payoffs.

In a voting game, the players will only join a coalition if they expect to gain from it. Here, the players are allowed to form binding agreements, and so there is often an incentive to work together to receive the largest total payoff. The problem then is how to split the total payoff between the players. In this context, Shapley [33] constructed a solution using an axiomatic approach. In particular, he defined a *value* for games to be a function that assigns to a game $\langle N, v\rangle$, a number $\varphi_i(N, v)$ for each i in N. This function satisfies three axioms [30,33]:

(1) *Symmetry*: The names of the players play no role in determining the value. That is, two players who are identical with respect to what they contribute to a coalition should have the same Shapley value.
(2) *Carrier*: The sum of $\varphi_i(N, v)$ for all players i in any carrier C equals $v(C)$. Here carrier C is simply a subset of N such that $v(S) = v(S \cap C)$ for any subset of players $S \subset N$.We obtain the error $e(\bar{\varphi}_i)$ by propagating the error $e(E\Delta_i^{X-1})$ to all coalitions between size $X = 1$ to $X = n$. This is done using the following error propagation rules [37]. Let x and y be two random variables with errors $e(x)$ and $e(y)$ respectively. Then, from [37] we have the following propagation rules :

$\mathcal{R}_2$ The error in the random variable $z = x + y$ is:

$$e(z) = e(x) + e(y)$$

$\mathcal{R}_3$ If $z = kx$ where the constant k has no error, then the error in z is:

$$e(z) = |k|e(x)$$

$\mathcal{R}_4$ The error in the random variable $z = x \times y$ is:

$$e(z) = e(x) + e(y)$$

Note that for $X = 1$ (i.e., player i is the first member of a coalition), $e(E\Delta_i^{X-1}) =$

0 since we know that a one player coalition can never win and i's marginal contribution to such a coalition is therefore known to be zero. Also, recall from Theorem 2, that a player's approximate Shapley value is the average of its approximate marginal contributions to coalitions of size $1 \leq X \leq n$. Hence, as per rules $\mathcal{R}_2$, $\mathcal{R}_3$, and $\mathcal{R}_4$, the absolute error ($e(\bar{\varphi}_i)$) is the average of the approximation errors $e(E\Delta_i^{X-1})$ for all coalitions between the sizes $X = 1$ and $X = n$.

(3) *Additivity*: This specifies how the values of different games must be related to one another. It requires that for any games $\varphi_i(N, v)$ and $\varphi_i(N, v')$, $\varphi_i(N, v) + \varphi_i(N, v') = \varphi_i(N, v + v')$ for all i in N.

Shapley showed that there is a *unique* function that satisfies these three axioms. This value gives a *fair* division of the gains of cooperation between the members of a coalition. Thus, one can think of the Shapley value as a measure of the utility of risk neutral players in a game [30].

Having given these intuitions, we now turn to their formalization. Specifically, we first introduce notation and then define the Shapley value. Let $\mathcal{S}$ denote the set $N - \{i\}$ and $f_i : \mathcal{S} \rightarrow 2^{N-\{i\}}$ be a random variable that takes its values in the set of all subsets of $N - \{i\}$, and has the probability distribution function (g) defined as:

$$g(f_i(\mathcal{S}) = S) = \frac{|S|!(n - |S| - 1)!}{n!}$$

The random variable f_i is interpreted as the random choice of a coalition that player i joins. Then, a player's Shapley value is defined in terms of its *marginal contribution*. Thus, the marginal contribution of player i to coalition S with $i \notin S$ is a function $\Delta_i v$ that is defined as follows:

$$\Delta_i v(S) = v(S \cup \{i\}) - v(S) \tag{1}$$

This means a player's marginal contribution to a coalition S is the increase in the value of S as a result of i joining it. A player's Shapley value is defined in terms of its marginal contribution as follows [30,33]:

Definition 1 *The Shapley value (φ_i) of the game $\langle N, v \rangle$ for player i is the average of its marginal contribution to all possible coalitions:*

$$\varphi_i = \sum_{S \subset N} \frac{|S|!(n - |S| - 1)!}{n!} \times \Delta_i v(S) \tag{2}$$

The Shapley value may be interpreted as follows. Suppose that all the players are arranged in some order, all orderings being equally likely. Then $\varphi_i(N, v)$ is the *expected marginal contribution*, over all orderings, of player i to the set of players who precede him.

Now, the method for finding a player's Shapley value depends on the definition of the value function (v). This function is different for different games, but here we focus specifically on the voting game because the computation of its Shapley value is computationally hard. Furthermore, voting games are an important way of modeling situations where there are multiple agents, different agents have different preferences, and they want to reach a consensus. For these games, the Shapley value gives an indication of how much influence each agent has on reaching a consensus.

2.2 *Weighted voting games*

Let n be a set of players that may, for example, represent shareholders in a company or members in a parliament. A weighted voting game [24] is then a game $G = \langle N, v \rangle$ in which:

$$v(S) = \begin{cases} 1 \text{ if } w(S) \geq q \\ 0 \text{ otherwise} \end{cases}$$

for some $q \in I\!R_+$ and $w_i \in I\!R_+$, where:

$$w(S) = \sum_{i \in S} w_i$$

for any coalition S. Thus, w_i is the number of votes that player i has and q is the number of votes needed to win the game (i.e., the *quota*).

For this game (denoted $\langle q; w_1, \ldots, w_n \rangle$), a player's marginal contribution is either zero or one. This is because the value of a coalition is either zero or one. A coalition with value zero is called a "losing coalition" and with value one a "winning coalition". If a player's entry to a coalition changes it from losing to winning, then the player's marginal contribution for that coalition is one; otherwise it is zero. A coalition S is said to be a *swing* for player i if S is losing but $S \cup \{i\}$ is winning.

A well known example of weighted voting is the Electoral College of the United States, where the players are the 50 states plus the District of Columbia; each player casts a number of votes equal to the number of that state's representatives plus senators. Another example is a company of shareholders where each shareholder casts a number of votes proportional to the number of shares he owns.

Speaking more broadly, however, a weighted voting game is just a special case of a more general setting called a k-majority game (for $k = 1$, we have the weighted voting game). This general version is defined as follows.

2.3 k-majority games

For the set N of n players, a weighted k-majority game ($v_1 \wedge \ldots \wedge v_k$) is a situation in which $v_t = [q^t; w_1^t, \ldots, w_n^t]$ for $1 \leq t \leq k$ are weighted voting games and

$$(v_1 \wedge \ldots \wedge v_k)(S) = \begin{cases} 1 \text{ if } w^t(S) \geq q^t \text{ for } 1 \leq t \leq k \\ 0 \text{ otherwise} \end{cases}$$

where $w^t(S) = \sum_{i \in S} w_i^t$.

The k-majority game finds application, for example, in the European Union enlargement. The Council of ministers of the European Union represents the national governments of the member states. The Council uses a voting system of qualified majority to pass new legislation. The Nice European Council established two decision rules for the European Union enlarged to 27 countries. One of these two rules is a weighted triple majority game $v_1 \wedge v_2 \wedge v_3$ described in the following example (taken from [5]):

Example 1 *Each member state represented in the future Council is considered an individual player. The players in the Council of the European Union enlarged to 27 countries are:* $\{$*Germany, United Kingdom, France, Italy, Spain, Poland, Romania, The Netherlands, Greece, Czech Republic, Belgium, Hungary, Portugal, Sweden, Bulgaria, Austria, Slovak Republic, Denmark, Finland, Ireland, Lithuania, Latvia, Slovenia, Estonia, Cyprus, Louxembourg, Malta*$\}$. *The decision rule is a weighted triple majority game* $v_1 \wedge v_2 \wedge v_3$, *where the three weighted voting games corresponding to votes, countries, and population are the following:*

$$\begin{aligned} v_1 = \{&255; 29, 29, 29, 29, 27, 27, 14, 13, 12, 12, 12, 12, 12, 10, 10, 10, 7, 7, 7, 7, 7, \\ &4, 4, 4, 4, 4, 3\} \\ v_2 = \{&14; 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\} \\ v_3 = \{&620; 170, 123, 122, 120, 82, 80, 47, 33, 22, 21, 21, 21, 21, 18, 17, 17, 11, \\ &11, 11, 8, 8, 5, 4, 3, 2, 1, 1\} \end{aligned}$$

The game v_3 *is defined by assigning to each country, a number of votes equal to the rate per thousand of its population over the total population and the quota represents 62% of the total population. So a vote will be favourable if it counts on the support of 14 countries with at least 255 votes, and at least 62% of the population.*

Although voting games have so far mostly been used in human contexts, they are now being increasingly studied in the context of multi-agent systems [29,32,35]. In systems composed of autonomous agents, each agent has a range of problem solving capabilities and resources at its disposal. While such agents are typically

DE-SHAPLEYVALUE

1. For each $S \subset N$ do the following

   For i from 1 to n do the following

      If $i \in N - S$ and $q - w_i \leq w(S) < q$ then $\varphi_i \leftarrow \varphi_i + |S|!(n - |S| - 1)!/n!$

2. Return φ_i

Table 1
Direct enumeration algorithm to find the Shapley value for player i.

self interested, often there are benefits to be obtained from pooling resources. In such cases, the agents need to cooperate and coordinate their activities to achieve joint goals. Since the individual agents are autonomous, different agents may have different preferences over these joint goals. In such scenarios, voting is an effective mechanism for the agents to reach consensus on what goals to achieve. Given this, in what follows, we will analyze voting games in the abstract without reference to their different contexts.

3 Related work

For a weighted voting game, a number of methods have been proposed for finding the Shapley value. These methods can be divided into two types. The first compute the *exact value*. The second compute the *approximate value*. These methods vary in their approach and in their computing requirements. None is universally ideal. Given this, in what follows, we discuss the main methods in each category and highlight their main advantages and limitations.

3.1 Exact methods

The following are the four main methods that can be classified as exact:

(1) Direct enumeration
(2) Generating functions [22,8,36]
(3) Conitzer and Sandholm's method [9]
(4) Ieong and Shoham's method [17,18]

In more detail, the *direct enumeration* method directly applies Equation 2 to compute the Shapley value for player i. The algorithm for doing this is described in Table 1. Since the number of subsets of the set ($N = \{1, 2, \ldots, n\}$) of players is 2^n, evaluating the Shapley value for player i has time complexity $\mathcal{O}(2^n)$. The disadvantage of this method is that it has exponential time complexity, but its advantage is that it is a simple algorithm and it can reasonably be applied for finding the exact value for games with a small number of players.

The *generating functions* method [21,22] was proposed by Mann and Shapley. This finds the exact Shapley value in terms of the coefficients of a polynomial generated by a function (see [39] for details on generating functions). The disadvantage of this method is that it uses a substantial amount of memory, but its advantage is that it has polynomial time complexity – $\mathcal{O}(Cn^2)$ – where C is the number of possible vote totals [6]. For games where every player has the same weight, there are $n+1$ possible vote totals. For games where each player has a unique weight, the number of possible vote totals may be 2^n. Therefore, this method is practical for games in which many players have the same weights. Methods based on generating functions have also been proposed in [8,36,1]. Generally speaking, these methods also tradeoff memory space for computation time and so cannot easily scale to larger numbers of agents.

Conitzer et al [9] use a method to find the exact value. However, their method can be used only if a characteristic function game is represented in a specific decomposed form. Also, this method has exponential time complexity and so it does not scale well.

Finally, the method proposed by Ieong et al [17,18] assumes that the Shapley value of a component of a given coalitional game is given by an oracle. Then, on the basis of this assumption, it aggregates these values to find the value for the overall game. The advantage of this method is that it has polynomial time complexity. However, its disadvantage is that it can be used only if the coalitional game is represented as a “marginal contribution net”.

To summmarize, all the above methods give the exact Shapley value, but they each have disadvantages; including requiring exponential time (as in the case with direct enumeration and, in some cases, with Mann and Shapley’s), a large memory space (as in the case of Mann and Shapley), or a specific representation [2] for the voting game (as is the case with Conitzer et al and Ieong et al). In order to overcome these problems, a number of approximation methods were developed.

3.2 Approximation methods

The methods that have been proposed to approximate the Shapley value are as follows:

(1) Monte Carlo simulation method [21]

[2] Note that transforming a voting game into these specific forms requires additional computational time.

(2) Multi-linear extension (MLE) method [25]
(3) Modified MLE method [20]
(4) Random permutation method [40]

The earliest approximation method was proposed by Mann and Shapley [21]. This method is based on Monte Carlo simulation and estimates the Shapley value from a random sample of coalitions. This is done as follows. Suppose a coalition S is selected by randomly sampling the players. Define a random variable X for each player. This X is one for player i if it is the swing player for the coalition, and zero otherwise. Then, the expectation of X is $E[X] = E[\frac{|S|!(n-1-|S|)!X}{n!}] = \varphi_i$ and the variance is $Var(X) = Var[\frac{|S|!(n-1-|S|)!X}{n!}] = \varphi_i(1-\varphi_i)$. Now take a sequence of m independent drawings, $X_1, \ldots, X_m$, with corresponding coalition sizes $|S_1|, \ldots, |S_m|$. Then, the estimated Shapley value ($\hat{\varphi}_i$) for i is:

$$\hat{\varphi}_i = \frac{1}{m}\sum_{j=1}^{m}\frac{|S_j|!(n-|S_j|-1)!}{n!}X_j$$

with variance $Var(\hat{\varphi}_i) = \varphi_i(1-\varphi_i)/m$. The variance decreases as m increases. The disadvantage of this method is that it does not give details of how the samples are to be drawn, which has a significant impact on the method's effectiveness. Given this, it it is hard to assess the accuracy of this method. However, the advantage is its linear time complexity.

For the sampling approach, Bachrach and Rosenschein provide an analysis of the error bounds and minimum number of samples required to achieve a given accuracy [4]. In more detail, they give randomized approximation methods for power indices such as the Shapley value, which can be used for any simple coalitional game. They show that their approximation methods approach the optimal, and give lower bounds for both deterministic and randomized approaches to computing power indices.

The MLE approximation method proposed by Owen [25] works as follows. Consider player i. Let T_i be a coalition such that T_i is losing, but $T_i \cup \{i\}$ is winning. Then, as per Equation 2, i's Shapley value is:

$$\varphi_i = \sum_{T_i}\frac{|T_i|!(n-|T_i|-1)!}{n!} \tag{3}$$

Equation 3 for the Shapley value can be rewritten by noting that the term inside the summation is a beta function:

$$B(t+1, n-t) = \frac{t!(n-t-1)!}{n!} = \int_0^1 x^t(1-x)^{n-t-1}dx \tag{4}$$

where $t = |T_i|$ is the number of players in T_i. The integrand on the right hand side of Equation 4, $x^t(1-x)^{n-t-1}$, can be interpreted as the probability that the random subset T_i appears, when x is the probability that any member joins T_i, assumed constant and independent for all players j, $j \in N - \{i\}$. Summing this expression over all swings gives the probability of a swing for i. Call this probability $f_i(x)$:

$$f_i(x) = \sum_{T_i} x^t(1-x)^{n-t-1}. \tag{5}$$

Integrating x out of Equation 4 gives the Shapley value, because substituting Equation 4 in Equation 3 gives:

$$\begin{aligned}\varphi_i &= \sum_{T_i} \int_0^1 x^t(1-x)^{n-t-1}dx = \int_0^1 \Big(\sum_{T_i} x^t(1-x)^{n-t-1}\Big)dx \\ &= \int_0^1 f_i(x)dx.\end{aligned} \tag{6}$$

Thus, the Shapley value can be evaluated by integrating $f_i(x)$. But as per Equation 5, this requires evaluating a function whose size doubles every time a new player is added. Obviously, this method has exponential time complexity. In order to overcome this problem, Owen approximated $f_i(x)$ as follows. Assuming that each player votes in the same way as i with probability x, independently of others, a random variable $v_i(x)$ is defined that counts the number of votes cast by others on the same side as i. Its mean ($\bar{\mu}_i$) and variance ($\bar{\nu}_i$) are:

$$\bar{\mu}_i(x) = E(v_i(x)) = xw(N - \{i\}) = xw(N) - xw_i \tag{7}$$

and

$$\begin{aligned}\bar{\nu}_i(x) &= Var(v_i(x)) = x(1-x)h(N - \{i\}) \\ &= x(1-x)h(N) - x(1-x)w_i^2,\end{aligned} \tag{8}$$

where $h(T) = \sum_{i \in T} w_i^2$ is the sum of squared weights.

In large games with many small weights and no large weights, $v_i(x)$ will be approximately normally distributed, and the desired swing probability:

$$f_i(x) = Pr[q - w_i \leq v_i(x) < q] \tag{9}$$

can be obtained approximately using the normal distribution function $\mathcal{N}(.)$ as follows:

$$f_i(x) \approx \mathcal{N}\left(\frac{q - \bar{\mu}_i(x)}{\bar{\nu}_i(x)}\right) - \mathcal{N}\left(\frac{q - \bar{\mu}_i(x) - w_i}{\bar{\nu}_i(x)}\right). \tag{10}$$

The advantage of this method is its linear time complexity.

The modified MLE method [20] is an extension of the above described MLE method that trades-off computational time in order to improve the error of approximation. In more detail, the modified MLE method combines the essential features of direct enumeration and MLE in order to improve the accuracy of the MLE method. Specifically, the players are divided into two subsets: major players with large weights $L = \{1, 2, \ldots, l\}$, and minor players $N - L$. This combined method treats the major players using enumeration as in the direct approach, but treats minor players using Owen's MLE approximation technique. Large values of l will improve accuracy, but will also increase computation time. The advantage of this method is that it generates a better approximation than Owen's MLE method, but its disadvantage is that it has exponential time complexity: $\mathcal{O}(2^l)$.

Finally, an an approximation method was proposed by Zlotkin and Rosenschein [40]. This is a *random permutation mechanism* where the players choose a random permutation and form the full coalition, one player after another, according to the chosen permutation. Here each player gets a utility equal to its contribution to the coalition at the time of joining it. If each permutation has equal chance of being chosen, then this mechanism gives each player an expected utility equal to its Shapley value. This method requires the players to agree on an all-or-nothing deal. The advantage of this method is its linear time complexity. However, for weighted voting games, getting the players to agree on an all-or-nothing deal may be an issue because, for these games, there is only one swing player for each possible coalition. So only one player gets a utility of one and all others get zero utility.

In summary, the existing literature on methods for finding the Shapley value have two key drawbacks. First, it only describes methods for finding an approximate Shapley value, but there is no associated error analysis available. Second, none of these methods can easily be extended to the generalized k-majority game. In order to overcome these drawbacks, we present new approximation methods for both weighted voting games and k-majority games. These methods are computationally efficient and have polynomial time complexity. Furthermore, we provide a comprehensive error analysis in terms of their worst and average case performance.

4 A new method for the approximate Shapley value

This section details the main contribution of the paper, namely a new method for finding the approximate Shapley value. As already stated, our approach is based on randomized algorithms, which are one of the most commonly used approaches for finding approximate solutions to problems whose exact solutions are hard to compute. In short, these algorithms tradeoff accuracy for computational time. More specifically, a randomized algorithm is one that, during some of its steps, makes random choices [3]. Moreover, since such algorithms generate approximate solutions, their performance is typically evaluated in terms of two criteria [3]: their time complexity and their error of approximation (i.e., the difference between the exact solution and its approximation).

Against this background, we present a new randomized algorithm for finding the approximate Shapley value for a weighted voting game. We then extend this method to k-majority games. Finally, we evaluate the approximation error for these algorithms; both theoretically to obtain an upper bound (in Section 5) and experimentally in order to obtain a more typical average case (in Section 6).

4.1 For a weighted voting game

The intuition behind the proposed method is as follows. As per Definition 1, in order to find a player's Shapley value, we first need to find its marginal contribution to all possible coalitions. For n players, there are 2^{n-1} possible coalitions. Finding a player's marginal contribution to each of these 2^{n-1} possible coalitions is computationally infeasible. So we do not attempt to find the marginal contribution to each possible coalition. Rather, we consider n random coalitions. The first coalition is of size one, the second is of size two, and so on. We find a player's approximate marginal contribution to each of these n coalitions. The average of all these marginal contributions gives the player's approximate Shapley value. Theorem 1 characterizes a player's approximate marginal contribution to a random coalition of size X, and Theorem 2 characertrizes its approximate Shapley value.

In what follows, φ_i ($\bar{\varphi}_i$) denotes the exact (approximate) Shapley value for player i for a weighted voting game. Also, the approximate marginal contribution of player i to a random coalition of size X is denoted $E\Delta_i^X$.

Theorem 1 *For an n player weighted voting game with mean weight μ and variance ν, player i's approximate marginal contribution ($E\Delta_i^X$) to a random coalition of size X ($1 \leq X \leq n$) is:*

$$E\Delta_i^X = \frac{1}{\sqrt{(2\pi\nu/X)}} \int_a^b e^{-X\frac{(x-\mu)^2}{2\nu}} dx. \tag{11}$$

where $a = (q - w_i)/X$, $b = (q - \epsilon)/X$, and w_i is player i's weight.

Proof: To find a player's approximate marginal contribution to a random coalition, we use the following rule from sampling theory. Let the players' weights in N be

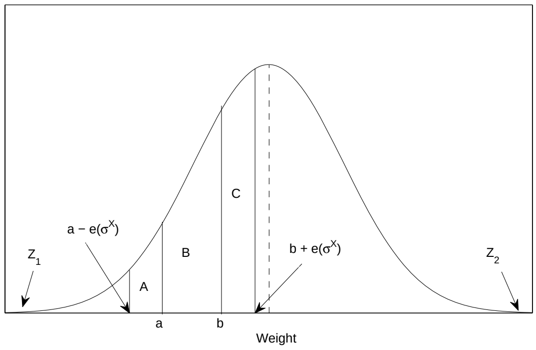

Fig. 1. A normal distribution for the players' weights in a coalition of size X.

denoted $w_1, w_2, \ldots, w_n$. Irrespective of how these weights are distributed, let the mean weight be μ and the variance ν. From this set (N) if we randomly draw a sample coalition, then the mean of the players' weights in the sample coalition is given by the following rule [16]:

> $\mathcal{R}_1$: If $w_1, w_2, \ldots, w_X$ is a random sample of size X drawn from 'any distribution' with mean μ and variance ν, then the sample mean (i.e., $\frac{1}{X}\sum_{i=1}^{X} w_i$) has an *approximate Normal distribution*, $\mathcal{N}$, with mean μ and variance $\frac{\nu}{X}$ (the larger the X the better the approximation [3]).

From Section 2.2, we know that for a weighted voting game, the marginal contribution of player i to a random coalition of size X is one if the total weight of the X players in the coalition is greater than or equal to $q - w_i$ but less than $q - \epsilon$ (where ϵ is an inifinitesimally small quantity). Otherwise, its marginal contribution is zero. Thus, the approximate marginal contribution of player i to a random coalition is the area under the curve defined by $\mathcal{N}(\mu, \frac{\nu}{X})$ in the interval $[a, b]$ where $a = (q - w_i)/X$ and $b = (q - \epsilon)/X$. This area is shown as the region B in Figure 1 (the dotted line in the figure is μ – the mean of the weights for a coalition of size X). If the mean weight in a coalition of size X is a, then the sum of weights of the coalition is $q - w_i$. Likewise, if the mean weight of a coalition of size X is b, then the sum of weights of the coalition is $q - \epsilon$. Hence, i's approximate marginal contribution to X is:

$$E\Delta_i^X = \frac{1}{\sqrt{(2\pi\nu/X)}} \int_a^b e^{-X\frac{(x-\mu)^2}{2\nu}} dx.$$

□

Note that, in order to find $E\Delta_i^X$, we do not actually draw sample coalitions. Rather, we use rule $\mathcal{R}_1$ to find the probability distribution (i.e., normal), the mean (i.e., μ),

[3] Also, for large X, any measurement done on a sample drawn with replacement is the same as that for a sample drawn without replacement [16].

and the error (i.e., ν/X) in the approximate weight in a random coalition of size X. The advantage of not having to draw samples will be explained in detail after presenting our method for finding an approximate Shapley value.

We now formulate the approximate Shapley value of player i in terms of its marginal contribution. This is done in Theorem 2.

Theorem 2 *For an n player weighted voting game, player i's approximate Shapley value is:*

$$\bar{\varphi}_i = \frac{1}{n} \sum_{X=1}^{n} E\Delta_i^{X-1} \tag{12}$$

where $E\Delta_i^X$ is as defined in Theorem 1.

Proof: Consider player i. In order to find i's Shapley value, we must consider all possible ways in which i can join in a coalition. For a game of n players, i can join a coalition as the Xth the member where $1 \leq X \leq n$. If i joins as the Xth member, there are $X-1$ players that precede it. Then i joins and then the remaining $n-X$ players join in. From Section 2.2, we know that the value of a coalition for a weighted voting game depends on the sum of weights of the players in the coalition. In other words, this value does not depend on the order in which the players joined the coalition. It follows that i's marginal contribution to the coalition of $X-1$ players that precede it depends on the weights of these players and not on the order in which they formed a coalition. Also, according to rule $\mathcal{R}_1$, the approximate weight in a random coalition of size $X-1$ depends on the coalition size $X-1$ and not on the actual players in it. Consequently, $E\Delta_i^{X-1}$ also depends only on X.

Now, there are $P(n-1, X-1)$ possible coalitions of $X-1$ players (where $P(n-1, X-1)$ denotes the number of permutations of $X-1$ players drawn from the set of $n-1$ players excluding player i) that precede i. Also, the remaining $n-X$ players that join after i can do so in $(n-X)!$ ways. Thus, there are $P(n-1, X-1) \times (n-X)!$ possible coalitions where i joins as the Xth member. Since i's approximate marginal contribution to a random coalition of size $X-1$ is $E\Delta_i^{X-1}$, its total marginal contribution to all possible coalitions where i joins a the Xth member is $P(n-1, X-1)(n-X)!E\Delta_i^{X-1}$. Given this, i's approximate total marginal contribution to all possible coalitions (i.e., for all possible values of X) is $\sum_{X=1}^{n} P(n-1, X-1)(n-X)!E\Delta_i^{X-1}$. It follows that i's approximate Shapley value, which is its average marginal contribution to all $n!$ possible coalitions (see Definition 1) is:

$$\begin{aligned}\bar{\varphi}_i &= \frac{1}{n!}\sum_{X=1}^{n} P(n-1, X-1)(n-X)! E\Delta_i^{X-1} \\ &= \frac{(n-1)!}{n!}\sum_{X=1}^{n} E\Delta_i^{X-1} \\ &= \frac{1}{n}\sum_{X=1}^{n} E\Delta_i^{X-1}\end{aligned}$$

□

Algorithm 1 ShapleyValueWVG(n, q, μ, ν, w_i)

n: Number of players
q: Quota for the game
μ: Mean weight of the players in N
ν: Variance in the weights of the players in N
w_i: Player i's weight

1: $T_i \Leftarrow 0$
2: **for** X = 0 to $n-1$ **do**
3: $a \Leftarrow (q - w_i)/X$; $b \Leftarrow (q - \epsilon)/X$
4: $E\Delta_i^X \Leftarrow \frac{1}{\sqrt{2\Pi\nu/X}} \int_a^b e^{-X\frac{(x-\mu)^2}{2\nu}}$ dx
5: $T_i \Leftarrow T_i + E\Delta_i^X$
6: **end for**
7: $\bar{\varphi}_i \Leftarrow T_i/n$
8: **return** $\bar{\varphi}_i$

Having got the approximate Shapley value, we are now ready to present this computation in Algorithm 1. In more detail, Step 1 does the initialization. In Step 2, we vary X between 0 and $n-1$ and repeatedly do the following. Player i's approximate marginal contribution to a random coalition of size X is found in Step 3. Step 4 finds the sum of these n marginal contributions. The average of these marginal contributions is found in Step 6 – and this is an approximate Shapley value for player i.

The key advantages of the method used in Algorithm 1 are as follows. First, it does not require making measurements on randomly drawn samples. This is because our method is based on $\mathcal{R}_1$ which gives an approximate distribution for the weights in a random coalition. Contrast our method with that of Mann and Shapley [21,22] which estimates an approximate Shapley value by making measurements on randomly drawn samples. The key disadvantage of actually drawing samples is that the estimate depends on how the samples are drawn; if we change the way in which samples are drawn, then the estimate changes. The method we propose is independent of any such details because it does not require the actual drawing of samples. The second advantage of our method is that it only requires the number of players, the quota, the mean weights, and the variance in the weights – the weights of the individual players are not needed. The third advantage of our method is that it can easily be extended to k-majority games. Having detailed the algorithm, we now consider its complexity.

Theorem 3 *For a game of n players, the time complexity of computing a player's approximate Shapley value ($\bar{\varphi}_i$) using Algorithm 1 is $\mathcal{O}(n)$.*

Proof: From Equation 11, we know that $E\Delta_i^X$ can be found in constant time. As per Equation 12, $E\Delta_i^{X-1}$ must be computed n times. This is done in the for loop of Step 2 in Algorithm 1. Since this for loop is repeated n times, the time complexity of computing a player's approximate Shapley value ($\bar{\varphi}_i$) using Algorithm 1 is $\mathcal{O}(n)$.
□

4.2 For a k-majority game

We now extend the method described in Algorithm 1 to k-majority games. The intuition behind the proposed method is as follows. As described in Section 2.3, a k-majority game is defined in terms of k weighted voting games v_g ($1 \leq g \leq k$). So we first find a player's approximate marginal contribution to v_g ($1 \leq g \leq k$). Given these k marginal contributions, we find the marginal contribution and then the Shapley value for a k-majority game. Before doing so, we introduce some notation.

For game v_g, let μ^g denote the mean weight of the players, ν^g the variance in their weights, and q^g the quota. For game v_g and player i, let $PL_i^g(S_X)$ denote the probability that the coalition S_X is losing but $S_X \cup \{i\}$ is winning. And, for game v_g, let $PW_i^g(S_X)$ denote the probability that the coalition $S_X \cup \{i\}$ is winning. Also, for a k-majority game, let $kE\Delta_i^X$ denote the approximate marginal contribution of player i to a random coalition S_X of size X. Finally, let φ_i^k ($\bar{\varphi}_i^k$) denote the exact (approximate) Shapley value for player i for a k-majority game.

Theorem 4 characterizes a player's approximate marginal contribution, and Theorem 5 a player's approximate Shapley value.

Theorem 4 *For an n player k-majority game with mean weight μ^g and variance ν^g ($1 \leq g \leq k$), player i's approximate marginal contribution ($kE\Delta_i^X$) to a random coalition S_X of size X is:*

$$kE\Delta_i^X = \sum_{j=0}^{k-1} \left(\Pi_{g=1}^{j}(1 - PL_i^g(S_X)) \times PL_i^{j+1}(S_X) \times \Pi_{f=j+2}^{k} PW_i^f(S_X) \right) \quad (13)$$

where

$$PL_i^g(S_X) = \frac{1}{\sqrt{(2\pi\nu^g/X)}} \int_{(q^g - w_i^g)/X}^{(q^g-\epsilon)/X} e^{-X\frac{(x-\mu^g)^2}{2\nu^g}} dx \tag{14}$$

and

$$PW_i^g(S_X) = \frac{1}{\sqrt{(2\pi\nu^g/X)}} \int_{(q^g - w_i^g)/X}^{\infty} e^{-X\frac{(x-\mu^g)^2}{2\nu^g}} dx. \tag{15}$$

Proof: Consider player i. From Section 2.3, we know that the approximate marginal contribution of player i to a coalition S_X for the game $v_1 \wedge \ldots \wedge v_k$ is 1 if the following conditions hold:

(1) there is at least one game v_g ($1 \leq g \leq k$) for which i is the swing player, and
(2) for each game v_g ($1 \leq g \leq k$), the value of $S_X \cup \{i\}$ is 1.

We find the probabilities PL_i^g and PW_i^g using the sampling rule $\mathcal{R}_1$. As per this rule, $PL_i^g(S_X)$ is the area under the normal distribution $N(\mu^g, \nu^g/X)$ between the limits $(q^g - w_i^g)/X$ and $(q^g - \epsilon)/X$:

$$PL_i^g(S_X) = \frac{1}{\sqrt{(2\pi\nu^g/X)}} \int_{(q^g - w_i^g)/X}^{(q^g-\epsilon)/X} e^{-X\frac{(x-\mu^g)^2}{2\nu^g}} dx \tag{16}$$

And $PW_i^g(S_X)$ is the area under the normal distribution $N(\mu^g, \nu^g/X)$ between the limits $(q^g - w_i^g)/X$ and ∞:

$$PW_i^g(S_X) = \frac{1}{\sqrt{(2\pi\nu^g/X)}} \int_{(q^g - w_i^g)/X}^{\infty} e^{-X\frac{(x-\mu^g)^2}{2\nu^g}} dx. \tag{17}$$

Given Equations 16 and 17, we find $kE\Delta_i^X$ by considering all possible ways in which i can be swing player. For $0 \leq j \leq k-1$, the probability that i is not swing player for games 1 to j, it is swing player for game $j+1$, and may or may not be swing player for games $j+2$ to k is:

$$\Pi_{g=1}^{j}(1 - PL_i^g(S_X)) \times PL_i^{j+1}(S_X) \times \Pi_{f=j+2}^{k} PW_i^f(S_X)$$

By summing the above expression over all possible j (i.e., between zero and $k-1$) we get i's approximate marginal contribution given in Equation 13. □

Algorithm 2 ShapleyValue-KMG(k, n, q, μ, ν, w_i)

k: The number of weighted voting games
n: Number of players
q: A k-element vector containing the quotas for the games
μ: A k element vector containing the mean weight of the players for the k games
ν: A k element vector containing the variance in the weights of the players for the k games
w_i: A k-element vector containing player i's weight for the k games

$T_i \Leftarrow 0$;
for $X = 0$ to $n - 1$ **do**
    sum $\Leftarrow 0$
    for $j = 0$ to $k - 1$ **do**
        prod $\Leftarrow 1$
        for $g = 1$ to j **do**
            $a \Leftarrow (q^g - w_i^g)/X$; $b \Leftarrow (q^g - \epsilon)/X$
            prod $\Leftarrow$ prod $\times \left(1 - \frac{1}{\sqrt{2\Pi\nu^g/X}} \int_a^b e^{-X\frac{(x-\mu^g)^2}{2\nu^g}} dx\right)$
        end for
        $a \Leftarrow (q^{j+1} - w_i^{j+1})/X$; $b \Leftarrow (q^{j+1} - \epsilon)/X$
        prod $\Leftarrow$ prod $\times \frac{1}{\sqrt{2\Pi\nu^{j+1}/X}} \int_a^b e^{-X\frac{(x-\mu^{j+1})^2}{2\nu^{j+1}}} dx$
        for $f = j + 2$ to k **do**
            $a \Leftarrow (q^f - w_i^f)/X$; $b \Leftarrow \infty$
            prod $\Leftarrow$ prod $\times \frac{1}{\sqrt{2\Pi\nu^f/X}} \int_a^b e^{-X\frac{(x-\mu^f)^2}{2\nu^f}} dx$
        end for
        sum $\Leftarrow$ sum + prod
    end for
    $kE\Delta_i^X \Leftarrow$ sum
    $T_i \Leftarrow T_i + E\Delta_i^X$
end for
$\bar{\varphi}_i^k \Leftarrow T_i/n$
return $\bar{\varphi}_i^k$

We now formulate an approximate Shapley value for player i in terms of its marginal contribution. This is done in Theorem 5.

Theorem 5 *For an n player k-majority game with mean weight μ^g and variance ν^g ($1 \leq g \leq k$), player i's approximate Shapley value is:*

$$\bar{\varphi}_i^k = \frac{1}{n} \sum_{X=1}^{n} kE\Delta_i^{X-1} \tag{18}$$

where $kE\Delta_i^X$ is as defined in Theorem 4.

Proof: As Theorem 2. ▫

The steps for computing $\bar{\varphi}_i^k$ are detailed in Algorithm 2. The time complexity of this method is formulated in Theorem 6.

Theorem 6 *The time complexity of Algorithm 2 is $\mathcal{O}(k^2n)$.*

Proof: The time to execute the for loop in Step 4 of Algorithm 2 is $\mathcal{O}(k^2)$. Since this for loop is within the for loop of Step 2 (which is executed n times), the time complexity of Algorithm 2 is $\mathcal{O}(k^2n)$. ▫

As already noted, the quality of an approximation method is evaluated on the basis of both its running time and its approximation error. To this end, we will now provide a comprehensive error analysis. This is done in three steps. First, we analytically find the upper bound on the error. This upper bound gives an indication of how our method performs in the *worst* case. Second, we provide an experimental analysis of the error in the *general* case. Third, we experimentally compare the error for our method with that for Owen's. In the following section, we first derive the formulae for error and carry out its worst case analysis for the methods proposed in Algorithms 1 and 2.

5 Worst case analysis of the approximation error

We first formalize the idea of *error* and then derive the formula for measuring it. The concept of error relates to an approximate measurement made of a quantity which has an exact value [37,7]. Obviously, it cannot be determined exactly how far off an approximation is from the exact value; if this could be done, it would be possible to just give the more accurate, corrected value. Thus, error has to do with *uncertainty* (i.e., *variance* or *standard error*) in measurements that nothing can be done about. If a measurement is repeated, the values obtained may differ and none of the results can be preferred over the others. However, although it is not possible to do anything about such an error, it can be characterized in terms of two essential components [37,7]:

(1) a numerical value giving the best "estimate" possible of the quantity measured, and
(2) an error, i.e., the degree of *uncertainty* or *variance* associated with this estimated value.

For example, if the approximate measurement of a given quantity is x and the approximation error is $e(x)$, the quantity would lie in the interval $x \pm e(x)$. For sampling based experiments, approximation error is defined as follows [37]:

Definition 2 *The approximation error (i.e., sampling error) in a set of measurements on random samples is the standard deviation for the set of measurements divided by the square root of the number of measurements.*

Since the exact value that corresponds to an approximation lies in the range $x \pm e(x)$, the term *standard error* is analogous to the algorithmic term *absolute error* which is defined as follows [3]:

Definition 3 *The absolute error of an approximation is the absolute difference between the approximate and its exact counterpart.*

The following section defines this error and uses it to evaluate the performance of the proposed randomized method. We first find this error for a weighted voting game and the k-majority game. Then we find the upper bound on these errors.

5.1 *Absolute error*

We first consider our approximation method for the weighted voting game. The error for this method depends on the error in the approximation rule $\mathcal{R}_1$ defined in the proof for Theorem 1. This error is defined as follows [37,7]:

Definition 4 *For rule $\mathcal{R}_1$, the absolute error ($e(\sigma^X)$) in a player's weight in a coalition S_X of size X is:*

$$
\begin{aligned}
e(\sigma^X) &= \sqrt{(}\nu/X)/\sqrt{(}X) \\
&= \sqrt{(}\nu)/X.
\end{aligned}
\tag{19}
$$

On the basis of Definition 4, we find the absolute error in a player's approximate marginal contribution ($e(E\Delta_i^X)$) and its approximate Shapley value ($e(\bar{\varphi}_i)$).

Theorem 7 *For an n player weighted voting game with mean weight μ and variance ν, the absolute error in $E(\Delta_i^X)$ with respect to its exact counterpart (denoted Δ_i^X) is:*

$$
\begin{aligned}
e(E\Delta_i^X) &= abs(E(\Delta_i^X) - \Delta_i^X) \\
&= \frac{1}{\sqrt{(}2\pi\nu/X)} \times \left(\int_{a-e(\sigma^X)}^{a} e^{\frac{-X(x-\mu)^2}{2\nu}} dx + \int_{b}^{b+e(\sigma^X)} e^{\frac{-X(x-\mu)^2}{2\nu}} dx \right)
\end{aligned}
\tag{20}
$$

where $a = (q - w_i)/X$ *and* $b = (q - \epsilon)/X$.

Proof: See Appendix B. □

On the basis of the $e(E\Delta_i^X)$, Theorem 8 characterizes the absolute error in the Shapley value for player i.

Theorem 8 *For an* n *player weighted voting game, if* $\bar{\varphi}_i$ *denotes player* i*'s approximate Shapley value that corresponds to the exact* φ_i*, then the absolute error of* $\bar{\varphi}_i$ *with respect to* φ_i *is:*

$$e(\bar{\varphi}_i) = abs(\varphi_i - \bar{\varphi}_i) = \frac{1}{n}\sum_{X=1}^{n} e(E\Delta_i^{X-1}) \tag{21}$$

Proof: See Appendix C. □

Given Theorem 8, the percentage error, PE_i, in player i's Shapley value for a weighted voting game is:

$$PE_i = 100 \times e(\bar{\varphi}_i)/\varphi_i \tag{22}$$

On the basis of Theorem 8, we now obtain the error for our approximation method for a k-majority game. Let $e(\sigma_g^X)$ be the error in the approximate weight of players in S_X for game g. Let $e(PL_i^g(S_X))$ and $e(PW_i^g(S_X))$ denote the errors in $PL_i^g(S_X)$ and $PW_i^g(S_X)$ respectively [4] . These two errors are obtained in the same way as we obtained $e(E\Delta_i^X)$ in Theorem 7. Thus, we have:

$$e(PL_i^g(S_X)) = \frac{1}{\sqrt{(2\pi\nu^g/X)}} \times \left(\int_{(q^g - w_i^g)/X - e(\sigma_g^X)}^{(q^g - w_i^g)/X} e^{-X\frac{(x-\mu^g)^2}{2\nu^g}} dx + \int_{(q^g-\epsilon)/X}^{(q^g-\epsilon)/X + e(\sigma_g^X)} e^{-X\frac{(x-\mu^g)^2}{2\nu^g}} dx \right) \tag{23}$$

and

$$e(PW_i^g(S_X)) = \frac{1}{\sqrt{(2\pi\nu^g/X)}} \times \int_{(q^g - w_i^g)/X - e(\sigma_g^X)}^{(q^g - w_i^g)/X} e^{-X\frac{(x-\mu^g)^2}{2\nu^g}} dx. \tag{24}$$

[4] See Equations 16 and 17 for a definition of $PL_i^g(S_X)$ and $PW_i^g(S_X)$.

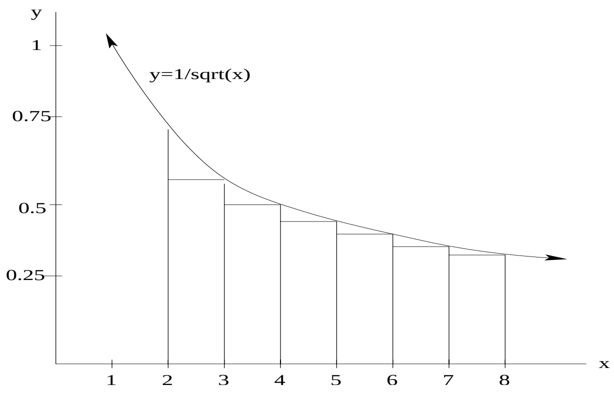

Fig. 2. Approximating sum ($\sum_{X=1}^{n} \frac{1}{\sqrt{X}}$) with definite integral.

For a k-majority game, let $e(kE\Delta_i^X)$ denote the absolute error in i's marginal contribution to a random coalition S_X, and let $e(\bar{\varphi}_i^k)$ denote the absolute error in i's Shapley value.

Theorem 9 *For an n player k-majority game, if $\bar{\varphi}_i^k$ denotes player i's approximate Shapley value that corresponds to the exact (φ_i^k), then the absolute error of $\bar{\varphi}_i^k$ with respect to φ_i^k is:*

$$e(\bar{\varphi}_i^k) = abs(\varphi_i^k - \bar{\varphi}_i^k) = \frac{1}{n} \sum_{X=1}^{n} e(kE\Delta_i^{X-1}) \tag{25}$$

Proof: See Appendix D. □

Having formulated $e(\bar{\varphi}_i)$ and $e(\bar{\varphi}_i^k)$, we now find the upper bound on these absolute errors.

5.2 Upper bound for weighted voting game

Theorem 10 characterizes the upper bound for $e(\bar{\varphi}_i)$.

Theorem 10 *For a weighted voting game of n players, the upper bound for the absolute error in the approximate Shapley value ($e(\bar{\varphi}_i)$) is $\mathcal{O}(1/\sqrt{n})$. Also, as $n \to \infty$, $e(\bar{\varphi}_i) \to 0$.*

Proof: See Appendix E. □

Theorem 10 shows that, as n increases, the upper bound on the error decreases (Figure 2 shows how $1/\sqrt{n}$ varies with n). This happens because as n increases, the error in Equation 19 decreases and, consequently, the error in the Shapley value decreases with n [5] .

5.3 Upper bound for k-majority games

On the basis of Theorem 10, we get the upper bound for the error for our randomized method for k-majority games.

Theorem 11 *For an n player k-majority game, the upper bound for $e(\bar{\varphi}_i^k)$ is $\mathcal{O}(k^2/\sqrt{n})$. Also, as $n \to \infty$, $e(\bar{\varphi}_i^k) \to 0$.*

Proof: See Appendix F. ▫

Here the error increases in k because, as per the method described in Section 4.2, we make k approximate measurements on a random coalition of size X ($1 \leq X \leq n$). For a weighted voting game, $k = 1$ (see Section 4.1), so we make a single approximate measurement on a coalition of size X. As k increases, the number of approximate measurements also increases and so does the error.

6 Experimental analysis of approximation error

For a weighted voting game, Theorem 10 gives the error in the worst case. However, in general, the error may well be less than this upper bound. Hence, we now focus on the general case and conduct an empirical analysis of the error for our approximation method. There are two objectives to this analysis:

(1) To compare the error for our method with that of Owen's.
(2) To analyze the effect of the parameters of a voting game on the error for our method.

We describe the former analysis in Section 6.2 and the latter in Section 6.3. For both, we focus on weighted voting games and not k-majority games. There are two reasons for this. First, from Section 5, we know the effect of k on the approximation error; $e(\bar{\varphi}_i^k)$ increases with k. This is because $e(kE\Delta_i^X)$ increases in k (see Equation D.1). Given this, we now want to find how the parameters n, q, and μ affect

[5] Note that we have found the bound for the absolute error for the Shapley value and this bound decreases with n. Here, it is interesting to note that a related concept for characterising the quality of an approximation is *performance ratio*. Roughly speaking, this is the ratio of an approximate solution and its exact counterpart [3]. The problem of approximating the Shapley value such that the approximation ratio is bounded by a constant is intractable unless P=NP [13]. In future, it would be interesting to obtain a similar result for the absolute error as well.

the percentage error PE_i (see Equation 22). The second reason for our focus on weighted voting games is that there are no pre-existing approximation methods for k-majority games. Hence, we can only compare the performance of these existing methods with our method for weighted voting games.

Given this, Section 6.1 describes the notation and experimental setting for the results presented in Sections 6.2 and 6.3.

6.1 Experimental Setting

We evaluate the approximation error for a range of voting games. Let $\mathcal{G}$ denote the set of all the games we consider. These are defined as follows. Recall that a weighted voting game is defined in terms of the parameters n, q, and w_i ($1 \leq i \leq n$). Also, recall that, for a given game, μ and ν denote the mean weight and the variance in weights respectively. Given this, in what follows, we consider a range of games by varying n, q, and μ such that the variance, ν, is always close to 1. In more detail, we vary the number of players between $n = 20$ (since approximation methods do not produce good results for games with a smaller number of players and, in any case, they are not necessary in such cases) and $n = 70$ (since the method in [22] becomes extremely slow for games with a larger number of players). For $20 \leq n \leq 70$, the players' weights are generated randomly in such a way that the variance in the weights is always between $\nu = 1$ and $\nu = 1.5$. Keeping the variance in this range, for each n, we generate a range of games by varying the mean weight between $\mu = 20$ and $\mu = 100$. Thus, in all, we consider 54 different cases (varying n between 20 and 70 in increments of 10, and for each n varying the mean weight between 20 and 100 in increments of 10).

The players' weights for the range of games for each of the 54 cases are shown in Figures H.1 to H.6 in Appendix H. For instance, Figures H.1(a) shows the player's weights for a range of games with $\mu = 21.1$ and $\nu = 1$. Likewise, for other figures in the appendix. For each of these 54 cases, we consider different games by varying the quota between $q > n\mu/2$ (i.e., a half majority weighted voting game) and $q > 2n\mu/3$ (i.e., a two thirds majority weighted voting game). Thus, we have a set of games for each of the 54 cases and $\mathcal{G}$ denotes the union of the sets of games for all the 54 cases).

We now introduce some notation for describing our experimental results. Recall that φ_i denotes player i's exact Shapley value. As before, for player i, let $\bar{\varphi}_i$, $e(\varphi_i)$, and PE_i denote the approximate Shapley value, the absolute error of approximation, and the percentage error respectively for our method. Analogously, let $\bar{\varphi}_i^O$, $eO(\varphi_i)$, and PE_i^O denote the approximate Shapley value, the absolute error, and the percentage error respectively for Owen's method. Then we have:

$$
\begin{aligned}
eO(\varphi_i) &= abs(\varphi_i - \bar{\varphi}_i^O) \quad \text{and} && (26)\\
PE_i^O &= eO(\varphi_i) \times 100/\varphi_i. && (27)
\end{aligned}
$$

For a game $G \in \mathcal{G}$, the average percentage error across all the players for our method is:

$$
APE_G = \frac{1}{n}\sum_{i=1}^{n} PE_i. \tag{28}
$$

For Owen's method, APE_G^O is defined analogously.

6.2 Comparison with Owen's method

Recall that Owen's method is defined in terms of a random variable given in Equations 7 and 8, while our method is defined in terms of a random variable given by the rule $\mathcal{R}_1$. Here we want to determine how this difference in the methods affects their performance in terms of their approximation errors.

For each game, $G \in \mathcal{G}$, we compare the exact Shapley value for each player with its approximate Shapley value generated by our method and by Owen's MLE method. For each game, we find the percentage error in the Shapley value for each player. Then, for each game, we find the average percentage error across all the players. Finally, for each n, we find the average percentage error across all the games.

Having outlined the experimental setting and process, we now turn to the actual results. For all $G \in \mathcal{G}$, APE_G and APE_G^O are shown in Appendix G. This appendix is comprised of 6 figures: G.1 to G.6. Each of these 6 figures is, in turn, comprised of 9 tables. Thus, in all there are 54 tables in Appendix G which correspond to the 54 cases mentioned Section 6.1. Each row of each table represents a game. Thus, $\mathcal{G}$ is the set of all the rows (i.e., games) in all the tables of Appendix G. The results of these 54 cases are shown Figures G.1 to G.6 of Appendix G. The players' weights for each of the 54 cases are shown in Figures H.1 to H.6 in Appendix H. Thus, the player's weights for Figure G.1 are as given in Figure H.1, for Figure G.2 in Figure H.2, and so on. For each game in Appendix G, the quota is given in the first column of the table, μ and ν are given at the top of the table, n is given at the bottom of the figure, and the players' weights are given in Appendix H. Note that, the games represented in a given table differ only in terms of their quotas. But the players' weights, n, μ, and ν are the same for all the games in a given table. In more detail, for each table in Appendix G, there is a corresponding graph in Appendix H that shows the players' weights. For instance, for all the games in G.1, the players' weights are as shown in H.1. Likewise, for the tables in G.2 to G.6, the players' weights are shown in H.2 to H.6 respectively.

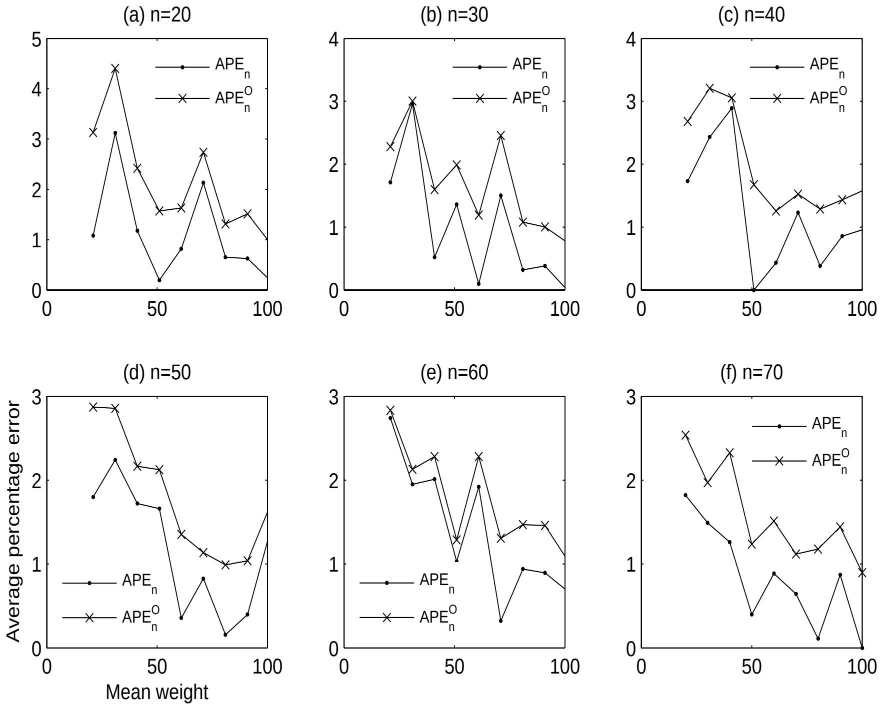

Fig. 3. A comparison of the average percentage error (for our method and Owen's) across all games with n players.

Consider the table in G.1(a) by way of explanation. For this table, $n = 20$, $\mu = 21.1$, $\nu = 1$, and the players' weights are as shown in Figure H.1(a) in Appendix H. For these fixed values of n, μ, ν, and weights, the quota is varied between 220 and 320. Likewise for each of the remaining figures in Appendix G. The last two columns of each table in Appendix G show a comparison of errors APE_G and APE_G^O. For each n, we find the average percentage error (for our method) across all the games in $\mathcal{G}$:

$$APE_n = \frac{1}{|\mathcal{G}_n|} \sum_{G \in \mathcal{G}_n} APE_G. \tag{29}$$

where $\mathcal{G}_n \subset \mathcal{G}$ denotes the set of games in $\mathcal{G}$ with n players. For Owen's method, APE_n^O is defined analogously. For different n, a comparison of APE_n and APE_n^O is shown in Figure 3. Recall that for our experiments, we vary the quota between $q > n\mu/2$ (i.e., a half majority weighted voting game) and $q > 2n\mu/3$ (i.e., a two thirds majority weighted voting game). Thus, for a given n and a given μ, the percentage error shown in Figure 3 is the average taken over this range of quotas. As seen in the figure, for each n and each μ, the error for our method is no worse than that for Owen's. These results show that, in comparison to Owen's method, our's performs better in terms of approximation error.

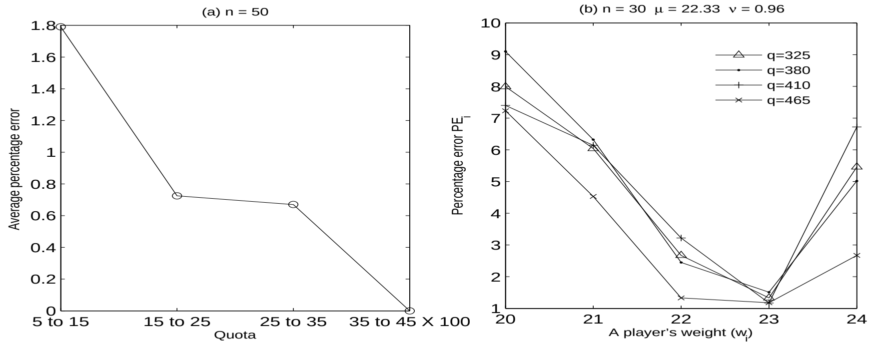

Fig. 4. The effect of the parameters of a voting game on the average percentage error.

6.3 Effect of the voting game parameters

We analyze the effect of ν, n, q, and w_i on the approximation error for our method. The effect of ν on PE_i is evident from Equation 22. As per this equation, PE_i depends on the absolute error $e(\bar{\varphi}_i)$. The error $e(\bar{\varphi}_i)$ depends on $e(E\Delta_i^X)$ (see Theorem 8), which, in turn, depends on $e(\sigma^X)$ (see Equation 19). As ν increases, $e(\sigma^X)$ increases. As a result, $e(\bar{\varphi}_i)$, and consequently PE_i increases.

Now consider the effect of n on approximation error. The error in the weights of a random coalition of size X (i.e., $e(\sigma^X)$ given in Equation 19) decreases in X and so does the error $e(E\Delta_i^X)$ (see Equation 20). Consequently PE_i decreases in n.

The effect of the other parameters (i.e, q and w_i) are not so immediately obvious from this equation for $e(\bar{\varphi}_i)$. Hence, for these, we conduct an experimental analysis.

- **Effect of** q. Recall from Section 4.1, that $a = (q - w_i)/X$ and $b = (q - \epsilon)X$. Thus, as q increases, both a and b increase. The error $e(E\Delta_i^X)$ is either the area of A or the area of C in Figure 1. As q increases, the three regions marked A, B, and C move to the right. As a result, $e(E\Delta_i^X)$ may increase or decrease depending on the coalition size X. Whether $e(E\Delta_i^X)$ increases or decreases is not obvious from Equation 20. Thus, we conduct an experimental analysis to determine the effect of q on PE_i. The setting for these experiments is as follows. We fix n and ν, and vary μ between 20 and 100. For each μ, we consider a range of games with different q. Specifically, we vary q between $n\mu/2$ (i.e., a half majority voting game) and $2n\mu/3$ (i.e., a two thirds majority voting game). For each game, we determine PE_i (where $1 \leq i \leq n$) and the average percentage error $\frac{1}{n}\sum_1^n PE_i$. We then split the set of all games into $1 \leq j \leq 4$ classes with class j containing those games for which the quota lies in the range $n\mu/2 + n\mu(j-1)/24$ to $n\mu/2 + n\mu j/24$. We then find the average percentage error for the games in each class. We then repeated the above experiments for different n in the range $20 \leq n \leq 70$. The effect of quota on the average percentage error was found to be the same for all n: the error decreased with q. The results of these experiments for $n = 50$ (corresponding to the data in the tables shown in Figure G.4) are plotted in Figure 4(a).

- **Effect of** w_i. Consider Equation 20, for which the limits of integration depend on $e(\sigma^X)$. These limits are $a = (q - w_i)/X$ and $b = (q - \epsilon)/X$. As w_i increases, a decreases but b remains unchanged. Also as w_i increases, the area of region C (see Figure 1) remains unchanged, the area of region B increases, and the area of region A may increase or decrease. The relation between this change in the areas A and B is again not obvious from the equations. Hence, we computed $e(\bar{\varphi}_i)$ for each voting game $G \in \mathcal{G}$. For each game, the error PE_i was found to increase with $abs(\mu - w_i)$. Consequently, for a given n, APE_n also increases with $abs(\mu - w_i)$. The results of experiments for $n = 30$, $\mu = 22.33$, and $\nu = 0.96$ are are plotted in Figure 4(b). For these fixed values of n, μ, and ν, and a range of different quotas, the figure shows how PE_i varies with w_i.

This analysis gives us the following key insights. First, as the number of players (n) increases, the average percentage error decreases. Second, as the quota (q) increases, the average percentage error decreases. Third, the error is different for players with different weights; players with weight closer to the mean weight (μ) have a lower error than those with weight further away from μ.

7 Conclusions and future work

The main advantage of the Shapley value as a solution concept in cooperative games is that it provides a solution that is both unique and fair. However the problem of finding this value for the voting game is #P-complete. In order to overcome this computational complexity, we presented a new approximation algorithm for computing the Shapley value for a weighted voting game. Like the current state of the art, our method has linear time complexity, but it is better in terms of its approximation error. We also empirically evaluated the error for our method and showed how the different parameters of a voting game affect it. Specifically, we showed the following effects. First, as the number of players in a voting game increases, the average percentage error across the players decreases. Second, as the quota increases, the average percentage error decreases. Third, the error is different for players with different weights; players with weight closer to the mean weight have a lower error than those with weight further away. We then extended our approximation method to the more general k-majority voting game and showed that, for n players, the method has time complexity $\mathcal{O}(k^2 n)$ and the upper bound on its approximation error is $\mathcal{O}(k^2/\sqrt{n})$.

The results are important because by devising a computationally efficient approximation method, that has a high degree of accuracy, we make it more attractive to use the Shapley value as the basis for computing solutions in a range of practical contexts. In particular, coalition formation techniques lie at the heart of virtual organizations, collective robotics, and agent teams. Thus, by providing such an algorithm, we start to open up the possibility of applying the principle approaches of cooperative game theory to such practical problems.

There are several interesting directions for future work. First, this work extended our approximation method for weighted voting games [15] to k-majority games. In future, we would like to generalize our method so that it works not just for the voting game, but also for other coalitional games. Second, although we found the upper bound on the approximation error for our method, it would be interesting to determine whether or not it is possible to find an approximate Shapley value in linear time, but with a better error bound. Third, the accuracy of our approximation method can be improved at the expense of computation time (using an approach similar to the MLE extension method). Thus, we would like to investigate how much time needs to be traded off in order to arrive at a more accurate result and whether an agent could make this tradeoff at run-time in order to reflect its current goals and available resources.

Acknowledgements

This paper is a substantially revised and extended version of our earlier work published in [15,14]. Specifically, we make the following extensions. We extend the randomised method (for a weighted voting game) presented in [15,14] to k-majority games, find the upper bounds for its approximation error, and also experimentally compare its error (for a weighted voting game) with the error for Owen's method. Finally, we show how the parameters of a voting game effect the approximation error.

We are very grateful to the anonymous reviewers of this paper. Their insightful and informative comments helped has to substantially improve the technical content and presentation of the paper.

References

[1] E. Algaba, J. M. Bilbao, J. R. Garcia, and J. J. Lopez. Computing power indices in weighted multiple majority games. *Mathematical Social Sciences*, 46:63–80, 2003.

[2] R. Aumann. Acceptable points in general cooperative n-person games. In *Contributions to the Theory of Games volume IV*. Princeton University Press, 1959.

[3] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and approximation: Combinatorial optimization problems and their approximability properties*. Springer, 2003.

[4] Y. Bachrach and J. Rosenschein. Approximating power indices. In *Proceedings of the Seventh International Conference on Autonomous Agents and Multi-Agent Systems*, 2008.

[5] J. M. Bilbao, J. R. Fernandez, N. Jimenez, and J. J. Lopez. Voting power in the european union enlargement. *Journal of Operational Research*, 143:181–196, 2002.

[6] J. M. Bilbao, J. R. Fernandez, A. J. Losada, and J. J. Lopez. Generating functions for computing power indices efficiently. *Top 8*, 2:191–213, 2000.

[7] P. Bork, H. Grote, D. Notz, and M. Regler. *Data Analysis Techniques in High Energy Physics Experiments*. Cambridge University Press, 1993.

[8] S. Brams and P. J. Affuso. Power and size: A new paradox. *Theory and Decision*, 7:29–56, 1976.

[9] V. Conitzer and T. Sandholm. Computing Shapley values, manipulating value division schemes, and checking core membership in multi-issue domains. In *Proceedings of the National Conference on Artificial Intelligence*, pages 219–225, San Jose, California, 2004.

[10] V. Conitzer and T. Sandholm. Complexity of constructing solutions in the core based on synergies among coalitions. *Artificial Intelligence Journal*, 170:607–619, 2006.

[11] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *An introduction to algorithms*. The MIT Press, Cambridge, Massachusetts, 2003.

[12] X. Deng and C. H. Papadimitriou. On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, 19(2):257–266, 1994.

[13] E. Elkind, L. A. Goldberg, P. Goldberg, and M. Wooldridge. Computational complexity of weighted threshold games. In *In Proceedings of the National Conference on Artificial Intelligence (AAAI-2007)*, 2007.

[14] S. S. Fatima, M. Wooldridge, and N. R. Jennings. An analysis of the Shapley value and its uncertainty for the voting game. In *Proc. 7th Int. Workshop on Agent Mediated Electronic Commerce*, pages 39–52, 2005.

[15] S. S. Fatima, M. Wooldridge, and N. R. Jennings. A randomized method for the Shapley value for the voting game. In *Proceedings of the Sixth International Conference on Autonomous Agents and Multi-Agent Systems*, pages 955–962, 2007.

[16] A. Francis. *Advanced Level Statistics*. Stanley Thornes Publishers, 1979. (pg 425).

[17] S. Ieong and Y. Shoham. Marginal contribution nets: A compact representation scheme for coalitional games. In *Proceedings of the Sixth ACM Conference on Electronic Commerce*, pages 193–202, Vancouver, Canada, 2005.

[18] S. Ieong and Y. Shoham. Multi-attribute coalition games. In *Proceedings of the Seventh ACM Conference on Electronic Commerce*, pages 170–179, Ann Arbor, Michigan, 2006.

[19] J. P. Kahan and A. Rapoport. *Theories of Coalition Formation*. Lawrence Erlbaum Associates Publishers, 1984.

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

[21] I. Mann and L. S. Shapley. Values for large games iv: Evaluating the electoral college by monte carlo techniques. Technical report, The RAND Corporation, Santa Monica, 1960.

[22] I. Mann and L. S. Shapley. Values for large games iv: Evaluating the electoral college exactly. Technical report, The RAND Corporation, Santa Monica, 1962.

[23] A. MasColell, M. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[24] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press, 1994.

[25] G. Owen. Multilinear extensions of games. *Management Science*, 18(5):64–79, 1972.

[26] C. H. Papadimitriou. *Computational Complexity*. Addison Wesley Longman, 1994.

[27] T. Rahwan and N. R. Jennings. An algorithm for distributing coalitional value calculations among cooperating agents. *Artificial Intelligence Journal*, 171:535–567, 2007.

[28] A. Rapoport. *N-person Game Theory: Concepts and Applications*. Dover Publications, Mineola, NY, 2001.

[29] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter*. MIT Press, 1994.

[30] A. E. Roth. Introduction to the shapley value. In A. E. Roth, editor, *The Shapley value*, pages 1–27. University of Cambridge Press, Cambridge, 1988.

[31] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohme. Coalition structure generation with worst case guarantees. *Artificial Intelligence Journal*, 111:209–238, 1999.

[32] T. Sandholm and V. Lesser. Coalitions among computationally bounded agents. *Artificial Intelligence Journal*, 94(1):99–137, 1997.

[33] L. S. Shapley. A value for n person games. In A. E. Roth, editor, *The Shapley value*, pages 31–40. University of Cambridge Press, Cambridge, 1988.

[34] O. Shehory and S. Kraus. A kernel-oriented model for coalition-formation in general environments: Implemetation and results. In *In Proceedings of the National Conference on Artificial Intelligence (AAAI-96)*, pages 131–140, 1996.

[35] O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence Journal*, 101(2):165–200, 1998.

[36] P. Tanenbaum. Power in weighted voting games. *The Mathematica Journal*, 7:58–63, 1997.

[37] J. R. Taylor. *An introduction to error analysis: The study of uncertainties in physical measurements*. University Science Books, 1982.

[38] W. V. Underhill. Finding bounds for definite integrals. *The College Mathematics Journal*, 15(5):426–429, 1984.

[39] H. S. Wilf. *Generatingfunctionology*. Academic Press, 1994.

[40] G. Zlotkin and J. Rosenschein. Coalition, cryptography, and stability: mechanisms foe coalition formation in task oriented domains. In *In Proceedings of the National Conference on Artificial Intelligence (AAAI-94)*, pages 432–437, 1994.

Appendix

A A summary of notation

N The set of players
n Number of players in N
N_{-i} The set of players excluding player i
q Quota for a game
S A subset of N
$\mathcal{S}$ The set $N - \{i\}$ where $i \in N$
s Number of players in S
$v(S)$ The value of coalition S
w_i The weight of player i
$w(S)$ The weight of coalition S
$\Delta_i v(S)$ The marginal contribution of player i to coalition S
φ_i The exact Shapley value for player i for a weighted voting game
T_i A coalition such that T_i is losing but $T_i \cup \{i\}$ is winning
t The number of players in T_i
μ The mean weight of players
ν The variance in players' weights
$E\Delta_i^X$ The approximate marginal contribution of player i to coalition X for a weighted voting game
$kE\Delta_i^X$ The approximate marginal contribution of player i to coalition X for a k-majority game
φ_i^k The exact Shapley value for player i for a k-majority game
$\bar{\varphi}_i$ Approximate Shapley value (for player i) generated by the proposed method for a weighted voting game
$\bar{\varphi}_i^k$ Approximate Shapley value (for player i) generated by the proposed method for a k-majority game
$PL_i^g(S_X)$ For game v_g and player i, the probability that the coalition S_X is losing but $S_X \cup \{i\}$ is winning
$PW_i^g(S_X)$ For game v_g, the probability that the coalition $S_X \cup \{i\}$ is winning
$e(\sigma^X)$ Absolute error in the sum of weights of a random coalition S_X of size X
$e(E\Delta_i^X)$ Absolute error in i's marginal contribution to coalition S_X for a weighted voting game

$e(kE\Delta_i^X)$ Absolute error in i's marginal contribution to coalition S_X for a k-majority game

$\bar{\varphi}_i^O$ Player i's approximate Shapley value generated by Owen's method for a weighted voting game

$e(\varphi_i)$ Absolute error in i's approximate Shapley value generated by the proposed method for a weighted voting game

$eO(\varphi_i)$ Absolute error in i's approximate Shapley value generated by Owen's method for a weighted voting game

PE_i Percentage error in i's approximate Shapley value generated by the proposed method for a weighted voting game

PE_i^O Percentage error in i's approximate Shapley value generated by Owen's method for a weighted voting game

$\mathcal{G}$ The set of all games on which the experiments are conducted

G An element of the set $\mathcal{G}$

APE_G Average percentage error in the approximate Shapley value generated by the proposed method for a weighted voting game; the average is taken over all the players in N for the game G

APE_G^O Average percentage error in the approximate Shapley value generated by Owen's method for a weighted voting game; the average is taken over all the players in N for the game G

APE_n Average percentage error in the approximate Shapley value generated by the proposed method for a weighted voting game; the average is taken over all the players over all the games in $\mathcal{G}$ for which there are n players

APE_n^O Average percentage error in the approximate Shapley value generated by Owen's method for a weighted voting game; the average is taken over all the players over all the games in $\mathcal{G}$ for which there are n players

B Proof of Theorem 7

The error ($e(E\Delta_i^X)$) is obtained by propagating the error in Equation 19 to the error in a player's approximate marginal contribution given in Equation 11. In this equation, a and b are the lower and upper limits for the approximate mean of the players' weights for a coalition of size X. Since the error in this mean is $e(\sigma^X)$, the actual values of a and b lie in the intervals $a \pm e(\sigma^X)$ and $b \pm e(\sigma^X)$ respectively. Hence, the error in Equation 11 is either the probability that the mean weight lies between the limits $a - e(\sigma^X)$ and a (i.e., the area under the curve defined by $\mathcal{N}(\mu, \frac{\nu}{X})$ between $a - e(\sigma^X)$ and a, which is the region A in Figure 1) or the probability that the mean weight lies between the limits b and $b + e(\sigma^X)$ (i.e., the area under the curve defined by $\mathcal{N}(\mu, \frac{\nu}{X})$ between b and $b + e(\sigma^X)$, which is the region C in Figure 1).

The area of region A in Figure 1 is:

$$\frac{1}{\sqrt{(2\pi\nu/X)}} \times \int_{a-e(\sigma^X)}^{a} e^{-X\frac{(x-\mu)^2}{2\nu}} dx$$

and that of C is:

$$\frac{1}{\sqrt{(2\pi\nu/X)}} \times \int_{b}^{b+e(\sigma^X)} e^{-X\frac{(x-\mu)^2}{2\nu}} dx$$

More specifically, the error $e(E\Delta_i^X)$ is at most the sum of the two areas A and C.

C Proof of Theorem 8

We obtain the error $e(\bar{\varphi}_i)$ by propagating the error $e(E\Delta_i^{X-1})$ to all coalitions between size $X = 1$ to $X = n$. This is done using the following error propagation rules [37]. Let x and y be two random variables with errors $e(x)$ and $e(y)$ respectively. Then, from [37] we have the following propagation rules :

$\mathcal{R}_2$ The error in the random variable $z = x + y$ is:

$$e(z) = e(x) + e(y)$$

$\mathcal{R}_3$ If $z = kx$ where the constant k has no error, then the error in z is:

$$e(z) = |k|e(x)$$

$\mathcal{R}_4$ The error in the random variable $z = x \times y$ is:

$$e(z) = e(x) + e(y)$$

Note that for $X = 1$ (i.e., player i is the first member of a coalition), $e(E\Delta_i^{X-1}) = 0$ since we know that a one player coalition can never win and i's marginal contribution to such a coalition is therefore known to be zero. Also, recall from Theorem 2, that a player's approximate Shapley value is the average of its approximate marginal contributions to coalitions of size $1 \leq X \leq n$. Hence, as per rules $\mathcal{R}_2$, $\mathcal{R}_3$, and $\mathcal{R}_4$, the absolute error ($e(\bar{\varphi}_i)$) is the average of the approximation errors $e(E\Delta_i^{X-1})$ for all coalitions between the sizes $X = 1$ and $X = n$.

D Proof of Theorem 9

We obtain the error $e(\bar{\varphi}_i^k)$ by propagating the error $e(kE\Delta_i^{X-1})$ to all coalitions between size $X = 1$ to $X = n$. The error $e(kE\Delta_i^X)$ is obtained using the error propagation rules $\mathcal{R}_2$, $\mathcal{R}_3$ and $\mathcal{R}_4$:

$$
\begin{aligned}
e(kE\Delta_i^X) = \sum_{j=0}^{k-1}\Bigg(&\sum_{g=1}^{j}(e(PL_i^g(S_X))) + e(PL_i^{j+1}(S_X)) + \\
&\sum_{f=j+2}^{k} e(PW_i^f(S_X))\Bigg)
\end{aligned} \tag{D.1}
$$

Given Equation D.1 and the fact that player i's approximate Shapley value is the average of its approximate marginal contribution to coalitions between size $X = 1$ and $X = n$, we get $e(\bar{\varphi}_i^k) = \frac{1}{n}\sum_{X=1}^{n} e(kE\Delta_i^{X-1})$.

E Proof of Theorem 10

We find a bound for the error in Shapley value by finding a bound for Equation 20. A bound for Equation 20 is found by first finding a bound for $\frac{1}{\sqrt{(2\pi\nu/X)}} \times \int_{a-e(\sigma^X)}^{a} e^{-X\frac{(x-\mu)^2}{2\nu}}dx$ and then for $\frac{1}{\sqrt{(2\pi\nu/X)}} \times \int_{b}^{b+e(\sigma^X)} e^{-X\frac{(x-\mu)^2}{2\nu}}dx$ and then summing them both. In order to obtain a bound on $\frac{1}{\sqrt{(2\pi\nu/X)}} \times \int_{a-e(\sigma^X)}^{a} e^{-X\frac{(x-\mu)^2}{2\nu}}dx$ we use the following rule from [38]:

> On some interval $[\bar{a}, \bar{b}]$, suppose that functions $\bar{f}$ and $\bar{g}$ are integrable, $\bar{g}$ never changes sign, and $m \leq \bar{f}(x) \leq M$. Then
>
> $$m\int_{\bar{a}}^{\bar{b}} \bar{g}(x)dx \quad \leq \quad \int_{\bar{a}}^{\bar{b}} \bar{f}(x)\bar{g}(x)dx \quad \leq \quad M\int_{\bar{a}}^{\bar{b}} \bar{g}(x)dx$$

We use this result to find the bound for $e(\bar{\varphi}_i)$ as follows. Let $h = e(\sigma^X)$, $\bar{g}(x) = 1$, $\bar{f}(x) = e^{-X\frac{(x-\mu)^2}{2\nu}}$, $\bar{a} = a - h$, and $\bar{b} = a$. This gives $\int_{a-h}^{a}\bar{g}(x)dx = h$, $m = e^{-X\frac{(a-\mu)^2}{2\nu}}$, and $M = e^{-X\frac{(a-h-\mu)^2}{2\nu}}$. Then using the above rule from [38], we get:

$$\int_{a-e(\sigma^X)}^{a} e^{-X\frac{(x-\mu)^2}{2\nu}}dx \leq he^{-X\frac{(a-h-\mu)^2}{2\nu}} \tag{E.1}$$

Since the upper bound for $e^{-X\frac{(a-h-\mu)^2}{2\nu}}$ is 1, we get:

$$
\begin{aligned}
\int_{a-e(\sigma^X)}^{a} e^{-X\frac{(x-\mu)^2}{2\nu}}dx &\leq h \\
&\leq e(\sigma^X)
\end{aligned} \tag{E.2}
$$

Also, from Equation 19, we know that $e(\sigma^X) = \sqrt{(\nu/X)}/\sqrt{(X)}$. Hence, Equation E.2 can be rewritten as:

$$\int_{a-e(\sigma^X)}^{a} e^{-X\frac{(x-\mu)^2}{2\nu}} dx \leq \sqrt{(\nu)}/X \tag{E.3}$$

It follows that

$$\frac{1}{\sqrt{(2\pi\nu/X)}} \int_{a-e(\sigma^X)}^{a} e^{-X\frac{(x-\mu)^2}{2\nu}} dx \leq \frac{1}{\sqrt{(2\pi X)}} \tag{E.4}$$

In the same way we get

$$\frac{1}{\sqrt{(2\pi\nu/X)}} \times \int_{b}^{b+e(\sigma^X)} e^{-X\frac{(x-\mu)^2}{2\nu}} dx \leq \frac{1}{\sqrt{(2\pi X)}} \tag{E.5}$$

From Equations 20, E.4, and E.5 we get:

$$e(E\Delta_i^X) \leq \frac{2}{\sqrt{(2\pi X)}} \tag{E.6}$$

Recall that $e(E\Delta_i^{X-1}) = 0$ for $X = 1$. Given this and Equations 21 and E.6, we get the bound for the error in Shapley value as:

$$\begin{aligned} e(\bar{\varphi}_i) &\leq \frac{1}{n} \sum_{X=1}^{n-1} \frac{2}{\sqrt{(2\pi X)}} \\ &\leq \frac{2}{n\sqrt{(2\pi)}} \sum_{X=1}^{n-1} \frac{1}{\sqrt{(X)}} \end{aligned} \tag{E.7}$$

Approximating summation with definite integral [11], we get the bound for $\sum_{X=1}^{n} \frac{1}{\sqrt{X}}$ as follows (see Figure 2):

$$\begin{aligned} \sum_{X=1}^{n} \frac{1}{\sqrt{X}} &\leq \int_{0}^{n} \frac{1}{\sqrt{x}} dx \\ &\leq 2\sqrt{n} \end{aligned} \tag{E.8}$$

Substituting Equation E.8 in Equation E.7, we get:

$$e(\bar{\varphi}_i) \leq 2\sqrt{\frac{2}{n\pi}} \tag{E.9}$$

In other words, $e(\bar{\varphi}_i) = \mathcal{O}(1/\sqrt{n})$. Also, from Equation E.9, it follows that as $n \to \infty$, $e(\bar{\varphi}_i) \to 0$.

F Proof of Theorem 11

From Equations 23, E.4, and E.5, we get $e(PL_i^g(S_X)) \leq \frac{2}{\sqrt{(2\pi X)}}$. And from Equations 24, E.4, and E.5, we get $e(PW^g(S_X)) \leq \frac{1}{\sqrt{(2\pi X)}}$. Substituting these two inequalities in Equation D.1, we get:

$$e(kE\Delta_i^X) \leq \frac{2k^2}{\sqrt{(2\pi X)}} \tag{F.1}$$

Substituting Equation F.1 in Equation 25, we get:

$$e(\bar{\varphi}_i^k) \leq 2k^2\sqrt{\frac{2}{n\pi}}. \tag{F.2}$$

Therefore, $e(\bar{\varphi}_i^k) = \mathcal{O}(k^2/\sqrt{n})$. It also follows that as $n \to \infty$, $e(\bar{\varphi}_i^k) \to 0$.

G Experimental Results

Results of experiments: a comparison of the error for our method and that for Owen's for the 54 cases mentioned in Section 6.2.

H Players' Weights for the Games

Data for experiments: the players' weights for the 54 cases mentioned in Section 6.2. This is data for the results given in Appendix G.

(a)	μ = 21.1	ν = 1
Quota	APE_G	APE^O_G
220	0	3.5
240	0	3.5
260	0	3.25
280	0.2	3
300	1.4	2.25
320	5	2

(b)	μ = 30.9	ν = 1.39
Quota	APE_G	APE^O_G
310	0	11
330	0	3
350	0	3
370	0	3
400	0	3
425	7	4
475	0	3

(c)	μ = 40.9	ν = 1.25
Quota	APE_G	APE^O_G
420	0	2
440	0	2
470	0	2
500	0	2
530	7.5	5
560	0	2
590	0	2
620	0	2

(d)	μ = 51.4	ν =1.24
Quota	APE_G	APE^O_G
520	1.2	0.6
600	0	1.8
650	0	1.8
700	0	1.8
750	0	1.8
800	0	1.8

(e)	μ = 60.9	ν =1.25
Quota	APE_G	APE^O_G
620	2	0.5
640	0	1.5
700	0	1.5
750	0	1.5
800	0	1.5
850	4.75	3.5
900	0	1.5
950	0	1.5

(f)	μ = 71.3	ν = 1.39
Quota	APE_G	APE^O_G
710	1	2
800	0	2.5
850	2	0
900	0	2.5
950	0	2.5
1000	15	13
1050	0	2.5
1100	0	2.5

(g)	μ =81.25	ν = 1.1
Quota	APE_G	APE^O_G
820	0	1
900	0.5	0.5
1000	0	1
1050	5	4.5
1100	0	1
1150	0	1
1200	0	1
1250	0	1

(h)	μ = 91.6	ν =1.34
Quota	APE_G	APE^O_G
910	6	5
1000	0	1
1050	0	1
1100	0	1
1150	0	1
1200	0	1
1250	0	1
1300	0	1
1350	0	1
1400	0	1

(i)	μ =101.4	ν = 1.24
Quota	APE_G	APE^O_G
1010	1.49	0.49
1100	0	1
1200	0	1
1300	0	1
1400	0	1
1500	0	1
1550	0	1

Fig. G.1. A comparison of the average percentage errors for games with 20 players.

(a)	μ =21.33	ν = 1
Quota	APE_G	APE^O_G
325	3.5	0.7
350	0	4
375	3.5	0.7
410	3.6	0.6
440	0	4
460	2	2
500	0	4

(b)	μ = 31.4	ν = 1.11
Quota	APE_G	APE^O_G
470	0	2
520	0.5	2
540	1	1
560	4	1
600	0	3
640	0.5	2.5
670	1.5	0.5
690	12.5	9
720	7	5

(c)	μ =41.2	ν = 1.23
Quota	APE_G	APE^O_G
620	1	1
700	1.75	0.25
750	0.5	2
800	0	2
850	0	2
900	0.5	1.5
950	0	2

(d)	μ =51.17	ν = 1.1
Quota	APE_G	APE^O_G
760	1.25	1.5
850	0	1.8
950	7	4
1000	0	1.8
1080	2	1.25
1150	0	2
1200	0	2

(e)	μ =61.13	ν = 1.32
Quota	APE_G	APE^O_G
920	0.5	0.6
1000	0	1.25
1100	0	1.25
1150	0.5	1.25
1225	0	1.25
1300	0	1.25
1400	0	1.25
1450	1	8

(f)	μ = 71.3	ν = 1.21
Quota	APE_G	APE^O_G
1100	0	1.5
1200	0	1.5
1300	0	1.5
1400	0	1.5
1500	10	9
1600	0	1.5
1700	0	1.5

(g)	μ = 81.2	ν =1.36
Quota	APE_G	APE^O_G
1200	0	1
1300	0	1
1400	0	1
1500	0	1
1600	2	1
1700	0	1
1800	2	7.5

(h)	μ = 91.1	ν =1.16
Quota	APE_G	APE^O_G
1400	0	1
1500	0	1
1600	0	1
1700	0	1
1800	0	1
1900	3	2
2000	0	1

(i)	μ =101.3	ν = 1.1
Quota	APE_G	APE^O_G
1500	0	0.75
1600	0	0.75
1700	0	0.75
1800	0	0.75
1900	0	0.75
2000	0	0.75
2100	0	0.75
2200	0	0.75
2300	0	0.75

Fig. G.2. A comparison of the average percentage errors for games with $n = 30$ players.

(a)	μ =21.18	ν = 1.2
Quota	APE_G	APE^O_G
425	2	1.9
475	0.2	3.9
500	0.2	3.8
550	5	1.6
600	1	3
650	1.5	2.25

(b)	μ = 31.1	ν = 1.7
Quota	APE_G	APE^O_G
640	0.3	2
700	0	2.5
750	7	2.5
800	0.2	2
850	0	2
900	9	6
950	0	3

(c)	μ =41.28	ν = 1.15
Quota	APE_G	APE^O_G
850	5	3
900	0	2
950	10	8
1000	0	2
1050	0	2
1100	0	2
1150	2	0.5
1200	8	6
1250	0	2

(d)	μ =51.1	ν = 1
Quota	APE_G	APE^O_G
1050	0	1.8
1100	0	1.8
1200	0	1.8
1300	0	1.8
1400	0	1.8
1500	0	1.8
1550	0	1.8

(e)	μ =61.2	ν = 1.16
Quota	APE_G	APE^O_G
1210	2.8	1.5
1300	0	1.5
1400	0.5	0.75
1500	0	1.5
1600	0.25	1.3
1700	0	1.5
1800	0	1.5
1850	0	1.5

(f)	μ = 71.2	ν = 1.16
Quota	APE_G	APE^O_G
1410	3	2
1450	5	4
1600	0	1
1700	0.2	1
1800	0	1
1900	0	1
2100	1.2	0.8
2200	0	1

(g)	μ = 81	ν =1.25
Quota	APE_G	APE^O_G
1750	0	1.2
1800	0	1.2
1850	0	1.2
1900	3.3	2
2050	0	1.2
2150	0	1.2
2250	0	1.2
2350	0.2	1.2
2400	0	1.3

(h)	μ = 91.3	ν =1.21
Quota	APE_G	APE^O_G
1900	0	0.5
2000	3.2	2.2
2100	5.8	5.5
2200	0	1
2300	0	1
2400	0	1
2500	0	1
2600	0	1
2700	0	1
2750	0	1

(i)	μ=101.23	ν = 1.7
Quota	APE_G	APE^O_G
2100	0	3
2200	0	3
2300	0	3
2400	0	3
2500	0	3
2600	0	3
2700	0	3
2800	0	3
2900	0	3
3000	0	3

Fig. G.3. A comparison of the average percentage error for games with $n = 40$ players.

(a)	μ =21.3	ν =1.36
Quota	APE_G	APE^O_G
550	4	4
590	2.5	3
630	0	3.5
660	3	4
720	3	4
750	0	20
820	0.5	0

(b)	μ = 31.2	ν = 1.2
Quota	APE_G	APE^O_G
800	0	3
850	1.5	1.5
900	0	3
950	0	3
1000	13	8
1050	0	3
1100	2	2.5
1150	4	2
1200	0	3

(c)	μ =41.34	ν = 1.38
Quota	APE_G	APE^O_G
1050	0	2
1100	9	6
1220	3	2
1300	0	2.5
1380	0	3
1450	0	3
1500	1	3
1600	0.5	2.5

(d)	μ =51.34	ν = 1.4
Quota	APE_G	APE^O_G
1300	0	2
1400	0	2
1500	0	2
1600	1	0.8
1700	3	2
1800	7	6
1900	0	2
2000	0	2

(e)	μ =61.14	ν = 1.28
Quota	APE_G	APE^O_G
1620	0.8	0.8
1750	0	1.7
1950	0	1.7
2050	2	0.5
2100	0	1.7
2250	0.2	1.7
2350	0	1.8

(f)	μ = 71.22	ν = 1.33
Quota	APE_G	APE^O_G
1800	0	1.2
1900	0	1.2
2000	3.8	2.5
2100	0	1
2200	1	0.5
2300	0	1
2400	0	1
2500	1.2	0.5
2600	0	1
2700	1.2	0

(g)	μ =81.44	ν =1
Quota	APE_G	APE^O_G
2050	0	1
2150	0	1
2250	0	1
2350	0	1
2450	1.8	0.8
2550	0	1
2650	0	1
2750	0	1
2850	0.1	0.9
2950	0	1
3050	0	1
3100	0	1

(h)	μ = 91.4	ν =1.4
Quota	APE_G	APE^O_G
2300	0	1
2500	0	1
2700	0	1
2900	0	1
3100	0.2	0.7
3300	2.4	1.5
3500	0	1.2

(i)	μ=101.38	ν = 1.43
Quota	APE_G	APE^O_G
2600	0	1
2750	0	1
3000	0.8	0
3200	2	1.5
3400	6	1
3600	0	1
3800	0	1
3850	0	1

Fig. G.4. A comparison of the average percentage errors for games with $n = 50$ players.

(a)	μ =21.4	ν =1
Quota	APE_G	APE^O_G
650	1	2.5
700	1.5	1.2
750	11	5
800	1	4
850	2	2
900	0.5	4
1000	1	4

(b)	μ= 31.05	ν = 1.05
Quota	APE_G	APE^O_G
900	0	2.5
1000	2.5	0.5
1050	2.3	0.5
1100	0	2.5
1150	3	2.5
1200	3	2.5
1280	3.2	2
1350	4.5	3.8
1410	0	3.8

(c)	μ =41.2	ν = 1.43
Quota	APE_G	APE^O_G
1250	0	2.5
1320	0.5	2
1420	2	0
1600	0	2.5
1700	9	6
1750	0	2.5
1800	0	2.5
1850	4	2

(d)	μ =51.25	ν = 1
Quota	APE_G	APE^O_G
1550	0	2
1620	0.2	1.8
1780	1.8	0
1820	4.4	3.4
1980	0	2.5
2050	0.5	2.5
2150	2	0.5
2250	0	1.8
2350	1	1

(e)	μ =61.3	ν = 1.24
Quota	APE_G	APE^O_G
1900	0	1
2000	7.5	6
2100	0	1
2200	0	1
2300	1	0
2400	0	1
2500	1	1
2600	0	1
2700	9	8

(f)	μ = 71.42	ν = 1.34
Quota	APE_G	APE^O_G
2150	2.5	1
2300	0	1
2450	0	1
2600	0	1
2800	0	1
2900	0	1
3100	0	1
3400	0	1

(g)	μ =81.2	ν =1.13
Quota	APE_G	APE^O_G
2450	0	1
2600	7	7.5
2750	0.5	0.8
2900	0	1
3150	0	1
3300	0	1
3450	0	1
3500	0	1
3700	0.5	0.2

(h)	μ = 91.5	ν =1.35
Quota	APE_G	APE^O_G
2750	0	1
3050	0	1
3200	0	1
3300	0	6
3500	7	1
3650	0	1
3800	1	0
3950	0	1
4100	0	1

(i)	μ=101.23	ν = 1.2
Quota	APE_G	APE^O_G
3100	0	1
3300	0.5	0.5
3500	1	0.5
3650	4	3
3800	0	1
4000	0	1
4200	0	1
4400	0	1
4600	0	1

Fig. G.5. A comparison of the average percentage errors for games with $n = 60$ players.

(a)	μ =21.24	ν =1.1
Quota	APE_G	APE^O_G
700	0.7	3
750	7	1.6
850	7	6
900	3	0.8
950	2.5	0.8
1000	2.5	0.8
1050	3	0.8
1100	3.5	0.6
1150	2	1

(b)	μ= 31.19	ν = 1.26
Quota	APE_G	APE^O_G
1100	2	1
1150	0.5	2.3
1200	2	2
1300	0	3
1400	5	2.8
1500	1	2.6
1600	1	2
1700	0.5	2

(c)	μ =41.23	ν = 1.26
Quota	APE_G	APE^O_G
1400	3	1
1500	0	2.2
1600	0.5	2
1700	0	2
1900	0	2
2000	0	2
2100	7.8	5.5
2200	0	2

(d)	μ =50.96	ν = 1
Quota	APE_G	APE^O_G
1850	0	1.5
1900	0	1.5
2000	0	1.5
2100	0.3	1
2200	0.7	0.5
2300	2.4	0.7
2400	0	1.5
2500	0	1.5
2600	0	1.5
2700	0.3	1.3

(e)	μ =61.23	ν = 1.1
Quota	APE_G	APE^O_G
2150	2.5	1
2250	0	1
2350	0	1
2450	0.3	0.8
2600	0	1
2700	0	1
3000	0.2	0.8
3100	0	0.9
3200	4.5	3.5

(f)	μ = 71.23	ν = 1.12
Quota	APE_G	APE^O_G
2500	2.5	1.5
2700	2	1
2900	0	1
3100	0	1
3300	0	1
3500	0.5	0.7
3700	0	1
3900	0	1

(g)	μ =81.16	ν =1.36
Quota	APE_G	APE^O_G
3000	0	1.4
3200	0	1.4
3400	0	1.4
3500	0.8	0.4
3700	0	1.4
3900	0	1.4
4100	0	1.4
4300	0	1.4

(h)	μ = 91.4	ν =1.13
Quota	APE_G	APE^O_G
3200	0	1
3400	0	1
3600	2	1
3800	0	1
3900	0	1
4200	0	1
4300	6.2	5.6
4400	0	1
4600	0	1
4800	0	1

(i)	μ=101.23	ν = 1.26
Quota	APE_G	APE^O_G
3600	0	0.9
3900	0	0.9
4200	0	0.9
4500	0	0.9
4800	0	0.9
5100	0	0.9
5300	0	0.9

Fig. G.6. A comparison of the average percentage errors for games with $n = 70$ players.

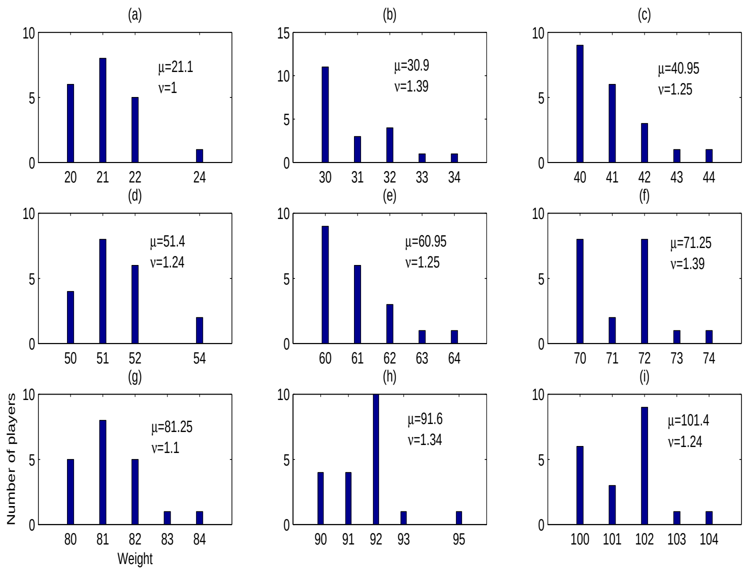

Fig. H.1. Players' weights for $n = 20$ players.

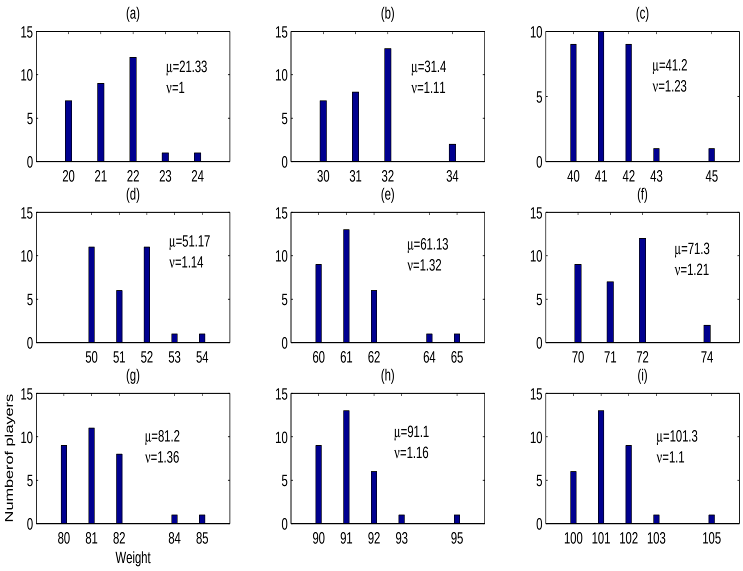

Fig. H.2. Players' weights for $n = 30$ players.

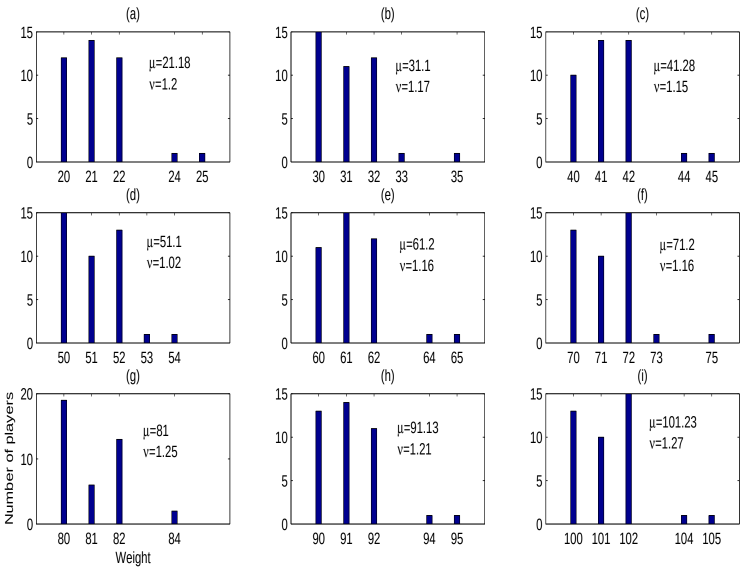

Fig. H.3. Players' weights for $n = 40$ players.

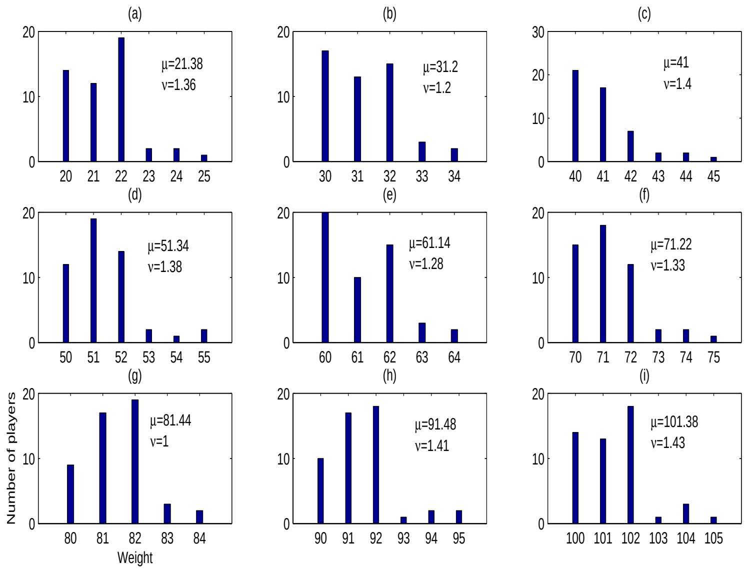

Fig. H.4. Players' weights for $n = 50$ players.

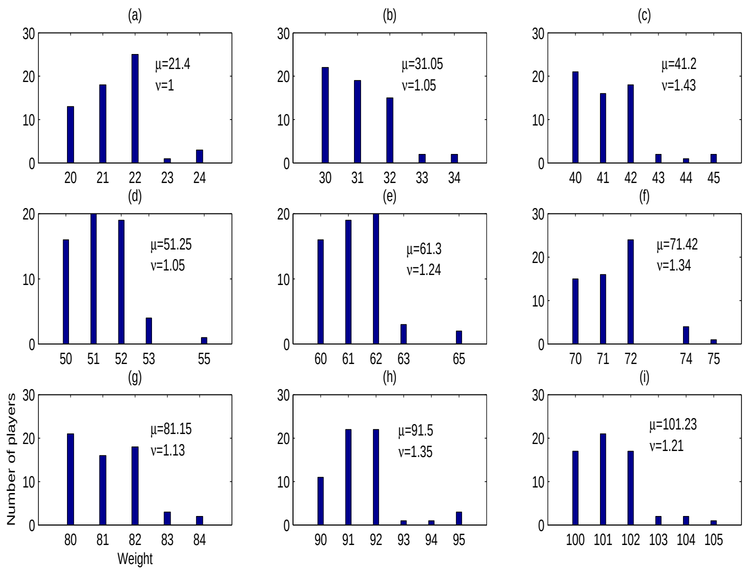

Fig. H.5. Players' weights for $n = 60$ players.

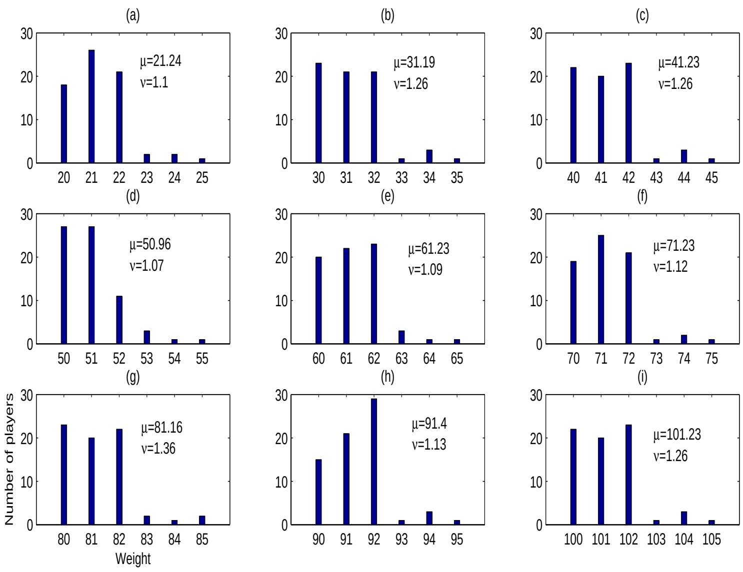

Fig. H.6. Players' weights for $n = 70$ players.