European Journal of Operational Research 279 (2019) 93–106

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Stochastics and Statistics

A stochastic approach to approximate values in cooperative games

Stefano Benati[a,*], Fernando López-Blázquez[b], Justo Puerto[b]

[a] Dipartimento di Sociologia e Ricerca Sociale, Universitá di Trento, Italy
[b] Instituto Universitario de Investigación Matemática (IMUS), Universidad de Sevilla, Spain

ARTICLE INFO

Article history:
Received 26 February 2018
Accepted 22 May 2019
Available online 25 May 2019

Keywords:
Game theory
Cooperative games
Values
Stochastic games
Statistical sampling

ABSTRACT

Computing additive values in cooperative games, like the Shapley value, is a hard task because, in general, it involves the summation of an exponential number of terms. We propose a new method, based on the stochastic approximation of deterministic games and sampling theory, to calculate a statistic estimate of these values and, at the same time, keeping under control estimation errors. We applied this technique to several well-known games and we show that in many cases we were able to improve previous results.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Let (N, v) be a cooperative game with set of players $N = \{1, \dots, n\}$ and characteristic function v with $v(\emptyset) = 0$. For simplicity throughout this paper we will assume that the characteristic function v takes values in the set of real numbers. The extension of our results to the case of vectorial characteristic functions is straightforward, see Puerto, Fernández, and Hinojosa (2008). Let G^N denote the class of N -person cooperative games. The main goal of this class of games is to allocate the grand coalition payoff $v(N)$ among the players in the game. There exist many allocation rules with a number of different properties and any of these rules may be considered as a solution concept of the game.

A well-accepted solution concept in cooperative game theory is the concept of value. A value $\alpha : G^N \to \mathbb{R}^n$ is a n-tuple of real numbers (one per agent in the grand coalition) where the ith component is the amount allocated to agent i in the game (N, v).

These values are determined by the properties required to them. Linearity is a desirable property that has been required to most of the values considered in the literature. Thus, we will be interested in studying values whose functional form fits into the following formula:

$$\alpha(\boldsymbol{v}_*) = \sum_{S \in \mathcal{U}} a(S) \circ v_*(S), \tag{1}$$

where $\circ$ is the Hadamard product of vectors, $a(S) = (a_1(S), \dots, a_n(S))^t$, $S \in \mathcal{U}$ are real n-vectors depending only on S, $\mathcal{U} = 2^N$ the family of all the coalitions of the game and $\boldsymbol{v}_* = (v_*(S))^t_{S \in \mathcal{U}}$, where $v_*(S) := (v_{1,*}(S, v), \dots, v_{n,*}(S, v))^t$ are n-vectors of real values depending on S, the characteristic function of the game and eventually on each player in the game. There are many examples of these values although the best known may be the Shapley value, the family of Semivalues (Dubey, Neyman, & Weber, 1981), the Least Squares values (Ruiz, Valenciano, & Zarzuelo, 1998) and the Multinomial values (Carreras & Puente, 2015; 2018).

- The Shapley value of the ith player, for the cooperative game (N, v), is defined as
$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{s!\,(n-s-1)!}{n!} (v(S \cup \{i\}) - v(S)), \tag{2}$$
where $|S| = s$ denotes the cardinal of the set S. The Shapley value corresponds to the form given in (1) with $a_i(S)$ equal to $\frac{s!(n-s-1)!}{n!}$, if $i \notin S$ and zero otherwise; and $v_{i,*}(S) := v(S \cup \{i\}) - v(S)$.
- The Semivalues, introduced by Dubey et al. (1981), are similar to the Shapley values but with different systems of coefficients:
$$\psi_i(v) = \sum_{S \subseteq N \setminus \{i\}} p_s (v(S \cup \{i\}) - v(S)),$$
with $p_s \geq 0$, $s = 0, \dots, n-1$, real numbers such that $\sum_{s=0}^{n-1} \binom{n-1}{s} p_s = 1$. The Semivalue ψ corresponds to (1) with $a(S) = p_s$, for $S \subset N \setminus \{i\}$; $a(S) = 0$ otherwise, and $v_{i,*}(S) := v(S \cup \{i\}) - v(S)$. Clearly the Shapley value is also a Semivalue.
- The family of Least Squares values introduced by Dragan (2006) and Ruiz et al. (1998) also fits to this framework (1).

* Corresponding author.
E-mail addresses: stefano.benati@unitn.it (S. Benati), lopez@us.es (F. López-Blázquez), puerto@us.es (J. Puerto).

https://doi.org/10.1016/j.ejor.2019.05.027
0377-2217/© 2019 Elsevier B.V. All rights reserved.

The Least Squares value for player i is defined as:

$$x_i(v) = \frac{v(N)}{n} + \frac{1}{n\kappa}(na_i^m(v) - \sum_j a_j^m(v)) \quad (3)$$

where $a_i^m(v) = \sum_{i\in S\subset N} m(s)v(S)$, $|S| = s$ and $\kappa = \sum_{s=1}^{n-1} m(s)\binom{n-2}{s-1}$. The reader is referred to (15) to check that $x_i(v)$ can expressed in the form (1).

- The mean value of the characteristic function of all coalitions containing player i,

$$\alpha_i(\boldsymbol{v}) = \frac{1}{2^{n-1}} \sum_{S:\ i\in S\subset U} v(S) \quad (4)$$

is clearly one more example of a value of the form given in (1).

In general, computing any value $\alpha(\boldsymbol{v}_*)$ of a cooperative game is #P-hard, since the number of evaluations needed to obtain it requires the evaluation of some expression for all the coalitions $S\subseteq N$ which is of the order $O(2^n)$. This issue becomes an unavoidable obstacle whenever the number of players is medium to large ($n > 30$). Examples of cooperative games with a large number of players can be found for instance in: Voting rules in the International Monetary Fund, microarrays (Moretti, 2010), neuroscience (Keinan, Sandbank, Hilgetag, Meilijson, & Ruppin, 2006), complex games as the ones defined in networks (van Campen, Hamers, Husslage, & Lindelauf, 2018), linear production games and parliaments. To avoid the difficulties of the large number of evaluations several authors have proposed different methodologies. Owen (1995) proposed a method to compute the Shapley value via a probabilistic approximation of the game by a normal distribution. A different approach, using generating functions, only valid for simple games, is given in Bilbao, Fernández, Jiménez, and López (2000). Preliminary approaches to approximate the Shapley values of cooperative games using sampling can be found in Mann and Shapley (1960), Fatima, Wooldridge, and Jenniggs (2006), Fernández, Mayor, Puerto, and Zafra (2004), Fernández and Puerto (2005), Leech (2003), and Matsui and Matsui (2000). Later, Keinan et al. (2006) apply similar techniques to estimate Shapley values of some neurocontroller games that appear in neuroscience. In Castro, Gómez, and Tejada (2009) and Castro, Gómez, Molina, and Tejada (2017) the authors revisit the above methodology and apply it to some standard cooperative games. Sampling applied to the Owen value is proposed in Saavedra-Nieves, García-Jurado, and Fiestras-Janeiro (2018).

In this paper we present a new method for approximating values of cooperative games. Rather than computing the exact values we concentrate on providing good estimations of these values with their corresponding measure of errors. To this end, we use a general stochastic approximation of a cooperative game. On this approximation, we define a probabilistic value that is seen as an estimator of the actual value of the original game. This estimator is unbiased, consistent and with controlled quadratic error. Our method provide a general methodology to calculate any value expressible as a weighted linear combination (1), and its estimated precision depends on sampling strategies, e.g, uniform, stratified, and so on, used to approximate $\alpha(\boldsymbol{v}_*)$. Our computational experiments show the generality of our approach: In some cases our theory provides new foundations to known empiric procedures, but in other cases we are able to devise new computational procedures that outperform previous approaches.

The paper is organized in six sections. After the introduction, the second section defines the stochastic approximation of a cooperative game, and proves its main statistical and distributional properties. In Section 3 we present our family of estimators and their properties. In Section 4 we show how different sampling schemes lead to specialized formulas of estimators and standard errors. Finally, in Section 5 we test the above mentioned estimators on five different well-known classes of cooperative games showing the efficiency of the method. The last section, namely Section 6 contains our conclusions.

2. The stochastic approximation of a cooperative game

The goal of this section is to define a stochastic approximation of a game (N, v) that allows us to estimate its values. The study of probabilistic or stochastic cooperative games is not new. The reader is referred to Fernández, Puerto, and Zafra (2002), Fernández and Puerto (2006), Granot (1977), Timmer (2006), Timmer, Borm, and Tijs (2003), and Timmer, Borm, and Tijs (2005) for different approaches and further analysis on this class of games.

Let $\mathcal{U}$ be the family of all the coalitions of the game and let $\mathcal{U}^\tau$ be the τ-fold cartesian product of $\mathcal{U}$. Therefore, any element $\mathbb{S}\in\mathcal{U}^\tau$, is a vector $\mathbb{S} = (S_1, \ldots, S_\tau)^t$ where each $S_i \subseteq N$. We assume further that we are given a probability distribution p on $\mathcal{U}$. Thus, for any $S\in\mathcal{U}$, $p(S)$ is the probability of choosing the coalition S. For simplicity, we will assume that $p(S) > 0$, for all $S\in\mathcal{U}$, although, as we will see later, this condition is not essential. This probability, p, induces on the space $\mathcal{U}^\tau$ the natural product probability.

We introduce a set of random vectors $\widetilde{v}_*(S) := (\widetilde{v}_{1,*}(S), \ldots, \widetilde{v}_{n,*}(S))$ associated with each coalition of the original game. For each coalition S and player $i\in N$, let us define the random variable $\widetilde{v}_{i,*}(S) := \widetilde{v}_{i,*}(S; \tau, p, \mathbb{S})$ as

$$\widetilde{v}_{i,*}(S) = \frac{M_S(\mathbb{S})}{\tau p(S)} v_{i,*}(S) \quad (5)$$

where $M_S := M_S(\mathbb{S}) = \#\{j : S_j\in\mathbb{S} \text{ and } S_j = S\}$ is the number of times that coalition S appears in the random vector of coalitions $\mathbb{S}$. Note that the random vector $\mathbb{M} = (M_S)_{S\in U}$ follows a multinomial distribution with parameters τ (number of elements in the sample $\mathbb{S}$) and $\mathbf{p} = (p(S))_{S\in\mathcal{U}}$ (vector of probabilities of selection for each possible set in the sample $\mathbb{S}$). We denote this fact by $\mathbb{M} \sim \mathcal{M}(\tau;\ \mathbf{p})$.

Theorem 1. *For all $S\in\mathcal{U}$ and $i\in N$,*

(a) $E\widetilde{v}_{i,*}(S) = v_{i,*}(S)$.

(b) $\mathrm{Var}\big(\widetilde{v}_{i,*}(S)\big) = \dfrac{1-p(S)}{\tau p(S)} v_{i,*}^2(S)$

(c) $\mathrm{Cov}(\widetilde{v}_{i,*}(S),\ \widetilde{v}_{i,*}(T)) = -\dfrac{v_{i,*}(S)\, v_{i,*}(T)}{\tau}$

(d) $\widetilde{v}_{i,*}(S) \xrightarrow{a.s.} v_{i,*}(S)$, *as* $\tau \longrightarrow \infty$.

(e) *Let* $\mathbf{v}_{i,*} = \big(v_{i,*}(S)\big)_{S\in\mathcal{U}}$ *and* $\widetilde{\mathbf{v}}_{i,*} = \big(\widetilde{v}_{i,*}(S)\big)_{S\in\mathcal{U}}$, *then* $\sqrt{\tau}\big(\widetilde{\mathbf{v}}_{i,*} - \mathbf{v}_{i,*}\big) \xrightarrow{d} \mathbf{Z} \sim \mathcal{N}_{2^n}(\mathbf{0},\ \boldsymbol{\Delta})$, *as* $\tau \longrightarrow \infty$, *where*

$$\boldsymbol{\Delta} := \mathrm{diag}\left(\frac{v_{i,*}^2(S)}{p(S)}\right)_{S\in\mathcal{U}} - \mathbf{v}_{i,*}\mathbf{v}_{i,*}^t$$

Proof. Recall that $\mathbb{M} = (M_S)_{S\in\mathcal{U}}$ follows a multinomial distribution, then for all $S\in\mathcal{U}$, $M_S \sim \mathit{Binomial}(\tau,\ p(S))$, so $EM_S = \tau p(S)$ and $\mathrm{Var}(M_S) = \tau p(S)(1-p(S))$. Also, for $S, T\in\mathcal{U}$ with $T\neq S$, $\mathrm{Cov}(M_S, M_T) = -\tau p(S)p(T)$.

(a) For all $S\subset N$,

$$E\widetilde{v}_{i,*}(S) = E\left(\frac{v_{i,*}(S)M_S}{\tau p(S)}\right) = \frac{v_{i,*}(S)}{\tau p(S)} EM_S = v_{i,*}(S).$$

(b) Using standard properties of the variance, we have

$$\mathrm{Var}(\widetilde{v}_{i,*}(S)) = \mathrm{Var}\left(\frac{v_{i,*}(S)M_S}{\tau p(S)}\right) = \left(\frac{v_{i,*}(S)}{\tau p(S)}\right)^2 \mathrm{Var}(M_S)$$
$$= \frac{1-p(S)}{\tau p(S)} v_{i,*}^2(S).$$

(c) We have,

$$\mathrm{Cov}(\widetilde{v}_{i,*}(S),\, \widetilde{v}_{i,*}(T)) = \mathrm{Cov}\left(\frac{v_{i,*}(S)M_S}{\tau\, p(S)},\, \frac{v_{i,*}(T)M_T}{\tau\, p(T)}\right) = -\frac{v_{i,*}(S)\, v_*(T)}{\tau^2 p(S)p(T)}\mathrm{Cov}(M_S,\, M_T) = -\frac{v_{i,*}(S)\, v_*(T)}{\tau}.$$

(d) It is straightforward from the fact that $M_S/\tau \xrightarrow{a.s.} p(S)$, as $\tau \uparrow +\infty$, for all $S \in \mathcal{U}$.

(e) The multinomial vector $\mathbb{M}$ is asymptotically normal, i.e., $\sqrt{\tau}\left(\frac{\mathbb{M}}{\tau} - \mathbf{p}\right) \xrightarrow{d} \mathbf{W}$, as $\tau \longrightarrow \infty$, with $\mathbf{W}$ following a 2^n-dimensional normal distribution with mean $\mathbf{0}$ and variance matrix Λ, i.e., $\mathbf{W} \sim \mathcal{N}_{2^n}(\mathbf{0},\, \Lambda)$, with $\Lambda = diag(\mathbf{p}) - \mathbf{p}\mathbf{p}^t$. On the other hand, observe that $\widetilde{\mathbf{v}}_{i,*} = \mathrm{diag}\left(\frac{v_{i,*}(S)}{\tau\, p(S)},\, S \in \mathcal{U}\right)\mathbb{M}$, then the asymptotic normality of $\widetilde{\mathbf{v}}_{i,*}$ follows from the asymptotic normality of the multinomial vector $\mathbb{M}$ and the limit covariance matrix follows from *(b)* and *(c)*.

□

In the statistical language, we may interpret $\widetilde{v}_{i,*}(S)$ as an '*estimator*' of the value $v_{i,*}(S)$ based on a sample of size τ. So, property *(a)* in Theorem 1, means that the '*estimator*' $\widetilde{v}_{i,*}(S)$ is unbiased for $v_{i,*}(S)$, for any $S \in \mathcal{U}$. Properties *(b)* and *(c)* allow us to determine variances and covariances between these '*estimators*'. Properties *(d)* and *(e)* (consistency and asymptotic normality) show how the '*estimators*' behave as the sample size increases: they approach the 'true values'. Note that here, we use '*estimator*' rather than estimator, because in the strict statistical sense an estimator must not depend on the estimated value. Anyway, we believe that this statistical interpretation is interesting as we will see in the following sections.

3. Estimation of values in a cooperative game

The main practical difficulty with quantities of the form given in (1) is that they are not easy to compute due to the fact that there are too many terms in the sum (usually 2^{n-1} or 2^n). The aim of this section is to provide approximations to these values based on the stochastic approximation to the game given in section 2.

Our approach is based on a sample of size τ, $\mathbb{S} = (S_1, \ldots, S_\tau)$ selected according to the probability scheme described in the previous section. The proposed stochastic approximation to $\alpha(\boldsymbol{v}_*)$ is given by:

$$\widetilde{\alpha(\boldsymbol{v}_*)} := \alpha\left(\widetilde{v}_*\right) = \sum_{S\in\mathcal{U}} a(S)\circ \widetilde{v}_*(S) = \sum_{S\in\mathcal{U}} \frac{a(s)\circ v_*(S)}{\tau\, p(S)} M_S = \left(\frac{1}{\tau}\sum_{S\in\mathbb{S}} \frac{a_i(S) v_{i,*}(S)}{p(S)}\right)_{i\in N}, \tag{6}$$

where $\circ$ is the Hadamard product of vectors and the last equality uses the fact that $M_S = 0$ for all $S \notin \mathbb{S}$. The advantage of using this approximation is that it involves a much smaller number of terms in the summation than the true expression of the value.

Now, we will study some properties of these approximations. In the sequel for any vector $x \in \mathbb{R}^n$, we denote $x^2 = (x_1^2, \ldots, x_n^2)^t = x \circ x$, i.e. the Hadamard product of x by itself.

Theorem 2. *The approximation $\widetilde{\alpha(\boldsymbol{v}_*)} = (\widetilde{\alpha_i(\boldsymbol{v}_*)})_{i=1}^n$ satisfies:*

(a) $E\widetilde{\alpha_i(\boldsymbol{v}_*)} = \alpha_i(\boldsymbol{v}_*)$, $\forall\, i \in N$.

(b) $Var(\widetilde{\alpha_i(\boldsymbol{v}_*)}) = \frac{1}{\tau}\left\{\sum_{S\in\mathcal{U}} \frac{a_i^2(S) v_{i,*}^2(S)}{p(S)} - \alpha_i^2(\boldsymbol{v}_*)\right\} := \frac{1}{\tau}\sigma_i^2, \forall i \in N.$

(c) $\widetilde{\alpha(\boldsymbol{v}_*)} \xrightarrow{a.s.} \alpha(\boldsymbol{v}_*)$, *as* $\tau \longrightarrow \infty$.

(d) $\sqrt{\tau}\left(\widetilde{\alpha_i(\boldsymbol{v}_*)} - \alpha_i(\boldsymbol{v}_*)\right)/\sigma_i \xrightarrow{d} Z \sim \mathcal{N}(0, 1)$, $\forall\, i \in N$.

Proof. These results are straightforward from Theorem 1 and the explicit expressions for $\alpha\left(\widetilde{v}_*\right)$ given in (6). □

We will follow the usual convention $0/0 = 0$. With this convention we can assign $p(S) = 0$ in those expressions of S with $a_i(S) = 0$. Doing so, expressions like $a(S) \circ \widetilde{v}_*(S) = M_S v_*(S)/(\tau\, p(S))$ and similar ones make sense.

Again, property *(a)* in Theorem 2 shows that our approximations are unbiased and property *(b)* means that as the sample size increases, i.e., $\tau \to \infty$, the approximation approaches (almost surely) the true value $\alpha(\boldsymbol{v}_*)$. The vector $\sigma^2 = (\sigma_1^2, \ldots, \sigma_n^2)^t$, in Theorem 2(b) is related to the precision of our approximations. Let us write

$$\sigma^2 := A_1 - A_2, \text{ where } A_1 := \sum_{S\in\mathcal{U}} \frac{a^2(S)\circ v_*(S)}{p(S)} \text{ and } A_2 := \alpha^2(\boldsymbol{v}_*),$$

and let us define

$$\widetilde{A}_1 := \frac{1}{\tau}\sum_{S\in\mathbb{S}} \frac{a^2(S)\circ v_*^2(S)}{p^2(S)}.$$

We have

$$E\widetilde{A}_1 := \frac{1}{\tau}E\left(\sum_{S\in\mathbb{S}} \frac{a^2(S)\circ v_*^2(S)}{p^2(S)}\right) = \frac{1}{\tau}E\left(\sum_{S\in\mathcal{U}} \frac{a^2(S)\circ v_*^2(S)}{p^2(S)} M_S\right) = E\left(\sum_{S\in\mathcal{U}}\left(\frac{a^2(S)\circ v_*(S)}{p(S)}\right)\circ \widetilde{v}_*(S)\right) = \sum_{S\in\mathcal{U}} \frac{a^2(S)\circ v_*(S)}{p(S)}\circ E\widetilde{v}_*(S) = A_1,$$

then the approximation $\widetilde{A_1}$ is unbiased for A_1 and as an approximation to σ^2 we propose

$$\widetilde{\sigma^2} := \widetilde{A}_1 - \widetilde{\alpha}^2(\boldsymbol{v}_*). \tag{7}$$

In practice, we should determine the sample size τ in such a way that $\widetilde{\alpha(\boldsymbol{v}_*)}$ is 'close enough' to $\alpha(\boldsymbol{v}_*)$. For that purpose, the normal approximation given in property *(d)* of the previous theorem is useful. For instance, suppose that τ will be determined under the condition that

$$P\left(\left|\widetilde{\alpha_i(\boldsymbol{v}_*)} - \alpha_i(\boldsymbol{v}_*)\right| > \varepsilon\right) \le \beta$$

where $\varepsilon > 0$ and $\beta \in (0, 1)$ are given. Using the normal approximation, we obtain

$$\tau \ge \left(\frac{\sigma_i}{\varepsilon} \quad \Phi^{-1}\left(1 - \frac{\beta}{2}\right)\right)^2 \tag{8}$$

where Φ^{-1} denotes the inverse of the distribution function of a $\mathcal{N}(0, 1)$ random variable.

The formula (8) for the sample size may have the drawback that the value of σ_i^2 is not known. The solution is that we must approximate σ_i^2 using a sample of size τ_1 (usually much smaller that τ). This smaller sample plays here the same role as the pilot samples (or training samples) frequently used in Statistics to obtain brief estimators of quantities of a secondary interest. These brief estimators are used to estimate some other quantities with more precision. As an approximation to the unknown variance, σ_i^2, we propose here the use of $\widetilde{\sigma_{i,\tau_1}^2}$, obtained from the expression given in (7) by using a pilot or training sample of size τ_1. Finally, the sample size, τ, is calculated by using (8) with σ_i replaced by its approximation $\widetilde{\sigma_{i,\tau_1}}$.

Also, the normal approximation can be used to obtain an (approximated) $(1-\beta)$- confidence interval for $\alpha(\boldsymbol{v}_*)$

$$\widetilde{\alpha(\boldsymbol{v}_*)} \pm \frac{\widetilde{\sigma}}{\sqrt{\tau}}\Phi^{-1}(1-\beta/2). \tag{9}$$

These intervals can be useful to obtain some information about the precision of the procedure used in the approximation.

4. Sampling strategies

A crucial step of the method implementation is to determine the sampling strategy, that is, values for $p(S)$ in formula (5).

We have considered the following strategies (although any other sampling mechanism can be easily accommodated):

- *Uniform sampling strategy (USS).* All the coalitions of interest have the same probability of being selected. For instance, if we are interested in all coalitions, then $p(S) = 2^{-n}$. If we are interested in all the coalitions containing a given player, then $p(S) = 2^{-(n-1)}$.
- *Weighted sampling strategy (WSS).* The coalitions are chosen in two steps. Firstly, the size of the coalition, s, is chosen at random. Then a coalition of size s is selected with uniform probability.
- *Stratified sampling strategy (SSS).* The set of all the coalitions U is partitioned in strata, where each stratum is formed by the coalitions of equal size s, e.g. $U_s = \{S : |S| = s\}$. A sample of size τ_s is selected with uniform probability from stratum U_s such that the sample of coalitions, $\mathbb{S}$, is of size $\tau = \sum_s \tau_s$. It is worth noting that stratified sampling allows, at least theoretically, to optimally allocate the sample size τ to strata, (optimally in the sense that appropriate values of τ_s can reduce the estimation standard error). See Castro et al. (2017) for details.

Next we show how sampling strategies are applied to stochastic values, obtaining as corollaries specific expression of variances and standard errors.

4.1. Estimation of the mean value

The simplest case is the estimation of (4) with uniform probability sampling in the family of sets containing the ith player. Therefore all formulas must be applied with $p(S) = 2^{-(n-1)}$, for S such that $i \in S$ and $p(S) = 0$ otherwise and the approximation proposed in (6) is:

$$\widetilde{\alpha}_i := \frac{1}{\tau}\sum_{S\in\mathbb{S}} v(S), \tag{10}$$

where $\mathbb{S} = (S_1, \ldots, S_\tau)$ is the random sample of size τ of the selected subsets. Using (7), we get

$$\widetilde{\sigma_i^2} = \frac{1}{\tau}\sum_{S\in\mathbb{S}} v^2(S) - \widetilde{\alpha}_i^{\,2}. \tag{11}$$

It is worth noting that these formulas straightforwardly correspond to the usual expression of the estimators of the mean and variance of a random variable.

4.2. Estimation of the Shapley value

The Shapley value is the most important value for cooperative games and for that reason, our main computational experiments apply the stochastic approximation to Shapley estimates. For the estimation of ith player's Shapley value, we will use two different approaches: The first relies on its original expression as a sequence average (2), the second relies on being a special case of the Least Squares formula (3).

4.2.1. Computing the Shapley value from the sequence formula

In order to apply our methodology to estimate the Shapley value using the original expressions (2), we can apply a WSS strategy. More specifically, as we will select coalitions not containing the ith player, we will choose firstly $s \in \{0, \ldots, n-1\}$, the size of the subset (possibly empty) that does not contain player-i with uniform probability $(1/n)$ and in the second stage we choose at random a subset of size s with uniform probability $(1/\binom{n-1}{s})$. Then,

$$p(S) = \frac{1}{n\binom{n-1}{s}} = \frac{s!(n-s-1)!}{n!}, \quad \text{for } S \text{ such that } i \notin S, \tag{12}$$

and $p(S) = 0$ otherwise. Then, the approximations given in (6) and (7) yield respectively,

$$\widetilde{\phi}_i = \frac{1}{\tau}\sum_{S\in\mathbb{S}} v_{*,i}(S) \quad \text{and} \quad \widetilde{\sigma_i^2} = \frac{1}{\tau}\sum_{S\in\mathbb{S}} v_{*,i}^2(S) - \widetilde{\phi}_i^{\,2}. \tag{13}$$

It is worth noting that these expressions correspond to the ones proposed in Castro et al. (2009), showing the generality of our approach.

4.2.2. Computing the Shapley value from the Least Squares formula

To estimate the Shapley value from (3), we first show how the Least Squares formula (3) corresponds to the general one (1), for which the approximation theory is developed.

Then, we will compare two implementations of the stochastic approximation, relying on different interpretations of the Least Squares formula that lead to two different sampling strategies: 1) a weighted sampling; and 2) a stratified sampling.

The Least Squares value of a cooperative game is, see Ruiz et al. (1998):

$$x_i(v) = \frac{v(N)}{n} + \frac{1}{n\kappa}(\underbrace{na_i^m(v)}_{1st\ \text{term}} - \underbrace{\sum_j a_j^m(v)}_{2nd\ \text{term}}), \quad \forall i \in N, \tag{14}$$

where $a_i^m(v) = \sum_{i\in S\subset N} m(s)v(S)$ with $|S| = s$ and $\kappa = \sum_{s=1}^{n-1} m(s)\binom{n-2}{s-1}$. Formula (14) can be rewritten as a weighted sum in the form of (1) (in which $\mathcal{U}$ are all non empty and proper subsets of N):

$$x_i(v) = \frac{v(N)}{n} + \sum_{S\in\mathcal{U}} a_i(S)v(S), \quad \forall i \in N. \tag{15}$$

In Eq. (15) the weights $a_i(S)$ are determined as follows: the values $v(S)$ appear in (14) for all $S \subset N$. If $i \in S$, the term $v(S)$ appears in the first term with weight $n\,m(s)$, then in the second term with weight $s\,m(s)$ (one for each $j \in S$). If $i \notin S$, $v(S)$ appears only in the second term with weight $s\,m(s)$. So that:

$$a_i(S) = \begin{cases} \frac{(n-s)m(s)}{n\kappa}, & i \in S \\ -\frac{sm(s)}{n\kappa}, & i \notin S. \end{cases} \tag{16}$$

The Shapley value is then obtained when:

$$m(s) = \frac{1}{n-1}\binom{n-2}{s-1}^{-1} \text{ and } \kappa = 1. \tag{17}$$

Let $\mathbb{S} = (S_1, \ldots, S_\tau)$ be a random sample of $\mathcal{U}$ of size τ, the stochastic approximation of the Shapley value is:

$$\tilde{\phi}_i(v) = \frac{v(N)}{n} + \frac{1}{\tau}\sum_{S\in\mathbb{S}} \frac{a_i(s)}{p(S)}v(S), \quad \forall i \in N. \tag{18}$$

Next, we discuss two different sampling procedures to approximate (18).

- Weighted sampling: Addends of formula (18) are sampled with different probabilities, according to their importance to determine $\tilde{\phi}_i(v)$.
- Stratified sampling: We decompose formula (18) into several addends, e.g., strata, that are separately estimated and then arithmetically aggregated.

The first procedure is the weighted sampling strategy (WSS), in which $\emptyset$ (the empty set) and the grand coalition N are excluded from the sampling scheme. The reason is that $a_i(\emptyset) = a_i(N) = 0$. So, the two stages for the selection of a coalition are: firstly, we choose the size of the subset $s \in \{1, \dots, n-1\}$ (excluding $\emptyset$ and N) with uniform probability $(n-1)^{-1}$. Secondly, we choose a subset of size s with uniform probability $\binom{n}{s}^{-1}$. Therefore:

$$p(S) = \frac{1}{(n-1)\binom{n}{s}}, \quad \text{for } S \text{ such that } i \in S, \tag{19}$$

and $p(S) = 0$ otherwise.

Therefore, from Eq. (16), taking advantage that:

$$\binom{n-2}{s-1}^{-1}\binom{n}{s} = \frac{n(n-1)}{s(n-s)} \tag{20}$$

we have that:

$$\frac{a_i(s)}{p(S)} = \begin{cases} \frac{n-1}{s}, & i \in S \\ -\frac{n-1}{n-s}, & i \notin S. \end{cases} \tag{21}$$

Then, formula (18) is:

$$\tilde{\phi}_i(v) = \frac{v(N)}{n} + \frac{1}{\tau}\left[\sum_{\substack{S\in\mathbb{S}\\ i\in S}}\left(\frac{n-1}{s}\right)v(S) - \sum_{\substack{S\in\mathbb{S}\\ i\notin S}}\left(\frac{n-1}{n-s}\right)v(S)\right], \quad \forall i \in N. \tag{22}$$

Therefore, when applying weighted sampling to the Least Squares, every time that a set S is drawn, the value $v(S)$ is used to calculate $\tilde{\phi}_i$ for every $i \in N$. But the weights $a_i(S)$ are different depending on the cases $i \in S$ or not.

The second approach is to apply stratified sampling. We start with a different Least Squares approximation, elaborating the summation (14) into separated terms $a_i^m(v)$ and observing that:

$$a_i^m(v) = \sum_{i\in S\subset N} m(s)v(S) = \sum_{s=1}^{n-1} \underbrace{\sum_{\substack{S\subset N\\ |S|=s\\ i\in S}} m(s)v(S)}_{a_{i,s}^m(v)}. \tag{23}$$

Then we apply stratified sampling to estimate $a_{i,s}^m(v)$, so that a sample set $\mathbb{S}_{i,s}$ of cardinality τ_{is} is drawn from $\mathcal{U}_{i,s}$ with uniform probability, and we have $\frac{1}{p(S)} = \binom{n-1}{s-1}$. The Shapley value is calculated by the Least Squares formula with $m(s) = \frac{1}{n-1}\binom{n-2}{s-1}^{-1}$, therefore the Shapley stochastic approximation is obtained through (23) with:

$$\widetilde{a_{i,s}^m(v)} = a_{i,s}^m(\tilde{v}) = \frac{m(s)}{p(S)\tau_{is}} \sum_{S\in\mathbb{S}_{i,s}} v(S) = \frac{1}{\tau_{is}(n-s)} \sum_{S\in\mathbb{S}_{i,s}} v(S) \tag{24}$$

We use $a_{i,s}^m(\tilde{v})$ to estimate $a_{i,s}^m(v)$, then we use the estimated value to calculate the estimated $a_i^m(\tilde{v})$, and finally we calculate the estimated $\tilde{\phi}_i$:

$$\tilde{\phi}_i(v) = \frac{v(N)}{n} + \frac{n-1}{n} a_i^m(\tilde{v}) - \frac{1}{n}\sum_{j:j\neq i} a_j^m(\tilde{v}), \quad \forall i \in N. \tag{25}$$

Regarding the standard error of the stratified sampling, we can estimate $\mathrm{Var}[a_{i,s}^m(\tilde{v})]$ using formula (7), and then, considering the linear combinations of formulas (24) and (25), we obtain:

$$\mathrm{Var}[a_i^m(\tilde{v})] = \sum_s \mathrm{Var}[a_{i,s}^m(\tilde{v})] \tag{26}$$

and:

$$\mathrm{Var}[\tilde{\phi}_i] = \left(\frac{n-1}{n}\right)^2 \mathrm{Var}[a_i^m(\tilde{v})] + \left(\frac{1}{n}\right)^2 \sum_{j\neq i} \mathrm{Var}[a_j^m(\tilde{v})]. \tag{27}$$

When implementing stratified sampling, it is worth noting that, according to Formula (24) and when we calculate $v(S)$, then this value can be used for all $i \in S \subset N$. Therefore in our implementations we fix τ_s and we will draw S from U_s. Then the actual values τ_{is}, that are necessary in (24), are sample dependent.

4.3. Improvements towards symmetry and efficiency: The projection method

It is well known that the Shapley values satisfy the property of efficiency, that is, $\sum_{i=1}^n \phi_i = v(N)$. In principle the estimated values, $\widetilde{\phi}_i$, do not satisfy the efficiency property. Instead, we have $\sum_{i=1}^n \mathbb{E}\widetilde{\phi}_i = v(N)$. Similarly, symmetry (equal Shapley values for equivalent players) is not guaranteed in our approximations due to the obvious random fluctuations in the sampling strategies. Anyway, efficiency and symmetry can be recovered after the estimation of $\widetilde{\phi} = (\widetilde{\phi}_1, \dots, \widetilde{\phi}_n)$ by solving the optimization problem

$$\begin{cases} \min_{x\in\mathbb{R}^n} \left\|x - \widetilde{\phi}\right\|^2 \\ \sum_{i=1}^n x_i = v(N) \\ x_i = x_j, \text{ if } i \text{ and } j \text{ are equivalent players} \end{cases}$$

where $\|\cdot\|$ denotes the Euclidean norm. It can be seen that the above problem is an easy convex quadratic optimization problem with linear constraints, for which solvers are available for all common platforms.

An argument in favour of the orthogonal projection is that it always improves the mean quadratic error. Indeed, let us denote by $\phi, \widetilde{\phi}$ and $\widetilde{\phi}_{ort}$, the vector of true Shapley values, the stochastic approximation and the orthogonal projection of $\widetilde{\phi}$ onto the hyperplane $\sum_{i\in N} x_i = v(N)$, (or another convenient vector space, as for instance symmetry conditions for equivalent players), respectively. Since the true Shapley value is in the hyperplane $\sum_{i\in N} x_i = v(N)$ then by the projection theorem $\|\phi - \widetilde{\phi}_{ort}\| \le \|\phi - \widetilde{\phi}\|$ and thus $\widetilde{\phi}_{ort}$ is closer to ϕ than $\widetilde{\phi}$, and, moreover, having the advantage that it is efficient. Using again the projection theorem and the property of monotony of expected values, we have: $\mathbb{E}\left(\|\phi - \widetilde{\phi}_{ort}\|^2\right) \le \mathbb{E}\left(\|\phi - \widetilde{\phi}\|^2\right)$, meaning that the orthogonal projection always improves the overall mean quadratic error.

5. Applications to some distinguished cooperative games

In this section we show how to implement our methodology to different applications. We consider the gloves game, the airport game, the voting game the linear production game and the assignment game and the different game values: the mean value (4), the Shapley value (2), and the Least Squares value (3). Finally, we apply formulas with distinct sampling strategies: uniform, WSS and SSS. To appreciate the computational time saved using our approximation, in all test we will report the sampling ratio $f = \tau/2^n$ (or $\tau/2^{n-1}$).

5.1. The gloves game

Suppose a game (N, v) with set of players $N = \{1, \dots, n_1, n_1+1, \dots, n_1+n_2\}$, where the first n_1 elements represent left hand gloves and the remaining n_2 are right hand gloves. Let $n=n_1+n_2$. For any subset $S\subseteq N$, we consider $v(S)=\min\left\{|S_{\text{left}}|,\ \left|S_{\text{right}}\right|\right\}$, where

$$S_{\text{left}}=S\cap\{1,\dots,n_1\}\,\text{and}\,S_{\text{right}}=S\cap\{n_1+1,\dots,n_1+n_2\}.$$

It can be checked that, for $i=1,\dots,n_1$ the mean value defined in (4) is

$$\alpha_i(v)=\sum_{k=0}^{\min\{n_1,n_2\}}kp_k$$

where

$$p_k=\frac{1}{2^{n-1}}\left\{\binom{n_2}{k}\sum_{j=k+1}^{n_1}\binom{n_1-1}{j-1}+\binom{n_1-1}{k-1}\sum_{j=k+1}^{n_2}\binom{n_2}{j}+\binom{n_1-1}{k-1}\binom{n_2}{k}\right\},\tag{28}$$

for $k=0,\dots,\min\{n_1,n_2\}$. A similar formula for $\alpha_i(v)$ with $i=n_1+1,\dots,n$ can be obtained by interchanging n_1 and n_2 in (28). (We assume the usual conventions $\sum_b^a\cdot=0$ and $\binom{a}{b}=0$ if $a<b$.)

Under the uniform sampling, in the set of coalitions containing the player i, the approximations given in (10) and (11) result in

$$\tilde{\alpha}_i=\frac{1}{\tau}\sum_{k=0}^{\min\{n_1,n_2\}}kW_k\quad\text{and}\quad\widetilde{\sigma_i^2}=\frac{1}{\tau}\sum_{k=0}^{\min\{n_1,n_2\}}k^2W_k-\tilde{\alpha}_i^2,$$

with W_k =*number of subsets with* $\min\left\{|S_{\text{left}}|,\ \left|S_{\text{right}}\right|\right\}=k$ *among the* τ *selected.* Note that $\boldsymbol{W}=\left(W_0,\dots,W_{\min\{n_1,n_2\}}\right)$ is a random vector following a multinomial $\mathcal{M}\left(\tau,p_1,\dots,p_{\min\{n_1,n_2\}}\right)$ distribution.

For instance, in the case $n_1=30$ and $n_2=15$ we illustrate our approximation with a sample size $\tau=1000$ for the mean value of players $i=1$ and $i=31$. The results are presented in the following table

Player	Exact (α_i)	$\tilde{\alpha}_i$	$\widetilde{\sigma_i^2}$	95%-confidence interval
1	7.4925	7.482	3.6957	(7.4782, 7.4858)
31	7.9812	7.949	3.1444	(7.9455, 7.9525)

These results show that the estimation is rather accurate even for a very small sampling fraction $f=1000/2^{44}=5.68\times10^{-11}$.

We also consider the Shapley value of this game. For $i=1,\dots,n_1$, and S such that $i\notin S$, we consider

$$v_{i,*}(S)=v(S\cup\{i\})-v(S)=\begin{cases}1, & |S_{\text{left}}|<|S_{\text{right}}|,\\ 0, & |S_{\text{left}}|\ge|S_{\text{right}}|.\end{cases}$$

The number of coalitions of size $s=|S|=\{1,\dots,n-1\}$ with $i\notin S$ and $|S_{\text{left}}|<|S_{\text{right}}|$ is $\sum_{t<s/2}\binom{n_1-1}{t}\binom{n_2}{s-t}$, then the corresponding Shapley value is

$$\phi_i(v)=\sum_{s=1}^{n-1}\frac{s!(n-s-1)!}{n!}\sum_{t<s/2}\binom{n_1-1}{t}\binom{n_2}{s-t},\quad\forall i\in N.$$

Under the sampling strategy WSS described in (12), the approximations given in (13) are

$$\tilde{\phi}_i=\frac{1}{\tau}U_i\quad\text{and}\quad\widetilde{\sigma_i^2}=\tilde{\phi}_i(1-\tilde{\phi}_i),$$

with U_i =*number of coalitions sampled for which* $|S_{\text{left}}|<|S_{\text{right}}|$ (recall that the sampling is performed in the set of coalitions not containing player i). We have that U_i is a binomial random variable, namely $U_i\sim Binomial(\tau,\ p_i)$, where $p_i=\frac{1}{n}\sum_{s=1}^{n-1}\sum_{t<s/2}\frac{\binom{n_1-1}{t}\binom{n_2}{s-t}}{\binom{n-1}{s}}$.

Again, the properties stated in Theorem 2 can be checked directly and for instance, we have

$$E\tilde{\phi}_i=p_i=\phi_i\quad\text{and}\quad Var(\tilde{\phi}_i)=\frac{1}{\tau}\phi_i(1-\phi_i,\quad\forall i\in N.$$

As a numerical example, suppose again that $n_1=30$ and $n_2=15$. Firstly, we run a pilot sample of size 500 and we obtained $\widetilde{\sigma_1^2}=0.0529$. This approximation to the variance was used in formula (8) along with the values $\epsilon=0.01$ and $\beta=0.05$, so that we obtain a sample size $\tau=2033$ ($f=2033/2^{44}=1.1510^{-10}$) and for this sample size we report the results presented in the following table

Player	Exact (ϕ_i)	$\tilde{\phi}_i$	$\widetilde{\sigma_i^2}$	95%-confidence interval
1	0.04475	0.0497	0.0472	(0.04949, 0.04991)
31	0.91050	0.9134	0.0787	(0.91363, 0.91417)

As compared with the previous estimates for the mean value, doubling the sample size, we get an improvement of the results of the standard errors of one order of magnitude.

Moreover, using the correction towards efficiency and symmetry, we solve the optimization problem

$$\min 30(x_1-\tilde{\phi}_1)^2+15(x_2-\tilde{\phi}_2)^2$$
$$30x_1+15x_2=15$$

so that the corrected values are: $\hat{x_1}=0.04458$ and $\hat{x_2}=0.09108$, which are even closer to the exact Shapley values.

5.2. The airport game

Let (N,v) be an airport game, with $N=\{1,\dots,n\}$, see Littlechild and Owen (1973). The characteristic function is defined as follows. Consider integers $0=n_0<n_1<\cdots<n_k=n$ and $0=c_0<c_1<\cdots<c_k$. For $i\in N$, define $c(i)=c_r$, if $i\in A_r:=\{n_{r-1}+1,\dots,n_r\}$. For $S\subset N$ the characteristic function is $v(S)=\max\{c(i):i\in S\}$.

Let $i\in N$ with $c(i)=c_r$, then elementary combinatorial arguments show that

$$\#\{S\subset N: i\in S\text{ and }v(S)=c_j\}=\begin{cases}2^{n_r-1}, & \text{if } j=r\\ 2^{n_j-1}-2^{n_{j-1}-1}-, & \text{if } j>r\end{cases}$$

then, the mean value, (4), is

$$\alpha_i(v)=\sum_{j=r}^{k}c_jp_j^{(i)},\tag{29}$$

with

$$p_j^{(i)}=\begin{cases}\dfrac{1}{2^{n-n_r}}, & if\, j=r\\ \dfrac{1}{2^{n-n_j}}-\dfrac{1}{2^{n-n_{j-1}}}, & if\, j>r.\end{cases}$$

Sampling τ coalitions with uniform probability in the set of coalitions containing the ith player, and using the approximations given in (10) and (11), we obtain

$$\tilde{\alpha}_i=\frac{1}{\tau}\sum_{j=r}^{k}c_jW_j^{(i)}\quad\text{and}\quad\widetilde{\sigma_i^2}=\frac{1}{\tau}\sum_{j=r}^{k}c_j^2W_j^{(i)}-\tilde{\alpha}_i^2$$

with $W_j^{(i)}$ =*number of selected coalitions with* $v(S)=c_j$ *among the* τ *selected.* Note that $\boldsymbol{W}^{(i)}=(W_r^{(i)},\dots,W_k^{(i)})\sim\mathcal{M}(\tau;p_r^{(i)},\dots,p_k^{(i)})$. It is well known, that under multinomial sampling the proportion $W_j^{(i)}/\tau$ is unbiased of its expected value, $p_j^{(i)}$, then $\tilde{\alpha}_i$ is unbiased for (29), in agreement with Theorem 2(a).

Table 1
Results for the mean value of the airport game with $\tau = 14036$.

Player	Exact (α_i)	$\widetilde{\alpha}_i$	$\widetilde{\sigma}_i^2$	95%-confidence interval
1	9.0854	9.0968	1.3322	(9.0966, 9.0969)
11	9.0854	9.0649	1.3528	(9.0647, 9.0651)
21	9.0854	9.0754	1.3277	(9.0753, 9.0756)
29	9.0854	9.0948	1.3251	(9.0947, 9.095)
35	9.0859	9.0726	1.352	(9.0724, 9.0728)
39	9.0938	9.089	1.2751	(9.0888, 9.0891)
41	9.125	9.1225	1.1196	(9.1224, 9.1227)
43	9.25	9.239	0.6977	(9.2389, 9.2391)
44	9.5	9.497	0.25	(9.4969, 9.4971)
45	10.0	10.0	0.0	(10.0, 10.0)

As a numerical example, we consider $n = 45$ players, $c_r = r$, for $r = 1, \ldots, 10$ and n_r given in the following table:

n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	n_9	n_{10}
10	20	28	34	38	40	42	43	44	45

We run a training sample of size $\tau_0 = 500$, obtaining an estimate of the variance of the mean value of the first player, $\widetilde{\sigma_1^2} = 1.1184$. This value along with $\epsilon = 0.05$ and $\beta = 0.05$ were used to determine the sample size $\tau = 14036$ ($f = 7.9810^{-10}$), see (8). For this sample size we obtain the results shown in Table 1.

Estimation of the Shapley value of the player $i \in A_r = \{n_{r-1} + 1, \ldots, n_r\}$.

Given a coalition $S \subset N \backslash \{i\}$, let us define $\ell(S) = \max\{j : S \cap A_j \neq \emptyset\}$ (for convenience, $\max \emptyset = 0$, so that $\ell(\emptyset) = 0$). Then,

$$v_{i,*}(S) = v(S \cup \{i\}) - v(S) = \begin{cases} 0, & \text{if } \ell(S) \geq r \\ c_r - c_{\ell(S)}, & \text{if } \ell(S) < r \end{cases}$$

and for $0 \leq j < r$

$$\#\{S : S \subset N \setminus \{i\},\ |S| = s \text{ and } \ell(S) = j\} = \binom{n_j}{s} - \binom{n_{j-1}}{s},\ s \geq 0,$$

(for convenience, $n_{-1} = -1$) then, after some algebra, the exact Shapley value is

$$\begin{aligned}\phi_i := \phi_i(v) &= \sum_{S \subseteq N-\{i\}} \frac{s!\,(n-s-1)!}{n!} v_{i,*}(S) \\ &= \sum_{s=0}^{n-1} \frac{s!\,(n-s-1)!}{n!} \sum_{j=0}^{r-1} \sum_{\substack{S \subset N-\{i\} \\ |S|=s \\ \ell(S)=j}} (c_r - c_j) \\ &= \sum_{s=0}^{n-1} \frac{s!\,(n-s-1)!}{n!} \sum_{j=0}^{r-1} (c_r - c_j)\left\{\binom{n_j}{s} - \binom{n_{j-1}}{s}\right\} \\ &= \sum_{s=0}^{n-1} \frac{s!\,(n-s-1)!}{n!} \sum_{j=0}^{r-1} \binom{n_j}{s}(c_{j+1} - c_j), \quad \forall i \in N.\end{aligned}$$

For the estimation of ϕ_i, for $i \in A_r$, we can take advantage of the fact that $v_{i,*}(S) = 0$ if $\ell(S) \geq r$, so that instead of the sampling scheme given in (12), we propose the following probability on the set of coalitions not containing player i,

$$p(S) = \begin{cases} \dfrac{1}{(n_{r-1}+1)\binom{n_{r-1}}{s}}, & \text{if } \ell(S) \leq r-1 \text{ and } |S| = s \in \{0, \ldots, n_{r-1}\} \\ 0, & \text{otherwise} \end{cases} \tag{30}$$

which can be seen as a two stages sampling such that in the first stage the size of the coalition (not containing player i), s, is selected with equal probability among $\{0, \ldots, n_{r-1}\}$ and in the second stage a coalition is selected with uniform probability within the set of coalitions of size s with $\ell(S) \leq r-1$. Define the random variables $W_{s,j}$ =*number of coalitions with* $\ell(S) = j$ *and* $|S| = s$ *among the* τ *selected*. Then, the approximations given in (6) and (7) yield, respectively

$$\widetilde{\phi}_i = \frac{(n_{r-1}+1)!}{\tau\, n!} \sum_{s=0}^{n_{r-1}} \frac{(n-s-1)!}{(n_{r-1}-s)!} \sum_{j=0}^{r-1} (c_r - c_j) W_{s,j},$$

$$\widetilde{\sigma_i^2} = \frac{1}{\tau}\left(\frac{(n_{r-1}+1)!}{n!}\right)^2 \sum_{s=0}^{n_{r-1}} \left(\frac{(n-s-1)!}{(n_{r-1}-s)!}\right)^2 \sum_{j=0}^{r-1} (c_r - c_j)^2 W_{s,j} - \widetilde{\phi}_i^2.$$

Table 2
Results for the Shapley value of the airport game with $\tau = 278581$.

Player	Exact (ϕ_i)	$\widetilde{\phi}_i$	$\widetilde{\sigma}_i^2$	99%-confidence interval
1	0.0222	0.0222	0.0	(0.0222, 0.0222)
11	0.0508	0.0509	0.0195	(0.0509, 0.0509)
21	0.0908	0.0915	0.0918	(0.0915, 0.0915)
29	0.1496	0.1492	0.244	(0.1492, 0.1493)
35	0.2405	0.2402	0.5169	(0.2402, 0.2402)
39	0.3834	0.3842	0.952	(0.3841, 0.3842)
41	0.5834	0.5827	1.5167	(0.5826, 0.5827)
43	0.9167	0.918	2.4164	(0.918, 0.918)
44	1.4167	1.4116	3.3844	(1.4116, 1.4116)
45	2.4167	2.4142	4.2718	(2.4142, 2.4142)

Similarly as in the previous example, we run a training sample of size $\tau_0 = 1000$ and obtained $\widetilde{\sigma_{45}^2} = 2.341$. For $\beta = 0.01$ and $\epsilon = 0.01$, we determined the sample size $\tau = 278581$ ($f = 1.5810^{-8}$). For this sample size we obtained the results reported in Table 2.

5.3. The voting game

The voting game is defined on a set N of $|N| = n$ players, characterized by voting weights $w_i, i = 1, \ldots, n$, and by the quota q necessary to approve an act. Therefore the characteristic function is $v(S) = 1$, if $\sum_{i \in S} w_i \geq q$, $v(S) = 0$ otherwise. The Shapley value is used in this framework to characterize players' voting power. We will estimate the Shapley value through the Least Squares formula, comparing weighted (WSS) and stratified (SSS) sampling first, and then comparing our results with the ones obtained by the algorithm *Two-Stage-St-ApproShapley-opt* proposed in Castro et al. (2017).

We can see that in this game Eq. (24) can be re-interpreted. Let $p_{i,s}$ be the probability that $v(S) = 1$ for S such that $i \in S$ and $|S| = s$, then:

$$\widetilde{a_{i,s}^m(v)} = \frac{1}{(n-s)} \underbrace{\left[\frac{1}{\tau_{is}} \sum_{S \in \mathbb{S}_{i,s}} v(S)\right]}_{\tilde{p}_{i,s}} \tag{31}$$

in which $\tilde{p}_{i,s}$ is the usual estimator of $p_{i,s}$. To calculate variances, necessary to obtain the estimator of standard errors from Eqs. (26) and (27), we can readily see that:

$$\mathrm{Var}[\widetilde{a_{i,s}^m(v)}] = \frac{1}{(n-s)^2} \frac{p_{i,s}(1-p_{i,s})}{\tau_{is}}. \tag{32}$$

5.3.1. An algorithmic framework for the application of the stratified sampling strategy in voting games

We will see that using the Least Squares formula of the Shapley value (25), with stratified sampling strategy outperforms previous approximations suggested so far. For this reason, this subsection describes an efficient implementation to compute the Shapley value using this sampling strategy. The pseudocode of this method is reported in Algorithm 1.

Algorithm 1: Estimating the Shapley value of the Voting Game.

Input: A Voting Game $G = \{q; w_1, w_2, \dots, w_n\}$, τ = the maximum number of $v(S)$ evaluation.
Output: $\tilde{\phi}_i, i = 1, \dots, n$: ith player's estimated Shapley value, $\tilde{\sigma}_i$, $i = 1, \dots, n$, the standard errors of $\tilde{\phi}_i$.

```
1  k_1 ← max{|S| : v(S) = 0 for all S} + 1
2  k_2 ← min{|S| : v(S) = 1 for all S} − 1
3  range ← k_2 − k_1 + 1
4  m_s^exp ← τ/(2 ∗ range)
5  for s ← k_1 to k_2 do
6      for i ← 1 to n do
7          v_is ← 0
8          τ_is ← 0

9  for s ← k_1 to k_2 do
10     for t ← 1 to m^exp do
11         S ← Sample(U_s)
12         for i ∈ S do
13             v_is ← v_is + v(S)
14             τ_is ← τ_is + 1

15 for s ← k_1 to k_2 do
16     for i ∈ N do
17         p̃_is ← v_is/τ_is
18         var[ã_is] ← (n − s)^−2 p̃_is (1 − p̃_is)

19 m_s^st ← Allocate(τ/2, Var[ã_is])
20 for s ← k_1 to k_2 do
21     for t ← 1 to m_s^st do
22         S ← Sample(U_s)
23         for i ∈ S do
24             v_is ← v_is + v(S)
25             τ_is ← τ_is + 1

26 for s ← k_1 to k_2 do
27     for i ∈ S do
28         p̃_is ← v_is/τ_is
29         ã_is = (n − s)^−1 p̃_is
30         var[ã_is] ← (n − s)^−2 p̃_is (1 − p̃_is)/τ_is

31 for s ← 1 to k_1 − 1 do
32     for i ← 1 to n do
33         p̃_is ← 0
34         ã_is ← 0
35         Var[ã_is] ← 0

36 for s ← k_2 + 1 to n − 1 do
37     for i ← 1 to n do
38         p̃_is ← 1
39         ã_is ← 1/(n − s)
40         Var[ã_is] ← 0

41 for i ← 1 to n do
42     ã_i ← Σ_{i=1}^{n−1} ã_is
43     Var[ã_i] ← Σ_{i=1}^{n−1} Var[ã_is]
44     φ̃_i ← 1/n + (1/n)(n ã_i − Σ_{i=1}^{n} ã_i)
45     Var[φ̃_i] = ((n−1)/n)^2 Var[ã_i] + (1/n)^2 (Σ_{j=1; j≠i}^{n} Var[a_j])
46     σ̃_i = Var[φ̃_i]^{1/2}
47     φ̃_i ← Project(φ̃_i)

48 return φ̃_i, σ̃_i for i = 1, …, n
```

We describe in the following the details of that implementation. Lines 1 and 2 are preliminary steps in which we are taking advantage that in voting games there are two indexes, k_1 and k_2, such that $v(S) = 0$ for all S, $|S| < k_1$, and $v(S) = 1$ for all S, $|S| > k_2$. In those cases the estimate of $p_{i,s}$ is trivial and no variance is to be accounted for. For $k_1 \le |S| \le k_2$, the necessary data structures are initialized in Lines 5–8.

Then we begin to sample strata U_s, see line 11 and line 22. When using stratified sampling, and with the aim of reducing the standard error, it is suggested to allocate a different number of sample units to each stratum, with more units allocated to the strata with the greatest variability, see Maleki, Tran-Thanh, Hines, Rahwan, and Rogers (2014) and Castro et al. (2017). We implement this observation dividing the estimate into two blocks. In the first block, beginning in line 9, strata U_s are sampled evenly. In the second block, beginning in line 20, more sample units are assigned to those strata with the greatest variances, according to the subroutine of line 19, that will be explained later. In each block, the sample size is one half the total size, as can be seen in line 4 and the input of subroutine *Allocate* in line 19.

In the first block, lines 9–14 are the sampling process and lines 15–18 are the pilot variances estimates. Note that in line 11, S is sampled from U_s, namely the family of cardinality s subsets, and that in line 13 the characteristic function $v(S)$ is used for all $i \in S$, controlling values τ_{is} in Line 14. When we look at the second block, we can recognize that lines 19–25 continue the sample process, lines 26–30 compile data structures for the case $k_1 \le s \le k_2$, and lines 31–35 and 36–40 compile data structures for the case $s < k_1$ and the case $s > k_2$, respectively. Finally, lines 41–46 are devoted to compute the estimate of the Shapley value $\tilde{\phi}_i$ and its standard error, $\tilde{\sigma}_i$ for all i. The output can be further refined through the quadratic projection of Line 45, described in Section 4.3. The algorithm has been coded in R, sampling in line 11 and 22 is done through the R subroutines.

Procedure *Allocate*, in line 19, implements the optimal sample size allocation described in Castro et al. (2017). It takes as input the pilot variances calculated in Line 18 and the remaining sample size $\tau/2$. The outputs are the optimal sample sizes m_s^{st} allocated to each stratum s. Note that we control values τ_s, not $\tau_{i,s}$ as in Castro et al. (2017), therefore, to apply the procedure described in Proposition 3.1 of Castro et al. (2017), we use the strata variability $\tilde{\sigma}_s^2 = \sum_i \mathrm{Var}[\tilde{a}_{is}]$.

5.3.2. Computational experiments on voting games

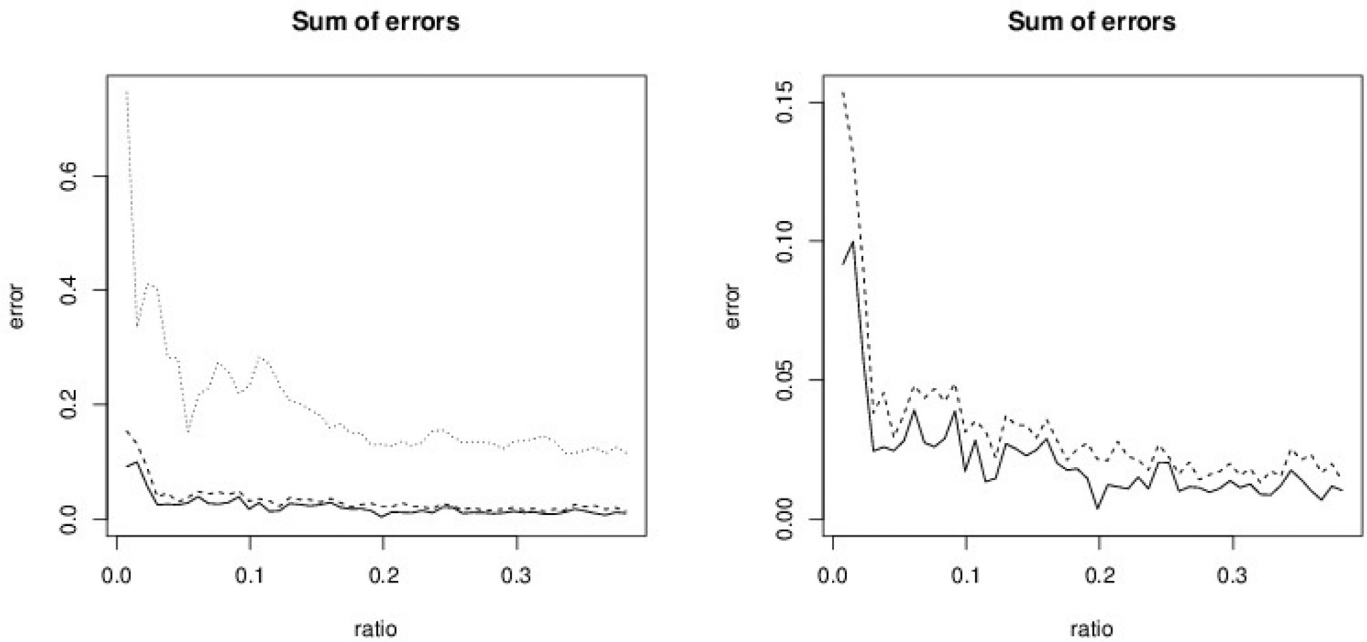
In what follows, we report on the two computational experiments that we have performed based on voting games. The first experiment considers a small size game with $n = 17$ players. Let

$$G = \{\text{quota} = 45;\ w = [11, 11, 9, 9, 8, 8, 5, 5, 4, 4, 3, 3, 3, 1, 1, 1, 1]\}.$$

As the number of players of this game is relatively small, we can calculate the exact Shapley value ϕ_i for all $i = 1, \dots, n$. In the first experiment we compare three algorithms: WSS and two versions of SSS, with and without the projection of Line 47 in Algorithm 1. Their performance is compared using the sum of absolute differences between true and estimated values: $error := \sum_{i=1}^{n} |\tilde{\phi}_i - \phi_i|$. Controlling for the sampling ratio $f = \tau/2^n$, the results are reported in Fig. 1. It can be seen that the SSS approximation error is much better than the one of WSS, and this result is confirmed in many other tests not reported here. Moreover, we zoom Fig. 1(a) in Fig. 1(b) to appreciate the effect of projection: On average, it decreases the approximation error by more than 20%.

Next we focus on SSS algorithm, applied to the same small voting game. In Table 3, we report an example of the computational results, calculated with parameter $\tau = 5000$ ($f = 0.076$). We compare three approximations: the first one is the plain SSS (without sample allocation and optimal projection); the second one is the SSS with optimal sample allocation; and the third one is SSS with optimal sample allocation and quadratic projection. Columns headed Error-(1/2/3) report the absolute differences between the true and estimated Shapley values, columns headed SD-(1/2) report the standard errors. We observe that estimates improve as we incorporate optimal sampling and projection. The average of the errors decreases around 17% when using optimal sampling; then after projection an additional gain of 20% is obtained. We can also see that the optimal sampling decreases the standard errors 9% on average. We repeat the test with other values of τ and for other voting games, finding that the outcomes are always very similar: Both subroutines have a positive effect, they both decreases the average error by at least 15% each, while optimal sampling decreases the average error too, with standard errors lowering of some 8–10%.

Our next experiment compares Algorithm 1 with the *Two-Stage-St-ApproShapley-opt* proposed in Castro et al. (2017), applied to the large voting game described in Owen (1995). The game is defined for $n = 51$ players and the exact Shapley values are available from the literature. The previous approach consists of approximating



(a) : Methods Comparison. (b) : Zoom of Figure (a)

Fig. 1. Sum of approximation errors of the small voting game. Dotted line: Weighted sampling (WSS), Broken line: Stratified sampling (SSS), Continuous line: Stratified sampling and quadratic projection.

Table 3
Exact and approximate Shapley values of the voting game, $n = 17$ and $\tau = 5000$.

Player	Weight	Exact	Error-1	Error-2	Error-3	SD-1	SD-2
1	11	0.132334	0.010916	0.002092	0.002522	0.005736	0.005259
2	11	0.132334	0.005683	0.002951	0.002522	0.005730	0.005083
3	9	0.105231	0.000922	0.007886	0.004111	0.005927	0.005305
4	9	0.105231	0.002580	0.000335	0.004111	0.005823	0.005067
5	8	0.092347	0.002371	0.001957	0.000129	0.005729	0.005343
6	8	0.092347	0.003773	0.002215	0.000129	0.005823	0.005331
7	5	0.055663	0.003794	0.009373	0.006787	0.005974	0.005348
8	5	0.055663	0.006420	0.004201	0.006787	0.005882	0.005412
9	4	0.044042	0.001207	0.002946	0.001609	0.005981	0.005398
10	4	0.044042	0.004047	0.000271	0.001609	0.005957	0.005306
11	3	0.032717	0.005768	0.005050	0.002458	0.005863	0.005459
12	3	0.032717	0.001003	0.001224	0.002458	0.005896	0.005440
13	3	0.032717	0.004767	0.001100	0.002458	0.005891	0.005345
14	1	0.010653	0.003417	0.001616	0.001181	0.005747	0.005429
15	1	0.010653	0.002225	0.006675	0.001181	0.005797	0.005363
16	1	0.010653	0.001750	0.003140	0.001181	0.005748	0.005221
17	1	0.010653	0.003519	0.000428	0.001181	0.005657	0.005462
Average	–	–	0.003774	0.003145	0.002495	0.005833	0.005328

the Shapley value through the average of players' marginal contributions to sequences. It is worth noting that to calculate the marginal contribution of one sample unit, the algorithm *Two-Stage-St-ApproShapley-opt* requires the evaluation of two values of $v(\cdot)$, namely $v(S)$ and $v(S \cup \{i\})$, while in Algorithm 1 when $v(S)$ is calculated, it contributes to the values of $p_{i,S}$ for all $i \in S$. Therefore we could argue that information is better exploited in the latter case, and it is expected that the Least Squares approximation should be better than *Two-Stage-St-ApproShapley-opt*. In Table 4, the two methods are compared, varying the sampling ratio $f = \tau/2^n$. We take again the absolute differences between Shapley values, now considering the mean and the maximum of these errors. Superscript $mean^{SEQ}$ stands for the results contained in Castro et al. (2017). Superscript $mean^{LS}$ stands for SSS followed by quadratic projection. Finally, superscript and subscript $mean_o^{LS}$ stands for SSS with optimal sample size and quadratic projection. Digits are multiplied 10^3, as in Castro et al. (2017).

Table 4
SEQ: results provided by Castro et al. (2017), LS: our results.

τ/n	10^3	10^4	10^5	10^6
$mean^{SEQ}$	2.608	0.836	0.223	0.061
max^{SEQ}	8.468	2.301	0.737	0.183
$mean^{LS}$	0.613	0.130	0.056	0.015
max^{LS}	1.916	0.583	0.628	0.068
$mean_o^{LS}$	0.461	0.177	0.043	0.017
max_o^{LS}	1.546	0.628	0.327	0.061

The reported results show that the Least Squares methods consistently outperform *Two-Stage-St-ApproShapley-opt*, as the order of magnitude of both error *mean* and *max* are different. Regarding the gains that can be obtained from the optimal allocation of the sample size, it can be seen that most of the time it actually improves the estimates: *mean* and *max* usually decrease, but not always, see the case of ratio 10^4. The reason to this is that, unreported in the table, the standard errors always decrease between 10 and 20% (thanks to the optimal size allocation), but the variability remains: A point estimate $\tilde{\phi}_i$ can get worse even though its standard error is lower.

5.4. The production game

Linear production games were introduced by Owen (1995). Each agent $i \in N$ owns a resource bundle $b_i \in \mathbb{R}^q_+$. The resources are used to produce m types of goods according to some technological constraint matrix $A \in \mathbb{R}^{q\times m}$. Goods are then sold at prices $p_1, \ldots, p_m$. When a coalition S is formed, its members pool their resources $b(S) = \sum_{i\in S} b_i$ to maximize the market value of their products. This class of games has become important among game theoretician by its implications in the cost sharing field and its equivalence to the class of all balanced games. The game characteristic function is:

$$v(S) = \max\left\{\sum_{j=1}^{m} p_j x_j :\ Ax \le b(S),\ x = (x_1, \ldots, x_m) \ge 0\right\}. \tag{33}$$

Preliminary experiments showed that, as in the Voting game, SSS is better than WSS and therefore we focus our analysis on the former. When applying SSS, Eq. (24) can be re-interpreted. Let $\mu_{i,s} = \binom{n-1}{s-1}^{-1}\sum_{S\in U_{is}} v(S)$ be the arithmetic mean of $v(S)$, for all S such that $i \in S$, $|S| = s$. Then:

$$\widetilde{a^m_{i,s}(v)} = \frac{1}{(n-s)}\underbrace{\left[\frac{1}{\tau_{is}}\sum_{S\in\mathbb{S}_{i,s}} \upsilon(S)\right]}_{\tilde{\mu}_{i,s}} \tag{34}$$

in which $\tilde{\mu}_{i,s}$ is the usual estimator of $\mu_{i,s}$. To calculate variances and standard errors of Eqs. (26) and (27), we can readily see that:

$$\mathrm{Var}[\widetilde{a^m_{i,s}(v)}] = \frac{1}{(n-s)^2}\frac{\left(\tau_{is}^{-1}\sum_{S\in\mathbb{S}_{i,s}} \upsilon(S)^2 - \tilde{\mu}^2_{i,s}\right)}{\tau_{is}}. \tag{35}$$

5.4.1. An algorithmic framework for the application of the stratified sampling strategy in linear production games

This subsection describes an efficient implementation to compute the Shapley value using this sampling strategy. The pseudocode of this method is reported in Algorithm 2. We describe in the following the details of that implementation.

Eqs. (34) and (35) are used in the SSS code, described in Algorithm 2. In line 1–5 some data structures are initialized. Next, in lines 6–11 we take advantage that the size of some strata U_s is so small that complete enumeration is better than sampling (it happens so when the stratum sample size τ/n is bigger than the cardinality of U_s). Evaluations of $v(S)$ by complete enumeration decrease the sample size τ in Line 12. Then, in lines 13, the remaining τ is allocated evenly between the rest of the strata. The sampling process is described in lines 14–20. It can be seen that, as in the Voting Game, every time that one value $v(S)$ is calculated, it is used to update data structures for all $i \in S$. In lines 21–28, values $\tilde{a}_{is}$ are calculated (note that we do not need to calculate their variability when they are calculated by complete enumeration: compare lines 21–25 with lines 26–28). Lines 29–35 compute the outputs $\tilde{\phi}$ and $\tilde{\sigma}_i$. With respect to the Voting game application, it is worth to note that Algorithm 2 does not include the optimal sample allocation. As a matter of fact, we implemented that too, but improvements were negligible.

Algorithm 2: Estimating the Shapley value of the Production Game.

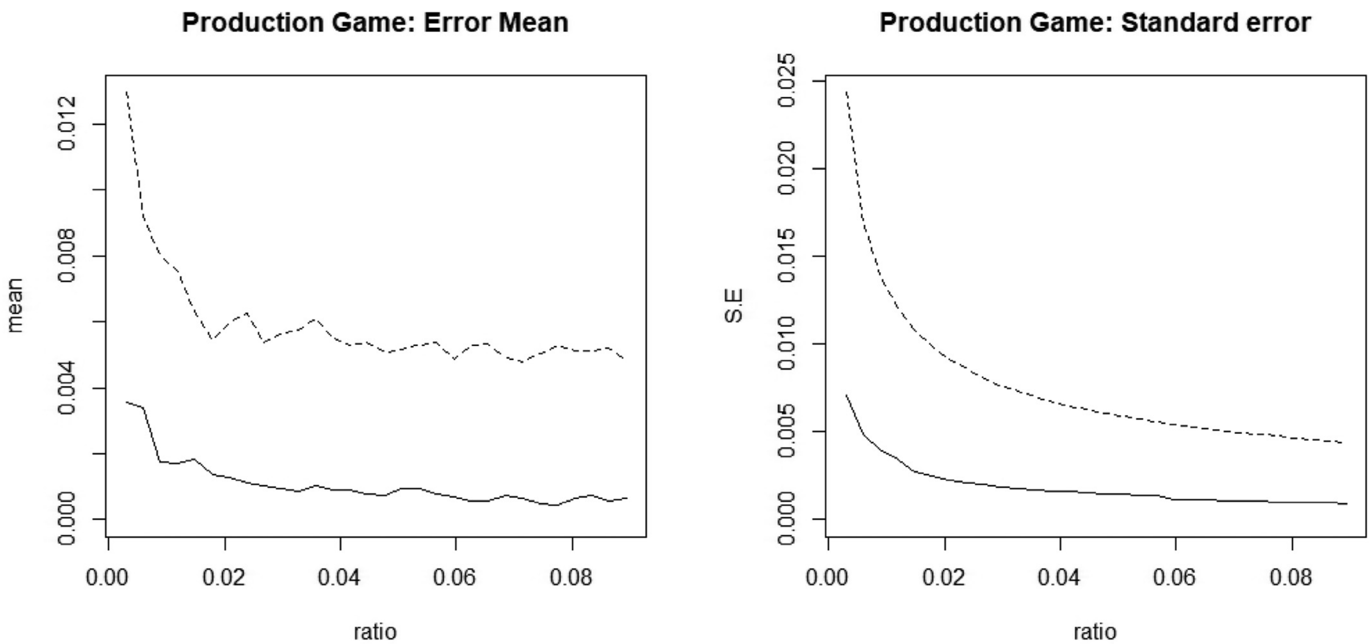
Input: A Production Game $A \in \mathbb{R}^{q\times m}, B \in \mathbb{R}^{q\times n}, p \in \mathbb{R}^m$, $\tau =$ the maximum number of $v(S)$ evaluation.
Output: $\tilde{\phi}_i, i = 1, \ldots, n$: ith player's estimated Shapley value, $\tilde{\sigma}_i, i = 1, \ldots, n$, the standard errors of $\tilde{\phi}_i$.

1 **for** $s \leftarrow 1$ ***to*** $n-1$ **do**
2   **for** $i \leftarrow 1$ ***to*** n **do**
3         $v_{is} \leftarrow 0$
4         $v^2_{is} \leftarrow 0$
5         $\tau_{is} \leftarrow 0$
6 $k_1 \leftarrow \max\{s : \binom{n}{s} \le \frac{\tau}{n-1}, s \le \frac{n}{2}\}$
7 $k_2 \leftarrow \min\{s : \binom{n}{s} \le \frac{\tau}{n-1}, s \ge \frac{n}{2}\}$
8 **for** $s \leftarrow 1$ ***to*** k_1 ***and*** $s \leftarrow k_2$ ***to*** $n-1$ **do**
9   **for** ***all*** $S \in U_s$, $i \in S$ **do**
10         $v_{is} \leftarrow v_{is} + v(S)$
11         $\tau_{is} \leftarrow \tau_{is} + 1$
12 $\tau \leftarrow \tau - \sum_{s\le k_1}\binom{n}{s} - \sum_{s\ge k_2}\binom{n}{s}$
13 $\tau_s \leftarrow \frac{\tau}{k_2-k_1-1}$
14 **for** $s \leftarrow k_1+1$ ***to*** k_2-1 **do**
15   **for** $t \leftarrow 1$ ***to*** τ_s **do**
16         $S \leftarrow$ **Sample**(U_s)
17   **for** $i \in S$ **do**
18             $v_{is} \leftarrow v_{is} + v(S)$
19             $v^2_{is} \leftarrow v^2_{is} + v(S)^2$
20             $\tau_{is} \leftarrow \tau_{is} + 1$
21 **for** $s \leftarrow k_1+1$ ***to*** k_2-1 **do**
22   **for** $i \leftarrow 1$ ***to*** n **do**
23         $\tilde{a}_{is} \leftarrow (n-s)^{-1}\,[v_{is}/\tau_{is}]$
24         $\tilde{a}^2_{is} \leftarrow v^2_{is}/\tau_{is}$
25         $\mathrm{Var}[\tilde{a}_{is}] \leftarrow (n-s)^{-2}\left[(\tilde{a}^2_{is} - (\tilde{a}_{is})^2)/\tau_{is}\right]$
26 **for** $s \leftarrow 1$ ***to*** k_1 ***and*** $s \leftarrow k_2$ ***to*** $n-1$ **do**
27   **for** $i \leftarrow 1$ ***to*** n **do**
28         $\tilde{a}_{is} \leftarrow (n-s)^{-1}\,[v_{is}/\tau_{is}]$
29 **for** $i \leftarrow 1$ ***to*** n **do**
30     $\tilde{a}_i \leftarrow \sum_{i=1}^{n-1}\tilde{a}_{is}$
31     $\tilde{\phi}_i \leftarrow v(N)/n + (1/n)(n\,\tilde{a}_i - \sum_{i=1}^{n}\tilde{a}_i)$
32     $\mathrm{Var}[\tilde{a}_i] \leftarrow \sum_{i=k_1+1}^{k_2-1}\mathrm{Var}[\tilde{a}_{is}]$
33     $\mathrm{Var}[\tilde{\phi}_i] = (\frac{n-1}{n})^2\mathrm{Var}[\tilde{a}_i] + (\frac{1}{n})^2(\sum_{j=1;j\ne i}^{n}\mathrm{Var}[\tilde{a}_j])$
34     $\tilde{\sigma}_i = \mathrm{Var}[\tilde{\phi}_i]^{1/2}$
35 **return** $\tilde{\phi}_i, \tilde{\sigma}_i$ ***for*** $i = 1, \ldots, n$

5.4.2. Computational experiments on linear production games

In the following we report our results on two cases considered for the linear production games. First of all, we consider a small problem with $n = 24$ players, $m = 12$ products, $q = 5$ production constraints ($n = 24$ as been chosen as the maximum game size allowing the exact computation of the Shapley value). All data for A, B, p are random (0–1)-uniform values. In the first experiment, we compare the exact Shapley value with the SSS estimates. In Table 5 results are reported for $\tau = 10^6$, $f = \tau/2^{24} \simeq 0.06$. As it can be seen, estimates are quite precise: The standard error is one order of magnitude smaller than the Shapley value and interval estimates contain the exact value. In the second experiment, we compare SSS with the algorithm *Two-Stage-St-ApproShapley-opt* described in Castro et al. (2017). In that method, for each sample unit S, player i's marginal gain must be calculated. Therefore two Linear programs, one for $v(S \cup \{i\})$ and one for $v(S)$, must be solved, and the computational time needed by *Two-Stage-St-ApproShapley-opt* are twice the ones needed by Algorithm 2. To a fair comparison, both methods are run with input τ corresponding to the number of times problem (33) is solved. In Fig. 2 we report the mean of the absolute errors and the mean of the standard error (the error variability), controlling for various sample ratios $\tau/2^{24}$. It can be seen that Algorithm 2 is much better than *Two-Stage-St-ApproShapley-opt*, as the absolute errors of the latter are between 4 and 10 times larger than the former, and the standard errors are between 3 and 5 times larger too. It can also be seen that as the ratio increases, both average and standard errors, seem to stabilize to a threshold.

The last experiment considers a game of $n = 80$ players. For this model size, the exact Shapley calculation is impossible. We run the



(a) Shapley Value Error Mean. (b) Shapley Value Standard Error Mean.

Fig. 2. Dashed line: *Two-Stage-St-ApproShapley-opt*, Continuous line: Least Squares approximation.

Table 5
Exact and approximate value of the Shapley value to the small linear production game with SSS and $f \simeq 0.06$.

Player	ϕ_i	$\tilde{\phi}_i$	SE[$\tilde{\phi}_i$]	95% Interval
1	1.0222	1.0070	0.0203	(0.9672, 1.0468)
2	0.5451	0.5429	0.0199	(0.5038, 0.5819)
3	0.5628	0.5593	0.0199	(0.5202, 0.5984)
4	0.8457	0.8384	0.0205	(0.7982, 0.8787)
5	1.0498	1.0431	0.0200	(1.0039, 1.0822)
6	0.7305	0.7187	0.0206	(0.6783, 0.7591)
7	0.5989	0.6151	0.0201	(0.5757, 0.6546)
8	0.8059	0.8101	0.0209	(0.7691, 0.8511)
9	0.8677	0.8923	0.0201	(0.8530, 0.9316)
10	1.1092	1.1196	0.0198	(1.0807, 1.1585)
11	0.5872	0.5902	0.0204	(0.5502, 0.6302)
12	0.5814	0.5925	0.0204	(0.5525, 0.6326)
13	0.8667	0.8635	0.0203	(0.8238, 0.9032)
14	0.6517	0.6391	0.0205	(0.5988, 0.6793)
15	0.3278	0.3032	0.0190	(0.2660, 0.3404)
16	0.2282	0.2497	0.0191	(0.2122, 0.2872)
17	0.6604	0.6392	0.0204	(0.5991, 0.6793)
18	0.9809	0.9706	0.0202	(0.9309, 1.0103)
19	0.3171	0.3072	0.0192	(0.2696, 0.3448)
20	1.2059	1.2015	0.0193	(1.1637, 1.2394)
21	1.0978	1.1097	0.0195	(1.0715, 1.1479)
22	0.9155	0.9144	0.0203	(0.8746, 0.9541)
23	0.6461	0.6742	0.0202	(0.6346, 0.7139)
24	0.8571	0.8597	0.0208	(0.8189, 0.9005)

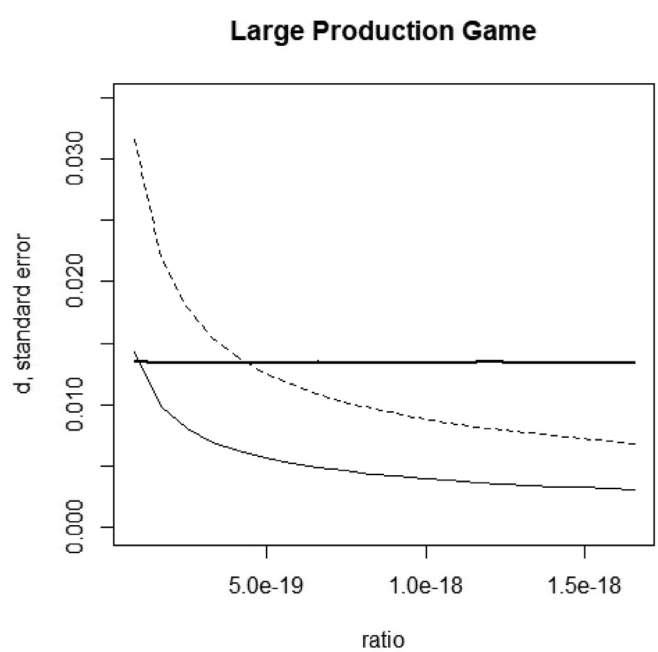
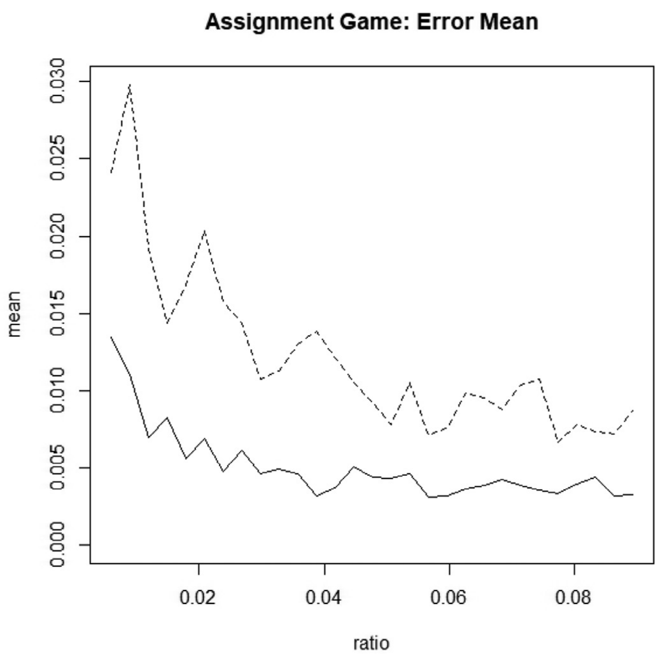


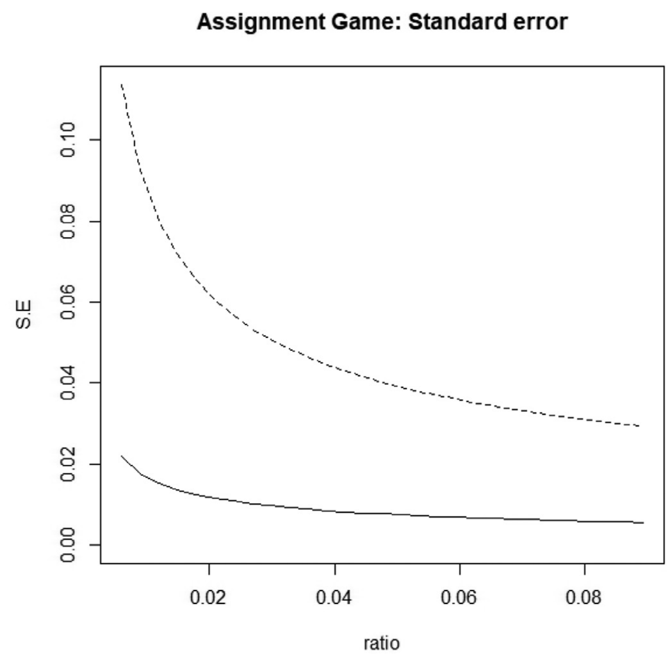
Fig. 3. Results of the Production Game with $n = 80$: Bold line: Average difference d of consecutive Shapley values, Continuous line: Average standard error of the Least Squares approximation, Dashed line: Average standard error of *Two-Stage-St-ApproShapley-opt*.

algorithms for various values of τ, ranging from $\tau = 10^5$ to $\tau = 20 \times 10^5$, corresponding to very small sampling ratios: the maximum being of order 10^{-18}. Computation time ranged from 10 seconds to 4 minutes, but consider that algorithms are implemented in R, that is an interpreted programming language much slower than C and Fortran. Since the exact Shapley values are not computable, Algorithm 2 and *Two-Stage-St-ApproShapley-opt* are compared by the average of standard errors. In Fig. 3, it can be seen that the former algorithm is uniformly better than the latter. Note also that after some relatively small value of f, the means of the standard errors remain rather stable, so that no further improvement in the precision of the estimators is obtained by increasing the sampling fraction, thus good estimators are provided by the Algorithm 2 even for relatively small sample sizes.

To control for the approximation precision, measured by standard errors, we check whether we can rank players according to their power with sufficient precision. So we calculated the average of the difference between two consecutive Shapley values.



(a) Shapley Value Error Mean.



(b) Shapley Value Standard Error Mean.

Fig. 4. Dashed line: *Two-Stage-St-ApproShapley-opt*, Continuous line: Least Squares approximation.

That is, we sort values $\tilde{\phi}_i$ in decreasing order to obtain $\tilde{\phi}_{i:n}$ as the value in position i in the ordered list of size n. Next we calculate $d_i = \tilde{\phi}_{i:n} - \tilde{\phi}_{i+1:n}$ for $i = 1, \ldots, n-1$ and finally the average of d_i's, $i = 1, \ldots, n-1$. Now, the intuition is that when the standard errors are smaller than the average difference, then interval estimates are more and more precise to the point that confidence intervals of the Shapley estimates do not overlap. In Fig. 3, it is apparent that, for a sufficiently large value of the sampling fraction, the standard errors are less than half the average of d_i's, meaning that the corresponding confidence intervals of the Shapley values do not overlap, then a perfect ranking of the strength of the players is possible in this example.

5.5. The assignment and the quadratic superadditive game

Algorithm 2 can be modified to deal with other games. As can be seen, the calculation of $v(S)$ and the data structure involved are independent on the game definition and one can readily substitute the production characteristic function with any other game characteristic function. We will see the results of two more peculiar applications: the Assignment Game and the Quadratic Game.

Assignment Games were introduced by Owen and Shapley and Shubik (1971). Each player $i \in N$ is a buyer/seller of an object. As a seller, player i values its object at c_i euros, while as a buyer, it values all other objects at h_{ij} euros. A transaction between two players i and j occurs on the condition that $h_{ij} - c_j > 0$. For any coalition S, it is supposed that they maximize the profit sum. Therefore, the characteristic function $v(S)$ is the solution to the assignment problem on the bipartite graph $G = (N_1, N_2, E)$, in which nodes are $N_1 = N_2 = S$ and the profit of arc ij, $i \in N_1$, $j \in N_2$ is $\max\{0, h_{ij} - c_j\}$.

We run some experiments to control whether our previous results are confirmed with some simulated data. For sellers, the value of each object c_i is a random number, drawn with uniform probability from the integer range [20; 40]. Buyers overestimate or underestimate all the other objects values with integers within the range [−5; +5]. Numerical results were similar to the ones obtained in the production game and we report only the experiment on the small problem. Here, we had $n = 24$, we calculated Shapley values by complete enumeration, we run Algorithm 2 changing the calculation of $v(S)$, and compare it with algorithm *Two-Stage-St-ApproShapley-opt*. In Fig. 4 we report the mean of the absolute errors and the mean of the standard error (the error variability), controlling for various sample ratios $\tau/2^{24}$. It can be seen that Algorithm (2) is much better than *Two-Stage-St-ApproShapley-opt*, as the absolute errors of the latter are between 2 and 3 times larger than the former, and the standard errors are 5 times larger too.

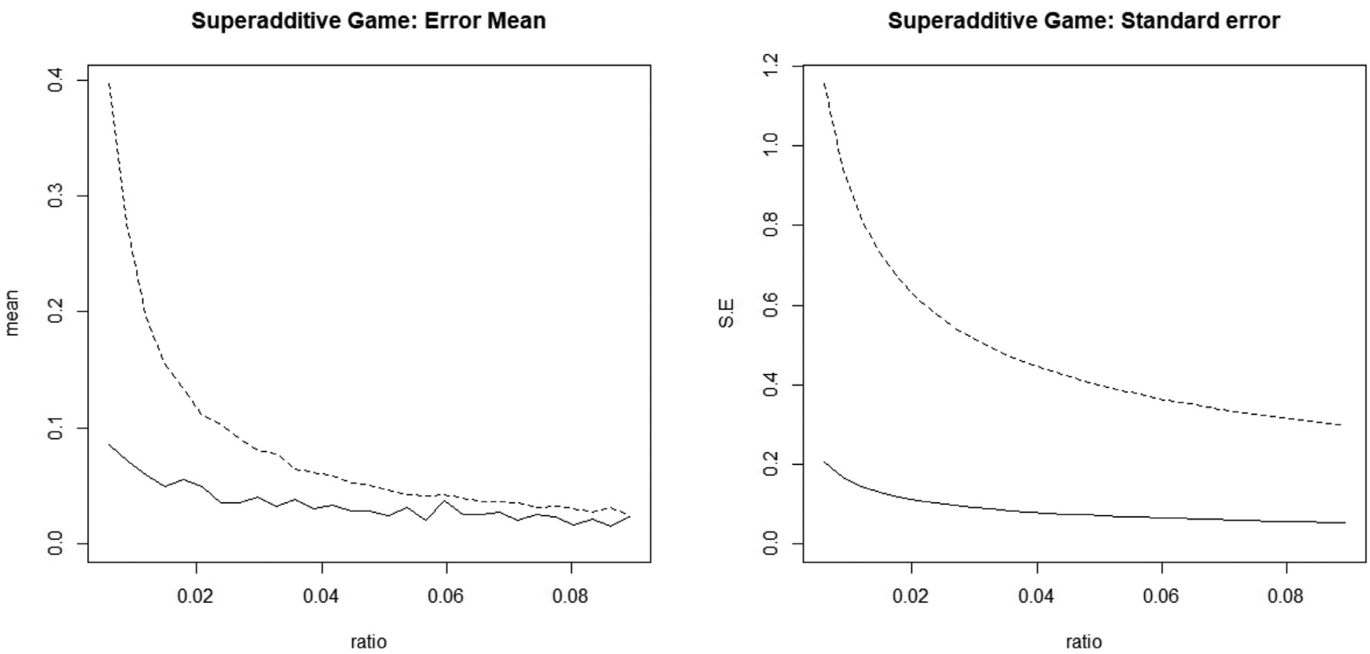
In the next, we are interested to quadratic games as instances of superadditive games on which it is imposed the minimum amount of structure. The characteristic function of quadratic games is defined as follows. Let a_i, $i = 1, \ldots, n$ be positive numbers, then $v(S) = \left(\sum_{i \in S} a_i\right)^2$, and $v(\emptyset) = 0$. To see that those games are superadditive, one can write the characteristic function as:

$$v(S) = \sum_{i \in S} a_i^2 + \sum_{\substack{i,j \in S \\ i<j}} 2a_i a_j$$

Then, one can readily see that, for all S, T such that $S \cap T = \emptyset$:

$$\begin{aligned} v(S) + v(T) &= \sum_{i \in S} a_i^2 + \sum_{\substack{i,j \in S \\ i<j}} 2a_i a_j + \sum_{i \in T} a_i^2 + \sum_{\substack{i,j \in T \\ i<j}} 2a_i a_j \\ &\le \sum_{i \in S} a_i^2 + \sum_{i \in T} a_i^2 + \sum_{\substack{i,j \in S \\ i<j}} 2a_i a_j + \sum_{\substack{i,j \in T \\ i<j}} 2a_i a_j + \sum_{\substack{i \in S \\ j \in T}} 2a_i a_j \\ &= v(S \cup T). \end{aligned}$$

As for the Matching game, we keep $n = 24$, and we draw values a_i from the uniform distribution ranging from 1 to 2. We calculate the exact Shapley value by complete enumeration and then we compare the accuracy of Algorithm 2 and *Two-Stage-St-ApproShapley-opt*. In Fig. 5 we report the mean of the absolute errors and the mean of the standard error, controlling for various sample ratios $\tau/2^{24}$. Again, it can be seen that Algorithm 2 is better than *Two-Stage-St-ApproShapley-opt*, as the absolute errors is



(a) Shapley Value Error Mean.

(b) Shapley Value Standard Error Mean.

Fig. 5. Dashed line: *Two-Stage-St-ApproShapley-opt*, Continuous line: Least Squares approximation.

always larger than the former, and the standard errors are much larger too.

6. Conclusions

We propose a methodology to calculate values of cooperative games that relies on the concept of stochastic approximation.

The idea is to replace the exponential number of terms of the value formula summation with just a sample of them. Applying probability concepts to the sample and to the reduced sum, we can prove that the expectation of the estimate is the actual value of the game and that its standard error keeps under control the difference between the estimate and the actual value. We proved the viability of our approach calculating values of different games with different sampling strategies, and we found that the sample ratio does not need to be large to obtain good approximations. In all our experiments, good approximations were obtained with ratios never greater than 0.1. Moreover, it is worth to note that standard errors depend on the sample size *and not* on the population size, (see Formula (13) for example), so it is expected that as the number of players grows larger, values can be approximated with even smaller ratios, too. Successful applications depend on the sampling strategy and the game characteristic function. Therefore, future research should consider different sampling strategies, such as different stratifications, and different classes of games, as for instance network games following the line in van Campen et al. (2018). Moreover, while the theoretical properties of many cooperative games are often well-understood, we are somewhat missing their economic applications. To date, one reason to this could have been the time complexity to calculate values, which has prevented the analysis of games whose number of players is more than minimal. We hope that the stochastic approximation will be the technique to foster new empirical analysis. An important application of Shapley and other values is to voting games, as testified in Badinger, Muhlbock, Nindl, and Reuter (2014), Barr and Passarelli (2009), and Pajala and Widgrèn (2004). There, values are used as player's power measure. Here we found that even though for large number of players exact values can not be obtained, estimated Shapley values are precise enough to rank players' power.

Acknowledgments

This research has been partially financed by the Spanish Ministry of Science and Technology projects FQM-331, MTM2016-74983-C02-01.

References

Badinger, H., Muhlbock, M., Nindl, E., & Reuter, W. H. (2014). Theoretical vs. empirical power indices: Do preferences matter? *European Journal of Political Economy, 36*, 158–176.
Barr, J., & Passarelli, F. (2009). Who has the power in the EU? *Mathematical Social Sciences, 57*, 339–366.
Bilbao, J. M., Fernández, J. R., Jiménez, A., & López, J. J. (2000). Generating functions for computing power indices efficiently. *TOP, 8*, 191–213.
van Campen, T., Hamers, H., Husslage, B., & Lindelauf, R. (2018). A new approximation method for the Shapley value applied to the WTC 9/11 terrorist attack. *Social Network Analysis and Mining, 8*(1), 3.
Carreras, F., & Puente, M. A. (2015). Coalitional multinomial probabilistic values. *European Journal of operational Research, 245*, 236–246.
Carreras, F., & Puente, M. A. (2018). A note on multinomial probabilistic values. *TOP, 26*, 164–186.
Castro, J., Gómez, D., Molina, E., & Tejada, J. (2017). Improving polynomial estimation of the Shapley value by stratified random sampling with optimum allocation. *Computer and Operations Research, 82*, 180–188.
Castro, J., Gómez, D., & Tejada, J. (2009). Polynomial calculation of the Shapley value based on sampling. *Computer and Operations Research, 36*, 1726–1730.
Dragan, I. (2006). The least squares value and the Shapley value for cooperative TU games. *TOP, 4*, 61–73.
Dubey, P., Neyman, A., & Weber, J. (1981). Value theory without efficiency. *Mathematics of Operations Research, 4*, 99–131.
Fatima, S. S., Wooldridge, M., & Jenniggs, N. R. (2006). Analysis of the Shapley value and its uncertainty for the voting game. *Lecture Notes in Artificial Intelligence, 3937*, 85–98.
Fernández, F. R., Mayor, J. A., Puerto, J., & Zafra, M. J. (2004). Sampling theory and Shapley value. In *Proceeding of the VI Spanish meeting of game theory*.
Fernández, F. R., & Puerto, J. (2005). Sampling theory and cooperative games. In *Proceeding of the worhshop on stochastic method in game theory*.
Fernández, F. R., & Puerto, J. (2006). Teoria de juegos multiobjetivos. Imagraf. Malaga.
Fernández, F. R., Puerto, J., & Zafra, M. J. (2002). Cores of stochastic cooperative games. *International Game Theory Review, 4*, 265–280.

Granot, D. (1977). Cooperative games in stochastic function form. *Management Science, 23*, 621–630.

Keinan, A., Sandbank, B., Hilgetag, C. C., Meilijson, I., & Ruppin, E. (2006). Axiomatic scalable neurocontroller analysis via the Shapley value. *Artificial Life, 12*, 333–352.

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

Littlechild, S., & Owen, G. (1973). A simple expression for the Shapley value in a special case. *Management Science, 20*, 370–372.

Maleki, S., Tran-Thanh, L., Hines, G., Rahwan, T., & Rogers, A. (2014). Bounding the estimation error of sampling-based Shapley value approximation. https://arxiv.org/pdf/1306.4265.pdf.

Mann, I., & Shapley, L. S. (1960). Values for large games IV:Evaluating the electoral collegue by monte-carlo techniques. Technical report The Reand Corporation, Santa Monica.

Matsui, T., & Matsui, Y. (2000). A survey of algorithms for calculating power indices of weighted majority games. *Journal of the Operations Research Society of Japan, 43*, 71–86.

Moretti, S. (2010). Statistical analysis of the Shapley value for microarray games. *Computers & Operations Reseach, 37*, 1413–1418.

Owen, G. (1995). *Game Theory*. California: Academic Press.

Pajala, A., & Widgrèn, M. (2004). A priori versus empirical voting power in the EU council of ministers. *European Union Politics, 5*, 73–97.

Puerto, J., Fernández, F. R., & Hinojosa, Y. (2008). Partially ordered cooperative games:extended core and Shapley value. *Annals of Operations Research, 158*, 143–159. 2008.

Ruiz, L. M., Valenciano, F., & Zarzuelo, J. M. (1998). The family of least squares values for transferable utility games. *Games and Economic Behavior, 24*, 109–130.

Saavedra-Nieves, A., García-Jurado, I., & Fiestras-Janeiro, M. G. (2018). Estimation of the Owen value based on sampling. In E. Gil, E. Gil, J. Gil, & M. Gil (Eds.), *The mathematics of the uncertain. studies in systems, decision and control: 142*. Cham: Springer.

Shapley, L. M., & Shubik, M. (1971). The assignment game I: The core. *International Journal of Game Theory, 1*, 111–130.

Timmer, J. (2006). The compromise value for cooperative games with random payoffs. *Mathematical Methods of Operations Research, 64*, 95–106.

Timmer, J., Borm, P., & Tijs, S. (2003). On three Shapley-like solutions for cooperative games with random payoffs. *International Journal of Game Theory, 32*, 595–613.

Timmer, J., Borm, P., & Tijs, S. (2005). Convexity in stochastic cooperative situations. *International Game Theory Review, 7*, 25–42.