

# Efficient computation of the Shapley value for Large-Scale Linear Production Games

Phuoc Hoang Le · Tri-Dung Nguyen · Tolga Bektas

Submitted: 31st March, 2017  
Re-submitted: 7th February, 2018

**Abstract** The Linear Production Game is concerned with allocating the total payoff of an enterprise among the owners of the resources in a fair way. With cooperative game theory providing a mathematical framework for sharing the benefit of the cooperation, the Shapley value is one of the widely used solution concepts as a fair measurement in this area. Finding the exact Shapley value for linear production games is, however, challenging when the number of players exceeds 30. This paper describes the use of linear programming sensitivity analysis for a more efficient computation of the Shapley value. The paper also proposes a stratified sampling technique to estimate the Shapley value for large-scale linear production games. Computational results show the effectiveness of the proposed methods compared to others.

**Keywords** Cooperative games · linear production game · payoff allocation · Shapley value · fairness.

## 1 Introduction

The Linear Production Game (LPG) provides a model for a collaboration arising in a production environment in which several independent decision makers are involved to make joint production maximisation based on their combined resources. Owen (1975) describes the LPG as a type of cooperative game with transferable utility (TU) that provides a framework to understand the cooperation among the resource owners (players) in such a context. The players work jointly to produce finished goods which can be sold at given market prices. Through working in the collaboration, the team can manufacture more products, and hence, can collectively generate more profit to the members.

One of the fundamental questions in cooperative game theory, including the LPG, is how to share the payoff/cost among the players in a fair way. Several

solution concepts have in the past been proposed to address this question, which include the core, the least core, the Shapley value and the Banzhaf index. We provide a brief review of these in Section 2.1. Among these, the Shapley value is one of the most popular solution concepts, particularly in cooperative games with TU, probably due to its relevant economics interpretation on marginal contributions and its desirable mathematical properties, i.e., it is a unique concept that satisfies four axioms of efficiency, dummy, symmetry and additivity. We provide further details about these properties in Section 2.2. Not only the Shapley value is interpreted as the payoff/cost distribution but also it suggests a measure of the power of the players in a voting game (Shapley & Shubik, 1954; Bilbao et al., 2000). The Shapley value has also been employed in TU games with graph-restricted communications as a measure of centrality to identify the important nodes in a social network (Gomez et al., 2003; Michalak et al., 2013).

In general, one cannot calculate the Shapley value in polynomial time for many coalition games due to the fact that its formulation involves an exponentially large number of terms. Such a computational difficulty also holds for structured games such as weighted voting games and minimum cost spanning tree games (see, Deng & Papadimitriou, 1994; Ando, 2012, for more detail). Finding the exact Shapley Value for large-scale cooperative games, including the LPG, is computationally intractable in general.

In this paper, linear programming sensitivity analysis is used to improve the process of recalculating the coalition values. We also construct a randomized algorithm for approximating the Shapley value of the LPG with a large number of players. This method combines linear programming sensitivity analysis and a stratified sampling technique. The contributions of this work are as follows:

- We derive an alternative closed form solution for calculating the exact Shapley value of a special class of Linear Production Games (in Section 3.1)
- We present a new method for its efficient computation that utilises linear programming sensitivity analysis (in Section 3.2).
- For large-scale Linear Production Games, we additionally use a stratified sampling technique to approximate the Shapley value (in Section 4).
- Finally, we provide numerical results to illustrate the effectiveness of the proposed methods compared to existing methods (in Section 5).

## 2 Preliminaries and Related Literature

In this section, we first provide some background on cooperative game theory and the LPG. We also provide a review of existing methods for computing the Shapley value.

### 2.1 Cooperative Games with Transferable Utility

We first introduce some solution concepts of cooperative games with TU. Consider a set  $\mathcal{N} = \{1, \dots, n\}$  of  $n$  players, and a *characteristic function*  $v : 2^{\mathcal{N}} \rightarrow \mathbb{R}$ , that maps each subset of players into a real value representing the payoff that the coalition is guaranteed to receive no matter what other players choose. We call  $\mathcal{G} = (\mathcal{N}, v)$  a *cooperative game*. Utility (payoff) is assumed to be transferable,

i.e., for any coalition  $S \subseteq \mathcal{N}$ , its total payoff is defined as  $v(S)$ , which can be transferred freely among its members.

Given a cooperative game  $(\mathcal{N}, v)$ , we are interested in finding an allocation  $\mathbf{x} \in \mathbb{R}^n$  to distribute the total payoff  $v(\mathcal{N})$  among individual players. An allocation (also called a payoff distribution)  $\mathbf{x}$  is *efficient* if  $\sum_{i \in \mathcal{N}} x_i = v(\mathcal{N})$ .

An important question in cooperative games is whether players are willing to join the grand coalition  $\mathcal{N}$ . For this to happen, there should exist an allocation  $\mathbf{x} \in \mathbb{R}^n$  with which each individual player is better off as compared to their standalone payoff. An allocation  $\mathbf{x}$  is called *individually rational* if  $x_i \geq v(\{i\}), \forall i \in \mathcal{N}$ .

An *imputation* is an allocation that is both efficient and individually rational. Individual rationality is not sufficient to guarantee that some players would prefer the grand coalition  $\mathcal{N}$  to a smaller coalition  $S \subsetneq \mathcal{N}$ . An allocation is called *stable* with respect to a coalition  $S$  if  $\sum_{i \in S} x_i \geq v(S)$ . The set of efficient allocations that are stable with respect to all coalitions  $S \subseteq \mathcal{N}$  is called the *core*.

However, the core might not exist. In that case, alternative solution concepts such as the least core and the Nucleous are designed to minimise the dissatisfaction among the coalitions. While the aforementioned solution concepts deal with the stability of the game, i.e., avoiding coalitions from breaking away from the grand coalition, the focus of this work is on another solution concept, called the Shapley value (Shapley, 1953), described next.

## 2.2 The Shapley Value

The Shapley value is among the most popular solution concepts in cooperative game theory for dividing the total payoff/cost between the players, assuming that they all collaborate. The allocation for player  $i$  is proportional to that player's contribution to the game, i.e., how much value, player  $i$  creates. The formula of the Shapley value for player  $i$  is the weighted average of all the marginal contributions of that player to all coalitions  $S \subseteq \mathcal{N} \setminus \{i\}$ .

More formally, the *Shapley value*  $\phi \in \mathbb{R}^n$  of a cooperative game  $G = (\mathcal{N}, v)$  is the allocation of payoff where

$$\phi_i = \sum_{S \subseteq \mathcal{N} \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} [v(S \cup \{i\}) - v(S)], \quad (1)$$

and where we use the short-hand notation  $v(S \cup \{i\}) - v(S)$  to denote the marginal contribution of player  $i$  to coalition  $S$ .

Weber (1988) provides another way to formulate the Shapley value as follows. Suppose the ordering of the arrival of players is represented as a permutation function  $\pi : \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\}$ , where  $\pi(j) = i$  means that player  $i$  joins the group in position  $j$ . Denote  $\Pi(\mathcal{N})$  as the set of all permutations  $\pi = (\pi(1), \pi(2), \dots, \pi(n))$  of players in  $\mathcal{N}$ , hence  $|\Pi(\mathcal{N})| = n!$ . For each  $\pi$  and  $i = \pi(j)$ , let  $\text{Pre}^i(\pi) := \{\pi(1), \pi(2), \dots, \pi(j-1)\}$  be the set of players who are the predecessors of player  $i$  in  $\pi$ . Given a permutation  $\pi$  that shows the ordering of the arrival of the players, the marginal contribution of player  $i = \pi(j)$  is the difference  $v(\pi(1), \dots, \pi(j)) - v(\pi(1), \dots, \pi(j-1))$ . The Shapley value  $\phi_i$  of a player  $i$  is the

average over all permutations  $\pi \in \Pi(\mathcal{N})$ , of the marginal contribution of the player  $i$  to all players who arrive before player  $i$ . Therefore, the value is calculated as:

$$\phi_i = \frac{1}{n!} \sum_{\pi \in \Pi(\mathcal{N})} [v(\text{Pre}^i(\pi) \cup \{i\}) - v(\text{Pre}^i(\pi))]. \quad (2)$$

In super-additive games such as the LPG, the condition  $v(i) \leq v(\text{Pre}^i(\pi) \cup \{i\}) - v(\text{Pre}^i(\pi))$  implies  $v(i) \leq \phi_i$ , i.e., the payoff allocation  $\phi$  satisfies the individual rationality property, making the Shapley value also an imputation allocation. The Shapley value defined in (1) is shown to be a unique division scheme that meets four desirable criteria (axioms). The first axiom (Efficiency) requires that players divide all the payoff/cost available to the grand coalition among themselves. The second axiom (dummy) requires that zero payoff is assigned to a player who does not contribute to any coalition. In the third axiom (symmetry), two players  $i, j$  are said to be symmetric in the game  $v$  if they make the same marginal contributions to any coalition. The axiom demands to pay equal shares to symmetric players. The last axiom (additivity) states that, for any two separate games  $G^1 = (\mathcal{N}, v^1)$  and  $G^2 = (\mathcal{N}, v^2)$ , the payoff of a player in the game  $G^1 \oplus G^2 := (\mathcal{N}, v^1 + v^2)$  should be equal to the sum of that player's payoffs in the two games.

Another special property of Shapley value for a fair payoff allocation is monotonicity (Young, 1985). This states that if we change a game such that the contributions of a player to all coalitions stays the same or increases then the final payoff for that player should not decrease. Young (1985) also shows that the Shapley value can be described simply by the three properties of efficiency, symmetry, and monotonicity for all cooperative games.

### 2.3 The Linear Production Game (LPG)

Consider a set  $\mathcal{N}$  of  $n$  players who own a set  $R$  of  $r$  resources available for the production of a set  $P$  of  $p$  products. Each player  $i$  has a vector of resources  $\mathbf{b}_i = (b_i^1, b_i^2, \dots, b_i^r)^T$ , i.e., that player has exactly  $b_i^k$  unit of resource  $k \in \{1, 2, \dots, r\}$ . Manufacturing product  $j \in \{1, 2, \dots, p\}$  requires  $a_{kj}$  units of resource  $k$ , for  $k \in \{1, 2, \dots, r\}$ . Product  $j$  can be sold in the market at price  $c_j$ .

For any arbitrary coalition  $S$  of players in  $\mathcal{N}$ , the combined resource vector is  $\mathbf{b}(S) = (b^1(S), b^2(S), \dots, b^r(S))^T = \sum_{i \in S} \mathbf{b}_i$ , where  $b^k(S) = \sum_{i \in S} b_i^k$ , for  $\forall k \in \{1, 2, \dots, r\}$ . The value of a coalition  $S$  is the maximum payoff this group can achieve with all the resources possessed by its members. Let the vector of production output be denoted as  $\mathbf{y} = (y_1, \dots, y_p)$  with  $y_p$  as the amount of product  $p$  manufactured. Then, the payoff  $v(S)$  assigned to a coalition  $S$  is defined as the total profit that the coalition can collectively gained, and can be calculated by solving the following linear program:

$$\begin{aligned} v(S) := \max & \sum_{j=1}^p c_j y_j \\ \text{s.t.} & \sum_{j=1}^p a_{kj} y_j \leq b^k(S), \quad \forall k = \{1, 2, \dots, r\}, \\ & y_j \geq 0, \quad \forall j = \{1, 2, \dots, p\}, \end{aligned} \quad (3)$$

or equivalently,  $v(S) := \max \{ \mathbf{c}^T \mathbf{y} \mid \mathbf{A} \mathbf{y} \leq \mathbf{b}(S), \mathbf{y} \geq 0 \}$ , where  $A = \{a_{kj}\}_{k=1, \dots, r}^{j=1, \dots, p}$  is the resource requirement matrix.

The dual problem of (3) can be derived as

$$v(S) := \min \{ \mathbf{b}(S)^T \mathbf{z} \mid \mathbf{A}^T \mathbf{z} \geq \mathbf{c}, \mathbf{z} \geq 0 \}. \quad (4)$$

Owen (1975) shows that the LPG is a totally balanced game and the LPG core is always non-empty. In addition, a core solution can be constructed by solving the dual of problem (3) for the grand coalition. Let  $\mathbf{z}^*$  be a dual solution of (4) when  $S = \mathcal{N}$ , then a payoff allocation scheme  $\mathbf{x} = (\mathbf{b}_i^T \mathbf{z}^*)_{i \in \mathcal{N}}$  is in the core. As the dual might have multiple optimal solutions, the set of core solutions corresponding to these optimal solutions are called the Owen set.

The characteristic function of the LPG is also super-additive; that is for any two disjoint coalitions  $S_1$  and  $S_2$ , we have  $v(S_1) + v(S_2) \leq v(S_1 \cup S_2)$ . Therefore, the grand coalition will be formed for the benefit of all players. The LPG is concerned with dividing the total payoff gained by the grand coalition to its players.

Linear Production Games have been widely studied over the past 40 years. Granot (1986) generalises Owen's LPG model without the additivity assumption of the resource function. Variations of this model include minimum cost spanning tree games (Granot et al., 1981), assignment games (Shapley & Shubik, 1971) and network synthesis games (Tamir, 1991). Some other extensions of linear production games are investigated in the works of Fernández et al. (2005) and Lozano (2013). Recently, Nishizaki et al. (2016) examined a linear production game with graph-restricted communication, developed the extended Owen solution and showed its geometric properties. The LPG is in fact belong to a class of games called the Operational Research Games (Borm et al., 2001), which include all games whose payoff of each coalition is represented as the optimal value of an optimisation problem. In the case of the LPG, this optimisation problem is a linear program.

## 2.4 Computing the Shapley Value

The expression (1) has an exponentially large number of terms and this implies that computing the Shapley value exactly is NP-hard in general (Deng & Papadimitriou, 1994). In fact, polynomial algorithms for computing this is only possible in special cases if we exploit the special structure of the games (Michalak et al., 2013; Ando, 2012). The formulations of the Shapley value suggest the use of Monte Carlo simulations. The first of such a method was proposed in Mann & Shapley (1960) to approximate the Shapley value, and then to analyse the Electoral College vote system in the UP Presidential elections.

### 2.4.1 Simple random sampling and stratified permutation sampling

With the Shapley formula (2) of permutations, (Castro et al., 2009) proposes a simple random technique to approximate the Shapley value. Among the  $n!$  possible permutations of the players, a sample is drawn and the sample means are used as approximated Shapley values. This technique is simple to implement compared to other randomized methods because of its formulation. The method has some

advantages such as being unbiased and having no error for dummy players. Moreover, its theoretical error of the approximation can be calculated in a probabilistic way.

Castro et al. (2017) presents an improved stratified permutation sampling technique to reduce the variances of the estimation. By noticing that the ordering when a player joins the grand coalition could be a major factor in contributing to the variances of the estimation, the authors stratify the population into those with fixed-position ordering. More specifically, for each player  $i \in \mathcal{N}$  and each position  $j = 1, \dots, n$ , there is a corresponding stratum containing all permutations having player  $i$  in the  $j$  position. For optimal sampling, the authors also develop advanced techniques for estimating the variance in each stratum before designing the optimal sample sizes for the strata accordingly. The authors also present the desirable features of the sampling technique.

#### 2.4.2 Stratified coalition sampling

While the methods of (Castro et al., 2009, 2017) sample over permutations, another possibility is to sample over coalitions. This sampling approach uses the coalitional formula (1) of the Shapley value and rewrites it in the the following form:

$$\begin{aligned}
 \phi_i &= \frac{1}{n!} \sum_{\pi \in \Pi(\mathcal{N})} [v(\text{Pre}^i(\pi) \cup \{i\}) - v(\text{Pre}^i(\pi))]. \\
 &= \sum_{S \subseteq \mathcal{N} \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} [v(S \cup \{i\}) - v(S)] \\
 &= \frac{1}{n} \sum_{k=0}^{n-1} \underbrace{\frac{1}{\binom{n-1}{k}} \sum_{|S|=k, i \notin S} [v(S \cup \{i\}) - v(S)]}_{\mathbb{E}(Y_i^k)}. \tag{5}
 \end{aligned}$$

Here, for each fixed  $i$ , let  $S$  be a random coalition with size  $k$  and that does not include player  $i$ . There are  $\binom{n-1}{k}$  such coalitions. If  $S$  is uniformly chosen from that sub-population and if we let  $Y_i^k$  be the corresponding random variable taking random value  $[v(S \cup \{i\}) - v(S)]$ , then the inner summation is exactly the expectation  $\mathbb{E}(Y_i^k)$  of  $Y_i^k$ . The idea of a random sampling technique is to draw random samples (i.e., coalitions) and replace that expectation with the sample average.

Formulation (5) suggests the use of Monte Carlo simulation based on stratified sampling. Each stratum  $k$  will be a set of all coalitions of the same size  $k$ . Maleki et al. (2013) propose an heuristic method to allocate the samples into different strata. The approach provides a non-asymptotic bound for the sampling error by using Hoeffdings inequality (Hoeffding, 1963, Theorem 2.) to estimate the error within each stratum.

In the sampling model of the Shapley value for player  $i$ , the population contains all marginal contributions  $v(S \cup \{i\}) - v(S)$ , where  $S$  is a subset of  $\mathcal{N} \setminus \{i\}$ . For each player  $i$ , this can be divided into  $n$  separate strata  $\{M_k^i\}_{k=0}^{n-1}$ , where stratum  $M_k^i$  contains all coalition of size  $k$  that does not contain player  $i$ . The total number of samples is allocated to  $n \times n$  strata by the population-proportional allocation

rule. For each stratum, we take the simple random sampling to approximate the expected value  $\mathbb{E}(Y_i^k)$ . More details on different stratified sampling techniques can be found in the book by [Cochran \(2007\)](#).

### 2.4.3 Learning stratified sampling

In this stratified algorithm, the Neyman allocation is applied to decide the numbers of sampling units taken for each stratum. A reinforcement learning algorithm is proposed in the paper by [O'Brien et al. \(2015\)](#) to estimate the variance in each stratum and to adjust the sample allocation among strata. The method also requires identifying the proportion of ‘exploration stage’ and ‘exploitation stage’ to determine the sampling allocation, which is a heuristic step in practice. In this implementation of the learning algorithm, we use the same coefficients  $\gamma$  and  $\beta$  of the previous paper for our sigmoid function.

## 3 An Exact Method for Finding the Shapley Value of LPGs

In this section, we first provide analytical properties of the Shapley value of the LPGs by exploiting the dual structure of the characteristic function. We then apply linear programming sensitivity analysis (LPSA) to provide an efficient computation of the marginal contribution of a player  $i$  to a coalition  $S$ .

### 3.1 Analytical Properties of the Shapley Value in LPGs

We first observe that, in reality, while the number of players might change, we often have a reasonably small and fixed number of resource types. We are interested in how the complexity of computing the Shapley value grows with the increase of the number of players. First, we formulate the dual of the characteristic function  $v(S)$  for each coalition  $S$  as follows,

$$v(S) := \min_{\mathbf{y} \in Y} \mathbf{b}(S)^T \mathbf{y}, \quad (6)$$

where  $Y = \{\mathbf{y} \in \mathbb{R}^r : \mathbf{A}^T \mathbf{y} \geq \mathbf{c}, \mathbf{y} \geq 0\}$  is the feasible space of the dual problem and let  $T$  be the number of extreme points of  $Y$ . It is interesting to note that this feasible set is independent on the set of players.

Let  $\mathbf{y}_i^*$ ,  $i = 1, \dots, n$ , be an optimal basic feasible solution of the dual problem (6) with the objective coefficients  $\mathbf{b}_i$ . For simplicity, we assume that the input data of the problem is chosen such that no coalition  $S$  would have multiple optimal solutions in this problem of computing  $v(S)$ . The assumption can be achieved by slightly perturbing the data of the LPG. We note that similar results derived in this work still applies even without this assumption, however, we need to be more careful in some of the statements.

If  $\mathbf{b}_i$  are all equal, then, by symmetry,  $\mathbf{y}_i^* = \mathbf{y}_j^*$ ,  $\forall i, j \in \mathcal{N}, i \neq j$ , and the Shapley value of all players should be the same and is equal to  $\mathbf{b}_i^T \mathbf{y}_i^*$ . It is noted, however, that the resource vectors  $\mathbf{b}_i, i = 1, \dots, n$ , do not have to be exactly the same for  $\mathbf{y}_i^* = \mathbf{y}_j^*$ ,  $\forall i, j \in \mathcal{N}, i \neq j$ . Instead, we show below a condition on  $\mathbf{b}_i, i = 1, \dots, n$ , for this to hold. Let us first provide some additional notation.

Let  $\mathbf{a}_1, \dots, \mathbf{a}_p$  be the columns of  $A$ . For each  $\mathbf{y} \in Y$ , let  $I(\mathbf{y}) \subseteq \{1, \dots, p\}$  be the set of active constraints at  $\mathbf{y}$ , i.e.,  $\mathbf{a}_j^T \mathbf{y} = c_j, \forall j \in I(\mathbf{y})$ . We denote  $A_{I(\mathbf{y})}$  as the set of active (tight) rows of  $A^T$  at  $\mathbf{y}$ . For each  $i = 1, \dots, n$ , let us define

$$\mathcal{K}_i = \left\{ - \sum_{l \in I(\mathbf{y}_i^*)} \gamma_l \mathbf{a}_l : \gamma_l \geq 0 \right\},$$

as the cone formed by the negative of the tight rows in  $A_{I(\mathbf{y}_i^*)}$ . We can show that  $(\mathcal{K}_i - \mathbf{y}_i^*)$  is indeed the polar cone of the support cone of  $(Y - \mathbf{y}_i^*)$  at  $\mathbf{0}$ .

We have the following results on the closed form solution of the Shapley value if the LPG has some special properties.

**Proposition 1** *If  $-\mathbf{b}_i \in \mathcal{K}_1$  for all  $i = 2, \dots, n$ , then the following results hold:*

- (a) *There exists vector  $\mathbf{y}^*$  such that  $\mathbf{y}_i^* = \mathbf{y}^*, \forall i = 1, \dots, n$ .*
- (b) *For all  $S \subseteq \mathcal{N}$ ,  $\mathbf{y}^* \in \operatorname{argmin}_{\mathbf{y} \in Y} \mathbf{b}(S)^T \mathbf{y}$ .*
- (c) *The characteristic function  $v$  of the LPG is additive and the closed form solution of the Shapley value is given by  $\phi_i = \mathbf{b}_i^T \mathbf{y}^*$ .*
- (d) *The Shapley value of the game lies in the core.*

*Proof*

- (a) The proof of this part is straightforward since as long as the negative of the objective vector  $-\mathbf{b}_i$  belongs to the cone  $\mathcal{K}_1$ , then the corresponding optimal dual solution is  $\mathbf{y}_i^* = \mathbf{y}_1^*$ .
- (b) Since  $\mathbf{y}_i^* = \mathbf{y}_j^*, \forall i, j \in \mathcal{N}, i \neq j$ , we have  $\mathcal{K}_i = \mathcal{K}, \forall i \in \mathcal{N}$ , for some cone  $\mathcal{K}$ . By the definition of  $\mathbf{y}_i^*$ , i.e.,  $\mathbf{y}_i^* \in \operatorname{argmin}_{\mathbf{y} \in Y} \mathbf{b}_i^T \mathbf{y}$ , and by the definition of

$\mathcal{K}_i$ , we have  $-\mathbf{b}_i \in \mathcal{K}_i \equiv \mathcal{K}$ . Therefore  $-\mathbf{b}(S) = \sum_{i \in S} (-\mathbf{b}_i) \in \mathcal{K}$  and hence

$$\mathbf{y}^* \in \operatorname{argmin}_{\mathbf{y} \in Y} \mathbf{b}(S)^T \mathbf{y}.$$

- (c) From part (b) we have

$$v(S) = \mathbf{b}(S)^T \mathbf{y}^* = \sum_{i \in S} \mathbf{b}_i^T \mathbf{y}^* = \sum_{i \in S} v(\{i\}).$$

Thus, the characteristic function  $v$  of the LPG is additive and  $v(S \cup \{i\}) - v(S) = v(\{i\})$ . We substitute into the Shapley formula and have the closed form solution as  $\phi_i = \mathbf{b}_i^T \mathbf{y}^*$ . The closed form solution of the Shapley value is given by  $\phi_i = \mathbf{b}_i^T \mathbf{y}^*$ .

- (d) This result directly follows from the additivity property of the game. For any coalition  $S$ , we have  $\sum_{i \in S} \phi_i = \mathbf{b}(S)^T \mathbf{y}^* = v(S)$ . Thus, coalition  $S$  has no incentive to leave the grand coalition and the solution is stable.

Proposition 1 is useful when the resource vectors  $\mathbf{b}_i, i = 1, \dots, n$ , are reasonably close to each other in directions such that they all belong to  $-\mathcal{K}_1$ .

Results in Proposition 1(d) show that the Shapley value lies in the core in the special case where all the resource vectors are sufficiently close to each other. In general, if that condition does not hold, we can construct examples where the Shapley value is no longer stable. However, we can still generalise results from Theorem 1 and show some partial-stability results as stated below.



**Proposition 2** For any coalition  $S$  whose resource vectors  $\mathbf{b}_j, j \in S$ , are sufficiently close to each other, i.e.,  $\mathbf{b}_j \in \mathcal{K}_k \forall j \in S$  for some  $k \in \{1, \dots, \Gamma\}$ , the coalition does not have any incentive to break away from the grand coalition if the Shapley value is used as the payoff distribution among the players.

*Proof* In order to prove that the coalition  $S$  does not have the incentive to break away from the Shapley value sharing scheme, we need to show that  $\sum_{i \in S} \phi_i \geq v(S)$ .

This could be done if we are able to prove that  $\phi_i \geq v(\{i\}) \forall i \in \mathcal{N}$  and that  $\sum_{i \in S} v(\{i\}) = v(S)$ .

In the first part, we have

$$\begin{aligned} \phi_i &= \sum_{S \subseteq \mathcal{N} \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} [v(S \cup \{i\}) - v(S)] \\ &\geq \sum_{S \subseteq \mathcal{N} \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} v(\{i\}) = v(\{i\}). \end{aligned}$$

Here, we have used the fact that the LPG is superadditive and hence any marginal contribution of player  $i$  to any coalition is at least  $v(\{i\})$ .

In the second part, we have  $\mathbf{b}(S) = \sum_{j \in S} \mathbf{b}_j \in \mathcal{K}_k$  since  $\mathbf{b}_j \in \mathcal{K}_k, \forall j \in S$ .

Therefore, the optimal dual solution  $\mathbf{z}^*$  of (4) for coalition  $S$  is the same with the optimal dual solution for all  $\mathbf{b}_j, j \in S$ . Therefore,

$$v(S) = \mathbf{b}(S)^T \mathbf{z} = \sum_{i \in S} \mathbf{b}_i^T \mathbf{z} = \sum_{i \in S} v(\{i\}).$$

□

If the resource vectors are sufficiently different such that  $\mathbf{y}_i^* \neq \mathbf{y}_j^*$  for some  $i, j \in \mathcal{N}, i \neq j$ , the result in Proposition 1 no longer holds and we need to work with the extreme points of  $Y$  which is fixed and independent of the set of players.

Let  $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots, \mathbf{y}^{(\Gamma)}$  be the extreme points of  $Y$  and let  $\mathcal{C}^{(1)}, \mathcal{C}^{(2)}, \dots, \mathcal{C}^{(\Gamma)}$  be the corresponding polar cones at these extreme points from which the objective vectors can vary without changing the optimal solutions. Each coalition  $S$  corresponds to an objective vector  $\mathbf{b}(S)$  with a corresponding optimal solution  $\mathbf{y}_S^*$  that belongs to the set  $\{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots, \mathbf{y}^{(\Gamma)}\}$ . The corresponding coalition value is  $v(S) = \mathbf{b}(S)^T \mathbf{y}_S^*$ .

Conversely, let  $H_l, l = 1, \dots, \Gamma$ , be the set of coalitions whose optimal dual variable coincides with  $\mathbf{y}^{(l)}$ , i.e.,

$$H_l = \left\{ S \subseteq \mathcal{N} : \mathbf{y}^{(l)} \in \operatorname{argmin}_{\mathbf{y} \in Y} \mathbf{b}(S)^T \mathbf{y} \right\}.$$

Let  $\gamma_k = \frac{k!(n - k - 1)!}{n!}$ , for  $k \in \{0, 1, \dots, n - 1\}$ . We can rewrite the Shapley value formulation as follows:

$$\begin{aligned}
\phi_i &= \sum_{k=0}^{n-1} \sum_{|S|=k} \gamma_k [v(S \cup \{i\}) - v(S)] \\
&= \sum_{k=0}^{n-1} \sum_{l_1=1}^{\Gamma} \sum_{l_2=1}^{\Gamma} \left\{ \sum_{|S|=k, S \in \mathbf{H}_{l_1}, S \cup \{i\} \in \mathbf{H}_{l_2}} \gamma_k [v(S \cup \{i\}) - v(S)] \right\} \\
&= \sum_{k=0}^{n-1} \sum_{l_1=1}^{\Gamma} \sum_{l_2=1}^{\Gamma} \left\{ \sum_{|S|=k, S \in \mathbf{H}_{l_1}, S \cup \{i\} \in \mathbf{H}_{l_2}} \gamma_k \left[ \mathbf{b}(S \cup \{i\})^T \mathbf{y}^{(l_1)} - \mathbf{b}(S)^T \mathbf{y}^{(l_2)} \right] \right\}. \quad (7)
\end{aligned}$$

*Remark 1* The alternative analytical formulation for the Shapley value in Equation (7) is useful for cases when the numbers of resources and products ( $r, p$ ) are relatively small compared to the number of players  $n$ . In the below, we will provide an example when the number of extreme points of  $Y$  is relatively small compared to the total number of coalitions.

It is also possible to analyse the structure of the problem to further reduce the calculation in Equation (7). We demonstrate one such possibility as follows. Let  $\mathcal{K}_{\mathcal{N}} = \text{conv}(\cup_{i \in \mathcal{N}} \mathcal{K}_i)$  be the smallest cone that contains all the cones  $\mathcal{K}_i, i \in \mathcal{N}$ . We then have the following result:

**Proposition 3** For each  $k = 1, \dots, \Gamma$ , if  $\mathcal{C}^{(k)} \cup \mathcal{K}_{\mathcal{N}} \neq \emptyset$ , then  $\mathbf{H}_k = \emptyset$ .

*Proof* Suppose by contradiction that  $\mathcal{C}^{(k)} \cup \mathcal{K}_{\mathcal{N}} \neq \emptyset$  but  $\mathbf{H}_k \neq \emptyset$ . This means there exists a coalition  $S$  such that  $\mathbf{y}^{(k)} \in \text{argmin}_{\mathbf{y} \in Y} \mathbf{b}(S)^T \mathbf{y}$ . Therefore,  $\mathbf{b}(S) \in \mathcal{C}^{(k)}$  and hence  $\mathbf{b}(S) \notin \mathcal{K}_{\mathcal{N}}$ . However, we have  $\mathbf{b}_j \in \mathcal{K}_{\mathcal{N}}, \forall i \in S$  and hence  $\mathbf{b}(S) \in \mathcal{K}_{\mathcal{N}}$ .  $\square$

The implication of Proposition (3) is that, by checking whether or not  $\mathcal{C}^{(k)} \cup \mathcal{K}_{\mathcal{N}} \neq \emptyset$ , we can find out a list of empty  $\mathbf{H}_k$  and hence can eliminate them from Equation (7) to speed up the calculation.

Consider, for example, the case of  $n = 30, p = 3$  and  $r = 2$ . Using formulation (3) would involve solving  $2^n \approx 10^9$  LPs, each with  $p$  decision variables and with  $(r + p)$  constraints. If we use Equation (7), we notice that  $Y$  has at most  $\binom{r+p}{p} = 10$  extreme points and the equation involves only  $10^2 = 100$  pairs of  $(l_1, l_2)$ . In each pair, we only need to evaluate linear functions of those coalitions that satisfy  $\{|S| = k, S \in \mathbf{H}_{l_1}, S \cup \{i\} \in \mathbf{H}_{l_2}\}$ . Let coalition  $S$  be rewritten as a binary vector  $\mathbf{z}(S) \in \{0, 1\}^n$  with  $z_i = 1$  if and only if player  $i$  is in the coalition. We also denote  $\mathbf{b}(S) = \mathcal{B}\mathbf{z}(S)$  where  $\mathcal{B}$  is a matrix of resources vectors in the columns.

The following remark can be used to speed up the process of finding the optimal solution for each coalition  $S$  by checking a set of linear constraints on  $\mathbf{z}(S)$ .

*Remark 2* The condition of  $S \in \mathbf{H}_l$  for  $l \in \{1, \dots, \Gamma\}$  is equivalent to:

- If the vertex  $\mathbf{y}^{(l)}$  is non-degenerate,  $(A_{I(\mathbf{y}^{(l)})})^{-1} \mathcal{B}\mathbf{z}(S) \geq 0$ , which is a set of linear constraints on  $\mathbf{z}(S)$ .

- If the vertex  $\mathbf{y}^{(l)}$  is degenerate, i.e., when  $A_{I(\mathbf{y}^{(l)})}$  is not a square matrix, we can list out all combinations of columns of  $A_I^{(l)} \subset A_{I(\mathbf{y}^{(l)})}$  such that each set of columns corresponds to a basis. Afterwards, the condition  $S \in H_l$  if and only if  $(A_I^{(l)})^{-1} \mathcal{B}z(S) \geq 0$ ,  $\forall A_I^{(l)} \subset A_{I(\mathbf{y}^{(l)})}$ .

We give a sketch proof here for the remark. More detailed explanation can be found in chapter 4 of the book by [Bertsimas & Tsitsiklis \(1997\)](#).

The condition  $S \in H_l$  is equivalent to  $\mathbf{y}^{(l)} \in \operatorname{argmin}_{\mathbf{y} \in Y} \mathbf{b}(S)^T \mathbf{y}$ . That means the

dual problem (6) has  $\mathbf{y}^{(l)}$  as one of its optimal solutions, hence, both conditions of feasibility and optimality must be satisfied. The feasibility condition of the vertex in the dual polyhedron is straightforward, however, we also need to check the optimality condition, which is the same as the feasibility condition of the primal problem. Therefore, if a basis  $B$  is optimal in the primal problem, the condition of  $S \in H_l$  is equivalent to  $B^{-1} \mathbf{b}(S) = (A_{I(\mathbf{y}^{(l)})})^{-1} \mathcal{B}z(S) \geq 0$  for the non-degenerate vertex  $\mathbf{y}^{(l)}$ .

Therefore, in the case of the polyhedron of the dual problem has a small number of vertices and there are a large number of game players, we can efficiently find the optimal solution for each coalition  $S$  by checking which value of  $l$  such that  $S \in H_l$ . This step is useful in finding the Shapley value quickly.

*Example 1* To provide an insight to the remark above, we provide an illustrative game with  $r = 2$  resources and  $p = 3$  as follows:

$$\begin{aligned} \text{Primal } (\mathbf{b}) := \max \quad & 3x_1 + 8x_2 + x_3 \\ \text{s.t.} \quad & 5x_1 + 19x_2 + 11x_3 \leq b_1, \\ & 13x_1 + 17x_2 + 23x_3 \leq b_2, \\ & x_1, x_2, x_3 \in \mathbb{R}_+. \end{aligned} \tag{8}$$

The dual problem and its feasible area are shown as following:

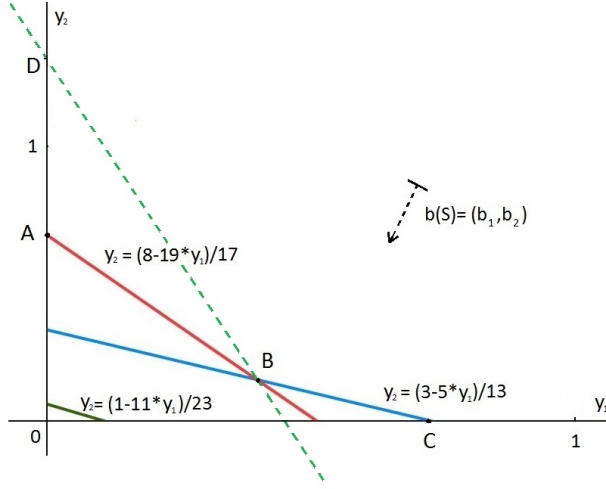
$$\begin{aligned} \text{Dual } (\mathbf{b}) := \min \quad & b_1 y_1 + b_2 y_2 \\ \text{s.t.} \quad & 5y_1 + 13y_2 \geq 3, \\ & 19y_1 + 17y_2 \geq 8, \\ & 11y_1 + 23y_2 \geq 1, \\ & y_1, y_2 \in \mathbb{R}_+. \end{aligned} \tag{9}$$

Suppose that there are only two types of players in our game, i.e.,  $m = 12$  players of the first type with the resource vector  $\mathbf{b} = (50, 37)$  and  $m' = 18$  players of the second type with the resource vector  $\mathbf{b}' = (28, 371)$ . The feasible area of the dual problem has three basic feasible solutions at three extreme points  $A = (0, 0.47)$ ,  $B = (0.33, 0.1)$ ,  $C = (0.6, 0)$ .

Following the arguments of remark 1, we have the resource constraint matrix

$$A = \begin{pmatrix} 5 & 19 & 11 & 1 & 0 \\ 13 & 17 & 23 & 0 & 1 \end{pmatrix},$$

from which we compute the set of active constraints and active rows at each vertices  $\mathbf{y}_A, \mathbf{y}_B, \mathbf{y}_C \in Y$ . In particular, we have  $I(\mathbf{y}_A) = \{2, 4\}$ ,  $I(\mathbf{y}_B) = \{1, 2\}$ , and



**Fig. 1** Dual space of the example with the resource vector  $b(S)$

$$I(\mathbf{y}_C) = \{1, 5\}, \text{ with } A_{I(\mathbf{y}_A)} = \begin{pmatrix} 19 & 1 \\ 17 & 0 \end{pmatrix}, A_{I(\mathbf{y}_B)} = \begin{pmatrix} 5 & 19 \\ 13 & 17 \end{pmatrix}, \text{ and } A_{I(\mathbf{y}_C)} = \begin{pmatrix} 5 & 0 \\ 13 & 1 \end{pmatrix}.$$

For each coalition  $S$  of the game, we need to check which  $l \in \{A, B, C\}$  s.t.  $S \in H_l$ . For the two resource vectors  $\mathbf{b} = (50, 37)$  and  $\mathbf{b}' = (28, 371)$ , the player of type  $\mathbf{b}$  belongs to  $H_A$  and the player of type  $\mathbf{b}'$  belongs to  $H_C$ . Now, considering a coalition  $S^*$  of eight players of type  $\mathbf{b}$  and one player of type  $\mathbf{b}'$ , i.e., the total resource vector of the coalition is  $\mathbf{b}(S^*) = (428, 667)$ . Among the three vertices that represent non-degenerate basic feasible solutions, we can check that the only satisfied inequalities  $(A_{I(\mathbf{y}_B)})^{-1}\mathbf{b}(S^*) \geq 0$  indicates  $S^* \in H_B$ , i.e.,  $B$  is the optimal vertex for the dual problem.

To demonstrate the effect of degeneracy on the dual and the way in which this issue could be overcome, we add the extra constraint  $14y_1 + 4y_2 \geq 5$  to the dual problem (9). The new feasible space has three vertices as  $\{D, B, C\}$  where  $D = (0, 1.25)$ . We have matrix  $A = \begin{pmatrix} 14 & 5 & 19 & 11 & 1 & 0 \\ 4 & 13 & 17 & 23 & 0 & 1 \end{pmatrix}$ , and the sets of active constraints are  $I(\mathbf{y}_C) = \{2, 6\}$ ,  $I(\mathbf{y}_B) = \{1, 2, 3\}$ , and  $I(\mathbf{y}_D) = \{1, 5\}$ .

In this case, the vertex  $B$  becomes degenerate with  $A_{I(\mathbf{y}_B)} = \begin{pmatrix} 14 & 5 & 19 \\ 4 & 13 & 17 \end{pmatrix}$ . We check that the second criteria of the Remark 2 satisfies for all basic representations  $A_I^{(B)} = \left\{ \begin{pmatrix} 14 & 5 \\ 4 & 13 \end{pmatrix}, \begin{pmatrix} 5 & 19 \\ 13 & 17 \end{pmatrix}, \begin{pmatrix} 19 & 14 \\ 17 & 4 \end{pmatrix} \right\}$  of the vertex  $B$ . Since  $(A_I^{(B)})^{-1}\mathcal{B}\mathbf{z}(S) \geq 0$  for all representations, this confirms that  $B$  is still the optimal vertex of the modified problem.

### 3.2 Linear Programming Sensitivity Analysis (LPSA)

The motivation of using LPSA is based on the following observation. Adding a player  $i$  to a coalition  $S$  of other players, in many cases, only slightly modifies the resource vector of the coalition. Sensitivity analysis dictates that the right-hand side values of the Linear Programming (LP) model can vary within certain limits without causing a change in the optimal solution. The LPSA is therefore used to calculate  $v(S \cup \{i\})$  and hence the marginal contribution value  $v(S \cup \{i\}) - v(S)$ , instead of the need to re-evaluate.

The LPSA is applied to calculate the marginal contribution  $v(S \cup \{i\}) - v(S)$  for each coalition  $S$  as follows. First, suppose that the calculation of  $v(S)$  can be done by solving the following standard LP:

$$(\mathbf{P}) \quad v(S) = \min\{\mathbf{f}^T \boldsymbol{\omega} \mid \mathbf{D}\boldsymbol{\omega} = \mathbf{b}(S), \boldsymbol{\omega} \geq 0\},$$

where  $\mathbf{f}$ ,  $\mathbf{D}$  are coefficients derived from  $(\mathbf{c}, \mathbf{A})$  after transforming problem (3) into the standard form with slack variables added.

Using the simplex method in tabular form, a basis  $B$  is optimal if the following two conditions hold (i) Feasibility:  $B^{-1}\mathbf{b}(S) \geq 0$  and (ii) Optimality:  $\mathbf{f}^T - \mathbf{f}_B^T B^{-1}\mathbf{D} \geq 0^T$ .

To calculate  $v(S \cup \{i\})$  in detail, let  $(\mathbf{P}')$  denote the new LP in which the right hand side coefficients  $\mathbf{b} := \mathbf{b}(S) = \{b^k(S)\}_{k=1}^r$  are changed to  $\mathbf{b} + \boldsymbol{\Delta} := \mathbf{b}(S \cup \{i\}) = \mathbf{b}(S) + \mathbf{b}_i = \{b^k(S) + \Delta^k\}_{k=1}^r$ , and where  $\boldsymbol{\Delta} := \{\Delta^k\}_{k=1}^r = \{b_i^k\}_{k=1}^r$ . The optimality condition of  $(\mathbf{P}')$  is still satisfied, as in  $(\mathbf{P})$ . However the new feasibility condition is  $B^{-1}(\mathbf{b} + \boldsymbol{\Delta}) \geq 0$  may not necessarily hold. Then, there are two cases to consider:

- If the resource vector  $\boldsymbol{\Delta} = \mathbf{b}_i$  satisfies the new feasibility condition then the optimal basis of the problem  $(\mathbf{P}')$  is still the same as in  $(\mathbf{P})$ . Hence, there is a closed form outcome of the problem  $(\mathbf{P}')$  as  $v(S \cup \{i\}) = \mathbf{f}_B^T B^{-1}(\mathbf{b} + \boldsymbol{\Delta})$ .
- Otherwise, the optimal solution and the basis of  $(\mathbf{P})$  are used as the starting point for the dual simplex algorithm to solve  $(\mathbf{P}')$  (see [Wolsey & Nemhauser, 1999](#), for details on the dual simplex method).

## 4 LPSA Randomized Algorithm

As discussed in the previous section, while we are able to exploit special structure of the linear production game to provide an improved formulation and perform faster recalculations of the coalition payoffs, the process might still require significant computational resource, especially for large  $n$ . Given the difficulty of the problem, resorting to approximation seems to be the only choice available. In this section, we propose a randomized algorithm combining LPSA and a stratified sampling technique to approximate the Shapley value.

The main drawbacks associated with the stratified sampling approaches reviewed in Section 2.4 include the need to keep track of  $n \times n$  strata, and the number of times the coalition payoffs are to be evaluated. In this section, we propose a new randomized method to take samples which will strengthen the effect of linear programming sensitivity analysis. To this end, we rewrite expression (5) by noticing that the summation indices  $S \subseteq \mathcal{N} \setminus \{i\}$  can be replaced with  $S \subseteq \mathcal{N}$ .

This does not affect the validity of the expression since even if  $i \in S$ , we would then have  $v(S \cup \{i\}) - v(S) = 0$ . We can therefore rewrite the Shapley value equation as follows,

$$\begin{aligned} \phi_i &= \sum_{\substack{S \subseteq \mathcal{N} \\ n-1 \\ k=0}} \frac{|S|!(n-|S|-1)!}{n!} [v(S \cup \{i\}) - v(S)] \\ &= \sum_{k=0}^{n-1} \frac{1}{n-k} \underbrace{\frac{1}{\binom{n}{k}} \sum_{|S|=k} [v(S \cup \{i\}) - v(S)]}_{\mathbb{E}(X_i^k)}. \end{aligned} \quad (10)$$

Here,  $S$  is written for each fixed  $i$  as a random coalition with size  $k$  and that does not include player  $i$ . There are  $\binom{n}{k}$  such coalitions. If  $S$  is uniformly chosen from that sub-population and if we let  $X_i^k$  be the corresponding random variable taking random value  $[v(S \cup \{i\}) - v(S)]$ , then the inner summation is exactly the expectation  $\mathbb{E}(X_i^k)$  of  $X_i^k$ . The idea of a random sampling technique is to draw random samples (i.e., coalitions) and replace that expectation with the sample average.

The main difference between this part and the stratified sampling methods in the literature (Maleki et al., 2013; O'Brien et al., 2015) is the fact that the drawing of the random sample is independent on any specific player. We can therefore approximate  $\mathbb{E}(X_i^k)$  for each and all players using the same sample of coalitions of size  $k$ . After taking a sample coalition  $S$ , we choose all players  $i \notin S$  one by one to form the bigger coalition  $S \cup \{i\}$ . The process saves  $(n - |S| - 1)$  times of recalculating  $v(S)$  for all different player  $i \notin S$ . In addition, instead of finding the marginal contribution  $v(S \cup \{i\}) - v(S)$  of that player  $i$  to coalition  $S$  directly, we utilise the Linear Programming sensitivity analysis to speed up the process.

While the benefits of the new sampling approach are clear from the description above, we should note two small drawbacks. First, the Shapley value estimates  $\hat{\phi}_i, i \in \mathcal{N}$ , are now dependent as they are calculated from the same set of samples. Second, the sizes of the resulting sub-sample  $M_i^k$  might not exactly be population-proportional allocations (eventhough these are in expectations). The first drawback of dependency among  $\hat{\phi}_i, i \in \mathcal{N}$  is not critical, however, as what we deem important for our purposes is the bias and the variances of these estimations. In particular, both factors are calculated separately for each player and hence does not affect the overall quality of the estimation. Similarly, as the whole process is an approximation in itself, the second drawback is not too significant either. We describe the full LPSA randomized scheme in Algorithm 1.

Here we apply a simple random method into the process of sampling from each stratum. The LPSA Randomized algorithm estimates the stratum mean  $\mu_i^k = \mathbb{E}(X_i^k)$  based on the sample average  $\bar{X}_i^k$ . Suppose that the number of samples taken from each stratum is sufficiently large such that the central limit theorem can be invoked to approximate the distribution of  $X_i^k$  by a Gaussian distribution with mean  $\mu_i^k$  and the standard deviation  $\sigma_i^k$ . From the formula  $\phi_i = \frac{1}{n} \sum_{k=0}^{n-1} \mathbb{E}(X_i^k)$ , the corresponding variances and expected values of stratified sampling for player  $i$  is

**Algorithm 1 LPSA Randomized Algorithm**


---

**Input:** The problem instance size  $(p, r, n)$  and LPG data  $(A, B, c)$   
**Initialize:** Divide total  $m$  samples into  $\{m_k\}_{k=0}^{n-1}$  strata by population-proportional allocation, where  $m_k$  is the sample size of stratum  $M_k$   
**for**  $k = 0 : (n - 1)$  **do**  
  **if** the stratum population  $\binom{n}{k} < m_k$  **then**  
    Compute  $v(S \cup \{i\}) - v(S)$  of all  $S \in M_k$   
    Find all marginal values for these strata instead of sampling  
  **else**  
    **for**  $j = 1 : m_k$  **do**  
      Randomly generate a coalition  $S$  of size  $k$   
      Solve the Linear Program and save the basis of the LP problem  
      **for** all player  $i$  not in the coalition  $S$  **do**  
        Rebuild the model with new resource vector of  $S \cup \{i\}$   
        Use LPSA to check if LP basis will change;  
        **if** the basic changes **then**  
          Use dual simplex method to solve LP problem with rhs  $\mathbf{b}(S \cup \{i\})$   
        **else**  
          Apply the closed form to calculate  $v(S \cup \{i\})$   
        **end if**  
        Add  $(v(S \cup \{i\}) - v(S))$  value into  $\mu_i^k$  and  $\sigma_i^k$  calculations of player  $i$   
      **end for**  
    **end for**  
  **end if**  
**end for**

---

calculated as follows:

$$\sigma_i^2 = \text{Var}(\bar{\phi}_i) = \sum_{k=0}^{n-1} \text{Var}\left(\frac{1}{n-k} \overline{X_i^k}\right) = \sum_{k=0}^{n-1} \frac{\text{Var}(\overline{X_i^k})}{(n-k)^2} = \sum_{k=0}^{n-1} \frac{(\sigma_i^k)^2}{(n-k)^2},$$

and

$$\begin{aligned} \mathbb{E}(\bar{\phi}_i) &= \sum_{k=0}^{n-1} \frac{1}{n-k} \mathbb{E}(\overline{X_i^k}) = \sum_{k=0}^{n-1} \frac{1}{n-k} \mu_i^k = \sum_{k=0}^{n-1} \frac{1}{n-k} \frac{1}{\binom{n}{k}} \sum_{|S|=k} [v(S \cup \{i\}) - v(S)] \\ &= \sum_{S \subseteq \mathcal{N}} \frac{|S|!(n-|S|-1)!}{n!} [v(S \cup \{i\}) - v(S)] = \phi_i. \end{aligned} \tag{11}$$

After the computation of the approximate Shapley value, we implement the process of balancing the total budget (Castro et al., 2017; O'Brien et al., 2015) to guarantee the efficiency axiom. The process uses sample averages and variances of the strata to find the maximum likelihood estimates of the Shapley value with budget constraints as follows:

$$\hat{\phi}_i = \bar{\phi}_i - \frac{\sigma_i^2}{\sum_{j=1}^n \sigma_j^2} (\bar{v}(\mathcal{N}) - v(\mathcal{N})),$$

where  $\bar{v}(\mathcal{N}) = \sum_{i=1}^n \bar{\phi}_i$ . We note that  $\mathbb{E}(\bar{v}(\mathcal{N})) = \sum_{i=1}^n \mathbb{E}(\bar{\phi}_i) = \sum_{i=1}^n \phi_i = v(\mathcal{N})$ . Moreover, all four properties of Shapley value are satisfied in expectation for the LPSA

Randomized algorithm as a result of the following calculations:

$$\mathbb{E}(\widehat{\phi}_i) = \mathbb{E}(\overline{\phi}_i) - \frac{\sigma_i^2}{\sum_{j=1}^n \sigma_j^2} (\mathbb{E}(\overline{v}(\mathcal{N})) - v(\mathcal{N})) = \mathbb{E}(\overline{\phi}_i) = \phi_i,$$

and

$$\sum_{i=1}^n \widehat{\phi}_i = \sum_{i=1}^n \overline{\phi}_i - (\overline{v}(\mathcal{N}) - v(\mathcal{N})) = v(\mathcal{N}).$$

In the next section, we will show the effect of combining the LPSA for computing the marginal contribution of a player to a coalition and the proposed stratified sampling approach to evaluate the Shapley value of LPGs.

## 5 Computational Experiments

This section reports the computational experiments undertaken to assess the performance of the proposed method. The algorithms are implemented in MATLAB and run on an Intel-core i5-4570 PC with 3.2 GHz CPU and 4GB RAM. We first describe data generation for small and large-scale instances. We then present results for small scale LPGs with  $n \in \{10, 13, 16, 19, 22, 25\}$ , for which the Shapley value can be computed exactly using a brute force method. We then extend the results to a set of larger size instances with up to 90 players and two types of resource vectors. This particular setting gives us the analytical form of the Shapley value, from which we can judge the accuracy of our approximation methods. We then show the effect of LP sensitivity analysis on the computational time of the LPSA randomized algorithm. Finally, we implement three other randomized algorithms in the literature for comparison purposes.

For small-scale instances, we use the *mean absolute percentage error* (MAPE) as a measure between the approximate and the exact Shapley values, calculated as:  $\text{MAPE}(\phi, \widehat{\phi}) = 100 \cdot \frac{1}{n} \sum_{i=1}^n \left| \frac{\widehat{\phi}_i - \phi_i}{\phi_i} \right|$ , where  $\phi$  is the real Shapley value and  $\widehat{\phi}$  is the approximate Shapley value.

### 5.1 Generating Problem Instances

Small scale LPG instances are generated as follows: For each tuple of  $n$  (the number of players),  $r$  (the number of resources) and  $p$  (the number of products), we generate a problem instance  $(A, B, c)$  of the linear production game. The elements of  $(A, B, c)$  are drawn randomly from a uniform distribution in the interval  $[0, 1]$ .

For large-scale instances, we consider a class of LPGs where there are only two types of players  $E$  and  $F$ , i.e., all players of the same type in the games have the same resource vector. For these instances, the Shapley value can be determined exactly in closed form without finding  $2^n$  values of  $v(S)$ . In particular, we generate a Linear production Game based on example 4.4 of [Bjorndal & Jornsten \(2009\)](#) with  $p = 5$  products and  $r = 10$  resources and  $n = 40$  players. The instances are characterized by the following parameter settings:



$$A = \begin{bmatrix} 7 & 3 & 5 & 2 & 1 \\ 6 & 9 & 9 & 5 & 10 \\ 6 & 3 & 3 & 4 & 3 \\ 9 & 5 & 4 & 2 & 1 \\ 3 & 6 & 10 & 2 & 4 \\ 4 & 5 & 1 & 3 & 8 \\ 4 & 3 & 4 & 2 & 3 \\ 7 & 9 & 1 & 1 & 7 \\ 5 & 8 & 9 & 3 & 2 \\ 2 & 6 & 3 & 10 & 2 \end{bmatrix}; c = \begin{bmatrix} 53 \\ 57 \\ 49 \\ 34 \\ 41 \end{bmatrix}; \mathbf{b}_E = \begin{bmatrix} 4 \\ 0 \\ 9 \\ 0 \\ 19 \\ 0 \\ 17 \\ 4 \\ 28 \\ 0 \end{bmatrix}; \mathbf{b}_F = \begin{bmatrix} 15 \\ 40 \\ 11 \\ 22 \\ 7 \\ 22 \\ 0 \\ 22 \\ 0 \\ 23 \end{bmatrix}. \quad (12)$$

If  $n$  is the total number of players, we calculate the number of players of type  $E$  as  $n_E = \lfloor 2n/5 \rfloor$  and the number of players of type  $F$  is  $n_F = n - n_E$ . In the problem instance where  $n = 40$ , therefore, there are  $n_E = 16$  players of type  $E$  and  $n_F = 24$  players of type  $F$ . Note that all coalitions  $S$  are composed of  $p_E$  players of type  $E$  and  $p_F$  players of type  $F$ , such that  $0 \leq p_E \leq n_E$  and  $0 \leq p_F \leq n_F$ . Therefore, the total number of coalition values  $v(S)$  needed to be evaluated is  $n_E \times n_F$  instead of  $2^n$ . This reduces the required computational effort for the experiment.

We compute the exact Shapley value of this game as  $\phi_E = 191.03$  and  $\phi_F = 57.88$ . In this situation, the value of the grand coalition is  $v(\mathcal{N}) = 4445.6$ . Although a coalition involving only one type of player does not result in any product, a balance between the two types of players generates the most outcome. Player  $E$  has a higher payoff compared to player  $F$  because of the smaller number of players.

For each configuration  $(n, r, p)$ , we generate problem instances  $(A, B, c)$  to test our approximation algorithm. With each problem instance, the approximation algorithm outputs different results for each run (trials). We compute the average value for these calculations in the final result, reported in the following section.

## 5.2 Effects of using LP Sensitivity Analysis

In this section, we computationally test the effect of using LP sensitivity analysis by generating LPG instances with  $p = 19$  products,  $r = 27$  resources, and with varied number of players. Suppose we have a total of 10000 samples to distribute in all the computation of the LPSA randomized algorithm. Let us denote  $P_{bnc}$  as the percentage of times when the calculation of the marginal contribution  $v(S \cup \{i\}) - v(S)$  does not change the basis (i.e., can be calculated in closed form). Let  $T_{SA}$  and  $T_{woSA}$  be the calculation times (in seconds), respectively, for the algorithms running with and without the LP sensitivity analysis. The exact time to compute the Shapley value is shown by  $T_{exact}$ .

Table 1 shows the results over 100 trials for the instances described above. When the number of players in the game is small, it is possible to compute the exact Shapley value without resorting to randomized methods (e.g., the number of coalitions is  $2^n < 10^4$  when  $n \leq 13$ ). However, as the number of players increases, the computational time of finding the correct Shapley value grows exponentially. In particular, the computational time of the exact Shapley value was more than ten hours for  $n \geq 25$ , indicated by the \* in the table.

$n$	13	16	19	22	25	28
$P_{bnc}(\%)$	7.68	12.37	15.9	14.16	15.83	16.39
$T_{SA}(s)$	41.02	37.8	35.84	35.82	34.75	34.42
$T_{woSA}(s)$	42.43	41.16	40.48	39.83	39.36	38.71
$T_{exact}(s)$	21.63	150.81	1415	9514	*	*

**Table 1** Average 100 trials for problem instances of small sizes with  $p = 19, r = 27$  and total 10000 samples.

For larger instances, we generate ten problem instances for each choice of  $n$  and implement 10 trials for each problem instance. Table 2 presents the computational results in a similar way to Table 1 for these instances, but without the exact Shapley value calculation statistics.

$n$	30	40	50	60	70	80	90
$P_{bnc}(\%)$	19.23	27.48	32.67	35.93	35.68	37.12	45.16
$T_{SA}(s)$	34.11	28.82	26.51	26.51	26.27	24.59	21.36
$T_{woSA}(s)$	38.64	37.86	37.02	35.87	38.43	35.25	35.87

**Table 2** Average 10 trials for 10 problem instances of large size with  $p = 19, r = 27$  and total 10000 samples.

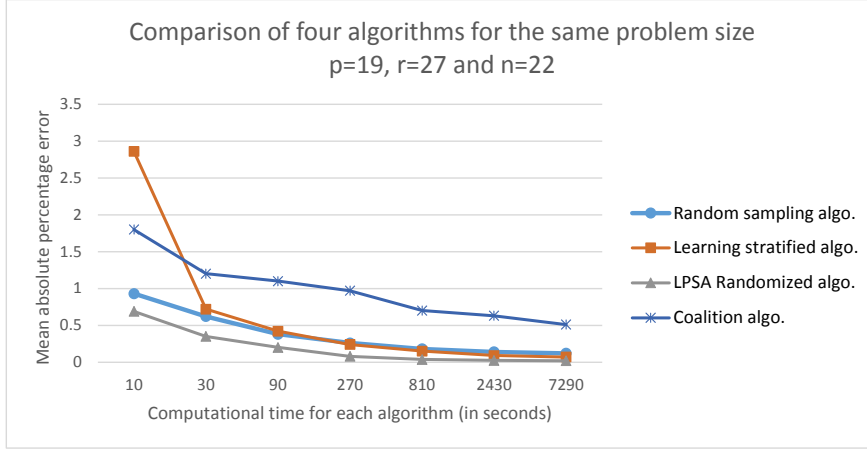
The results shown in Tables 1 and 2 suggest that as the number  $n$  of players increases, the use of sensitivity analysis is more effective as each player can contribute proportionally more to a small group compared to a bigger one; making it beneficial to use sensitivity analysis in the case of large games. For example, when  $n$  goes beyond 30,  $P_{bnc}$  shows that the time to compute marginal contributions can be reduced in more than 20% of the total 10000 samples. For this reason, computational time of the LPSA is significantly less than that without the use of sensitivity analysis.

### 5.3 Effects of using the Stratified Sampling technique

This section presents comparison results with three other approximation algorithms for comparison that we implemented, namely simple random sampling (adapted from Castro et al., 2009), stratified coalition sampling (adapted from Maleki et al., 2013), and learning stratified sampling (adapted from O'Brien et al., 2015). We apply the Linear Programming Sensitivity Analysis technique for the calculation of marginal contributions in all algorithms. A brief review of the different steps among these sampling techniques was given in Section 2.4.

For comparison purposes, we generate random instances  $(A, B, c)$  with  $p = 19, r = 27$  and  $n = 22$ . The Shapley value is computed by four different algorithms using MAPE to compare the sampling errors. Figure 2 plots the four MAPEs against time to compare convergence of the algorithms. In the implementation, we use different number of samples for each player. The total number of samples to

use for each sampling algorithm depends on the running time of that algorithm; hence there is the trade-off between the accuracy and computational time.

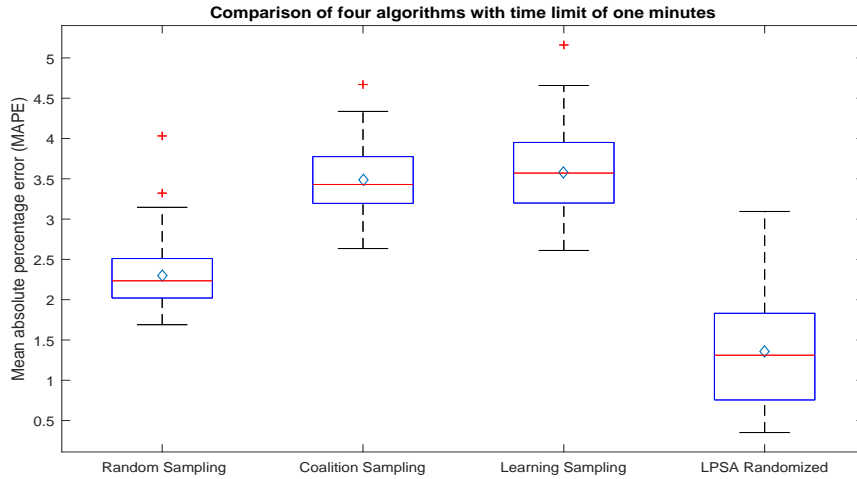


**Fig. 2** Comparison of four algorithms using different sampling techniques

Figure 2 shows that the MAPEs of all algorithms decrease to less than 1% as each algorithm runs more than five minutes, but the number of samples also increases proportionally. However, there are some differences in computational times among these algorithms, where the coalition sampling is not competitive compared to the three other algorithms. The LPSA randomized algorithm performs the best compared to others with respect to the MAPE, particularly when the computational time is limited.

Figure 3 shows a boxplot which presents the spread of sampling errors for the four different algorithms under the same running time for large-scale problem instances with  $p = 5, r = 10$  and  $n = 40$  using the parameters shown by (12). These results show that, on average, the LPSA randomized algorithm is the best sampling algorithm compared to others in terms of the MAPE for computing the Shapley value. The coalition sampling and the Learning stratified sampling do not seem to be as effective for this particular set of problem instances as the simple random sampling and LPSA randomized algorithms. We note that a stratified version of the random sampling algorithm is likely to improve the results obtained with the random sampling algorithm, but the extent to which this version would to compete with the LPSA is an interesting question that we leave for further research.

Finally, Table 3 shows how the MAPE changes for four algorithms with the number  $n$  of players, where ten problem instances are generated for each value  $n$  and the averages of ten runs are presented. The results suggest that, under a fixed amount of running time, the LPSA randomized algorithm is, on average, the best compared to the other three.



**Fig. 3** The MAPE box-plot of four algorithms to approximate the Shapley value of a large-scale LPG

Number of players	30	40	50	60	70	80	90
Random sampling algo.	2.48	3.81	3.77	4.51	4.78	3.94	5.03
Coalition sampling algo.	3.93	5.44	5.78	6.50	6.73	6.14	6.63
Learning algo.	4.26	5.88	5.10	5.70	5.53	7.18	6.98
LPSA Randomized algo.	1.56	1.37	2.11	2.39	2.52	2.93	3.65

**Table 3** Average MAPE of 10 trials for 10 problem instances of size  $p = 5, r = 10$  with 5 minute time limit.

#### 5.4 Error Estimation in the LPSA Randomized algorithm

In this section, we present some numerical results to show the combined effect of both LPSA and the stratified sampling method, and show that they work well in combination. The test instances include some large-scale games with two types of players as described in section 5.1.

Initially, we present the results of the experiments with the LPSA Randomized algorithm if the sampling budget and number of players in the game are modified. When the total number of players  $n$  ranges from 30 to 90, the total sampling budget  $m$  increases from 1000 to  $3^5 \cdot 1000$ . We then describe the MAPEs of the algorithm showing the computational error between the real Shapley values and the calculated values in Table 4. In general, a game with a fixed number of player has smaller error if there are more samples to approximate the Shapley value. It can be seen from the table that as the number of players increases, the sampling budget also needs to increase to keep the sampling error small.

Additional results for instances with  $p = 5, r = 10$  and  $n = 40$  are shown in Table 5, where the numbers of players for two types  $E$  and  $F$  are varied. For these instances, we also use the *root mean squared error* (RMSE) as a quadratic

$m \setminus n$	30	40	50	60	70	80	90
1000	10.48	13.76	15.36	14.48	11.96	13.74	16.39
3000	7.59	8.78	8.20	10.32	8.76	12.22	11.12
9000	4.63	4.09	5.52	4.71	6.75	6.18	6.29
27000	2.18	2.52	2.62	2.33	3.80	4.57	3.87
81000	1.66	1.26	1.75	1.41	1.66	1.58	1.47
243000	0.69	0.68	0.87	0.93	0.85	1.15	0.97

**Table 4** MAPEs of LPSA Randomized algorithm for large-scale problem instances with different sample sizes

scoring rule which measures the average magnitude of the error, calculated as:

$\text{RMSE}(\phi, \hat{\phi}) = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{\phi}_i - \phi_i)^2}$ , where  $\phi$  is the real Shapley value and  $\hat{\phi}$  is the approximate Shapley value. The real Shapley values and the average MAPE and RMSE for the LPSA is given in Table 5.

# players of $(E, F)$	(5,35)	(10,30)	(15,25)	(20,20)	(25,15)	(30,10)	(35,5)
Shapley value of $E$	279.3	256.6	211.8	95.5	18.5	4.7	1.2
Shapley value of $F$	1.9	11.3	41.8	145.5	215.1	232.4	238.2
$v(\mathcal{N})$	1463	2905	4221	4820	3689	2465	1233
MAPE	14.56	7.42	2.1	0.64	4.55	16.98	28.58
RMSE	0.87	1.85	1.54	0.81	2.31	1.84	0.95

**Table 5** The approximation errors of LPSA Randomized algorithm on large-scale problem instances with 5 minutes time limit.

As can be seen from Table 5, as the number of players of one type is reduced, the percentage of values that player type can claim from the  $v(\mathcal{N})$  in the Shapley value increases. Moreover, the grand coalition value is largest when there is a balance between two types of players. The reason that some problem instances have large MAPEs is due to the fact that the Shapley values of one player become very small, and any slight changes in the value of that approximate Shapley value will affect the MAPE of the algorithm. The RMSE also confirm this observation as the errors are quite small.

## 6 Conclusions

This paper described new computational methods for calculating the Shapley value of Linear Production Games. We showed how to find a closed form solution of the Shapley value in some particular cases of the game. For larger-scale LPGs, we proposed a randomized algorithm combining both Linear Programming sensitivity analysis and a stratified sampling method. The computational results that compared the LPSA randomized method with three other sampling techniques in the

literature indicated the superiority of the proposed algorithm. We expect that such LPSA randomized algorithm could be used to approximate the Shapley value for some other cooperative games, such as those associated with minimum spanning trees, assignment and network synthesis.

### Acknowledgement

The authors thank two anonymous reviewers for their very constructive comments. The first author gratefully acknowledges the Faculty Scholarship provided by Southampton Business School.

### References

- Ando, K. (2012). Computation of the Shapley value of minimum cost spanning tree games: #P-hardness and polynomial cases. *Japan Journal of Industrial and Applied Mathematics*, 29, 385–400.
- Bertsimas, D. & Tsitsiklis, J. N. (1997). *Introduction to Linear Optimization*. Massachusetts: Athena Scientific Belmont.
- Bilbao, J.M., Fernandez, & J.R., Losada, A.J., Lopez, J.J. (2000). Generating functions for computing power indices efficiently. *Top*, 8(2), 191–213.
- Bjorndal, E., & Jornsten, K. (2009). Lower and upper bounds for linear production games. *European Journal of Operational Research. Production, Manufacturing and Logistics*, 196(2), 476–486.
- Borm, P, Hamers, H., & Hendrickx, R. (2001). Operations research games: A survey. *TOP*, 9(2), 139–216.
- Castro, J., Gómez, D., & Tejada J. (2009). Polynomial calculation of the Shapley value based on sampling. *Computers and Operations Research*, 36(5), 1726–1730.
- Castro, J., Gómez, D., Monila, E., & Tejada J. (2017). Improving polynomial estimation of the Shapley value by stratified random sampling with optimum allocation. *Computers and Operations Research*, 82, 180–188.
- Cochran, W.G. (2007). *Sampling Techniques*, 3rd Wiley Student Edition.
- Deng, X., & Papadimitriou, C.H. (1994). On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, 19(2), 257–266.
- Fernández, F.R., Fiestras-Janeiro, M.G., Garca-Jurado, I., & Puerto, J. (2005). Competition and Cooperation in Non-Centralized Linear Production Games. *Annals of Operations Research*, 137(1), 91–100.
- Gómez, D., González-Arangüena, E., Manuel, C., Owen, G., Pozo, M., & Tejada, J. (2003). Centrality and power in social networks: a game theoretic approach. *Mathematical Social Sciences*, 46(1), 27–54.
- Granot, D., & Huberman, G. (1981). Minimum cost spanning tree games. *Mathematical Programming*, 21, 1–18.
- Granot, D. (1986). A generalized linear production model: a unifying model. *Mathematical Programming*, 34, 212–222.
- Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables, *Journal of the American Statistical Association*, 58(301), 13–30.

- Lozano, S. (2013). DEA production games. *European Journal of Operational Research*, 231(2), 405–413.
- Maleki, S., Tran, T.L., Hines, G., Rahwan, T., & Rogers, A. (2013). Bounding the Estimation Error of Sampling-based Shapley Value Approximation. *CoopMAS 2014, AAMAS 14*.
- Mann, I., & Shapley, L.S. (1960). Values for large games, IV: Evaluating the Electoral College by Monte Carlo Techniques. The Rand Corporation, Technical report.
- Michalak, T.P., Aadithya, K.V., Szczepanski, P.L., Ravindran, B., & Jennings, N.R. (2013). Efficient Computation of the Shapley Value for Game-Theoretic Network Centrality. *Journal of Artificial Intelligence Research*, 46, 607-650.
- Nishizaki, I., Hayashida, T., Shintomi, Y. (2016). A core-allocation for a network restricted linear production game. *Annals of Operations Research*, 238(1), 389–410.
- O'Brien, G., El Gamal, A., & Rajagopal, R. (2015). Shapley Value Estimation for Compensation of Participants in Demand Response Programs, *IEEE Transactions on Smart Grid*, 6(6), 2837–2844.
- Owen, G. (1975). On the core of linear production games. *Mathematical Programming*, 9, 358–370.
- Shapley, L.S. (1953). A value for n-person games. *Contributions to the Theory of Games II*, 307–317, Princeton University Press.
- Shapley, L.S., & Shubik, M. (1954). A Method for Evaluating the Distribution of Power in a Committee System. *American Political Science Review*, 48, 787–792.
- Shapley, L.S., & Shubik, M. (1971). The assignment game I: the core. *International Journal of Game Theory*, 1, 111–130.
- Tamir, A. (1991). On the core of network synthesis games. *Mathematical Programming*, 50(1), 123–135.
- Weber, R.J. (1988). Probabilistic values for games. In A. Roth (Ed.), *The Shapley value: Essays in honor of Lloyd S. Shapley*. Cambridge University Press, 101–119.
- Wolsey, L.A., & Nemhauser, G.L. (1999). *Integer and Combinatorial Optimization*, John Wiley & Sons.
- Young, H. P. (1985). Monotonic solutions of cooperative games, *International Journal of Game Theory*, 14(2), 65–72.