Sharing delay costs in stochastic scheduling problems with delays

J.C. Gonçalves-Dosantos[1], I. García-Jurado, J. Costa

Grupo MODES, Departamento de Matemáticas, Universidade da Coruña,
Campus de Elviña, 15071 A Coruña, Spain

Abstract

An important problem in project management is determining ways to distribute amongst activities the costs that are incurred when a project is delayed because some activities end later than expected. In this study, we address this problem in stochastic projects, where the durations of activities are unknown but their corresponding probability distributions are known. We propose and characterise an allocation rule based on the Shapley value, illustrate its behaviour by using examples, and analyse features of its calculation for large problems.

Keywords. Project management, scheduling, stochastic durations, delay cost, cooperative game theory, Shapley value.

1 Introduction

Project management is a field within operations research that provides managers with techniques to select, plan, execute, and monitor projects. An important issue in project management is time management, which generally call for careful planning of project activities to meet various project delivery dates, especially the final delivery date. Normally, a delay in the final delivery date incurs a cost that is often specified by contract. Sometimes, projects are not developed by one agent but a group of agents. When there is a delay in one of such joint projects, the manner of allocating the delay cost amongst the several participating agents may not be clear. This paper deals with the problem of sharing delay costs in a joint project by using cooperative game theory. We consider that the study of this problem from

[1]Corresponding author. E-mail: juan.carlos.goncalves@udc.es.

the point of view of game theory is very pertinent, since the legal systems of many countries contemplate the need for those responsible for the delay in the execution of a contract to compensate those harmed by the damage resulting from such delay. For example, article 1101 of the Civil Code currently in force in Spain states that "those who, in the performance of their obligations, incur in malice, negligence or arrears are subject to compensation for damages caused". However, the regulation of how such damages are compensated, especially in the case of concurrent fault, is generally not very developed, so that legal agents may require external assistance from the academic and scientific world to support their arguments.

In the last few years, several papers have been written proposing and studying allocation rules for delay costs. Bergantiños and Sánchez (2002) proposed a rule based on the serial cost-sharing problem. Brânzei et al. (2002) provided two rules using, respectively, a game theoretical and bankruptcy-based approach. In Castro et al. (2007), the core of a class of transferable utility cooperative game (in short, a TU-game) arising from a delay cost-sharing problem was studied. In Bergantiños et al. (2018), a consistent rule based on the Shapley value was introduced and analysed. Estévez-Fernández et al. (2007) and Estévez-Fernández (2012) dealt with some classes of TU-games associated with projects whose activities might have been delayed or advanced by generating delay costs or acceleration benefits of the corresponding projects. Curiel (2011) studied situations in which companies can cooperate in order to decrease the earliest completion time of a project that consists of several tasks. Cooperative game theory is used to model those situation, and conditions for the core of the corresponding games are non-empty are derived. In San Cristóba (2014) a practical example is given of the use of cooperative games to allocate delay costs between the different activities in a project. Finally, Briand and Billaut (2011), Briand et al. (2017) and Bergantiños and Lorenzo (2019) adopted a non-cooperative approach and addressed some strategic aspects in project scheduling where players responsible for activities can choose strategies that affect their durations. All these papers tackle *deterministic scheduling problems with delays*. One such problem is that of a delayed deterministic project. By *deterministic project*, we mean a set of activities to be performed with respect to an order of precedence and a description of their estimated durations; by *delayed deterministic project*, we mean a project that has been performed, description of the observed durations of the activities according to which the project has lasted longer than expected, and cost function that indicates the delay cost associated with the durations of the activities.

A natural extension of deterministic problems with delays can be found in *stochastic scheduling problems with delays*, which we introduce and analyse in this study. In our extension we deal with stochastic projects, in which activity durations are described by giving their probability distributions rather than their estimates (as is done in the deterministic case). To the best of our knowledge, these problems have not been treated in literature, although Castro et al. (2014) considered the problem of allocating slacks in a stochastic PERT network,[1] which is a related but different problem. Tanimoto et al. (2000) introduced a variation of the Shapley value for stochastic cost games, their model being an alternative to the stochastic cooperative games in Suijs et al. (1999). These two papers deal with general TU-games (not with the particular class we are considering in this paper) and, though they might have some connections with our approach, they are concerned with the different problem of how to allocate the risk according to the risk acceptance level for each player in a general cooperative game whose characteristic function is stochastic. Herroelen and Leus (2005) surveyed literature on project management under uncertainty. In a stochastic scheduling problem with delays, the manager has a description of the probability distributions of the random variables modelling the durations of the activities instead of simply their estimated durations. In most cases, managers have information about random variables—for instance, their empirical distributions—based on the durations of similar activities in past projects of the same type.

The remainder of this paper is organised as follows. In Section 2, we motivate the interest of the stochastic scheduling problems with delays and introduce them formally; we also discuss the main differences between the deterministic approach, usually adopted in literature, and our novel stochastic approach. In Section 3, we propose an allocation rule based on the Shapley value in this context and characterize it using the property of balancedness; basically, this property states that, for every pair of activities i and j, the effect of the elimination of i on the allocation to j is equal to the effect of the elimination of j on the allocation to i. We also show that the Shapley rule in this context satisfies a list of interesting properties and illustrate its performance by using two examples and a simulation experiment. Finally, Section 4 addresses some computational issues related to our rule. In particular, we illustrate the implementation of the estimation of the Shapley rule through its pseudocode, from which it is easy to check that its computational complexity is $O(n^4)$ and, moreover, we show by examples that it is possible to estimate the Shapley rule for stochastic scheduling problems with delays in an acceptable time, even if there are hundreds of activities, by using a desktop computer and free software.

[1] PERT is the acronym of Program Evaluation and Review Technique, a tool used in project management, first developed by the United States Navy in the 1950s.

2 The problem

In this section, we describe the problem with which we deal. We first formally introduce a deterministic scheduling problem with delays mainly following Bergantiños et al. (2018):

Definition 2.1. *A deterministic scheduling problem with delays P is a tuple $(N, \prec, x^0, x, C)$ where:*

- *N is the finite non-empty set of activities.*
- *$\prec$ is a binary relation over N satisfying asymmetry and transitivity. For every $i, j \in N$, we interpret $i \prec j$ as "activity j cannot start until activity i has finished".*
- *$x^0 \in \mathbb{R}^N$ is the vector of planned durations. For every $i \in N$, x_i^0 is a non-negative real number indicating the planned duration of activity i.*
- *$x \in \mathbb{R}^N$ is the vector of actual durations. For every $i \in N$, x_i is the non-negative duration of activity i.*[2]
- *$C : \mathbb{R}^N \to \mathbb{R}$ is the delay cost function. We assume that C is non-decreasing (i.e., $y_i \leq z_i\ \forall i \in N \Rightarrow C(y) \leq C(z)$),*[3] *and that $C(x^0) = 0$.*

We denote by $\mathcal{P}^N$ the set of deterministic scheduling problems with delays with player set N, and by $\mathcal{P}$, the set of deterministic scheduling problems with delays.

Note that the first three items of a deterministic scheduling problem with delays characterise a project. Operational researchers have developed several methodologies for project management. In particular, the minimum duration of a project $(N, \prec, x^0)$, provided that all restrictions imposed by $\prec$ are satisfied, can be obtained as the solution of a linear programming problem, and thus, can be easily computed. We denote the minimum duration of $(N, \prec, x^0)$ by $d(N, \prec, x^0)$. Alternatively, $d(N, \prec, x^0)$ can be calculated using a project planning methodology like PERT (see, for instance, Hillier and Lieberman (2001) for details on project planning). The delay cost function C in Definition 2.1 is rendered in a general way but typically depends on the minimum duration of the project, i.e., $C(y) = c(d(N, \prec, y))$ for a non-decreasing function $c : \mathbb{R} \to \mathbb{R}$ with $c(d(N, \prec, x^0)) = 0$.

In a deterministic scheduling problem with delays P, the main question to be answered is how to allocate $C(x)$ amongst the activities in a fair way. This issue has been taken up, for instance, in Bergantiños et al. (2018); they introduce the Shapley rule in this context.

Definition 2.2. *A rule for deterministic scheduling problems with delays is a map φ on $\mathcal{P}$ that assigns to each $P = (N, \prec, x^0, x, C) \in \mathcal{P}^N$ a vector $\varphi(P) \in \mathbb{R}^N$ satisfying:*

[2] In Bergantiños et al. (2018) it is assumed that $x_i \geq x_i^0$ for all $i \in N$.

[3] Bergantiños et al. (2018) does not assume that C is non-decreasing.

1. *Efficiency (EFF).* $\sum_{i \in N} \varphi_i(P) = C(x)$.
2. *Null Delay (ND).* $\varphi_i(P) = 0$ *when* $x_i = x_i^0$.

Take a deterministic scheduling problem with delays $P \in \mathcal{P}^N$. We denote by v^P the TU-game with set of players N given by

$$v^P(S) = C(x_S, x^0_{N \setminus S})$$

for all $S \subseteq N$ (where $x_S, x^0_{N \setminus S}$ denotes the vector in $\mathbb{R}^N$ whose i-th component is x_i if $i \in S$ or x_i^0 if $i \in N \setminus S$).

Definition 2.3. *The Shapley rule for deterministic scheduling problems with delays Sh is defined by* $Sh(P) = \Phi(v^P)$, *where* $\Phi(v^P)$ *denotes the proposal of the Shapley value for* v^P.

For those unfamiliar with cooperative game theory, a TU-game is a pair (N, v) where N is a non-empty finite set, and v is a map from 2^N to $\mathbb{R}$ with $v(\emptyset) = 0$. We say that N is the player set of the game and v is the characteristic function of the game, and we usually identify (N, v) with its characteristic function v. We denote by G^N the set of all TU-games with player set N, and by G the set of all TU-games. The Shapley value is a map Φ that associates with every TU-game (N, v) a vector $\Phi(v) \in \mathbb{R}^N$ satisfying $\sum_{i \in N} \Phi_i(v) = v(N)$ and providing a fair allocation of $v(N)$ to the players in N. The explicit formula of the Shapley value for every TU-game (N, v) and every $i \in N$ is given by:

$$\Phi_i(v) = \sum_{S \subseteq N \setminus i} \frac{(|N| - |S| - 1)! \; |S|!}{|N|!} (v(S \cup i) - v(S)).$$

Since its introduction by Shapley (1953), the Shapley value has proved to be one of the most important rules in cooperative game theory and has applications in many practical problems (see, for instance, Flores et al. (2007)).

Bergantiños et al. (2018) showed that the Shapley value has good properties in this context and provided an axiomatic characterisation of their Shapley rule by using a consistency property. In this paper, we introduce a generalization of the model and the Shapley rule described above by assuming that the durations of the activities are stochastic. Let us first introduce and motivate interest in our model.

Definition 2.4. *A stochastic scheduling problem with delays SP is a tuple* $(N, \prec, X^0, x, C)$ *where:*

- *N is the finite non-empty set of activities.*
- *$\prec$ is a binary relation over N satisfying asymmetry and transitivity.*
- *$X^0 \in \mathbb{R}^N$ is a vector of independent random variables. For every $i \in N$, X_i^0 is a non-negative random variable describing the duration of activity i.*

- *$x \in \mathbb{R}^N$ is the vector of actual non-negative durations.*
- *$C : \mathbb{R}^N \to \mathbb{R}$ is the delay cost function. We assume that C is non-negative and non-decreasing.*

We denote by $\mathcal{SP}^N$ the set of stochastic scheduling problems with delays with player set N, and by $\mathcal{SP}$ the set of all stochastic scheduling problems with delays.

Like in the deterministic case, the first three items of a stochastic scheduling problem with delays characterize a stochastic project. The minimum duration of a stochastic project $(N, \prec, X^0)$ is a random variable whose distribution is, in general, difficult to obtain from a theoretic point of view, but easy to estimate using simulation techniques. Note that in a stochastic scheduling problem with delays, the durations are non-negative random variables instead of non-negative numbers. In general, the duration of an activity can now take any non-negative real value, and a condition generalising $C(x^0) = 0$ as in Definition 2.1 cannot be stated. In the stochastic setting, a delay in an activity is unclear. If the actual duration of an activity is longer than the upper bound of its distribution support, it has thus been delayed. Moreover, if its duration is in the 99th percentile of the distribution of its duration, one may think that it has been delayed somewhat. However, what should we think when its actual duration is in the 56th percentile? In the deterministic setting, we can clearly observe when an activity has been delayed. Another novelty in the stochastic setting is that an activity may somehow be delayed, but it may also somehow be ahead of schedule (for instance, when its duration is in the first percentile). In the deterministic setting, by contrast, the case $x_i < x_i^0$ is generally discarded. In any case, although we propose our model in general, our objective is to distribute delay costs when they occur (because $P(x_i \geq X_i^0)$ is large, at least for some $i \in N$), and in situations in which there should not be delays a priori, in the sense that $P(C(X^0) = 0)$ is large.

In Definitions 2.1 and 2.4 we assume that C is a non-negative function. As it was remarked in Section 1, some papers consider that activities can be delayed or advanced and, consequently, there may be delay costs or acceleration benefits. We do not take such an approach in this article but, if we do (for instance dropping the non-negativeness of C), the analytical results should not change significantly.

We give next the definition of a rule in this setting. As the meaning of a delay is not clear, this definition does not contain a kind of null delay property, as in Definition 2.2.

Definition 2.5. *A rule for stochastic scheduling problems with delays is a map ψ on $\mathcal{SP}$ that assigns to each $SP = (N, \prec, X^0, x, C) \in \mathcal{SP}^N$ a vector $\psi(SP) \in \mathbb{R}^N$ satisfying $\sum_{i \in N} \psi_i(SP) = C(x)$.*

A first approach to deal with a stochastic scheduling problem with delays is to build from it an associated deterministic problem. More precisely, for a given $SP = (N, \prec, X^0, x, C) \in \mathcal{SP}^N$, it is natural to associate with it the problem $\overline{SP} = (N, \prec, E(X^0), x, C)$, where $E(X^0) = (E(X_i^0))_{i \in N}$, $E(X_i^0)$ denotes the mathematical expectation of random variable X_i^0. This approach encounters a technical obstacle: $\overline{SP}$ is not always a deterministic scheduling problem with delays in the sense of Definition 2.1 because $C(E(X^0))$ may be different from zero. This obstacle can be overcome with small adjustments in the definition of an associated deterministic problem. Besides, in many particular examples, we do not encounter this obstacle. In any case, this approach is not the most appropriate because it does not use all the relevant information given in the original problem. Let us illustrate this shortcoming in the following example:

Example 2.1. *Consider the stochastic scheduling problem with delays $SP = (N, \prec, X^0, x, C)$ given by:*

N	1	2
$\prec$	-	-
X^0	$U(0,10)$	$U(2,8)$
x	7	7

and, for every $y \in \mathbb{R}^N$,

$$C(y) = \begin{cases} 0 & \text{if } d(N, \prec, y) \leq 6, \\ d(N, \prec, y) - 6 & \text{otherwise.} \end{cases}$$

Note that for all $i \in N$ the i-th column displays:

- *Activities that precede activity i. In this example, $\prec = \emptyset$, i.e., the two activities can be carried out simultaneously. In general, the row corresponding to $\prec$ only shows the immediate precedences, i.e., some elements of $\prec$, but the entire $\prec$ can be easily obtained as the smallest transitive binary relation over N that contains the given elements of $\prec$. An illustration of this can be found in Example 3.1.*
- *The distribution of X_i^0. In this case, X_1^0 and X_2^0 are random variables with a uniform distribution of $U(0,10)$ and $U(2,8)$, respectively.*
- *x_i, the duration of i; in this case, $x = (7,7)$.*

Note that in this example, $E(X_1^0) = E(X_2^0) = 5$, and activities 1 and 2 are indistinguishable in $\overline{SP}$. Hence, the anonymity property satisfied by the Shapley rule for deterministic scheduling problems with delays (see Bergantiños et al. (2018)) implies that $Sh(\overline{SP}) = (\frac{1}{2}, \frac{1}{2})$. However, activities 1 and 2 are actually distinguishable in SP because the expected duration of the project conditioned to $x_1 = 7$ is $E(C(7, X_2^0)) = 13/12$ and the expected duration of the project conditioned to $x_2 = 7$ is $E(C(X_1^0, 7)) = 29/20 > 13/12$.

It seems that a fair rule should take this into account and allocate to activity 2 *a larger part of the delay cost.*

In the next section, we provide a rule for stochastic scheduling problems with delays that overcomes the technical obstacle described above and, more importantly, the drawback described in Example 2.1.

3 Shapley rule for stochastic scheduling problems with delays

In this section, we define and study the Shapley rule for stochastic scheduling problems with delays. Take a stochastic scheduling problem with delays $SP \in \mathcal{SP}^N$. We denote by v^{SP} the TU-game with set of players N given by

$$v^{SP}(S) = E(C(x_S, X^0_{N\setminus S}))$$

for all non-empty $S \subseteq N$,[4]

Definition 3.1. *The Shapley rule for stochastic scheduling problems with delays SSh is defined by* $SSh(SP) = \Phi(v^{SP})$, *where* $\Phi(v^{SP})$ *denotes the proposal of the Shapley value for* v^{SP}.

This rule inherits many properties of the Shapley value. For instance, it is easy to check that it satisfies the correspondingly modified versions of the properties proved in Bergantiños et al. (2018) for the Shapley rule for deterministic scheduling problems with delays. Let us remember some of those properties.

We start with some notation. Take a finite set N. A permutation of N is a bijective map $\pi : N \to N$. Denote by Π_N the set of permutations of N.

Anonimity. A rule for stochastic scheduling problems with delays ψ satisfies anonimity if for all $SP = (N, \prec, X^0, x, C)$, all $\pi \in \Pi_N$ and all $i \in N$, it holds that

$$\psi_i\,(SP) = \psi_{\pi(i)}\,(SP^\pi)$$

where SP^π denotes the problem $(N, \prec_\pi, \pi(X^0), \pi(x), C^\pi)$ given by:

- For all $i, j \in N$, $i \prec_\pi j$ if and only if $\pi(i) \prec \pi(j)$,
- $\pi(X^0)$ is the vector of random variables whose i-th component is $X^0_{\pi^{-1}(i)}$,
- $\pi(x)$ is the vector in $\mathbb{R}^N$ whose i-th component is $x_{\pi^{-1}(i)}$,
- $C^\pi(\pi(y)) = C(y)$ for all $y \in \mathbb{R}^N$.

[4]As in all TU-games, we define $v^{SP}(\emptyset) = 0$.

Cost additivity. A rule for stochastic scheduling problems with delays ψ satisfies cost additivity if for all $SP = (N, \prec, X^0, x, C)$ and all $SP' = (N, \prec, X^0, x, C')$, then

$$\psi_i\left(SP + SP'\right) = \psi_i\left(SP\right) + \psi_i\left(SP'\right)$$

for all $i \in N$, where $SP + SP' = (N, \prec, X^0, x, C + C')$ and $(C + C')(y) = C(y) + C'(y)$ for all $y \in \mathbb{R}^N$.

Monotonicity A rule for stochastic scheduling problems with delays ψ satisfies monotonicity if for all $SP = (N, \prec, X^0, x, C)$ and all $SP' = (N, \prec, X^0, x', C)$ such that $x_i \leq x'_i$ and $x_j = x'_j$ for some $i \in N$ and for all $j \in N \setminus i$, then

$$\psi_i\left(SP\right) \leq \psi_i\left(SP'\right)$$

Equal responsability for two. A rule for stochastic scheduling problems with delays ψ satisfies equal responsability for two if for all $SP = (\{1, 2\}, \prec, X^0, x, C)$, then

$$\psi_i\left(SP\right) = E(C(x_i, X_j^0)) + \frac{1}{2}\left(C(x_i, x_j) - E(C(x_i, X_j^0)) - E(C(x_j, X_i^0))\right)$$

for all $i, j \in \{1, 2\}$ with $i \neq j$.

Scale Invariance. A rule for stochastic scheduling problems with delays ψ satisfies scale invariance if for all $SP = (N, \prec, X^0, x, C)$ and all $\lambda \in (0, \infty)^N$, we have

$$\psi\left(N, \prec, X^0, x, C\right) = \psi\left(N, \prec, \lambda X^0, \lambda x, C^\lambda\right)$$

where $C^\lambda : \mathbb{R}^N \to \mathbb{R}$ is given by $C^\lambda(\lambda y) = C(y)$ for all $y \in \mathbb{R}^N$, $\lambda X^0 = (\lambda_i X_i^0)_{i \in N}$ and $\lambda y = (\lambda_i y_i)_{i \in N}$.

Independence of Irrelevant Delays. A rule for stochastic scheduling problems with delays ψ satisfies independence of irrelevant delays if, for all $SP = (N, \prec, X^0, x, C)$ such that

$$E(C(x_{S \cup i}, X^0_{N \setminus (S \cup i)})) = E(C(x_S, X^0_{N \setminus S})$$

for $i \in N$ and for all $S \subseteq N \setminus i$, then $\psi_i\left(SP\right) = 0$.

Note that the "independence of irrelevant delays" above is a kind of null agent property, in the sense that an activity i that satisfies the condition of the property can be seen as a null agent and therefore, according to the property, should receive a zero allocation.

The next result states that the Shapley rule for stochastic scheduling problems with delays satisfies all the properties above. Its proof is very similar to that of Theorem 2 of Bergantiños et al. (2018) and, therefore, is omitted.[5]

Theorem 3.1. *The Shapley rule for stochastic scheduling problems with delays satisfies anonimity, cost additivity, monotonicity, equal responsability for two, scale invariance and independence of irrelevant delays.*

Next we focus on a different property of the Shapley value and how to adapt it to our context: the balancedness property.

A rule for stochastic scheduling problems with delays satisfies the balancedness property if it treats all pairs of activities in a balanced way, which more precisely means that for every pair of activities i and j, the effect of the elimination of i on the allocation to j (according to the rule) is equal to the effect of the elimination of j on the allocation to i. To write this property formally, consider a stochastic scheduling problem with delays $SP = (N, \prec, X^0, x, C) \in \mathcal{SP}^N$, with $|N| \geq 2$, and $i \in N$. Now, we define the resulting problem if activity i is eliminated $SP_{-i} \in \mathcal{SP}^{N \setminus i}$ by

$$SP_{-i} = (N \setminus i, \prec_{-i}, X^0_{-i}, x_{-i}, C_{-i})$$

where:

- $\prec_{-i}$ is the restriction of $\prec$ to $N \setminus i$,
- X^0_{-i} is the vector equal to X^0 after deleting its i-th component,
- x_{-i} is the vector equal to x after deleting its i-th component, and
- $C_{-i} : \mathbb{R}^{N \setminus i} \to \mathbb{R}$ is given by $C_{-i}(y) = E(C(y, X^0_i))$, for all $y \in \mathbb{R}^{N \setminus i}$.

We now formally write the balancedness property.

Balancedness. A rule for stochastic scheduling problems with delays ψ satisfies the balancedness property when

$$\psi_i(SP) - \psi_i(SP_{-j}) = \psi_j(SP) - \psi_j(SP_{-i})$$

for all $SP \in \mathcal{SP}^N$, all finite N, and all $i, j \in N$ with $i \neq j$.

The following theorem shows that the balancedness property characterises the Shapley rule.

Theorem 3.2. *The Shapley rule is the unique rule for stochastic scheduling problems with delays that satisfies the balancedness property.*

[5]The proof of Theorem 3.1 is available to readers upon request to the authors.

Proof. Let us first check that the Shapley rule satisfies the balancedness property. Take $SP = (N, \prec, X^0, x, C) \in \mathcal{SP}^N$ and $i, j \in N$ with $i \neq j$. Then,

$$SSh_i(SP) - SSh_i(SP_{-j}) = \Phi_i(v^{SP}) - \Phi_i(v^{SP_{-j}}), \tag{1}$$

$$SSh_j(SP) - SSh_j(SP_{-i}) = \Phi_j(v^{SP}) - \Phi_j(v^{SP_{-i}}). \tag{2}$$

Now, for every $k \in N$, $v^{SP_{-k}}$ is a TU-game with set of players $N \setminus k$. For every non-empty $S \subseteq N \setminus k$, [6]

$$\begin{aligned} v^{SP_{-k}}(S) &= E_{N\setminus(S\cup k)}(C_{-k}(x_S, X^0_{N\setminus(S\cup k)})) \\ &= E_{N\setminus(S\cup k)}(E_k(C(x_S, X^0_{N\setminus(S\cup k)}, X^0_k))) \end{aligned}$$

Now, the independence of the components of X^0 implies that

$$v^{SP_{-k}}(S) = E_{N\setminus S}(C(x_S, X^0_{N\setminus S})) = v^{SP}(S).$$

Note that for every $S \subseteq N \setminus k$, $v^{SP}(S) = v^{SP}_{-k}(S)$, where $v^{SP}_{-k} \in G^{N\setminus k}$ denotes the restriction of the TU-game $v^{SP} \in G^N$ to $N \setminus k$. Hence,

$$v^{SP_{-k}} = v^{SP}_{-k} \text{ for all } k \in N. \tag{3}$$

Considering (3) and that Myerson (1980) proved that the Shapley value of a TU-game satisfies a balancedness property, the equations in (1) and (2) are equal. This implies that the Shapley rule satisfies the balancedness property.

Suppose now that there exists another rule $R \neq SSh$ for stochastic scheduling problems with delays that satisfies the balancedness property. As $R \neq SSh$, there must exist $SP = (N, \prec, X^0, x, C) \in \mathcal{SP}$ with $R(SP) \neq SSh(SP)$. Assume that SP is minimal, in the sense that: (a) $|N| = 1$, or (b) $|N| \geq 2$ and $R(SP_{-i}) = SSh(SP_{-i})$ for every $i \in N$.[7] Note that $|N| \neq 1$ because otherwise, $R(SP) = C(x) = SSh(SP)$; hence, $|N| \geq 2$. Take $i, j \in N$ with $i \neq j$. As R and SSh satisfy the balancedness property, then

$$R_i(SP) - R_j(SP) = R_i(SP_{-j}) - R_j(SP_{-i}),$$

$$SSh_i(SP) - SSh_j(SP) = SSh_i(SP_{-j}) - SSh_j(SP_{-i}).$$

Now, considering the minimality of SP,

$$R_i(SP) - R_j(SP) = SSh_i(SP) - SSh_j(SP)$$

[6] To facilitate the reading of this proof, when dealing with the mathematical expectation of a random vector, we explicitly indicate the components of the vector to which the mathematical expectation refers.

[7] This assumption is without loss of generality because if $SP \in \mathcal{SP}^N$ is not minimal, we can eliminate one by one the elements of N until we have a minimal SP' with $R(SP') \neq SSh(SP')$.

or, equivalently, $R_i(SP) - SSh_i(SP) = A \in \mathbb{R}$, i.e. it does not depend on i. But then, $A = 0$ because $\sum_{j\in N} R_j(SP) = C(x) = \sum_{j\in N} SS_j(SP)$. This implies that $R(SP) = SSh(SP)$, and the proof is concluded. □

Next we illustrate the performance of the Shapley rule in two examples. Note first that the Shapley rule behaves in Example 2.1 as desired. For the stochastic scheduling problem with delays SP, we can easily check that:

- $v^{SP}(1) = E(C(7, X_2^0)) = 13/12$,
- $v^{SP}(2) = E(C(X_1^0, 7)) = 29/20$,
- $v^{SP}(N) = C(7, 7) = 1$,

and then, $SSh(SP) = (0.31666, 0.68333)$. Thus, activity 2 receives a larger part of the delay cost, as it should. Note that in this example, $SSh(SP)$ can be easily exactly calculated. In general, SSh cannot be exactly calculated, but can be estimated using simulation techniques. Consider now a new example that is slightly more complex.

Example 3.1. *Consider the stochastic scheduling problem with delays* $SP = (N, \prec, X^0, x, C)$ *given by:*

N	1	2	3	4	5
$\prec$	-	1	-	1,3	2
X^0	t(1,2,3)	t(1/2,1,3/2)	t(1/4,1/2,9/4)	t(3,4,5)	exp(1/2)
x	2.5	1.25	2	4.5	3

and, for every $y \in \mathbb{R}^N$,

$$C(y) = \begin{cases} 0 & \text{if } d(N, \prec, y) \leq 6.5, \\ d(N, \prec, y) - 6.5 & \text{otherwise}, \end{cases}$$

where $t(a, b, c)$ *denotes the triangular distribution with parameters a (minimum), b (mode), and c (maximum), and exp(α) denotes the exponential distribution with parameter α (i.e., with mean* $1/\alpha$*). As we remarked in Example 2.1, the table does not give the entire binary relation* $\prec$ *but only the immediate precedences. For instance, because 1 precedes 2, 2 precedes 5 and* $\prec$ *is transitive, then 1 must precede 5; however, the table only indicates that 2 precedes 5. The entire* $\prec$ *is easily obtained as the smallest transitive binary relation over N that contains the given elements of* $\prec$*. In this case, the table displays*

$$(1, 2), (1, 4), (3, 4), (2, 5)$$

and then

$$\prec = \{(1, 2), (1, 4), (1, 5), (3, 4), (2, 5)\}.$$

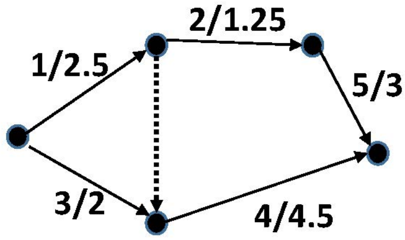

Figure 1: PERT graph of the project in Example 3.1

In some cases, it is more instructive to give the PERT graph representing the precedences instead of the precedences and $\prec$*. The PERT graph in this example is given in Figure 1, where, for each arc, we indicate the activity that it represents and the duration of this activity according to* x*; the dotted arc corresponds to a fictitious activity, that is needed to build a graph representing the precedences in this project. Fictitious activities always have zero duration. It is easy to check that* $d(N, \prec, x) = 7$ *(remember that the duration of a project is equal the duration of its longest path in the PERT graph), and then* $C(x) = 0.5$*. To allocate this cost amongst the activities in a fair way, note first that* $E(X^0) = (2, 1, 1, 4, 2)$*, and thus, all activities have a delay with respect to their expected durations. If we take a naive approach, i.e., if we allocate the delay cost by using the Shapley rule for* $\overline{SP} = (N, \prec, E(X^0), x, C)$*, we have*

$$Sh(\overline{SP}) = (0.27083, 0.02083, 0, 0.18750, 0.02083).$$

At first sight, this is a reasonable allocation of the delay cost. Activities 1 and 4 belong to the longest path in project $(N, \prec, x)$*, and thus, receive most of the delay cost. The cost allocated to activity 1 is greater than that allocated to activity 4 because activity 1 also belongs to a path with a duration greater than* 6.5 *(the path 1-2-5 has duration* 6.75*). Activity 3 only belongs to one path with duration* 6.5*, and produces no delay cost. Therefore, it pays* 0*. However, note that this allocation does not consider the probability distributions of the durations of the activities but only their averages. For instance, the duration of activity 5 follows an exponential distribution, the support for which is* $[0, \infty)$*. This means that its duration can be very long, and therefore, can produce a longer delay. However, its duration is not very long; so, in a sense, activity 5 contributes to a lack of delay in the project. This is captured by the Shapley rule for stochastic scheduling problems with delays. Using elementary simulation techniques,* v^{SP} *can be estimated in a good way and then* $SSh(SP)$ *can be calculated; the result is*

$$SSh(SP) = (0.28960, 0.09834, 0.07641, 0.20659, -0.17095).$$

It should be noted that this allocation differs from $Sh(\overline{SP})$ *primarily in that activity 5* *receives a kind of reward for not being too late, where this reward is paid by activities 1, 2, and 4, which last longer than expected and belong to paths whose durations entail a delay cost.*

We now use a small simulation experiment indicating that, on the average, when x is drawn from X^0, the cost allocation provided by SSh causes activity 5 to pay the largest part of the delay cost. We then realise that SSh tends to allocate the delay cost to activities 1, 4, and 5, but that it is very sensitive to the durations of the activities. We simulated 1,000 times the durations of the activities such that the 1,000 corresponding durations of the projects were greater than 6.5, *i.e. we simulated $(x^i)_{i\in\{1,\ldots,1000\}}$, each x^i_j being an observation of X^0_j, all drawn independently and in such a way that $C(x^i) > 0$. Thus, we obtained 1,000 stochastic scheduling problems with delays $SP^i = (N, \prec, X^0, x^i, C)$ as well as their 1,000 associated proposals of the Shapley rule $SSh(SP^i)$. We then calculated*

$$\sum_{i\in\{1,\ldots,1000\}} \frac{SSh(SP^i)}{1000} = (0.12857, 0.06844, 0.06686, 0.10757, 0.93790), \tag{4}$$

where the average observed cost was 1.30935*. Note that (4) showed that, in effect, when there are positive delay costs in an implementation of the stochastic project $SP = (N, \prec, X^0)$ the delay cost function being C, the cost allocation provided by SSh primarily burdens activity 5. This suggests that the vector of actual durations x that we handle in this example could be considered atypical because $SSh_5(SP) < 0$. Figure 2 confirms it. It displays the density estimations of the variables Z^1_i (solid line) and Z^2_i (dotted line), $i \in \{1, \ldots 5\}$, such that*

- *Z^1_i is the i-th component of $Sh((N, \prec, E(X^0), X, C))$, where X denotes the random variable corresponding to an observation of X^0; and*
- *Z^2_i is the i-th component of $SSh((N, \prec, X^0, X, C))$, where X denotes the random variable corresponding to an observation of X^0.*

Note that the scales of the five graphics in Figure 2 are different, which is a relevant feature to interpret them. It is not possible to adjust the scales while maintaining the informative graphics. It is interesting to note that the variables Z^1_i and Z^2_i are significantly different for each i, which strengthens the interest of the rule SSh. Finally, Table 1 displays the percentage of times that each of the activities (in columns) received non-negative or negative allocations (in rows) according to Sh and SSh. Again, activity 5 shows the largest discrepancies between the deterministic and the stochastic scenario, because if its planned duration equals its mean duration (as we assume occurs in the deterministic scenario) the marginal contribution of activity 5 to each possible coalition in the corresponding game cannot be negative, and thus $Sh_5((N, \prec, E(X^0), x, C)) \geq 0$ for all x; this does not happen in the stochastic scenario in which its duration is described by X^0_5 and not by $E(X^0_5)$.

Example 3.1 raises one controversial property of *Sh* and *SSh*: they can propose

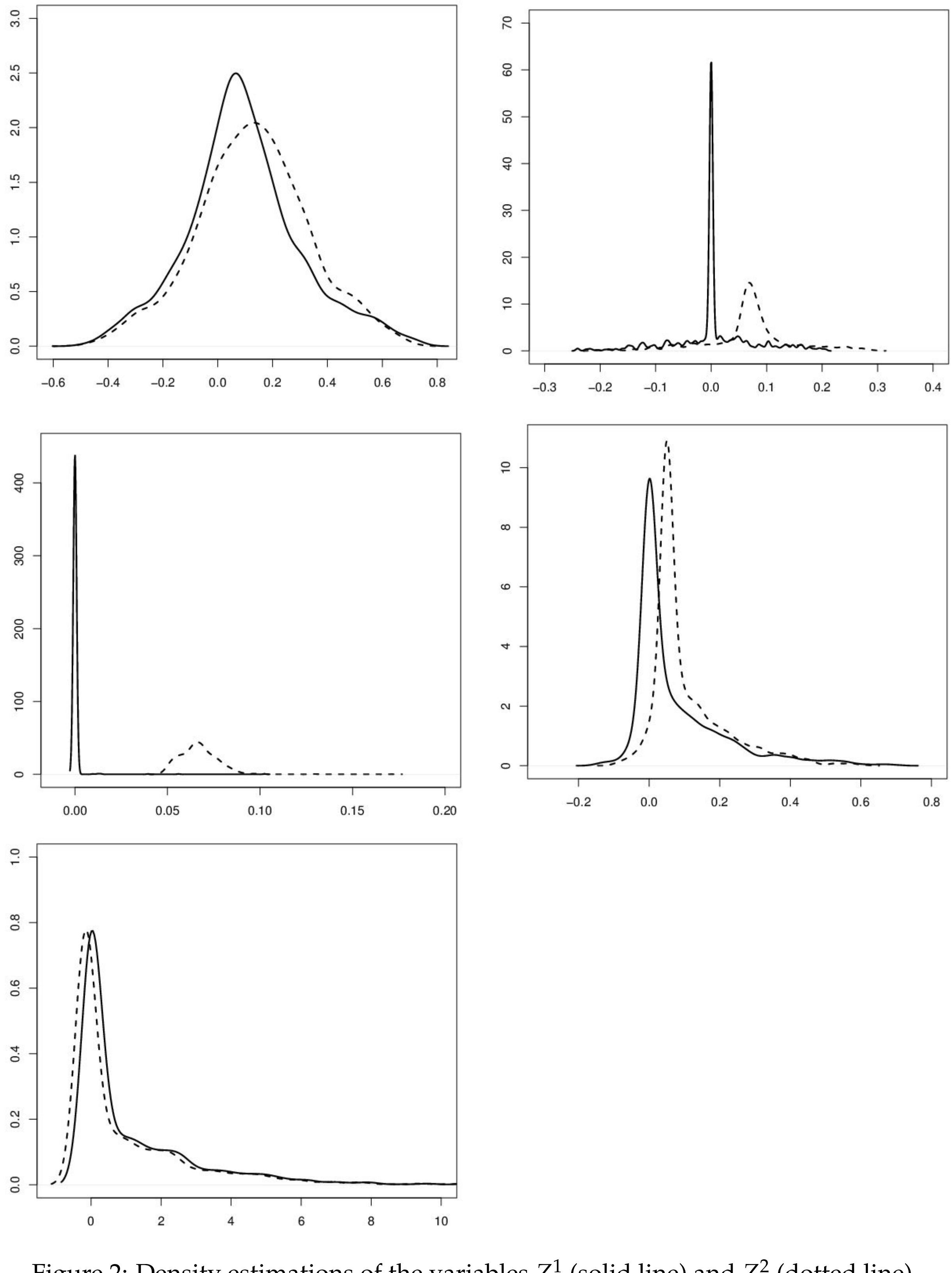

Figure 2: Density estimations of the variables Z_i^1 (solid line) and Z_i^2 (dotted line)

Sh	1	2	3	4	5
≥ 0	70.5	74.9	100	91.1	100
< 0	29.5	25.1	0.0	8.9	0

SSh	1	2	3	4	5
≥ 0	75.0	85.5	100	95.5	48.3
< 0	25	14.5	0.0	4.5	51.7

Table 1: Positive and negative payments for the Sh rule (left) and SSh rule (right)

negative allocations to some activities. This can be seen as a counter-intuitive feature, mainly because we are dealing with the issue of how to allocate delay costs when these occur. However, it must be borne in mind that in the context we are studying, even if we are only interested in "delay costs", there will inevitably be activities whose observed durations contribute positively to the occurrence of such costs and others whose observed durations contribute negatively and, then, it is not so counter-intuitive for a rule to propose negative allocations to some activities. In any case, it is clear that in some scenarios it will be inadmissible to allocate negative values to some activities even though their participation has contributed to reducing the final delay of the project and, therefore, its delay cost.

Let us see now what condition concerning non-negativity can we prove for Sh and how can it be extended to SSh.

Theorem 3.3. *(a) Let $P = (N, \prec, x^0, x, C)$ be a deterministic scheduling problem with delays. Then, for every $i \in N$,*

$$x_i \geq x_i^0 \;\Rightarrow Sh_i(N, \prec, x^0, x, C) \geq 0.$$

(b) Let $SP = (N, \prec, X^0, x, C)$ be a stochastic scheduling problem with delays. Then, for every $i \in N$,

$$C(x_i, y_{N\backslash i}) \geq E(C(X_i^0, y_{N\backslash i}))\; \forall\, y_{N\backslash i} \in \mathbb{R}^{N\backslash i} \Rightarrow SSh_i(N, \prec, x^0, x, C) \geq 0.$$

Proof. (a) Take $i \in N$ such that $x_i \geq x_i^0$. Since C is non-decreasing then, for all $S \subseteq N\backslash i$,

$$v^P(S \cup i) = C(x_i, x_S, x^0_{N\backslash(S\cup i)}) \geq C(x_i^0, x_S, x^0_{N\backslash(S\cup i)}) = v^P(S).$$

Hence $Sh_i(P) = \Phi_i(v^P) \geq 0$.

(b) Take $i \in N$ such that $C(x_i, y_{N\backslash i}) \geq E(C(X_i^0, y_{N\backslash i}))$ for all $y_{N\backslash i} \in \mathbb{R}^{N\backslash i}$. The independence of the components of X^0 implies that, for all $S \subseteq N\backslash i$,

$$\begin{aligned} v^{SP}(S \cup i) &= E(C(x_i, x_S, X^0_{N\backslash(S\cup i)})) \geq E_{N\backslash(S\cup i)}(E_i(C(X_i^0, x_S, X^0_{N\backslash(S\cup i)}))) \\ &= E(C(X_i^0, x_S, X^0_{N\backslash(S\cup i)})) = v^{SP}(S). \end{aligned}$$

Hence $SSh_i(SP) = \Phi_i(v^{SP}) \geq 0$. □

It may be worth noting that the sufficient condition in subparagraph (b) extends, in some way, the condition in subparagraph (a). We cannot transfer directly to the stochastic case condition $x_i \geq x_i^0$ because it is not clear how to compare a real number (x_i) with a random variable (X_i^0); on the other hand, $x_i \geq E(X_i^0)$ is not a sufficient condition for the non-negativity of SSh. However, since C is non-decreasing, $x_i \geq x_i^0$ implies that $C(x_i, y_{N\setminus i}) \geq C(x_i^0, y_{N\setminus i})$. Now, if we replace x_i^0 with X_i^0 and we take the mathematical expectation, we do obtain a sufficient condition (in view of Theorem 3.3). It is not difficult to check that an alternative sufficient condition is that $x_i \geq z$ for all real number z in the support of the random variable X_i^0. In words, every condition of the type "x_i is sufficiently large in view of the distribution of X_i^0" guarantees the non-negativity of SSh.

4 Computational Analysis

The calculation of the Shapley value has, in general, an exponential complexity. Although equivalent expressions with polynomial complexity can be used in some game classes, this is not the case for the class of games with which we are dealing. Calculating the Shapley value in our context is impossible in practise, even for a moderate number of activities. For example, if the number of activities is 100, there are 2^{100} coalitions in which the characteristic function must be evaluated. However, in spite of these difficulties, we are still strongly interested in the Shapley value for its good properties in our particular context as it was discussed in the previous sections. As an alternative to exact calculation, Castro et al. (2009) proposed an estimate of the Shapley value in polynomial time using a sampling process.

In addition, to estimate the Shapley rule for stochastic scheduling problems with delays we also need to calculate $v^{SP}(S) = E(C(x_S, X_{N\setminus S}^0))$, with $S \subseteq N$. In some simple cases, these mathematical expectations can be calculated in a simple way using the properties of order statistics; but in general, we need to use simulations to approximate v^{SP}.

The aims of this section are twofold: First, to illustrate the implementation of the estimation of the Shapley rule for stochastic scheduling problems with delays through its pseudocode, from which it is easy to check that the computational complexity of our rule is $O(n^4)$; and second, to show by examples that it is possible to estimate the Shapley rule for stochastic scheduling problems with delays in an acceptable time, even if there are hundreds of activities, by using a desktop computer and free software. The error in the two phases of estimation is tracked a posteriori through the estimation of variance and central limit theorem.

Next we are going to show the pseudocode of our implementation; we do it routine to routine. The first routine aims to reorder the precedence matrix. If there are n activities $1, 2, \ldots, n$, the binary relation $\prec$ can be written as an $n \times n$ matrix named *precedence* in which $precedence_{ij} = 1$ means that i precedes j. We want to permute the set of activities in order that if $i \prec j$ then $i < j$. Note that this task can always be carried out and allows for faster calculation. Given a matrix P, we denote its i-th row by $P_{i,\cdot}$ and its i-th column by $P_{\cdot,i}$.

Organise precedence matrix ———————————————

- **Begin**

  $P = precedence, index = NULL$

  While number of P's columns > 0

  Take all $i \in n$ such that $\sum_{j=1}^{n} P_{ij} = 0$

  $index = (index, i)$

  $P = P \backslash P_{i,\cdot}$ and $P = P \backslash P_{\cdot,i}$

  end

  $precedence = precedence_{index,index}$

- **end**

The next routine computes the early times for a deterministic scheduling problem when the duration of the activities is given by x^0. The early time of an activity is the earliest that this activity can begin.

Early times ———————————————

- **Begin**

  $early.times_i = 0 \ \forall i \in N$

  Organise precedence matrix

  $I = \{i \in n$, such that $\sum_{j=1}^{n} precedence_{ji} \neq 0\}$

  For $i \in I$

  $prec = \{j \in n / precedence_{ji} = 1\}$

  $early.times_i = \max\{x^0_{prec} + early.times_{prec}\}$

  end

- **end**

Let us consider a deterministic scheduling problem with delays with delay cost function, for every $y \in \mathbb{R}^N$, given by:

$$C(y) = \begin{cases} 0 & \text{if } d(N, \prec, y) \leq \delta, \\ d(N, \prec, y) - \delta & \text{otherwise.} \end{cases}$$

We obtain an estimation of the Shapley rule in polynomial time. The algorithm consists of taking $m \in \mathbb{N}$ permutations of the set of players N with equal probability (Castro et al., 2009). We denote by Π_N the set of permutations of N. We then calculate $|N|$ real numbers as follows:

$$\pi^j \in \Pi_N \text{ where } \pi^j = (\pi^j_1, ..., \pi^j_{|N|}) \text{ and } j \in \{1, ..., m\}$$

$$x(\pi^j)_i = v(Pre^i(\pi^j) \cup i) - v(Pre^i(\pi^j))$$

where $Pre^i(\pi^j)$ denotes the set of activities that precede activity i in the permutation π^j, i.e., $Pre^i(\pi^j) = \{\pi^j_1, \ldots, \pi^j_{k-1} \mid i = \pi^j_k\}$. So, $x(\pi^j) \in \mathbb{R}^{|N|}$ is the corresponding marginal contributions vector. Finally, the estimated value of the Shapley value is:

$$\hat{Sh}_i = \frac{1}{m} \sum_{j \in 1}^{m} x(\pi^j)_i$$

for all $i \in N$.

When we address the stochastic version of the problem, we can use nearly the identical procedure to that in the deterministic case; but in this new situation, we also need to estimate the TU-game. For this, we simulate the TU-game $m_1 \in \mathbb{N}$ times and take the average of these values.

The key part of the next routine is the computation of $v(S \cup i)$, where $S = Pre^i(\pi)$, given by

$$E(C(x_{Pre^i(\pi) \cup i}, X^0_{N \setminus (Pre^i(\pi) \cup i)})).$$

We compute $d(N, \prec, y)$ as the maximum of the sums of the early times of the activities and their durations.

Estimation of Shapley rule in the stochastic case ———

- **Begin**

  Determine m and m_1

  $Cont = 0$, $\hat{Sh}_i = 0$, $time_i = 0\ \forall i \in N$ and $v_j = 0\ \forall j \in m_1$

  For $j \in m_1$

  $\hat{X}^0_{j,\cdot} = sample(X^0)$

  end

  Organise precedence matrix

  While $cont < m$

  Take $\pi \in \Pi_N$ with probability $\frac{1}{n!}$

  For $i \in n$

  For $j \in m_1$

  Early times of $(x_{Pre^i(\pi) \cup i}, \hat{X}^0_{j, N \setminus (Pre^i(\pi) \cup i)})$

$v_j = \max\{\max\{early.times + (x_{Pre^i(\pi)\cup i}, \hat{X}^0_{j,N\setminus(Pre^i(\pi)\cup i)})\} - \delta, 0\}$

end

$time_i = mean(v)$

end

$\hat{Sh}_{\pi_1} = \hat{Sh}_{\pi_1} + time_1$

$\hat{Sh}_{\pi_i} = \hat{Sh}_{\pi_i} + time_i - time_{i-1}\ \forall i \in N\setminus 1$

$cont = cont + 1$

end

$\hat{Sh} = \frac{\hat{Sh}}{m}$

- **end**

To gain insight into the computation time needed to obtain a solution, we selected five problems[8] with a number of activities ranging from 10 to 1,000. We ran the problems on a PC with a 3.70 GHz Core i7-8700K, and 64 GB of RAM on an Ubuntu 64-bits. The programming language used was **R** x64 3.4.4. It is freely available under the GNU General Public License. To improve performance in terms of time, we used the packages **Rcpp** and **parallel**. The package Rcpp was used to write in C the function *early times* and parallel was used to parallelise the estimation of the Shapley value by using six cores of our computer.

Table 2 shows the computation times, in seconds, of the five problems, with 10, 30, 100, 300, and 1,000 activities, respectively (in columns). The TU-game was approximated using $m_1 = 1000$ simulations, while $m = 1000$ and 10000 estimates (in rows) were used for the Shapley rule.

	10	30	100	300	1000
1000	18	120	1033	7801	118770
10000	211	1329	11941	80521	1277377

Table 2: Computation times in seconds

Table 2 illustrates that it is possible to estimate the Shapley rule for stochastic scheduling problems with delays in an acceptable time, even if there are hundreds of activities, by using a desktop computer and free software.

Table 3 shows an estimation of errors, both in the approximation of the characteristic function and Shapley rule by using $m = 1000$ and 10000 (in rows). As

[8]These problems were too large to be included in this paper. They can be downloaded from http://dm.udc.es/profesores/ignacio/stochasticprojects

above, the columns display the number of activities of the corresponding problems. All errors are relative and in percent.[9] A significance level of $\alpha = 0.05$ was used in these estimates. The error in $v^{SP}(S)$ is different for every $S \subseteq N$, and therefore, we display the average of 1,000 coalitions chosen in a random way. In the Shapley rule, each activity has an error, and the table shows the average of all activities.

	10	30	100	300	1000
v^{SP}	2.18	2.96	4.64	2.28	0.83
1000	12.92	13.49	19.37	27.88	12.92
10000	4.17	4.27	6.13	8.82	4.09

Table 3: Errors for v^{SP} and the Shapley rule

Table 3 illustrates that the estimations of the games v^{SP} are rather good, whereas the estimations of the Shapley rule are acceptable and improve significantly with the size of m. In conclusion, we can affirm that a real problem of great dimension can be solved in a satisfactory way in a reasonable time using our methodology.

Acknowledgements

This work has been supported by the MINECO grants MTM2014-53395-C3-1-P and MTM2017-87197-C3-1-P, and by the Xunta de Galicia through the ERDF (Grupos de Referencia Competitiva ED431C-2016-015 and Centro Singular de Investigación de Galicia ED431G/01). The authors also acknowledge the useful comments of two anonymous referees.

Compliance with ethical standards

Conflict of interest There is no potential conflicts of interest.
Ethical standard Research do not have human participants and/or animals.

References

Bergantiños G, Lorenzo L (2019). How to apply penalties to avoid delays in projects. European Journal of Operational Research 275, 608-620.

[9] As is common in statistical methodology, the relative error in percent of the estimation of a parameter θ is given by $z_{\alpha/2}\frac{s}{\sqrt{n}}\frac{100}{\theta}$, where s is the square root of the sample variance.

Bergantiños G, Sánchez E (2002). How to distribute costs associated with a delayed project. Annals of Operations Research 109, 159-174.
Bergantiños G, Valencia-Toledo A, Vidal-Puga J (2018). Hart and Mas-Colell consistency in PERT problems. Discrete Applied Mathematics 243, 11-20.
Brânzei R, Ferrari G, Fragnelli V, Tijs S (2002). Two approaches to the problem of sharing delay costs in joint projects. Annals of Operations Research 109, 359-374.
Briand C, Billaut JC (2011). Cooperative project scheduling with controllable processing times: a game theory framework. Proceedings of the ETFA2011 IEEE Conference, pp. 1-7.
Briand C, Ngueveu SU, Sucha P (2017). Finding an optimal Nash equilibrium to the multi-agent project scheduling problem. Journal of Scheduling 20, 475-491.
Castro J, Gómez D, Tejada J (2007). A project game for PERT networks. Operations Research Letters 35, 791-798.
Castro J, Gómez D, Tejada J (2009). Polynomial calculation of the Shapley value based on sampling. Computers & Operations Research 36, 1726-1730.
Castro J, Gómez D, Tejada J (2014). Allocating slacks in stochastic PERT networks. CEJOR 22, 37-52.
Herroelen W, Leus R (2005). Project scheduling under uncertainty: survey and research potentials. European Journal of Operational Research 165, 289-306.
Curiel I (2011). Project management games. International Game Theory Review 13, 281-300.
Hillier FS, Liberman GJ (2001). Introduction to Operations Research. McGraw-Hill.
Estévez-Fernández A (2012). A game theoretical approach to sharing penalties and rewards in projects. European Journal of Operational Research 216, 647-657.
Estévez-Fernández A, Borm P, Hamers H (2007). Project games. International Journal of Game Theory 36, 149-176.
Flores R, Molina E, Tejada J (2007). Evaluating groups with the generalized Shapley value. 4OR, 1-32.
Myerson RB (1980). Conference structures and fair allocation rules. International Journal of Game Theory 9, 169-182.
San Cristóba JR (2014). Cost allocation between activities that have caused delays in a project using game theory. Procedia Technology 16, 1017-1026.
Shapley LS (1953) A value for n-person games. In: Kuhn HW, Tucker AW (eds) Contributions to the Theory of Games II. Princeton University Press, Princeton.
Suijs J, Borm P, de Waegenaere A, Tijs S (1999). Cooperative games with stochastic payoffs. European Journal of Operational Research 113, 193-205.
Tanimoto K, Okada N, Tatano H (2000). Project risk allocation through contingent cost allocation. Proceedings of the 2000 IEEE International Conference on Systems,

Man and Cybernetics, pp. 512-517.