# Optimization and allocation in some decision problems with several agents or with stochastic elements

Author: Juan Carlos Gonçalves Dosantos

Doctoral Thesis / 2021

Supervisors:

Julián Costa Bouzas
Ignacio García Jurado

Programa de doctorado en estadística e investigación operativa

UNIVERSIDADE DA CORUÑA

# OPTIMIZATION AND ALLOCATION IN SOME DECISION PROBLEMS WITH SEVERAL AGENTS OR WITH STOCHASTIC ELEMENTS

# OPTIMIZATION AND ALLOCATION IN SOME DECISION PROBLEMS WITH SEVERAL AGENTS OR WITH STOCHASTIC ELEMENTS

JUAN CARLOS GONÇALVES DOSANTOS

This dissertation is the doctoral thesis of Juan Carlos Gonçalves Dosantos under the supervision of Julián Costa Bouzas and Ignacio García Jurado, from the University of A Coruña.

Departamento de Matemáticas
Universidade da Coruña
December, 2020

El acto público de defensa y mantenimiento de esta tesis doctoral se realizó el día 19 de febrero de 2021, en la Facultad de Informática de la *Universidade da Coruña*, ante el siguiente tribunal.

| | |
|---|---|
| Presidente: | Andrés Jiménez Losada |
| Vocal: | Arantza Estévez Fernández |
| Secretaria: | María Luisa Carpente Rodríguez |

La tesis obtuvo la máxima calificación de sobresaliente *cum laude* y la mención internacional; fue realizada en la modalidad de compilación de artículos y defendida por videoconferencia.

# Agradecimientos

Quiero expresar mi más sentido agradecimiento a mis directores de tesis, Julián Costa Bouzas e Ignacio García Jurado, por la constante ayuda, apoyo y tiempo que me han dedicado a lo largo de esta etapa predoctoral. Esto no hubiese sido posible sin vosotros. A José María Alonso Meijide y Balbina Casas Méndez por su ayuda y contribución con magnificas aportaciones. Y a Fernando Bernstein por hacer que mi estancia en Estados Unidos fuese lo más agradable y cómoda posible.

Además agradezco la colaboración en distintas etapas de la tesis de María Luisa Carpente Rodríguez, Greg DeCroix, Aratza Estévez Fernández, Herbert Hamers, Andrés Jiménez Losada, Estela Sánchez Rodríguez y Joaquín Sánchez Soriano.

A mis padres, ya que el llegar hasta aquí ha sido posible gracias a su esfuerzo y dedicación. A mi familia, amigos y compañeros de trabajo. Y a ti, por acompañarme, Carmen.

Este trabajo ha sido financiado por el FEDER, la Agencia Española de Investigación del Gobierno de España, la Xunta de Galicia, el CITIC y el grupo MODES de la Universidade da Coruña a través de los proyectos MTM2014-53395-C3-1-P, MTM2017-87197-C3-1-P, ED431C-2016-015, ED431C-2017/38 y ED431G 2019/01.

# Contents

# Resumen

En esta memoria se abordan diversos problemas de decisión que surgen en la gestión de proyectos, en la teoría de juegos cooperativos y en la optimización de rutas de vehículos.

Empezamos estudiando el problema del reparto de los costes de demora en un proyecto. En un contexto estocástico en el que suponemos que las duraciones de las actividades son variables aleatorias, proponemos y estudiamos una regla de reparto basada en el valor de Shapley. Además, presentamos un paquete de R que permite un control integral del proyecto, incluyendo la nueva regla de reparto.

A continuación, proponemos y caracterizamos axiomáticamente nuevas soluciones igualitarias en el contexto de los juegos cooperativos con una estructura coalicional. E introducimos un nuevo valor, utilizando una propiedad de jugadores necesarios, para juegos cooperativos, que posteriormente extendemos y caracterizamos dentro del marco de los juegos cooperativos con una estructura coalicional.

Por último, presentamos un algoritmo en dos pasos para resolver problemas de rutas de vehículos con multi-compartimentos y demandas estocásticas. Este algoritmo obtiene una solución inicial mediante una heurística constructiva y, a continuación, utiliza una búsqueda tabú para mejorar la solución. Utilizando datos reales, llevamos a cabo un análisis del comportamiento del algoritmo.

# Resumo

Nesta memoria abórdanse diversos problemas de decisión que xorden na xestión de proxectos, na teoría de xogos cooperativos e na optimización de rutas de vehículos.

Empezamos estudando o problema da repartición dos custos de demora nun proxecto. Nun contexto estocástico no que supoñemos que as duracións das actividades son variables aleatorias, propoñemos e estudamos unha regra de repartición baseada no valor de Shapley. Ademais, presentamos un paquete de R que permite un control integral do proxecto, incluíndo a nova regra de repartición.

A continuación, propoñemos e caracterizamos axiomaticamente novas solucións igualitarias no contexto dos xogos cooperativos cunha estrutura coalicional. E introducimos un novo valor, utilizando unha propiedade de xogadores necesarios, para xogos cooperativos, que posteriormente estendemos e caracterizamos dentro do marco dos xogos cooperativos cunha estrutura coalicional.

Por último, presentamos un algoritmo en dous pasos para resolver problemas de rutas de vehículos con multi-compartimentos e demandas estocásticas. Este algoritmo obtén unha solución inicial mediante unha heurística construtiva e, a continuación, utiliza unha búsqueda tabú para mellorar a solución. Utilizando datos reais, levamos a cabo unha análise do comportamento do algoritmo.

# Abstract

This dissertation addresses some decision problems that arise in project management, cooperative game theory and vehicle route optimization.

We start with the problem of allocating the delay costs of a project. In a stochastic context in which we assume that activity durations are random variables, we propose and study an allocation rule based on the Shapley value. In addition, we present an R package that allows a comprehensive control of the project, including the new rule.

We propose and characterize new egalitarian solutions in the context of cooperative games with a coalitional structure. Also, using a necessary player property we introduce a new value for cooperative games, which we later extend and characterize within the framework of cooperative games with a coalitional structure.

Finally, we present a two-step algorithm for solving multi-compartment vehicle route problems with stochastic demands. This algorithm obtains an initial solution through a constructive heuristic and then uses a tabu search to improve the solution. Using real data, we evaluate the performance of the algorithm.

# Preface

## Precedents

When a problem is detected, decision making is essential in order to deal with it correctly. Here is where Operations research (OR) appears, which allows to maximise the effectiveness of the decision making. Operations research is as old as humanity, but it is not until the Second World War when it acquires scientific autonomy. Given the scarcity of resources, it became necessary to allocate them, as best as possible, to the various military activities that made up each operation. Therefore, within the allied armies, it was decided to bring together groups of scientists to advise on the optimization of military operations; in this process, operations research was born. Given the great success of OR in the different operations, after the war, it is introduced in industrial problems of business or govern as a way to obtain a better relation cost/profit.

Investigate about operations, as its name suggest, deals with a large and diverse combination of topics. However, we can stand out a series of classical problems and models all of them of special relevance and, at the same time, general. Among them, the linear programming (see, for instance Hillier and Lieberman, 2002) is, maybe, the most used tool in OR given its great versatility. This tool is used in problems that seek to assign, in the most efficient way, a group of limited resources to different tasks that compete between them. The assignment is done by taking into account a combination of restrictions and to maximizing the profit. Associated to the linear programming the simplex method proposed by Dantzig in 1947[1] is an algorithm to solve this kind of problems, effective and fast, even to big size problems. Two particular problems of linear programming are the problems of transportation and assignment. The transportation problem, first formalized in Monge (1781) and being solved mathematically in Tolstoi (1930), consists of optimizing the way

[1]Dantzig (1963) is the first book written by George Dantzig where this methodology appears.

of transporting the goods from the origins to the different destinations at the minimum cost. On the other hand, the assignment problem (Kuhn, 1955), consists of assigning different tasks to people (it is important to stand out that the assignment problem is a special case of the transportation problem). Although both of them are modulable as linear programming problems, the elevated number of restrictions and variables makes that the resolution with the simplex method requires a high computational effort; therefore, some specific algorithms to these kind of problems were arise. These two particular problems enter in the subcategory of OR called network optimization models, due to their connections with graphs and networks. Other network optimization problems are the shortest path (Dijkstra, 1959), the minimum spanning tree (Kruskal, 1956) or the maximum flow problem (Ford y Fulkerson, 1956). We also emphasize a problem of network modelling of great relevance throughout this work, the PERT (Program Evaluation and Review Techniques) problem, or also known as CPM (Critical Path Method); see Malcolm et al. (1959) and, for a more complete revision, Punmia and Khandelwal (2002). In this type of problems it is formulated and studied a project as a directed network, with only one initial and final nodes, formed by a group of activities with relationships of precedence among them.

All the problems introduced until now are deterministic, but OR also deals with stochastic problems. For instance, in the problems of queueing theory presented by Erlang (1909), (see a more complete revision in Gross, 2008), queuing lines are studied to analyze characteristics like the time that each client spends at the queue, the time that each server is occupied, the length of the queue, etc. It is also analyzed how a system has to be designed in order to reduce the time the users spend in it, without increasing the cost of maintenance of the system.

Previously we have mentioned linear programming models. These type of models are characterized by linear functions. This restricts the range of problems that we can analyze. For this reason, the nonlinear mathematical programming (for a complete reference see Bazaraa et al., 2013) considers more general models. However, even though some of those models can be resolved in an efficient way, as the convex models, for the most part of the cases it is not possible to solve them in an exact way. This is why the approximation algorithms and the heuristic algorithms are used to obtain good solutions.

OR does not just contemplate problems in which one agent make decisions about a problem to optimize the corresponding benefits. There are situations in which we find a conflict between various agents, or players, and for each decision vector, a different scenery and result is obtained. Game theory (Von Neumann and Morgenstern, 1947) studies this kind of situations from a mathematical point of view and contains two sub-theories: the cooperative and the non-cooperative. In the non-cooperative game theory the players make decisions based on their sets of available strategies and, given the natural competition between the players, the final result of each one depends on the decision made by all of them. In a different way, in the cooperative game theory the players look for setting up coalitions in order to maximize the earnings obtained, and a “fair” distribution among the different players is one of the main objectives of the problem; the Shapley value (Shapley 1953) is one of the best known and used allocation rules in cooperative game theory.

In addition, it has to be highlighted that a big part of the advance that operations research experimented throughout its (short) history has been thanks to the computer programming. The advance of computers has been essential to solve large problems. In addition, the creation of specific free software libraries allows all operations researchers to program the resolution of the problems they face in a relatively simple way. This causes an advance even faster in the development of OR. Because of its great benefits for research and its importance in this thesis, we must particular mention the free software R (R Core Team, 2020). R is a statistical software formed by a set of very flexible tools that allow the users adequate them in every moment for its personal use. Additionally, it is formed by packages, or libraries: a group of functions designed by the users and accessible for the whole community by a simple download. This causes a daily increase in its functionality.

The general objectives of this thesis are to develop diverse tools to tackle new variants of some operations research problems, both deterministic and stochastic. In addition, it also aims to study the good behaviour of the tools already mentioned, in a practical and theoretical way, and to create libraries in the statistic software R with these tools and other already existing in the literature in order to make them available to the scientist community.

## General summary of the thesis

The present thesis is developed in six chapters. The first one focuses on the distribution of delay costs in stochastic projects already finished (Gonçalves-Dosantos et al., 2020c). Project management is a field which develops techniques in order to select, plan, execute and supervise projects. One of the first topics of interest regarding the planning of projects is time management, with the main objective of completing the project and the various activities that are part of it on the delivery date. However, sometimes the projects can be delayed so the cost associated to them has to be defrayed among the different agents responsible for the project execution. In the literature there are diverse ways of addressing the problem: from proportional rules based on bankruptcy problems (Brânzei et al., 2002) to rules based on game theory (for instance, Bergantiños et al., 2018, and Castro et al., 2007). All of them suppose that the duration of each activity is deterministic. Nevertheless, it is natural to consider this duration as a stochastic variable, following some distribution model which enriches it and approaches it more to reality. Additionally, considering the advantages of using rules based on cooperative game theory instead of rules based on bankruptcy problems, we have chosen to extend the game suggested by Bergantiños et al. (2018) to the stochastic context and use the Shapley value as an allocation rule. Finally, we characterize the rule introduced in the context of these project games and we propose an approximation algorithm based on Castro et al. (2009) in order to get it by simulation in projects with a high number of activities.

Chapter 2 develops a package of project management for R: ProjectManagement (Gonçalves-Dosantos et al., 2020a). It is a tool that enables free and computationally efficient management of projects. Given the scarcity of software relative to project management and the high cost of licenses for most existing software, we believe that an R package will be a great help to any user with special needs in project management. ProjectManagement allows the user to handle a project in order to get a calendar of it; this means: the expiration date of the project, the starting and ending times of each activity, as well as the slack that each activity has without delaying the project. As for the cost of delayed projects, the package has a catalog of possible probability distributions for activity durations and of allocation rules for cost sharing, including both proportional rules and the Shapley value. All of this is also analyzed by a stochastic point of view, using the suggested methodology from Chapter 1. Finally, ProjectManagement includes the resources management which permits the reduction of the project's expiration time by increasing the resources and, therefore, the cost. Additionally, it also contains a redistribution of the timetable to a more uniform level of resources consumption, and a new management of the project, taking into account a maximum limit of resources per time unit.

We have seen how game theory and the Shapley value can be used to solve problems that appear in project management when the delay costs are allocated. Now, from a more theoretical point of view, in Chapter 3 we study and characterize some new values for cooperative games. We extend the egalitarian values to the context of cooperative games with a priori unions (Alonso-Meijide et al., 2020). The egalitarian values are based on an equal distribution of the benefits obtained among the cooperating players. A large amount of theoretical literature proposes different variants of egalitarian solutions. For instance van den Brink (2007) or Casajus and Hüttner (2014) compare some egalitarian values with the Shapley value. Moreover, in van den Brink and Funaki (2009) and in van den Brink et al. (2016) we can see a few characterizations of the equal division value and also of the equal surplus division value. However, these values have never been introduced and studied in cooperative games with a priori unions. A game with a priori unions (Owen, 1977) is a cooperative game to which a partition of the group of players is added, i.e. an a priori coalition structure that conditions the negotiation between them and consequently produces a change in the outcome of the negotiation. It is in Owen (1977) where the Shapley value is extended to these type of games giving rise to the Owen value. Once it is extended the equal division value and three possible variants of the equal surplus division value for cooperative games with a priori unions, we characterize these new values with similar properties as the ones used on the original values. Eventually, in order to check the good behaviour of these values, these are applied to an example that arises in the allocation of costs corresponding to the installation of an elevator in an apartment building.

In Chapter 4 we study new characterizations for the values proposed in the previous chapter (Gonçalves-Dosantos and Alonso-Meijide, 2020), in a similar way to how the Owen value is characterized on Vázquez-Brage et al. (1997). In addition, two new values and its respective characterizations are introduced in order to extend the equal surplus division value. One of them is the value obtained by applying the procedure proposed by Owen (1977) to obtain the Owen value from the Shapley value but, in this new situation, starting from the equal surplus division value. The second extension arises looking for a coalitional value for the equal surplus division that satisfies the property of balanced contributions. In order to compare these new values, between them and with those already introduced on Chapter 3, we propose a similar example to the one used in the former chapter.

In previous chapters, we have obtained extensions of different values already existing in a specific framework of cooperative games. In Chapter 5, considering a special kind of players, the necessary player, and their corresponding allocations we propose a new value for cooperative games (Gonçalves-Dosantos et al., 2020d). A necessary player is a player without whom the worth for any coalition is zero. In Alonso-Meijide et al. (2019a) and Béal and Navarro (2020) these players are used to characterize the Shapley value, the Banzhaf value (Banzhaf III, 1964) and the equal surplus division value. In order to be able to theoretically compare this new value with others already existing in the literature a characterization of it is provided. In addition, its behaviour is compared with that of other values in a practical example. Finally, this value is extended to the context of cooperative games with a priori unions and characterized using properties of necessary players. We end with two characterizations of the Owen and Banzhaf-Owen values (Owen, 1982) using, again, the necessary players in a similar way as Alonso-Mejide et al. (2019a) characterize the Shapley and Banzhaf values.

In Chapter 6 we solve a real problem of networks with stochastic elements from a more practical point of view than on chapter 1 (Gonçalves-Dosantos and Casas-Mendez, 2020). The vehicle routing problems, introduced in Dantzig and Ramser (1959), study the design of a group of routes of minimum cost for a fleet of vehicles that should attend the demand of a group of clients scattered in different locations. Many other models arise from this basic model when we consider different restrictions such as the vehicles fleet with diverse capacities, time windows in which the clients must be attended, various collection and delivery points, etc. In this chapter we analyze the multi-compartment vehicle routing problems. These problems are characterized by the existence of various incompatible products to be delivered so the vehicle fleet has various independent multi-compartments to avoid mixing between them. In the existing literature for these models it is not taken into account important aspects of real life that are often random. In this sense, we are going to consider that the demands of the clients are stochastic variables. The motivation of these models comes from a real problem that appears in an agricultural cooperative in Galicia. In this cooperative four different types of feed are produced for farm animals. It has a fleet of vehicles of various compartments with different capacities and in each compartment only one type of feed can be transported. Given the high complexity of this type of models an algorithm in two steps is suggested. First, it is used a constructive heuristic based on the algorithm of Clarke and Wright (1964), and then we improve this initial solution by a tabu search (Glover, 1989 and Glover, 1990). Some results are shown through real data, to finally carry out a study of simulation to test the behaviour of this algorithm.

## Conclusions

The first chapter looks at the problem of sharing the costs of delays in projects when activity durations are stochastic. A cooperative game has been devised to determine the influence of activities on project delay. On the basis of this game, a rule of fair allocation of delay costs using the Shapley value is proposed. The comparison with the deterministic approach is satisfactory, obtaining different costs for activities with equal average durations but different distributions. An estimation algorithm of the allocation rule has been proposed, obtaining good results in admissible times and with acceptable errors. The second chapter proposes a project management software of free use, with an online manual to support it.

In Chapters 3 and 4 the egalitarian values are extended to the context of cooperative games with a priori unions. With an acceptable number of coherent properties, the different values have been characterized and the independence of the properties that characterize each value has been proved.

Chapter 5 introduces a new value for cooperative games using a property for necessary players that corrects the properties for necessary players satisfied by the Shapley and Banzhaf values. Additionally, the chapter provides new characterizations for the Owen and Banzhaf-Owen values using only three properties.

Finally, Chapter 6 solves a real problem of vehicle routing with multi-compartment and stochastic demands. Given the high computational complexity of these problems, a two-step algorithm has been proposed. The constructive algorithm takes into account the fact that demands are stochastic variables when selecting customers and, if necessary, the return to the depot to reload necessary goods. In the second step a tabu search improves the initial solution, selecting routes at random and exchanging one or two customers between the routes. In an example with real data, the solutions obtained have been compared with those of the deterministic case. The distance made by the vehicles coincides with the distance in the deterministic case when all the demands (in our case they are stochastic) can be satisfied without returning to the deposit. In any case, the distance traveled in the stochastic scenario increases, on average, approximately a 6% because of the randomness. Finally, in the simulation carried out, the solutions obtained in the first and second steps of the algorithm are compared. The initial solutions are achieved in less than one second and with good results. In the second step, in reasonable times, the initial solution is improved up to 10%.

## Future research lines

To finish, we indicate some open tasks that we have intention of deal with in the future. In relation to the allocation of delay costs in stochastic projects, new allocation rules can be suggested in order to be compared, both theoretically and in practice, with the rule used in Gonçalves-Dosantos et al. (2020d). In addition, we aim to analyze projects with a priori unions, considering that there may be groups of activities managed by the same player; in this context, an allocation rule based on the Owen value can be proposed, both for the case of deterministic durations and for the case of stochastic durations. With respect to the ProjectManagement package we plan to implement new functions, ranging from the methodology proposed above to the creation of a graphical interface for a more intuitive use of the package.

A new topic of interest in project management where various players intervene is to analyze various issues in relation to the effort they make. It can be considered situations in where each one of the players has the control over the time it takes to complete their activities, for instance by allocating more or less resources to carry them out. In this type of situation, the players can generate income by allocating part of the initial resources to other projects, but this may involve additional costs due to penalties for delay. In this context a non-cooperative game appears between the players who choose delays for their individual activities. We plan to look for equilibria of this game, in which players choose to delay their activities either simultaneously or sequentially, depending on the location of their activities within the project. We have started working on this topic during a visit to the Duke University on the autumn of 2019 and we are preparing an article with Fernando Bernstein (Duke University) and Greg DeCroix (University of Wisconsin-Madison) that we expect to complete in a near future.

In egalitarian values for cooperative games with a priori unions, we are willing to obtain new characterizations taking into account the ones proposed by Ferrières (2017) for standard cooperative games. In addition, egalitarian values can be extended to and characterize for graph-restricted cooperative games in the sense of Borm et al. (1992).

When the property for necessary players of Chapter 5 is combined with the properties of additivity, symmetry and null player, a non-efficient value arises. This value can be compared to the Shapley and Banzhaf values in the class of microarray games (Lucchetti et al., 2010), which allows, among other things, to identify the genes which are responsible for a certain disease by using a gene expression data matrix.

Finally, in the vehicle routing problem with multi-compartment and stochastic demands new metaheuristics can be developed in order to improve the solutions obtained and/or the calculation times, based on ant colonies

(Rajappa et al., 2016), simulated annealing (Xiao et al., 2014) or genetic algorithms (Vidal et al., 2013), among others.

## References

Albino D, Scaruffi P, Moretti S, Coco S, Truini M, Di Cristofano C, Cavazzana A, Stigliani S, Bonassi S, Tonini GP (2008). Identification of low intratumoral gene expression heterogeneity in neuroblastic tumors by genomewide expression analysis and game theory. Cancer 113, 1412-1422.

Algaba E, Fragnelli V, Sánchez-Soriano J (2019). Handbook of the Shapley value. CRC Press.

Alonso-Meijide JM, Carreras F, Fiestras-Janeiro MG, Owen G (2007). A comparative axiomatic characterization of the Banzhaf-Owen coalitional value. Decision Support Systems 43, 701-712.

Alonso-Meijide JM, Casas-Méndez B, Fiestras-Janeiro G, Holler MJ (2011). The Deegan-Packel index for simple games with a priori unions. Quality an Quantity 45, 425-439.

Alonso-Meijide JM, Casas-Méndez B, González-Rueda A, Lorenzo-Freire S (2014). Axiomatic of the Shapley value of a game with a priori unions. Top 22, 749-770.

Alonso-Meijide JM, Costa J, García-Jurado I (2019a). Null, Nullifying, and Necessary Agents: Parallel Characterizations of the Banzhaf and Shapley Values. Journal of Optimization Theory and Applications 180, 1027-1035.

Alonso-Meijide JM, Costa J, García-Jurado I (2019b). Values, Nullifiers and Dummifiers. Handbook of the Shapley Value (E Algaba, V Fragnelli, J Sánchez-Soriano eds.), CRC Press, 75-92.

Alonso-Meijide JM, Costa J, García-Jurado I, Gonçalves-Dosantos JC (2020). On egalitarian values for cooperative games with a priori unions. TOP 28, 672-688.

Alonso-Meijide JM, Fiestras-Janeiro G (2002). Modification of the Banzhaf value for games with a coalition structure. Annals of Operations Research 109, 213-227.

Ambrosino D, Sciomachen A (2007). A food distribution network problem: a case study. IMA Journal of Management Mathematics 18, 33-53.

Amer R, Carreras F, Giménez JM (2002). The modified Banzhaf value for games with a coalition structure: an axiomatic characterization. Mathematical Social Sciences 43, 45-54.

Archetti C, Campbell AM, Speranza MG (2016). Multicommodity vs. singlecommodity routing. Transportation Science 50, 461-472.

Arín J, Kuipers J, Vermeulen D (2003). Some characterizations of egalitarian solutions on classes of TU-games. Mathematical Social Sciences 46, 327-345.

Aumann RJ, Drèze J (1974). Cooperative games with coalition structures. International Journal of Game Theory 3, 217-237.

Avella P, Boccia M, Sforza A (2004). Solving a fuel delivery problem by heuristic and exact approaches. European Journal of Operational Research 152, 170-179.

Banzhaf III JF (1964). Weighted voting doesn't work: A mathematical analysis. Rutgers Law Review, 19, 317.

Bazaraa MS, Sherali HD, Shetty CM (2013). Nonlinear programming: theory and algorithms. John Wiley and Sons.

Béal S, Navarro F (2020). Necessary versus equal players in axiomatic studies. Preprint.

Béal S, Rémila E, Solal P (2019). Coalitional desirability and the equal division value. Theory and Decision 86, 95-106.

Benati S, López-Blázquez F, Puerto J (2019). A stochastic approach to approximate values in cooperative games. European Journal of Operational Research 279, 93-106.

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

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.

Berhan E, Beshah B, Kitaw D, Abraham A (2014). Stochastic vehicle routing problem: A literature survey. Journal of Information and Knowledge Management 13, 1450022.

Bertsimas DJ, Simchi-Levi D (1996). A new generation of vehicle routing research: robust algorithms, addressing uncertainty. Operations Research 44, 286-303.

Borm P, Owen G, Tijs S (1992). On the position value for comunication situations. SIAM Journal on Discrete Mathematics 5, 305-320.

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, 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.

Brown GG, Ellis C, Graves GW (1987). Real-time, wide area dispatch of mobil tank trucks. Interfaces 17, 107-120.

Brown GG, Graves GW (1981). Real-time dispatch of petroleum tank trucks." Management Science 27, 19-32.

Caramia M, Guerriero F (2010). A milk collection problem with incompatibility constraints. Interfaces 40, 130-143.

Cano-Berlanga S (2017). GameTheory: Cooperative Game Theory. URL https://CRAN.R-project.org/package=GameTheory. R package version 2.7.

Carnell R (2019). triangle: Provides the Standard Distribution Functions for the Triangle Distribution. URL https://CRAN.R-project.org/package=triangle. R package version 0.12.

Carreras F, Puente A (2015). Coalitional multinomial probabilistic values. European Journal of Operational Research 245, 236-246.

Casajus A (2010). Another characterization of the Owen value without the additivity axiom. Theory and Decision 69, 523-536.

Casajus A, Hüttner F (2014). Null, nullifying, or dummifying players: The difference between the Shapley value, the equal division value, and the equal surplus division value. Economics Letters 122, 167-169.

Casas-Méndez B, García-Jurado I, van den Nouweland A, Vázquez-Brage M (2003). An extension of the $\tau$-value to games with coalition structures. European Journal of Operational Research 148, 494-513.

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 and Operations Research 36, 1726-1730.

Castro J, Gómez D, Tejada J (2014). Allocating slacks in stochastic PERT networks. CEJOR 22, 37-52.

Chajakis ED, Guignard M (2003). Scheduling deliveries in vehicles with multiple compartments. Journal of Global Optimization 26, 43-78.

Chun Y, Park B (2012). Population solidarity, population fair-ranking and the egalitarian value. International Journal of Game Theory 41, 255-270.

Christofides N, Mingozzi A, Toth P, Sandi C (1979). Combinatorial Optimization, chapter 11. Chichester: John Wiley.

Clarke G, Wright J (1964). Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 12, 568-581.

Coelho LC, Laporte G (2015). Classification, models and exact algorithms for multicompartment delivery problems. European Journal of Operational Research 242, 854-864.

Cordeau JF, Laporte G, Savelsbergh MW, Vigo D (2007). Vehicle routing. In Handbooks in Operations Research and Management Science (Barnhart C, Laporte G, eds.) North-Holland: Elsevier 14, 367-428.

Costa J (2016). A polynomial expression of the Owen value in the maintenance cost game. Optimization 65, 797-809.

Crettez B, Deloche R (2019). A law-and-economics perspective on cost-sharing rules for a condo elevator. Review of Law and Economics, DOI: 10.1515/rle-2016-0001.

Csardi G, Nepusz T (2006). The igraph software package for complex network research. InterJournal, Complex Systems:1695. URL http://igraph.org.

Curiel I (2011). Project management games. International Game Theory Review 13, 281-300.

Dantzig GB (1963). Linear programming and extensions. Princeton University Press.

Dantzig GB, Ramser JH (1959). The truck dispatching problem. Management Science 6, 80-91.

Derigs U, Gottlieb J, Kalko J, Piesche M, Rothlauf F, Vogel U (2011). Vehicle routing with compartments: Applications, modelling and heuristics.

OR Spectrum 33, 885-914.

Derigs U, Pullmann M, Vogel U (2013). Truck and trailer routing problems, heuristics and computational experience. Computers and Operations Research 40, 536-546.

Dietzenbacher B, Borm P, Hendrickx R (2017). The procedural egalitarian solution. Games and Economic Behavior 106, 179-187.

Dijkstra EW (1959). A note on two problems in connexion with graphs. Numerische mathematik 1, 269-271.

Driessen TSH, Funaki Y (1991). Coincidence of and collinearity between game theoretic solutions. OR Spectrum 13, 15-30.

Dror M, Ball MO, Golden BL (1985). Computational comparison of algorithms for inventory routing. Annals of Operations Research 4, 3-23.

Dror M (2016). Vehicle routing with stochastic demands: models and computational methods. International Series in Operations Research and Management Science 46, 625-649.

Dror M, Trudeau P (1986). Stochastic vehicle routing with modified savings algorithm. European Journal of Operational Research 23, 228-235.

Dutta B (1990). The egalitarian solution and reduced game properties in convex games. International Journal of Game Theory 19, 153-169.

Dutta B, Ray D (1989). A concept of egalitarianism under participation constraints. Econometrica 57, 615-635.

El Fallahi A, Prins C, Calvo RW (2008). A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. Computers and Operations Research 35, 1725-1741.

Erlang AK (1909). The theory of probabilities and telephone conversations. Nyt Tidsskrift for Matematik B 20, 33-39.

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.

Evdokimov IV, Tsarev RY, Yamskikh TN, Pupkov AN (2018). Using pert and gantt charts for planning software projects on the basis of distributed digital ecosystems. In Journal of Physics: Conference Series, 1074, 012127.

Ferrières S (2017). Nullified equal loss property and equal division values.

Theory and Decision 83, 385-406.

Flores R, Molina E, Tejada J (2007). Evaluating groups with the generalized Shapley value. 4OR 1-32.

Ford LR, Fulkerson DR (1956). Maximal flow through a network. Canadian journal of Mathematics 8, 399-404.

Fragnelli V, Iandolino A (2004). Cost allocation problems in urban solid wastes collection and disposal. Mathematical Methods of Operations Research 59, 447-463.

Gendreau M, Laporte G, Séguin R (1996). Stochastic vehicle routing. European Journal of Operational Research 88, 3-12.

Glover F (1989). Tabu search-part I. ORSA Journal on Computing 1, 190-206.

Glover F (1990). Tabu search-part II. ORSA Journal on Computing 2, 4-32.

Gonçalves-Dosantos JC, Alonso-Meijide JM (2020). New results on egalitarian values for games with a priori unions. Preprint.

Gonçalves-Dosantos JC, Casas-Méndez BV (2020). A two-stage heuristic algorithm for a class of multi-compartmentvehicle routing problems with stochastic demands. Preprint.

Gonçalves-Dosantos JC, García-Jurado I, Costa J (2020a). ProjectManagement: an R Package forManaging Projects. The R Journal 12, 419-436.

Gonçalves-Dosantos JC, García-Jurado I, Costa J (2020b). ProjectManagement: Management of Deterministic and Stochastic Projects. URL https://CRAN.R-project.org/package=ProjectManagement. R package version 1.3.3.

Gonçalves-Dosantos JC, García-Jurado I, Costa J (2020c). Sharing delay costs in stochastic scheduling problems with delays. 4OR - A Quarterly Journal of Operations Research, 18, 457-476.

Gonçalves-Dosantos JC, García-Jurado I, Costa J, Alonso-Meijide JM (2020d). Necessary players and values. Preprint.

Goodson JC (2015). A priori policy evaluation and cyclic-order-based simulated annealing for the multicompartment vehicle routing problem with stochastic demands. European Journal of Operational Research 241, 361-369.

Grabisch M, Kojadinovic I, Meyer P (2015). kappalab: Non-Additive

Measure and Integral Manipulation Functions. URL https://CRAN.R-project .org/package=kappalab. R package version 0.4-7.

Gregoriou G, Kirytopoulos K, Kiriklidis C (2013). Project management educational software (promes). Computer Applications in Engineering Education 21, 46-59.

Gross D (2008). Fundamentals of queueing theory. John Wiley and Sons.

Guitián de Frutos RM, Casas-Méndez, BV To appear. Routing problems in agricultural cooperatives: A model for optimization of transport vehicle logistics. IMA Journal of Management Mathematics. DOI: 10.1093/imaman/dpy010.

Gurram P, Kwon H, Davidson C (2016). Coalition game theory-based feature subspace selection for hyperspectral classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 9, 2354-2364.

Hall NG (2012). Project management: Recent developments and research opportunities. Journal of Systems Science and Systems Engineering 21, 129-143.

Hegazy T (1999). Optimization of resource allocation and leveling using genetic algorithms. Journal of construction engineering and management 125, 167-175.

Henke T, Speranza MG, Wäscher G (2018). A branch-and-cut algorithm for the multi-compartment vehicle routing problem with exible compartment sizes. Annals of Operations Research. DOI: 10.1007/s10479-018-2938-4.

Herroelen W, Leus R (2005). Project scheduling under uncertainty: survey and research potentials. European Journal of Operational Research 165, 289-306.

Hillier FS, Lieberman GJ (2002). Introduction to Operations Research. McGraw-Hill.

Hsu C-I, Hung S-F, Li H-C (2007). Vehicle routing problem with time-windows for perishable food delivery. Journal of Food Engineering 80, 465-475.

Hu XF (2019). The weighted Shapley-egalitarian value for cooperative games with a coalition structure. TOP, DOI: 10.1007/s11750-019-00530-4.

Hu XF, Xu GJ, Li DF (2019). The egalitarian efficient extension of the Aumann-Drèze value. Journal of Optimization Theory and Applications 181, 1033-1052.

Huang SH (2015). Solving the multi-compartment capacitated location routing problem with pickup-delivery routes and stochastic demands. Computers and Industrial Engineering 87, 104-113.

Ju Y, Borm P, Ruys P (2007). The consensus value: A new solution concept for cooperative games. Social Choice and Welfare 28, 685-703.

Karush W (1939). Minima of functions of several variables with inequalities as side constraints. Master's thesis. Department of Mathematics, University of Chicago.

Kelley D (2018). plan: Tools for Project Planning. URL https://CRAN.R-project.org/package=plan. R package version 0.4-3.

Kelley JE (1961). Critical-path planning and scheduling: Mathematical basis. Operations Research 9, 296-320, 1961.

Khmelnitskaya AB, Yanovskaya EB (2007). Owen coalitional value without additivity axiom. Mathematical Methods of Operations Research 66, 255-261.

Klijn F, Slikker M, Tijs S, Zarzuelo J (2000). The egalitarian solution for convex games: some characterizations. Mathematical Social Sciences 40, 111-121.

Kruskal JB (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematical society, 7, 48-50.

Kuhn HW (1955). The Hungarian method for the assignment problem. Naval research logistics quarterly, 2, 83-97.

Kuznietsov KA, Gromov VA, Skorohod, VA (2016). Cluster-based supply chain logistics: a case study of a Ukrainian food distributor. IMA Journal Managment Mathematics 28, 553-578.

Laporte G (2009). Fifty years of vehicle routing. Transportation Science 43, 408-416

Lorenzo-Freire S (2016). On new characterizations of the Owen value. Operations Research Letters 44, 491-494.

lp_solve, Konis K, Schwendinger F (2020). lpSolveAPI: R Interface to lp_solve Version 5.5.2.0. URL https://CRAN.R-project.org/package=lpSolve API. R package version 5.5.2.0-17.7.

Lucchetti R, Moretti S, Patrone F, Radrizzani P (2010). The Shapley and Banzhaf values in microarray games. Computers and Operations Research

37, 1406-1412.

Malcolm DG, Roseboom JH, Clark CE, Fazar W (1959). Application of a technique for research and development program evaluation. Operations research 7, 646-669.

Mendoza JE, Castanier B, Guéret C, Medaglia AL, Velasco N (2010). A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. Computers and Operations Research 37, 1886-1898.

Mendoza JE, Castanier B, Guéret C, Medaglia AL, Velasco N (2011). Constructive heuristics for the multicompartment vehicle routing problem with stochastic demands. Transportation Science 45, 346-363.

Monge G (1781). Mémoire sur la théorie des déblais et des remblais. Histoire de l'Académie Royale des Sciences de Paris.

Moretti S, Patrone F (2008). Transversality of the shapley value. Top 16, 1-41, 2008.

Muñoz JC (2015). PlotPrjNetworks: Useful Networking Tools for Project Management. URL https: //CRAN.R-project.org/package=PlotPrjNetworks. R package version 1.0.0.

Muyldermans L, Pang G (2010). On the benefits of co-collection: Experiments with a multi-compartment vehicle routing algorithm. European Journal of Operational Research 206, 93-103.

Myerson RB (1980). Conference structures and fair allocation rules. International Journal of Game Theory 9, 169-182.

Oppen, J. and L∅kketangen, A (2008). A tabu search approach for the livestock collection problem. Computers and Operations Research 35, 3213-3229.

Osman I (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. Annals of Operations Research 41, 421-451.

Ostermeier M, Hübner A (2018). Vehicle selection for a multi-compartment vehicle routing problem. European Journal of Operational Research 269, 682-694.

Owen G (1975). Multilinear extensions and the Banzhaf value. Naval Research Logistics Quarterly 22, 741-750.

Owen G (1977) Values of games with a priori unions. Mathematical Economics and Game Theory (R Henn, O Moeschlin, eds.), Springer, 76-88.

Owen G (1981) Modification of the Banzhaf-Coleman index for games with a priori unions. Power, voting, and voting power. Physica, Heidelberg, 232-238.

Oyola J, Arntzen H, Woodru D (2016). The stochastic vehicle routing problem, a literature review. http://www.optimizationonline.org/DB FILE/2016/01/5299.pdf.

Özdamar L, Ulusoy G (1995). A survey on the resource-constrained project scheduling problem. IIE transactions 27, 574-586, 1995.

Pandelis DG, Kyriakidis EG, Dimitrakos TD (2012). Single vehicle routing problems with a predefined customer sequence, compartmentalized load and stochastic demands. European Journal of Operational Research 217, 324-332.

Pollaris H, Braekers K, Caris A, Janssens G, Limbourg S (2015). Vehicle routing problems with loading constraints: State-of-the-art and future directions. OR Spectrum 37, 297-330.

Punmia BC, Khandelwal KK (2002). Project Planning and Control with PERT and CPM. Firewall media.

R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Rajappa GP, Wilck JH, Bell JE (2016). An Ant Colony Optimization and Hybrid Metaheuristics Algorithm to Solve the Split Delivery Vehicle Routing Problem. International Journal of Applied Industrial Engineering (IJAIE) 3, 55-73.

Ruiz LM, Valenciano F, Zarzuelo JM (1998). The family of least square values for transferable utility games. Games and Economic Behavior 24, 109-130.

Saavedra-Nieves A, García-Jurado I, Fiestras-Janeiro G (2018). Estimation of the Owen value based on sampling. The Mathematics of the Uncertain: A Tribute to Pedro Gil (E Gil, E Gil, J Gil, MA Gil, eds.), Springer, 347-356.

Salas-Morera L, Arauzo-Azofra A, García-Hernández L, Palomo-Romero JM, César HM (2013). Ppcproject: An educational tool for software project management. Computers and Education 69, 181-188.

San Cristóbal JR (2014). Cost allocation between activities that have caused delays in a project using game theory. Procedia Technology 16, 1017-

1026.

Schmeidler D (1969). The nucleolus of a characteristic function form game. SIAM Journal on Applied Mathematics 17, 1163-1170.

Schmitz K, Mahapatra R, Nerur S (2019). User engagement in the era of hybrid agile methodology. IEEE Software 36, 32-40.

Selten R (1972). Equal share analysis of characteristic function experiments. Contributions to Experimental Economics III. (Sauermann H, ed.), Mohr Siebeck, 130-165.

Shapley LS (1953). A value for n-person games. Contributions to the Theory of Games 2, 307-317.

Sievert C (2020). Interactive Web-Based Data Visualization with R, plotly, and shiny. Chapman and Hall/CRC. ISBN 9781138331457. https://plotly-r.com.

Silvestrin PV, Ritt M (2017). An iterated tabu search for the multi-compartment vehicle routing problem." Computers and Operations Research 81, 192-202.

Strumbelj E, Kononenko I (2010). An efficient explanation of individual classifications using game theory. Journal of Machine Learning Research 11, 1-18.

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.

Tatarakis A, Minis I (2009). Stochastic single vehicle routing with a predefined customer sequence and multiple depot returns. European Journal of Operational Research 197, 557-571.

Tijs S (1981). Bounds for the core and the $\tau$-value. Game Theory and Mathematical Economics (O Moeschlin, D Pallaschke eds.), North Holland, 123-132.

Tolstoi AN (1930). Methods of finding the minimal total kilometrage in cargo transportation planning in space. TransPress of the National Commissariat of Transportation 1, 23-55.

van den Brink R (2007). Null or nullifying players: the difference between the Shapley value and equal division solutions. Journal of Economic Theory

136, 767-775.

van den Brink R, Funaki Y (2009) Axiomatizations of a class of equal surplus sharing solutions for TU-games. Theory and Decision 67, 303-340.

van den Brink R, Chun Y, Funaki Y, Park B (2016). Consistency, population solidarity, and egalitarian solutions for TU-games. Theory and Decision 81, 427-447.

van den Brink R, van der Laan G, Moes N (2015). Values for transferable utility games with coalition and graph structure. Top 23, 77-99.

van der Bruggen R, Gruson L, Salomon M (1995). Reconsidering the distribution structure of gasoline products for a large oil company. European Journal of Operational Research 81, 460-473.

Vázquez-Brage M, García-Jurado I, Carreras F (1996). The Owen value applied to games with graph-restricted communication. Games and Economic Behavior 12, 42-53.

Vázquez-Brage M, van den Nouweland A, García-Jurado I (1997). Owen's coalitional value and aircraft landing fees. Mathematical Social Sciences 34, 273-286.

Vidal T, Crainic TG, Gendreau M, Prins C (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. Computers and operations research 40, 475-489.

Von Neumann J, Morgenstern O (1947). Theory of games and economic behavior, 2nd rev.

Wang W, Sun H, van den Brink R, Xu G (2019). The family of ideal values for cooperative games. Journal of Optimization Theory and Applications 180, 1065-1086.

Xia Y, Fu Z, Pan L, Duan F (2018). Tabu search algorithm for the distance-constrained vehicle routing problem with split deliveries by order. Plos One 13, e0195457. DOI: 10.1371/journal.pone.0195457.

Xiao Y, Zhao Q, Kaku I, Mladenovic N (2014). Variable neighbourhood simulated annealing algorithm for capacitated vehicle routing problems. Engineering Optimization, 46, 562-579.

Yang Z, Zhang Q (2015). Resource allocation based on DEA and modified Shapley value. Applied Mathematics and Computation 263, 280-286.

# Chapter 1

# Sharing delay costs in stochastic scheduling problems with delays

JC Gonçalves-Dosantos, I García-Jurado, J Costa

4OR ISSN: 1614-2411 Volume: 18 Pages: 457-476

## 1.1 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 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 Cristóbal (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.

## 1.2 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 1.1.** 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 1.1 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 $c(d(N, \prec, x^0)) = 0$.

[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.

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 1.2.** A rule for deterministic scheduling problems with delays is a map $\varphi$ 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:

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 1.3.** 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 $\Phi$ 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 1.4.** 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 1.1 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 1.1 and 1.4 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 1.2.

**Definition 1.5.** A rule for stochastic scheduling problems with delays is a map $\psi$ 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 1.1 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 1.1.** 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 1.2.

- 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 1.1.

## 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 1.6.** 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 $\Pi_N$ the set of permutations of $N$.

**Anonimity.** A rule for stochastic scheduling problems with delays $\psi$ 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 $SP^\pi$ 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 $y \in \mathbb{R}^N$.

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

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

$, X^0, x, C')$, 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 $\psi$ 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 $j \in N\backslash i$, then

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

**Equal responsability for two.** A rule for stochastic scheduling problems with delays $\psi$ 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 $\psi$ 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 $\psi$ satisfies independence of irrelevant delays if, for all $SP = (N, \prec, X^0, x, C)$ such that

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

for $i \in N$ and for all $S \subseteq N\backslash i$, 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 1.1.** *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 $\psi$ 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.

[5]The proof of Theorem 1.1 is available to readers upon request to the authors.

**Theorem 1.2.** *The Shapley rule is the unique rule for stochastic scheduling problems with delays that satisfies the balancedness property.*

*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,

$$SSh_i(SP) - SSh_i(SP_{-j}) = \Phi_i(v^{SP}) - \Phi_i(v^{SP_{-j}}), \tag{1.1}$$

$$SSh_j(SP) - SSh_j(SP_{-i}) = \Phi_j(v^{SP}) - \Phi_j(v^{SP_{-i}}). \tag{1.2}$$

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,

$$v^{SP_{-k}} = v^{SP}_{-k} \text{ for all } k \in N. \tag{1.3}$$

Considering (1.3) and that Myerson (1980) proved that the Shapley value of a TU-game satisfies a balancedness property, the equations in (1.1) and (1.2) 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

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

[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 $R(SP') \neq SSh(SP')$.

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

Now, considering the minimality of $SP$,

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

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 1.1 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 1.2.** 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 $\exp(\alpha)$ denotes the exponential distribution with parameter $\alpha$ (i.e., with mean $1/\alpha$). As we remarked in Example 1.1, the table does not give the entire binary relation $\prec$ but only the immediate precedences. For instance, because 1 precedes 2, 2 precedes 5

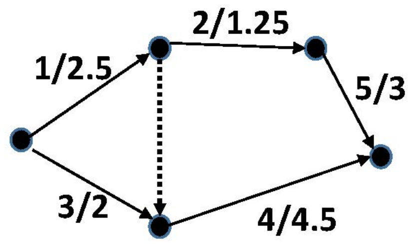


Figure 1.1: PERT graph of the project in Example 1.2.

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

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

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 1.1, 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

$$\sum_{i\in\{1,\ldots,1000\}} \frac{SSh(SP^i)}{1000} = (0.12857, 0.06844, 0.06686, 0.10757, 0.93790), \quad (1.4)$$

where the average observed cost was 1.30935. Note that (1.4) 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 1.2 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

- $Z_i^2$ 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 1.2 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 $Z_i^1$ and $Z_i^2$ are significantly different for each $i$, which strengthens the interest of the rule $SSh$. Finally, Table 1.1 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 $X_5^0$ and not by $E(X_5^0)$.

| $Sh$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $\geq 0$ | 70.5 | 74.9 | 100 | 91.1 | 100 |
| $< 0$ | 29.5 | 25.1 | 0.0 | 8.9 | 0 |

| $SSh$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $\geq 0$ | 75.0 | 85.5 | 100 | 95.5 | 48.3 |
| $< 0$ | 25 | 14.5 | 0.0 | 4.5 | 51.7 |

Table 1.1: Positive and negative payments for the $Sh$ rule (left) and $SSh$ rule (right).

Example 1.2 raises one controversial property of $Sh$ and $SSh$: they can propose 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 $SSh$.

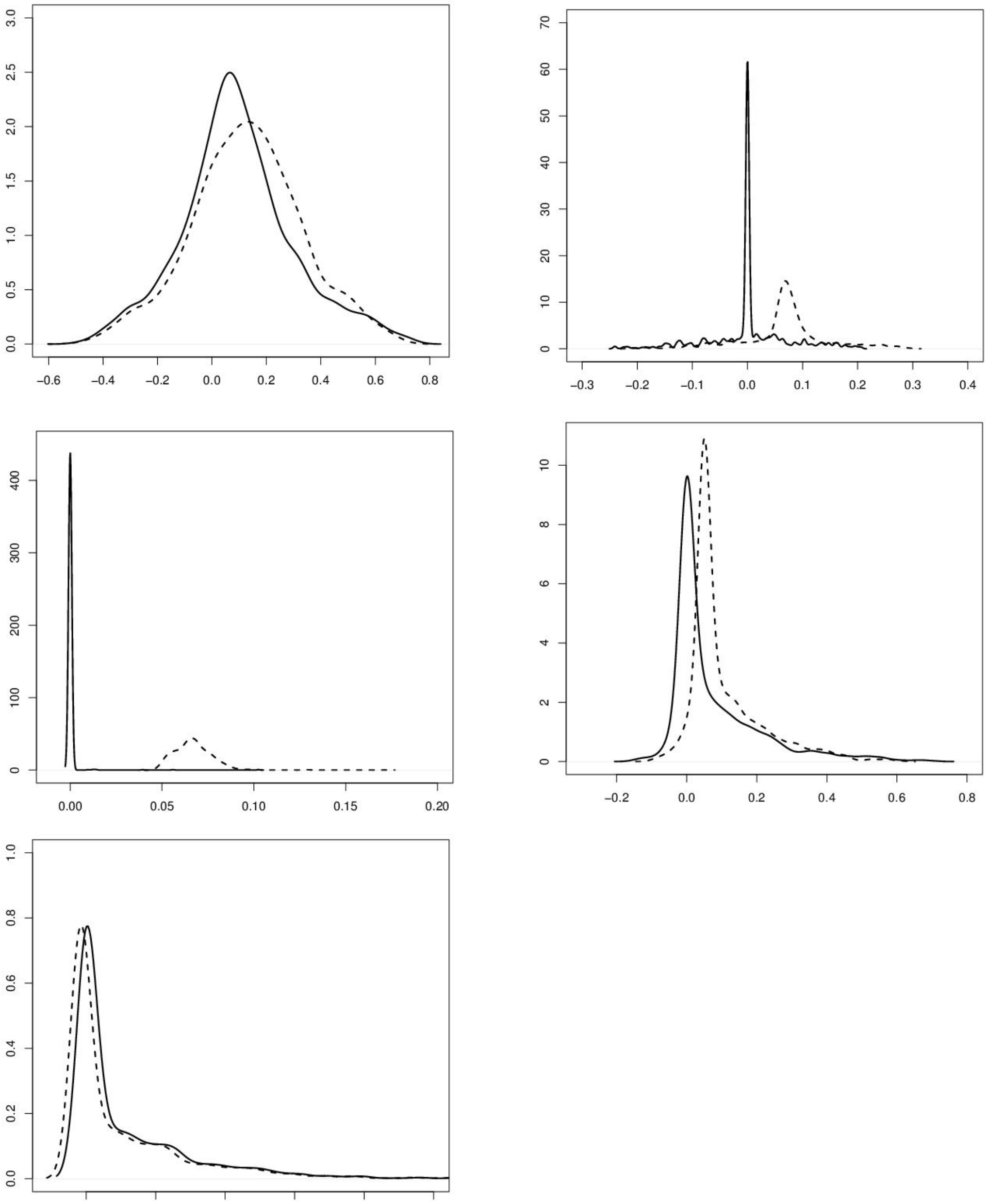


Figure 1.2: Density estimations of the variables $Z_i^1$ (solid line) and $Z_i^2$ (dotted line).

**Theorem 1.3.** *(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 $C(x_i, y_{N\backslash i}) \geq C(x_i^0, y_{N\backslash i})$. 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 1.3). 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 $SSh$.

## 1.4 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 $v^{SP}(S) = E(C(x_S, X^0_{N\setminus S}))$, 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 $\Pi_N$ the set of permutations of $N$. We then calculate $|N|$ real numbers as follows:

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

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

where $Pre^i(\pi^j)$ denotes the set of activities that precede activity $i$ in the permutation $\pi^j$, i.e., $Pre^i(\pi^j) = \{\pi^j_1, \dots, \pi^j_{k-1} \mid i = \pi^j_k\}$. 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

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

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

$cont = cont + 1$

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 1.2 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 1.2: Computation times in seconds.

Table 1.2 illustrates that it is possible to estimate the Shapley rule for stochastic scheduling problems with delays in an acceptable time, even if

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

there are hundreds of activities, by using a desktop computer and free software.

Table 1.3 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 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 1.3: Errors for $v^{SP}$ and the Shapley rule.

Table 1.3 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.

# References

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

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

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

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 Cristóbal 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.

# Chapter 2

# ProjectManagement: an R Package for Managing Projects

JC Gonçalves-Dosantos, I García-Jurado, J Costa

The R Journal ISSN: 2073-4859 Volume: 12 Pages: 419-436

## 2.1 Introduction

Project management is an important body of knowledge and practices that comprises the planning, organisation and control of resources to achieve one or more pre-determined objectives. The most commonly used methods for project planning are PERT (Program Evaluation and Review Technique model) and CPM (Critical Path Method). PERT/CPM analyses the tasks involved in completing a project, especially the time needed to complete each task, and computes the minimum time needed to complete the total project. Through the data obtained in the analysis of the project, PERT/CPM identifies the critical activities, which are those for which any disturbance in its duration modifies the minimum time of execution of the project. Also, it obtains the times that can be assigned to non-critical activities, called slacks, in addition to their fixed durations, to give them flexibility. Project management often deals with the problem of redistribution of resources. Sometimes it is convenient to reduce the time of an activity by increasing the assigned costs. Other times, when the availability of resources is limited in a period of time, it may be necessary to level the use of those resources. These situations require a re-planning of the project.

Even with good project management, once the project has been carried out and the actual durations of the activities are known, there can be a delay in the completion time of the project. When the delay generates an additional cost, ways are needed to distribute the cost of the delay among the different tasks involved. To solve this problem we can use cooperative game theory and rules based on bankruptcy problems.

The essential elements related to project management can be found in Castro et al. (2007) or in Hillier and Lieberman (2001). Project management techniques have been widely used in all fields of engineering. Hall (2012) reviews the impact that such techniques have in various fields and their broad business opportunities. Their fields of application vary from classical construction and engineering to information technology and software development, including modern agile methods. Schmitz et al. (2019) also argue the usefulness of traditional project management techniques in the context of agile methodology. Evdokimov et al. (2018) include a case study that shows the current relevance of project management techniques in software development. Özdamar and Ulusoy (1995) present a survey of the problem of resource constraints. To distribute the delay cost of the project among the activities, Brânzei et al. (2002) provide two rules using, respectively, a game theoretical and a bankruptcy-based approach, and Bergantiños et al. (2018) introduce and analyse a consistent rule based on the Shapley value.

A well-known project management software is Microsoft Project. This tool is designed to create and control a project, through the allocation of resources to tasks, the management of budget and workloads, as well as monitoring developments. Microsoft Project is not open source and its license is fee-based. Other project management applications have been created as free software, such as OpenProj, PpcProject or ProMes (Gregoriou et al., 2013). In Salas-Morera et al. (2013) we can see a useful comparison of these applications.

The aforementioned tools are written in Java or Phyton. To the best of our knowledge, there are only two packages in R available for project management. PlotPrjNetworks (Muñoz, 2015) and plan (Kelley, 2018) are packages that offer the user the creation of a Gantt diagram for the visualization of the project structure. In our opinion, a tool was missing to manage a project from its development to its control. We believe that such a tool would be useful for the user community because it could be integrated with other tools developed in R, it could be easily modified to suit the specific needs of each user, and it could be wrapped into a graphical interface.

In this paper, we introduce ProjectManagement[1] (Gonçalves-Dosantos et al., 2020b), a new R package that provides the necessary tools to manage projects in a broad sense. It calculates the critical activities, the slack of each activity, the minimum duration of the project and the early and last times of each activity. It plots a graph of the project and the schedule. The package also allows cost management to reduce the minimum project time, as well as resource management. Once the actual durations of the activities are known, it is possible to distribute the delay generated in the project among the different activities. When activity durations are considered random variables, the package provides additional functionality. In particular, it calculates the average duration of the project and the criticality index of each activity. It plots a representation of the project duration distribution and the early and last times of the activities. And it calculates several allocation proposals of the delay cost when the project has been completed and the actual duration of the activities is known.

The paper is organized as follows. First, we recall the basic definitions of project management and present different ways to distribute the delay cost when durations are assumed to be known and when they are random variables. Then, we provide a description of ProjectManagement. Finally, we illustrate the use of the package by way of examples.

## 2.2 Project management

In this section we discuss the basic concepts of deterministic and stochastic projects with a special focus on allocating the delay cost among the project activities. The aim of this section is to provide a brief (and quick) survey of the methodologies implemented in the R package ProjectManagement that we introduce later, as well as to indicate the main bibliographical sources in which interested readers can deepen their knowledge of each of these methodologies.

Let $X$ be a finite non-empty set and $N$ be a set of ordered pairs $(x_1, x_2)$, with $x_1, x_2 \in X$ and $|N| = n$. A directed graph is a pair $G = (X, N)$, where $X$ is the set of nodes and $N$ is the set of arcs. We say that an arc $i = (x_{i,1}, x_{i,2}) \in N$ starts at node $x_{i,1} \in X$ and ends at $x_{i,2} \in X$. A node $x_s \in X$ is a source node if there is no arc $i \in N$ such that $x_{i,2} = x_s$. A node $x_e \in X$ is a sink node if there is no arc $i \in N$ such that $x_{i,1} = x_e$. A cycle is a set of arcs $i_0, i_1, ..., i_m \in N$ such that $x_{i_j,2} = x_{i_{j+1},1}$, with $j \in \{0, ..., m-1\}$,

[1]https://github.com/Juan-Goncalves-Dosantos/ProjectManagement.git

and $x_{i_m,2} = x_{i_0,1}$. To illustrate the concept of directed graph consider the following example. Take graph $G = (X, N)$ given by $X = \{a, b, c, d\}$ and $N = \{1 = (a, b), 2 = (a, c), 3 = (b, d), 4 = (c, d)\}$. The diagram representing this graph is depicted in Figure 2.1. This graph has one source ($a$) and one sink ($d$). Arc 3, for instance, starts at node $b$ and ends at node $d$. This graph has no cycles. However, if we add an arc $5 = (d, a)$, the resulting graph has two cycles: $1, 3, 5$ and $2, 4, 5$.

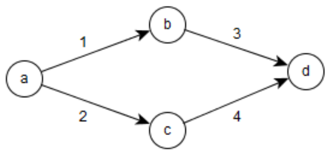


Figure 2.1: Diagram of the directed graph $G = (X, N)$. The circles represent the nodes and the arrows represent the arcs. This is the standard way of depicting a graph.

A deterministic project $P$ is a tuple $P = (G, x^0)$, where $G = (X, N)$ is a directed graph without cycles, with one source node and one sink node, and $x^0 \in \mathbb{R}^n_+$ is the vector of non-negative planned durations. In this context, $N$ represents the set of activities in the project. We denote by $\mathcal{P}^N$ the family of all deterministic projects with set of activities $N$, and by $\mathcal{P}$ the family of all deterministic projects.

In a deterministic project $P = (G, x^0) \in \mathcal{P}^N$, you can calculate the minimum duration of $P$, denoted by $D(G, x^0)$, i.e. the minimum time the project needs to complete all activities taking into account the structure of the graph. This time can be obtained as the solution of a linear programming problem, and thus, can be easily computed. Alternatively, $D(G, x^0)$ can be calculated using a project planning methodology like PERT (see, for instance, Hillier and Lieberman (2001) for details on project planning).

Given a node $x \in X$, we define the set of immediate predecessors of $x$ as the set of activities ending in $x$, $Pred(x) = \{i \in N / x_{i,2} = x\}$, and the immediate successors of $x$ as $Suc(x) = \{i \in N / x = x_{i,1}\}$. We define the earliest time $D_i^E(G, x^0)$ of an activity $i \in N$ as the minimum time required to complete all immediate predecessor activities of $x_{i,1}$, i.e. the *earliest start time* the activity $i$ can start taking into account the graph

$$D_i^E\left(G, x^0\right) = \max_{j \in Pred(x_{i,1})} \{D_j^E\left(G, x^0\right) + x_j^0\}.$$

The *latest completion time* $D_i^L(G, x^0)$ of an activity $i \in N$ is the latest point in time when the activity can end without delaying the project

$$D_i^L\left(G,x^0\right)=\begin{cases}\max\limits_{j\in N;Suc(x_{j,2})=\emptyset}\{D_j^E\left(G,x^0\right)+x_j^0\} & if \quad Suc\left(x_{i,2}\right)=\emptyset,\\ \\ \min\limits_{j\in Suc(x_{i,2})}\{D_j^L\left(G,x^0\right)-x_j^0\} & otherwise.\end{cases}$$

It is easy to see that $D_i^E\left(G,x^0\right)\leq D_i^L\left(G,x^0\right)$ for all $i\in N$. Also, we can calculate the minimum duration of a project, using the earliest start times, as $D\left(G,x^0\right)=\max_{i\in N}\{D_i^E\left(G,x^0\right)+x_i^0\}$.

We define the slack $S_i\left(G,x^0\right)$ of an activity $i\in N$ as the maximum time, in addition to $x_i^0$, that $i$ can use to complete its task without delaying the project

$$S_i\left(G,x^0\right)=D_i^L\left(G,x^0\right)-D_i^E\left(G,x^0\right)-x_i^0.$$

If the slack for an activity is equal to 0, then this activity is critical, i.e. any perturbation in its time modifies the duration of the project. We can also define two other types of slack. The free slack of an activity is the maximum amount of time that this activity can be delayed without causing a delay in the project or in the earliest time of the other activities. The free slack of an activity can be calculated as

$$FS_i\left(G,x^0\right)=\min_{j\in Suc(x_{i,2})}\{D_j^E\left(G,x^0\right)\}-D_i^E\left(G,x^0\right)-x_i^0.$$

The independent slack of an activity is the maximum time that the activity duration can be increased without affecting the times of others activities

$$IS_i\left(G,x^0\right)=\max\{\min_{j\in Suc(x_{i,2})}\{D_j^E\left(G,x^0\right)\}-D_i^L\left(G,x^0\right)-x_i^0,0\}.$$

Given the slack of an activity, we define the *latest start time* as the latest time that an activity can start without delaying the project

$$D_i^{EL}\left(G,x^0\right)=D_i^E\left(G,x^0\right)+S_i\left(G,x^0\right)$$

and the *earliest completion time* as the earliest time in which an activity can end if it starts in its earliest start time

$$D_i^{LE}\left(G,x^0\right)=D_i^L\left(G,x^0\right)-S_i\left(G,x^0\right).$$

Besides the schedule of a project, we can manage the resources allocated to the activities. The minimal cost expediting or MCE method (Kelley, 1961) considers that the duration of some activities can be reduced by increasing the resources allocated to them and thus the implementation costs. An MCE problem is a tuple $(P, \bar{x}^0, c, D)$, where $P$ is a deterministic project, $\bar{x}^0 \in \mathbb{R}^n_+$ is the vector of minimum durations, that is, for each activity $i \in N$, $\bar{x}^0_i$ is the minimum duration that the activity can take if the resources allocated to carry it out are increased, $c \in \mathbb{R}^n$ is the vector of unit costs, that is, for each activity $i \in N$, $c_i$ is the cost of accelerating a unit of time the duration of $i$, and $D$ is the minimum duration of the project we are trying to achieve, with $D < D(G, x^0)$. This problem can be solved as a linear programming problem.

Two other interesting problems that arise from the management of resources are the levelling and the allocation (Hegazy, 1999). These problems take into account that in order for activities to be carried out in the estimated time, a certain level of resources must be used. The problem of levelling of resources is to find a schedule that allows to execute the project in its minimum duration time $D(G, x^0)$ whilst the use of resources is as uniform as possible over time. In the problem of allocation of resources, the level of resources available in each period of time is limited. The aim is to find the minimum duration time and a schedule for the execution of the project taking into account this resource constraint. Given the complex nature of these problems, their exact resolution is computationally demanding. The most common practice is to use heuristic methods to solve them.

Once the project is completed, we can know the actual (observed) duration of the activities and, therefore, whether there has been a delay in the project, that is, whether the actual duration of the project has been different than expected. We define a deterministic project with delays as a tuple $CP = (G, x^0, x, C)$, where $(G, x^0)$ is a deterministic project, $x \in \mathbb{R}^n_+$ is the vector of actual duration of the activities, and $C : \mathbb{R}_+ \to \mathbb{R}$ is the delay cost function. We assume that $C$ only depends on the duration of the project, it is a non-decreasing function, and $C(D(G, x^0)) = 0$. In practice, the most commonly used functions, for a vector $y \in \mathbb{R}^n_+$, are

$$C(D(G, y)) = D(G, y) - \delta \tag{2.1}$$

with $\delta \in \mathbb{R}_+$, for example $\delta = D(G, x^0)$.

We denote by $\mathcal{CP}^N$ the family of all deterministic projects with delays with set of activities $N$, and by $\mathcal{CP}$ the family of all deterministic projects with delays.

In a deterministic project with delays $CP \in \mathcal{CP}^N$, we may need to allocate $C(D(G, x))$ among the activities. This can be useful for several reasons. For example, it can serve as an incentive for those responsible for the activities that have been delayed to be more diligent in similar projects that we may carry out with them in the future; or it can be a mechanism to distribute among those responsible for the activities that have been delayed the financial penalty that the project manager has contractually guaranteed. Brânzei et al. (2002) propose two rules based on bankruptcy problems to address this problem: the Proportional rule and the Truncated Proportional rule. Although they define these rules for the case $x_i \geq x_i^0$, we do not consider this restriction. These rules are only defined when the sum of the individual delays is not zero.

The Proportional rule for deterministic scheduling problems with delays $\phi$ is defined, for each $i \in N$, by

$$\phi_i = \frac{x_i - x_i^0}{\sum_{j \in N} x_j - x_j^0} \cdot C\left(D\left(G, x\right)\right).$$

The Truncated Proportional rule for deterministic scheduling problems with delays $\bar{\phi}$ is defined, for each $i \in N$, by

$$\bar{\phi}_i = \frac{\min\{x_i - x_i^0, C\left(D\left(G, x\right)\right)\}}{\sum_{j \in N} \min\{x_j - x_j^0, C\left(D\left(G, x\right)\right)\}} \cdot C\left(D\left(G, x\right)\right).$$

In Bergantiños et al. (2018), the problem of allocating the delay costs is addressed in the context of cooperative game theory using a Shapley rule. As we illustrate later in an example, the Shapley rule allocates the delay costs in a more sensible way than the proportional rules, at least in some cases. It is much more costly to compute it but, in general, the extra effort is worthwhile. A TU-game is a pair $(N, v)$ where $N$ is a finite non-empty set, and $v$ is a map from $2^N$ to $\mathbb{R}$ with $v(\emptyset) = 0$. We say that $N$ is the player (activity) set of the game and $v$ is the characteristic function of the game, and we usually identify $(N, v)$ with its characteristic function $v$. The Shapley value, an allocation rule in cooperative game theory, is a map $\Phi$ that associates to each 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 $v(N)$ among 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

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

Since its introduction by Shapley (1953), the Shapley value has proved to be one of the most important rules in cooperative game theory and to have applications in many practical problems (see, for instance, Moretti and Patrone (2008)).

Bergantiños et al. (2018) define the Shapley rule for deterministic projects with delays $Sh$ as $Sh(CP) = \Phi\left(v^{CP}\right)$, where for all $CP = (G, x^0, x, C) \in \mathcal{CP}^N$:

- $v^{CP}$ is the TU-game with set of players $N$ given by

$$v^{CP}(S) = C\left(D\left(G, \left(x_S, x^0_{N\setminus S}\right)\right)\right)$$

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

- $\Phi\left(v^{CP}\right)$ denotes the proposal of the Shapley value for $v^{CP}$.

The calculation of the Shapley value has, in general, an exponential complexity. In this context, its exact calculation is impossible in practice, even for a moderate number of activities. 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 practical terms, this estimate is a reasonable solution.

Next, we introduce a generalization of the model and the rules described above. It follows the results in Gonçalves-Dosantos et al. (2020a). If instead of $x^0_i$, the planned duration of activity $i \in N$ , we consider the non-negative random variable $X^0_i$ describing the duration of $i$, we can define a stochastic project $SP$ as tuple $SP = (G, X^0)$. Unlike in the deterministic setting, the duration of activities, the duration of the project, as well as the early and last times are now random variables instead of fixed numbers.

A stochastic project with delays is a tuple $SCP = (G, X^0, x, C)$, where $(G, X^0)$ is a stochastic project, $x$ is the vector of actual durations, and $C : \mathbb{R}_+ \to \mathbb{R}$ is the delay cost function. We assume that $C$ is non-decreasing and $C(D(G, 0)) = 0$, where $0 \in R^n$ is the vector with all components equal to zero. Proportional rules can be extended to stochastic projects with delays in a straightforward way.

The Stochastic Proportional rule for deterministic scheduling problems with delays $\phi$ is defined, for each $i \in N$, by

$$\phi_i = \frac{x_i - E\left(X^0_i\right)}{\sum_{j\in N} x_j - E\left(X^0_j\right)} \cdot C\left(D\left(G, x\right)\right).$$

The Stochastic Truncated Proportional rule for deterministic scheduling problems with delays $\bar{\phi}$ is defined, for each $i \in N$, by

$$\bar{\phi}_i = \frac{\min\{x_i - E\left(X_i^0\right), C\left(D\left(G,x\right)\right)\}}{\sum_{j\in N} \min\{x_j - E\left(X_j^0\right), C\left(D\left(G,x\right)\right)\}} \cdot C\left(D\left(G,x\right)\right).$$

Also, we can extend the Shapley rule to the stochastic context. Let us see two extensions of the rule. The Shapley rule for stochastic projects with delays $SSh$ is defined by $SSh\left(SCP\right) = \Phi\left(v^{SCP}\right)$, where

- $v^{SCP}$ is the TU-game with set of players $N$ given by

  $$v^{SCP}\left(S\right) = E\left(C\left(D\left(G,\left(x_S, X^0_{N\setminus S}\right)\right)\right)\right)$$

  for all non-empty $S \subseteq N$,[2] and

- $\Phi\left(v^{SCP}\right)$ denotes the proposal of the Shapley value for $v^{SCP}$.

As an alternative to the previous rule, the Shapley rule in two steps for stochastic projects with delays $SSh2$ is defined by $SSh2\left(SCP\right) = \Phi\left(v_1^{SCP}\right) + \Phi\left(v_2^{SCP}\right)$, where

- $v_1^{SCP}$ is the TU-game with set of players $N$ given by

  $$v_1^{SCP}\left(S\right) = E\left(C\left(D\left(G,\left(x_S, X^0_{N\setminus S}\right)\right)\right)\right) - E\left(C\left(D\left(G, X^0\right)\right)\right)$$

  for all $S \subseteq N$,

- $v_2^{SCP}$ is the TU-game with set of players $N$ given by

  $$v_2^{SCP}\left(S\right) = E\left(C\left(D\left(G,\left(X^0_S, 0_{N\setminus S}\right)\right)\right)\right)$$

  for all $S \subseteq N$, where $0 \in \mathbb{R}^n$ is the vector with all components equal to zero, and

- $\Phi\left(v_1^{SCP}\right)$ and $\Phi\left(v_2^{SCP}\right)$ denote the proposal of the Shapley value for $v_1^{SCP}$ and $v_2^{SCP}$.

In general, the calculation of $v^{SCP}$, $v_1^{SCP}$ and $v_2^{SCP}$ is very complex. In our package, we use simulations to approximate these characteristic functions.

[2] As in all TU-games, we define $v^{SCP}\left(\emptyset\right) = 0$.

## 2.3 ProjectManagement package

ProjectManagement is a new R package that allows the user to address different tasks in project management. The user can obtain the duration of a project and a schedule of activities, and can plot this schedule for a better understanding of the problem. When the actual durations of each activity are observed, the package proposes several allocations of the delay cost, if there was any, among the activities. In the stochastic context, the package estimates the average duration of the project and plots the density functions of the following random variables: duration of the project, and early and last times of the activities. As in the deterministic case, it can make an allocation of the delay cost, if any.

The following dependencies of the package must be taken into account: triangle (Carnell, 2019), plotly (Sievert, 2020), igraph (Csardi and Nepsusz, 2006), kappalab (Grabisch et al., 2015), GameTheory (Cano−Berlanga, 2017) and lpSolveAPI (lp_solve et al., 2020). The first one is used for calculations with triangular distributions, the second one to plot interactive graphics, the third one to plot graphs, the next two are related to game-theoretic concepts and the last one to solve linear programming problems.

The functions incorporated in the package can be seen in Table 2.1. Note that for projects of more than 10 activities, functions `delay.pert` and `delay.stochastic.pert` will approximate the Shapley value through a sampling process. Table 2.2 describes the complete list of parameters used by the functions. Tables 2.3, 2.4 and 2.5 state which arguments use each function.

ProjectManagement allows the user to plot the activities on nodes graph of the Project (AON). Originally, in the PERT methodology, projects are represented by activities on arcs graphs (AOA). This is the representation we have used in this paper up to now. Both AON and AOA representations are widely used in the literature, each having some advantages over the other in particular circumstances. For automatically drawing the network of a project, the AON representation is more appropriate because it is computationally much more efficient. This is why we have incorporated it into the `dag.plot` function. This representation will be useful mainly for the user to check that he has entered the precedence matrices correctly, which are the ones that really characterize the project.

| Function | Description |
|---|---|
| `dag.plot` | Plots the AON graph of a project. |
| `delay.pert` | Calculates the delay cost of a deterministic project and allocates it among the activities. |
| `delay.stochastic.pert` | Calculates the delay cost of a stochastic projectand allocates it among the activities. |
| `early.time` | Calculates the earliest start time for each activity. |
| `last.time` | Calculates the latest completion time for each activity. |
| `levelling.resources` | Calculates the schedule of the project so that the consumption of resources is as uniform as possible. |
| `mce` | Calculates the costs per activity needed to accelerate the project. |
| `organize` | Relabels the activities of a project (if $i$ precedes $j$ then $i \leq j$). |
| `rebuild` | Builds a type 1 precedence matrix. |
| `resource.allocation` | Calculates the project schedule so that resource consumption does not exceed the maximum available per time period. |
| `schedule.pert` | Calculates the duration of a project and the schedule of each activity, and plots the schedule and the AON graph. |
| `stochastic.pert` | Calculates the average duration of a stochastic project, the criticality index of each activity, and the density functions of the duration of the project, early times and last times. |

Table 2.1: Summary of functions in ProjectManagement.

ProjectManagement also allows the user to choose from four different types of immediate precedences between the activities.

- Type 1: Finish to start (FS). If an activity $i \in N$ precedes type 1 to $j \in N$, then $j$ cannot start until activity $i$ has finished.
- Type 2: Start to start (SS). If an activity $i \in N$ precedes type 2 to $j \in N$, then $j$ cannot start until activity $i$ has started.
- Type 3: Finish to finish (FF). If an activity $i \in N$ precedes type 3 to $j \in N$, then $j$ cannot finish until activity $i$ has finished.
- Type 4: Start to finish (SF). If an activity $i \in N$ precedes type 4 to $j \in N$, then $j$ cannot finish until activity $i$ has started.

The relationships between the types of dependencies are as follows: Type 1 implies type 2, type 2 implies type 4, type 1 implies type 3 and finally type 3 implies type 4. Considering these relations, if one activity precedes another by more than one type, it is only necessary to indicate the one with the strongest character.

The user can indicate types 1 or 2 in the `"prec1and2"` parameter (see Table 2.2) using the values 1 or 2 respectively, and types 3 or 4 in `"prec3and4"` using 3 or 4 respectively. Note that cycles can not exist.

| Parameter | Description |
|---|---|
| `duration` | Vector with the expected duration for each activity. |
| `prec1and2` | Matrix indicating precedence type 1 or type 2 between the activities (Default=`matrix(0)`). |
| `prec3and4` | Matrix indicating precedence type 3 or type 4 between the activities (Default=`matrix(0)`). |
| `observed.duration` | Vector with the actual duration for each activity. |
| `delta` | Value indicating the maximum time that the project can take without delay (see equation 2.1). This value is only used with the default cost function. |
| `distribution` | Vector with the distribution function of the duration for each activity. It can be normal, triangular, exponential, uniform, beta, t-Student, F distribution, chi-squared, gamma, Weibull, binomial, Poisson, geometric, hypergeometric, and empirical. |
| `values` | Matrix with the arguments of the distribution function of the duration for each activity. By rows the activities, and by columns the arguments. |
| `percentile` | Value used to calculate the maximum time allowed for the duration of the project without delay. This value is only used if no `delta` value is assigned. |
| `compilations` | Number of simulations that the function uses for estimation (`Default=1000`). |
| `cost.funtion` | Delay cost function. If this value is not added, the package uses equation 2.1. |
| `early.times` | Vector with the early time for each activity. |
| `PRINT` | Logical parameter indicating if the schedule and the AON graph are depicted (`Default=TRUE`). |
| `plot.activities.times` | Vector of selected activities from which it is shown the distribution of their early and last times (`Default=NULL`). |
| `minimum.durations` | Vector with the minimum duration an activity can take even if the resources are increased. |
| `critical.activities` | Vector with the critical activities to represent them in a different color in the AON graph (`Default=NULL`). |
| `duration.project` | Value indicating the minimum time sought in the project (`Default=NULL`). |
| `activities.costs` | Vector indicating the cost of accelerating a unit of time the duration for each activity. |
| `resources` | Vector indicating the necessary resources for each activity per period of time. |
| `int` | Value indicating the duration of each period of time (Default=1). |
| `max.resources` | Value indicating the maximum number of resources that can be used in each period of time. |

Table 2.2: Summary of parameters in ProjectManagement.

| Function | duration | prec1and2 | prec3and4 | observed.duration | delta | distribution | values | percentile |
|---|---|---|---|---|---|---|---|---|
| dag.plot | | ✓ | ✓ | | | | | |
| delay.pert | ✓ | ✓ | ✓ | ✓ | ✓ | | | |
| delay.stochastic.pert | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| early.time | ✓ | ✓ | ✓ | | | | | |
| last.time | ✓ | ✓ | ✓ | | | | | |
| levelling.resources | ✓ | ✓ | ✓ | | | | | |
| mce | ✓ | ✓ | ✓ | | | | | |
| organize | | ✓ | ✓ | | | | | |
| resource.allocation | ✓ | ✓ | ✓ | | | | | |
| schedule.pert | ✓ | ✓ | ✓ | | | | | |
| stochastic.pert | | ✓ | ✓ | | | ✓ | ✓ | ✓ |

Table 2.3: Arguments used by each function in ProjectManagement.

| Function | cost.function | compilations | early.times | PRINT | plot.activities.times | minimum.durations |
|---|---|---|---|---|---|---|
| dag.plot | | | | | | |
| delay.pert | ✓ | | | | | |
| delay.stochastic.pert | ✓ | ✓ | | | | |
| early.time | | | | | | |
| last.time | | | ✓ | | | |
| levelling.resources | | | | | | |
| mce | | | | | | ✓ |
| organize | | | | | | |
| resource.allocation | | | | | | |
| schedule.pert | | | | ✓ | | |
| stochastic.pert | | ✓ | | | ✓ | |

Table 2.4: Arguments used by each function in ProjectManagement.

| Function | critical.activities | duration.project | activities.costs | resources | int | max.resources |
|---|---|---|---|---|---|---|
| dag.plot | ✓ | | | | | |
| delay.pert | | | | | | |
| delay.stochastic.pert | | | | | | |
| early.time | | | | | | |
| last.time | | | | | | |
| levelling.resources | | | | ✓ | ✓ | |
| mce | | ✓ | ✓ | | | |
| organize | | | | | | |
| resource.allocation | | | | ✓ | ✓ | ✓ |
| schedule.pert | | | | | | |
| stochastic.pert | | | | | | |

Table 2.5: Arguments used by each function in ProjectManagement.

## 2.4 Examples

ProjectManagement is available for download from CRAN. To use the package you will need to load it at the beginning of the session, usually by typing

```
> library("ProjectManagement")
```

Next we analyse the following deterministic project with 10 activities. Their durations and precedence relations are given in Table 2.6.

| $N$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Immediate precedence type 1 | - | - | - | 2 | 3 | 3 | 1, 4 | 2 | 5, 8 | 6 |
| Immediate precedence type 2 | - | - | - | - | 4 | - | - | - | - | - |
| Immediate precedence type 3 | - | - | - | - | - | 8 | - | - | - | - |
| Immediate precedence type 4 | - | - | - | - | - | - | 9 | - | - | - |
| Durations | 2 | 1.5 | 1 | 4.5 | 2 | 2.5 | 3 | 4 | 2 | 5 |

Table 2.6: Example of a deterministic project.

We start by introducing the data set characterizing the project. We use the function `dag.plot` for depicting its AON graph. Figure 2.2 shows it; the green blocks contain the activities and the precedences are represented by arrows. The blocks $S$ and $E$ are the source and sink nodes, respectively. Note that the precedences type 1 are arrows without label, precedences type 2 are labeled as SS, precedences type 3 as FF, and precedences type 4 as SF.

```
> prec1and2<-matrix(0,nrow=10,ncol=10)
> prec1and2[1,7]<-1; prec1and2[2,4]<-1; prec1and2[2,8]<-1;
> prec1and2[3,5]<-1; prec1and2[3,6]<-1; prec1and2[4,7]<-1;
> prec1and2[5,9]<-1; prec1and2[6,10]<-1; prec1and2[8,9]<-1;
> prec1and2[4,5]<-2

> prec3and4<-matrix(0,nrow=10,ncol=10)
> prec3and4[8,6]<-3; prec3and4[9,7]<-4

> dag.plot(prec1and2,prec3and4)
```

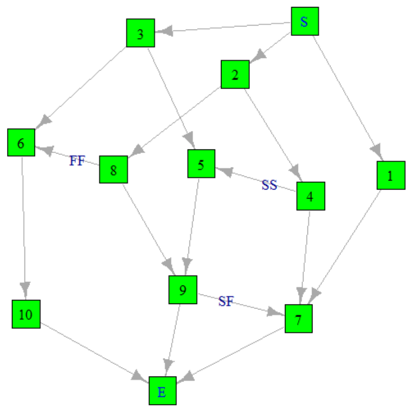


Figure 2.2: AON graph of the project. In an AON graph the activities are embodied in the nodes (squares) and the precedences of the various types, FS, SS, FF, SF, in the arcs (arrows).

Using `schedule.pert`, we obtain the project schedule, i.e. the minimum time needed to complete all activities and the early and last times. Also, we can plot the schedule. Let us see it:

```
> duration<-c(2,1.5,1,1.5,2,2.5,3,4,2,5)
> schedule.pert(duration,prec1and2,prec3and4)
'Total duration of the project'
[1] 10.5

[[2]]
```

| Activities | Duration | Earliest start | Latest start | Earliest completion |
|---|---|---|---|---|
| 1 | 2.0 | 0.0 | 5.5 | 2.0 |
| 2 | 1.5 | 0.0 | 0.0 | 1.5 |
| 3 | 1.0 | 0.0 | 2.0 | 1.0 |
| 4 | 1.5 | 1.5 | 6.0 | 3.0 |
| 5 | 2.0 | 1.5 | 6.5 | 3.5 |
| 6 | 2.5 | 2.0 | 3.0 | 5.5 |
| 7 | 3.0 | 3.0 | 7.5 | 6.0 |
| 8 | 4.0 | 1.5 | 1.5 | 5.5 |
| 9 | 2.0 | 5.5 | 8.5 | 7.5 |
| 10 | 5.0 | 5.5 | 5.5 | 10.5 |

```
Activities  Latest completion  Slack  Free Slack  Independent Slack
1                 7.5            5.5      1.0            0.0
2                 1.5            0.0      0.0            0.0
3                 3.0            2.0      0.0            0.0
4                 7.5            4.5      0.0            0.0
5                 8.5            5.0      2.0            0.0
6                 5.5            2.0      2.0            0.0
7                10.5            4.5      4.5            0.0
8                 5.5            0.0      0.0            0.0
9                10.5            3.0      3.0            0.0
10               10.5            0.0      0.0            0.0

[[3]]
```

In this output we can see the total duration (10.5 units) of the project as well as other relevant information for each activity. Figure 2.3 depicts the different times of each activity using a colour coding. If we click on the points on the graph, a label indicates which activity and time it belongs to. Also, if we double click on a section of the legend we can see the data related to it, as in Figure 2.4 with the last times for each activity (another double click restarts the graph). In the output, the plot depicted in Figure 2.3 is saved as an object on [[3]]; this allows the users to manipulate the plot according to their needs. Finally, Figure 2.5 shows the AON graph of the project where critical activities are represented in red.

Next, suppose we are interested in shortening the duration of the project. The `mce` function is used for this purpose. Let us use the function with the following input data: the minimum duration for each activity even if the resources are increased

$$\bar{x}^0 = (1, 1, 0.5, 1, 1, 2, 2, 3, 1, 3)$$

and the costs per unit time to shorten each activity

$$c = (1, 2, 1, 1, 3, 2, 1, 2, 3, 5)\,.$$

```
> minimum.durations<-c(1,1,0.5,1,1,2,2,3,1,3)
> activities.costs<-c(1,2,1,1,3,2,1,2,3,5)
```

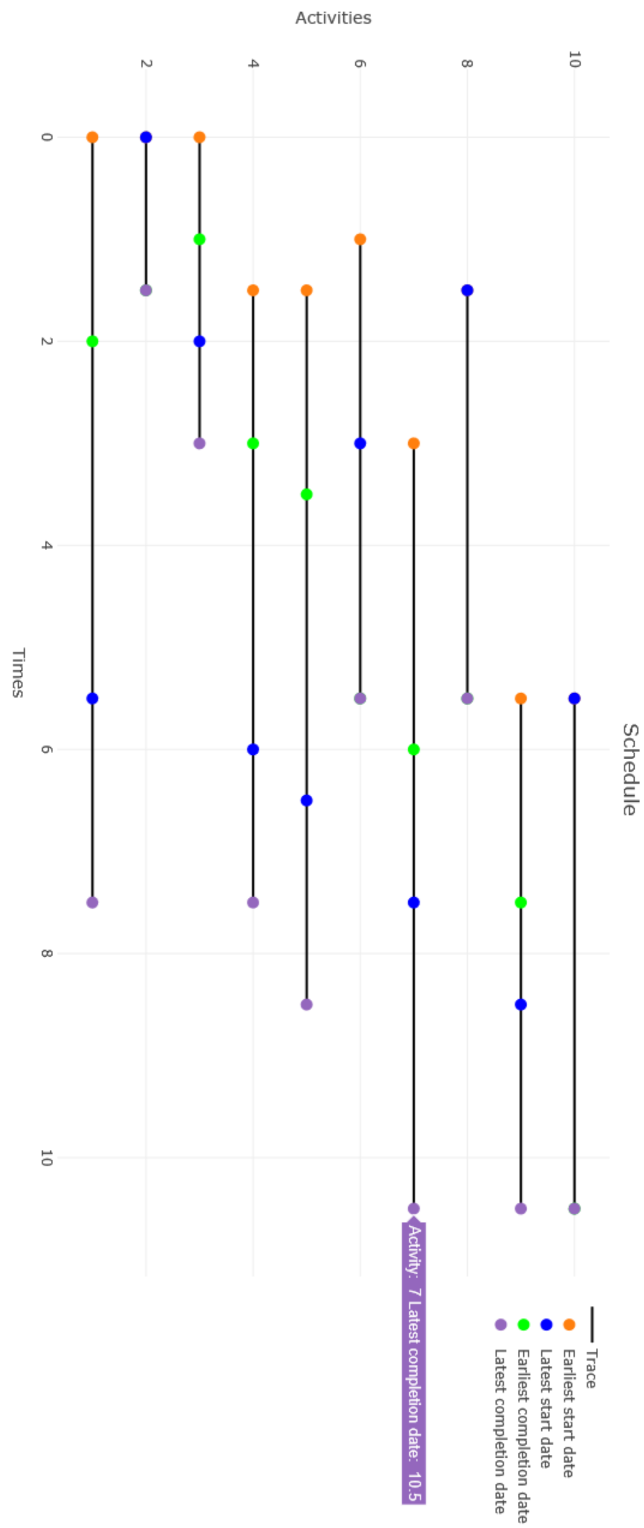


Figure 2.3: This figure shows an interactive graphic that displays the schedule of the project. If we move the mouse over the highlighted points of the segments, pop-up tags are generated with information about the activities.

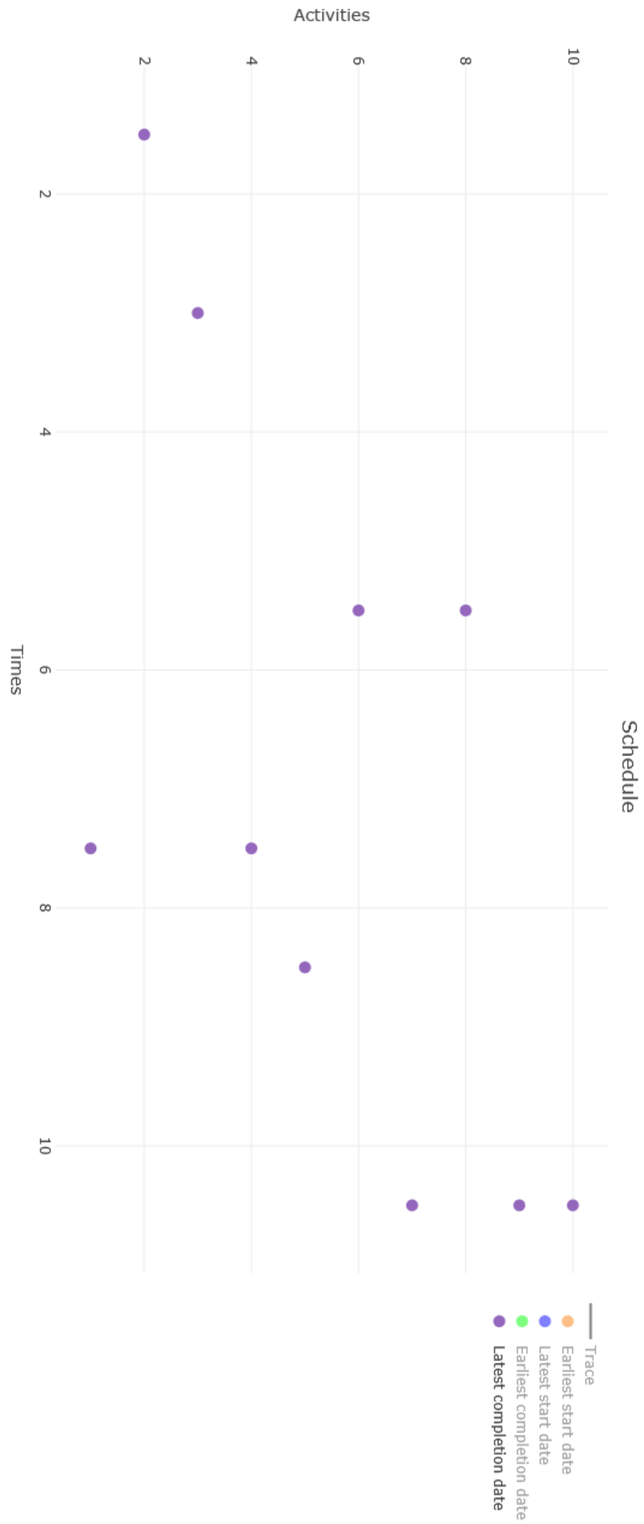


Figure 2.4: Latest completion times. This interactive graphic is the result of double clicking on the “Latest completion date” section of the legend.

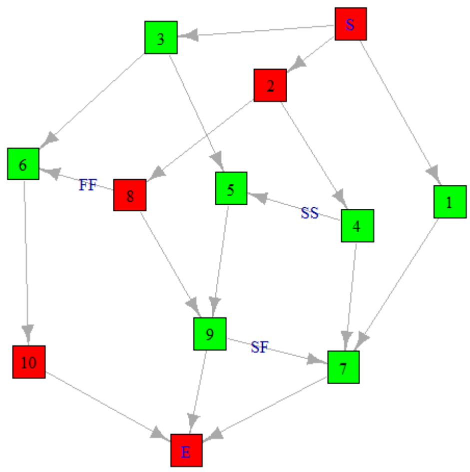


Figure 2.5: AON graph of the project. In an AON graph the activities are embodied in the nodes (squares) and the precedences of the various types, FS, SS, FF, SF, in the arcs (arrows). Nodes in red indicate critical activities.

```
> mce(duration,minimum.durations,prec1and2,prec3and4,
activities.costs,duration.project=NULL)
necessary negative increase
1: 0.5
Read 1 item
Project duration =
[1] 10.0  9.5  9.0  8.5  8.0  7.5  7.0

Estimated durations =               Costs per solution  =


 2.0  2.0  2.0  2.0  2.0  2.0  2.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 1.5  1.5  1.0  1.0  1.0  1.0  1.0     0.0  0.0  1.0  1.0  1.0  1.0  1.0
 1.0  1.0  1.0  1.0  1.0  1.0  1.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 1.5  1.5  1.5  1.5  1.5  1.5  1.5     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 2.0  2.0  2.0  2.0  2.0  2.0  2.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 2.5  2.5  2.5  2.5  2.5  2.5  2.5     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 3.0  3.0  3.0  3.0  3.0  3.0  3.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 3.5  3.0  3.0  3.0  3.0  3.0  3.0     1.0  2.0  2.0  2.0  2.0  2.0  2.0
 2.0  2.0  2.0  2.0  2.0  2.0  2.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 5.0  5.0  5.0  4.5  4.0  3.5  3.0     0.0  0.0  0.0  2.5  5.0  7.5   10
```

The parameter `duration.project=NULL` means that we do not indicate a minimum duration of the project, so the function asks us for a decrease of the duration of the project to obtain all possible solutions. We have considered it convenient a decrease of 0.5 units of time. Therefore, we have obtained that the project can reduce its minimum duration to 10, 9.5, 9, 8.5, 8, 7.5 and 7. For each possible duration of the project, we have the durations of each activity (duration per column and activity per row), as well as the cost needed to reduce their times to these durations.

Suppose now that to complete the project each activity needs the amount of resources

$$(6, 6, 6, 3, 4, 2, 1, 2, 3, 1)\,,$$

and we are interested in obtaining a new schedule with a uniform consumption of resources over time. To do this we use the function `levelling.resources` in such a way

```
> resources<-c(6,6,6,3,4,2,1,2,3,1)
> levelling.resources(duration,prec1and2,prec3and4,resources,int=0.5)
Earliest start times =
[1] 3.5 0.0 2.0 2.0 6.5 3.0 5.5 1.5 8.5 5.5
Resources  by period=
[1]  6  6  6  2 11 11  7 10 10 10 10  2  2  6  6  6  6  4  4  4  4
```

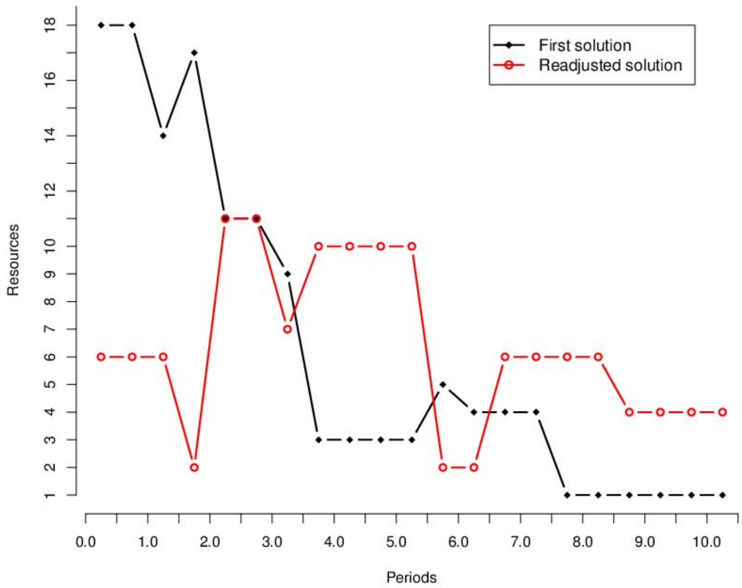


Figure 2.6: This graphic shows the resource consumption according to the initial scheduling (in black) and according to the scheduling after leveling (in red). The x-axis represents time and the y-axis represents resource consumption.

As we can see, the function returns the new earliest start times of the activities and the resources consumed in each period with the new schedule, where time periods start at 0 and end at 10.5 with an increase of 0.5 time units. Figure 2.6 represents the resources required in each period of time, before and after the readjustment.

To conclude with the analysis of resources, consider that the maximum amount of resources available in each period is 10. We use the `resource.allocation` function in this situation.

```
> max.resources<-10
> resource.allocation(duration,prec1and2,prec3and4,resources,
max.resources,int=0.5)

Project duration =
[1] 11
Earliest start times =
[1] 6.0 0.0 1.5 2.5 4.0 2.5 8.0 1.5 6.0 5.5
Resources  by period =
[1]  6  6  6  8  8  7  7  7  8  8  6  5 10 10 10 10  2  2  2  2  2  1
```

With the new restriction, the minimum duration of the project becomes 11 instead of 10. The output includes the new earliest start times for each activity and the consumption of resources by period (note that the last period is now 11).

Continuing the example, we now analyse the allocation of delays. The function `delay.pert` shows if there has been a delay in the project and, in that case, allocates it among the activities. Let us see it using the delay cost function

$$C\left(D\left(G,y\right)\right)=\begin{cases} 0 & \text{if } D\left(G,y\right)\leq 10.5, \\ D\left(G,y\right)-10.5 & \text{otherwise,} \end{cases}$$

and the (observed) actual durations

$$x=\left(2.5,1.5,2,2,2,6,4,6,3,5.5\right).$$

```
> observed.duration<-c(2.5,1.5,2,2,2,6,4,6,3,5.5)
> cost.function<-function(x){return(max(x-10.5,0))}

> delay.pert(duration,prec1and2,prec3and4,observed.duration,
delta=NULL,cost.function)

There has been a delay of =  3
```

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| The proportional rule | 0.150 | 0.000 | 0.300 | 0.150 | 0.000 |
| The truncated proportional rule | 0.158 | 0.000 | 0.316 | 0.158 | 0.000 |
| Shapley rule | 0.000 | 0.000 | 0.333 | 0.000 | 0.000 |

| | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|
| The proportional rule | 1.050 | 0.300 | 0.600 | 0.300 | 0.150 |
| The truncated proportional rule | 0.947 | 0.316 | 0.632 | 0.316 | 0.158 |
| Shapley rule | 1.083 | 0.000 | 1.083 | 0.000 | 0.500 |

The output shows that there is a delay in the project of 3 units. As there is a delay, we proceed to make the allocation using three rules: proportional, truncated proportional and Shapley. We can see the differences between the three rules, especially in activities 1, 4, 7 and 9. While the proportional and truncated proportional rules assign a positive payment, the Shapley rule does not assign costs to these activities. This is due to the fact that, although they fall behind the planned duration, they do not affect the overall delay of the project. Note that if the project has more than ten activities, `delay.pert` does not calculate the Shapley rule; instead, it asks the user if he wants to calculate an estimate of its value.

Let us now assume that we are in a stochastic context, with additional information on planned durations being random variables. Using the the package's function `stochastic.pert` with the following random variables to describe the duration of the activities

$$\begin{aligned} X^0 = (&t\,(1,2,3)\,, \exp{(2/3)}\,, t\,(1/2,5/4,5/4)\,, t\,(1/4,7/4,5/2)\,, t\,(1,2,3)\,,\\ &t\,(1,3/2,5)\,, t\,(1,1,7)\,, t\,(3,4,5)\,, t\,(1/2,5/2,3)\,, t\,(1,6,8))\,, \end{aligned}$$

where $t\,(a,b,c)$ denotes the triangular distribution with parameters $a$, $b$, and $c$, and $\exp{(\alpha)}$ denotes the exponential distribution with parameter $\alpha$, we can obtain relevant information about the project. Note that with the argument `plot.activities.times=c(7,8)` we indicate the activities for which we want to estimate the densities of their earliest and latest start and completion times; in this example we have requested only such densities for activities 7 and 8.

```
> values<-matrix(c(1,3,2,2/3,0,0,1/2,5/4,5/4,1/4,5/2,7/4,1,3,2,1,5,
3/2,1,7,1,3,5,4,1/2,3,5/2,1,8,6),nrow=10,ncol=3,byrow=T)
> distribution<-c("TRIANGLE","EXPONENTIAL",rep("TRIANGLE",8))
> stochastic.pert(prec1and2,prec3and4,distribution,values,
percentile=0.95,plot.activities.times=c(7,8))
```

```
Average duration of the project =  10.64242
Percentile duration of the project =  14.21999
```

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Criticality index | 0.6 | 88 | 11.4 | 2 | 0.1 | 11.3 | 2.6 | 86 | 4 | 93.4 |

In the output we can see the average duration of the project and the percentile duration of the project. The percentile duration of the project shows the value of such that the probability that the duration of the project is smaller than $d$ equals the variable percentile introduced by the user (see Table 2.2); in this case `percentile=0.95`. In addition, we obtain the criticality index by activity, that is, the proportion of times that an activity is critical. An activity is critical when it has zero slack. Figure 2.7 plots estimations of the density function of the project duration, the earliest start time and the latest completion time of activities 7 and 8.

We proceed now to the allocation of the delay cost in the stochastic model using the function `delay.stochastic.pert`. To be able to compare the results, we will use the same delay cost function as in the deterministic case. As expected, there are noticeable differences in the allocations between the two models, as the stochastic model makes use of more complex information.

```
> delay.stochastic.pert(prec1and2,prec3and4,distribution,values,
observed.duration,percentile=NULL,delta=NULL,cost.function)
Total delay of the stochastic project =  3
```

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Stochastic Shapley rule | 0.072 | -0.076 | 0.570 | 0.074 | 0.072 |
| Stochastic Shapley rule 2 | 0.006 | 0.034 | 0.520 | 0.013 | 0.007 |
| The proportional payment | 0.150 | 0.000 | 0.300 | 0.150 | 0.000 |
| The truncated proportional payment | 0.156 | 0.000 | 0.316 | 0.158 | 0.000 |

| | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|
| Stochastic Shapley rule | 1.665 | 0.083 | 0.119 | 0.073 | 0.348 |
| Stochastic Shapley rule 2 | 1.621 | 0.023 | 0.250 | 0.010 | 0.513 |
| The proportional payment | 1.050 | 0.300 | 0.600 | 0.300 | 0.150 |
| The truncated proportional payment | 0.947 | 0.316 | 0.632 | 0.316 | 0.158 |

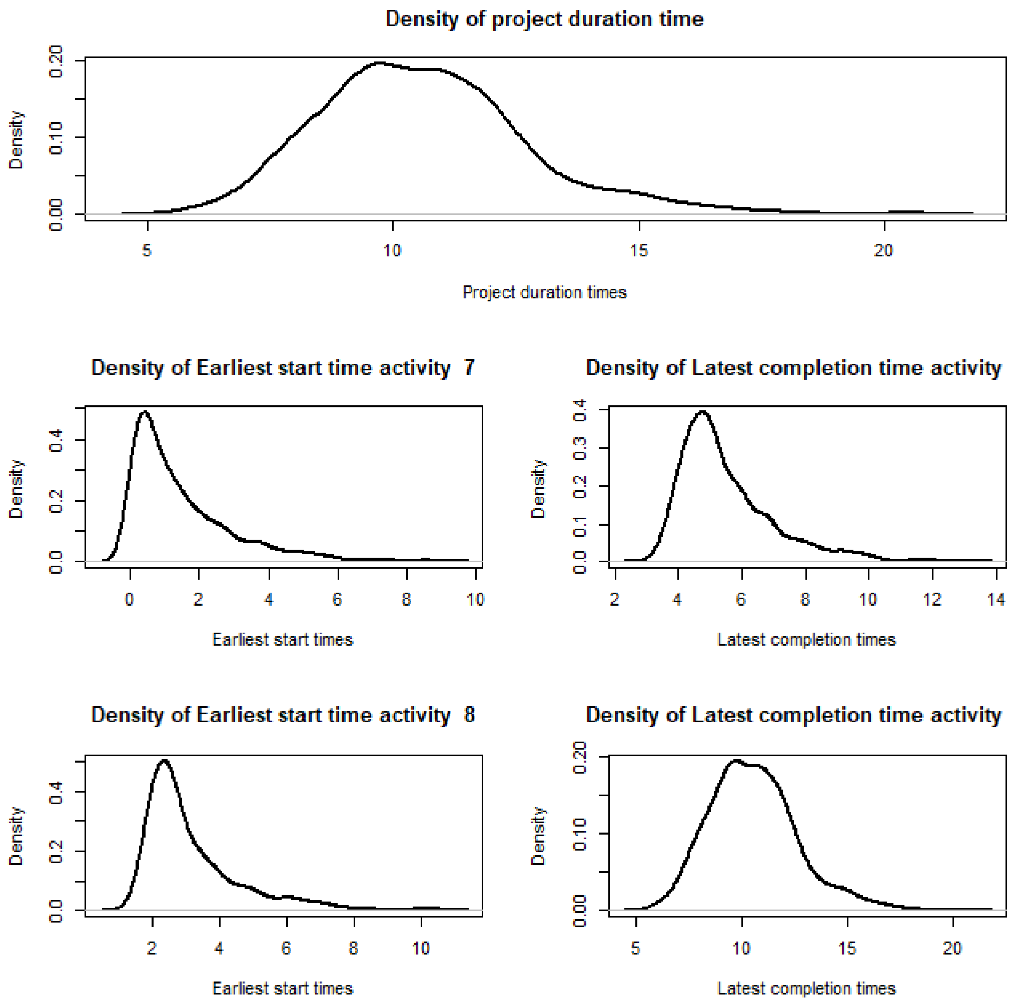


Figure 2.7: Density estimation of project duration time and earliest start and latest completion times for activities 7 and 8.

Finally, to illustrate the runtime of previously used functions, Table 2.7 shows the time (in seconds) needed to compute several problems. We have selected a variety of projects with 2, 4, 6, 8 and 10 activities, and we have run the different routines on a computer with Intel Core i5 − 7200U and 12 GB of RAM.

| Activities | 2 | 4 | 6 | 8 | 10 |
|---|---|---|---|---|---|
| `delay.pert` | 0.00 | 0.00 | 0.00 | 0.03 | 0.11 |
| `delay.stochastic.pert` | 0.06 | 0.44 | 1.68 | 6.23 | 30.58 |
| `early.time` | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| `last.time` | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| `levelling.resources` | 0.00 | 0.00 | 0.00 | 0.02 | 0.03 |
| `mce` | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 |
| `organize` | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| `resources.allocation` | 0.00 | 0.00 | 0.00 | 0.01 | 0.02 |
| `schedule.pert` | 0.08 | 0.11 | 0.13 | 0.11 | 0.12 |
| `stochastic.pert` | 0.02 | 0.03 | 0.05 | 0.04 | 0.04 |

Table 2.7: Runtime in seconds of ProjectManagemet functions.

# References

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.

Cano-Berlanga S (2017). GameTheory: Cooperative Game Theory. URL https://CRAN.R-project.org/package=GameTheory. R package version 2.7.

Carnell R (2019). triangle: Provides the Standard Distribution Functions for the Triangle Distribution. URL https://CRAN.R-project.org/package=triangle. R package version 0.12.

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 and Operations Research 36, 1726-1730.

Csardi G, Nepusz T (2006). The igraph software package for complex network research. InterJournal, Complex Systems:1695. URL http://igraph.org.

Gonçalves-Dosantos JC, García-Jurado I, Costa J (2020a). Sharing delay costs in stochastic scheduling problems with delays. 4OR - A Quarterly Journal of Operations Research, 18, 457-476.

Gonçalves-Dosantos JC, García-Jurado I, Costa J (2020b). ProjectManagement: Management of Deterministic and Stochastic Projects. URL https://CRAN.R-project.org/package=ProjectManagement. R package version 1.3.3.

Grabisch M, Kojadinovic I, Meyer P (2015). kappalab: Non-Additive Measure and Integral Manipulation Functions. URL https://CRAN.R-project.org/package=kappalab. R package version 0.4-7.

Gregoriou G, Kirytopoulos K, Kiriklidis C (2013). Project management educational software (promes). Computer Applications in Engineering Education, 21, 46-59.

Hall NG (2012). Project management: Recent developments and research opportunities. Journal of Systems Science and Systems Engineering, 21, 129-143.

Hegazy T (1999). Optimization of resource allocation and leveling using genetic algorithms. Journal of construction engineering and management, 125, 167-175.

Hillier FS, Lieberman GJ (2002). Introduction to Operations Research. McGraw-Hill.

Kelley D (2018). plan: Tools for Project Planning. URL https://CRAN.R-project.org/package=plan. R package version 0.4-3.

Kelley JE (1961). Critical-path planning and scheduling: Mathematical basis. Operations Research, 9, 296-320, 1961.

lp_solve, Konis k, Schwendinger F (2020). lpSolveAPI: R Interface to lp_solve Version 5.5.2.0. URL https://CRAN.R-project.org/package=lpSolveAPI. R package version 5.5.2.0-17.7.

Moretti S, Patrone F (2008). Transversality of the shapley value. Top, 16, 1-41, 2008.

Muñoz JC (2015). PlotPrjNetworks: Useful Networking Tools for Project Management. URL https: //CRAN.R-project.org/package=PlotPrjNetworks. R package version 1.0.0.

Özdamar L, Ulusoy G (1995). A survey on the resource-constrained project scheduling problem. IIE transactions, 27, 574-586, 1995.

Salas-Morera L, Arauzo-Azofra A, García-Hernández L, Palomo-Romero JM, César HM (2013). Ppcproject: An educational tool for software project management. Computers and Education, 69, 181-188.

Schmitz K, Mahapatra R, Nerur S (2019). User engagement in the era of hybrid agile methodology. IEEE Software, 36, 32-40.

Shapley LS (1953). A value for n-person games. Contributions to the Theory of Games, 2, 307-317.

Sievert C (2020). Interactive Web-Based Data Visualization with R, plotly, and shiny. Chapman and Hall/CRC. ISBN 9781138331457. https://plotly-r.com.

# Chapter 3

# On egalitarian values for cooperative games with a priori unions

JM Alonso-Meijide, J Costa, I García-Jurado, JC Gonçalves-Dosantos

TOP ISSN: 1863-8279 Volume: 28 Pages: 672-688

## 3.1 Introduction

Many economic problems deal with situations in which several agents cooperate to generate benefits or to reduce costs. Cooperative game theory studies procedures to allocate the resulting benefits (or costs) among the cooperating agents in those situations.

One of the most commonly used allocating procedures is the Shapley value, introduced in Shapley (1953) and analyzed more recently in Moretti and Patrone (2008) or in Algaba et al. (2019). Very often, however, agents cooperate on the basis of a kind of egalitarian principle according to which the benefits will be shared equitably. For instance, Selten (1972) indicates that egalitarian considerations explain in a successful way observed outcomes in experimental cooperative games.

In recent years, the game theoretical literature has dealt with several egalitarian solutions in cooperative games. For instance, van den Brink (2007) provides a comparison of the equal division value and the Shapley value, and Casajus and Hüttner (2014) compare those two solutions with the equal surplus division value (studied first in Driessen and Funaki, 1991). In van den Brink and Funaki (2009), Chun and Park (2012), van den Brink et al. (2016), Ferrières (2017) and Béal et al. (2019) several axiomatic characterizations of the equal division and equal surplus division values are provided. Ju et al. (2007) introduce and characterize the consensus value, a new solution that somewhat combines the Shapley value and the equal division rule. Dutta and Ray (1989) introduce the egalitarian solution for cooperative games, closely related to Lorenz dominance, that considers cooperating agents who believe in equality as a desirable social goal and negotiate accordingly; this solution was later characterized by Dutta (1990), Klijn et al. (2000) and Arín et al. (2003), and modified by Dietzenbacher et al. (2017).

Another stream of literature in cooperative game theory started in Owen (1977), where a variant of the Shapley value for games with a priori unions is introduced and characterized. In a game with a priori unions there exists a partition of the set of players, whose classes are called unions, that is interpreted as an a priori coalition structure that conditions the negotiation among the players and, consequently, modifies the fair outcome of the negotiation. There is a large literature concerning the Owen value and its applications; just to cite some recent papers, Lorenzo-Freire (2016) provides new axiomatic characterizations of the Owen value, Costa (2016) deals with an application in a cost allocation problem, and Saavedra-Nieves et al. (2018) propose a sampling procedure to approximate it. Not only the Shapley value but also other values have been modified for the case with a priori unions. For instance, Alonso-Meijide and Fiestras-Janeiro (2002) deal with the Banzhaf value for games with a priori unions, Casas-Méndez et al. (2003) introduce the $\tau$-value for games with a priori unions, Alonso-Meijide et al. (2011) study the Deegan-Packel index for simple games with a priori unions, and Hu et al. (2019) introduce an egalitarian efficient extension of the Aumann-Drèze value (Aumann and Drèze, 1974). Finally, the literature of games with a priori unions has developed in many other directions. For instance, Alonso-Meijide et al. (2014) analyze an extension of the Shapley value for games with a priori unions alternative to the Owen value, Vázquez-Brage et al. (1996) and van den Brink et al. (2015) introduce and study values for games with graph-restricted communication and a priori unions, and Hu (2019) deals with a weighted value for games with a priori unions.

In this paper we modify the equal division value and the equal surplus division value for games with a priori unions. In Section 2 we illustrate the interest of our study describing a cost allocation problem that arises in the installation of an elevator in an apartment building. In Section 3 we define and characterize the equal division rule for games with a priori unions. In Section 4 we introduce and characterize three alternative extensions of the equal surplus division rule for games with a priori unions. In Section 5 we include some final remarks.

## 3.2 An example

In this section we consider an example where the owners of apartments in a building have agreed to install an elevator and share the corresponding costs. This example is inspired by a problem analyzed in Crettez and Deloche (2019) from the point of view of French legislation. The French Law on Apartment Ownership of Buildings does not provide a precise method for sharing the cost of an improvement but indicates that the co-owners must pay "in proportion to the advantages" they will receive. In the case of elevators in France, Crettez and Deloche (2019) indicate that there is a de facto sharing method that they call the *elevator rule*. The elevator rule associates a parameter $\lambda_i = 1+(i-1)/2$ to each floor $i$ and distributes the total cost among the floors proportionally to those parameters; then the cost allocated to each floor is divided equally among its apartments. In their paper Crettez and Deloche study the elevator rule and other proposals in the spirit of the French legislation.

However, Crettez and Deloche (2019) explain that in other European countries the legislation is based on principles of egalitarian character. For example, in the Netherlands each of the owners of the apartments must "participate for an equal part in the debts and costs which are for account of all apartments owned pursuant to law or the internal arrangements, unless the internal arrangements provide for another proportion of participation."

The Spanish Horizontal Property Law 49/1960 (modified by the Act 8/2013) indicates that "to each apartment or local will be attributed a quota of participation in relation to the total of the value of the building (...). This quota will serve as a module to determine the participation in the burdens and benefits due to the community." These quotas generally depend on the surface area of each apartment but can take into account other aspects.

In a particular example, let us see how the Dutch and Spanish rules would share the costs of installing an elevator. Consider the following three-storey building with no apartments or offices on the ground floor: on the first floor there is a single apartment of 180 square meters, on the second floor there are two apartments, one of 100 and other one of 90 square meters, and on the third floor there are three apartments of 60 square meters each. The second floor has a slightly larger area because one of the two apartments on the floor has an additional gallery. Suppose now that the cost of installing the elevator is 120 (in thousands of euros), 50 of which correspond to the machine, 40 to the works to make the hollow of the elevator, and 30 to the works to be done on each floor to allow access to the elevator (10 in each of them). Table 3.1 below shows the distribution of costs for each of the apartments according to the Dutch and Spanish rules (the latter with quotas for each apartment given by its surface). Notice that both rules are based on egalitarian principles and can be interpreted as the equal division rule; the difference is that in the case of the Dutch rule the subjects that receive the equitable distribution are the apartments, whereas in the case of the Spanish rule the equitable distribution subjects are the quota units.[1] Notice that the same egalitarian spirit of these rules can be maintained despite changing the equitable distribution subjects. For instance, it would be natural to consider a kind of two-step equitable distribution subjects, where the subjects in the first step are the floors and the subjects in the second step are the apartments (in the case of the Dutch rule) or the quota units (in the case of the Spanish rule). This would result in the distribution of costs shown in Table 3.2 below. Observe that this variation arises from considering that the floors of the building naturally give rise to a structure of a priori unions in the sense of Owen (1977) and, thus, the convenience of extending the equal division value for games with a priori unions emerges spontaneously in this example. We do it formally in Section 3. Table 3.2 also displays the distribution proposed by the elevator rule that is by definition a two-step rule.

There are other possible variations of these Dutch and Spanish rules with and without the structure of a priori unions when using the equal surplus division value instead of the equal division value. Thus, the convenience of extending the equal surplus division value for games with a priori unions can also be motivated on the basis of this example. We do it in Section 4, where we also analyse in more depth how the equal surplus division value for games with a priori unions can be applied in the example we have discussed in this section.

| | Dutch rule | Spanish rule |
|---|---|---|
| 3rd floor | 20 20 20 | 13.0909 13.0909 13.0909 |
| 2nd floor | 20 20 | 21.8182 19.6364 |
| 1st floor | 20 | 39.2727 |

Table 3.1: Distribution according to the Dutch and Spanish rules.

| | Dutch rule | Spanish rule | elevator rule |
|---|---|---|---|
| 3rd | 13.3333 13.3333 13.3333 | 13.3333 13.3333 13.3333 | 17.7777 17.7777 17.7777 |
| 2nd | 20 20 | 21.0526 18.9474 | 20 20 |
| 1st | 40 | 40 | 26.6666 |

Table 3.2: Distribution according to the two-step Dutch and Spanish rules and to the elevator rule.

[1] In this example the quota units are the square meters of the apartments. For the approach we adopt to be meaningful, the quota unit numbers must be integers.

## 3.3 The equal division value for TU-games with a priori unions

In this section we extend the equal division value for TU-games to the more general setup of TU-games with a priori unions. To start with, we introduce the basic concepts and notations we use in this paper.

A transferable utility cooperative game (from now on a TU-game) is a pair $(N, v)$ where $N$ is a finite set of $n$ players, and $v$ is a map from $2^N$ to $\mathbb{R}$ with $v(\emptyset) = 0$, that is called the characteristic function of the game. In the sequel, $\mathcal{G}_N$ will denote the family of all TU-games with player set $N$ and $\mathcal{G}$ the family of all TU-games. A value for TU-games is a map $f$ that assigns to every TU-game $(N, v) \in \mathcal{G}$ a vector $f(N, v) = (f_i(N, v))_{i\in N} \in \mathbb{R}^N$ with $\sum_{i\in N} f_i(N, v) = v(N)$.[2]

As it was remarked in the introduction, sometimes agents cooperate on the basis of a kind of egalitarian principle according to which the benefits will be shared equitably. This gives rise to the equal division value $ED$ that distributes $v(N)$ equally among the players in $N$. Formally, the equal division value $ED$ is defined for every $(N, v) \in \mathcal{G}$ and for all $i \in N$ by

$$ED_i(N, v) = \frac{v(N)}{n}.$$

Now denote by $P(N)$ the set of all partitions of $N$. A TU-game with a priori unions is a triplet $(N, v, P)$ where $(N, v) \in \mathcal{G}$ and $P = \{P_1, \dots, P_m\} \in P(N)$. The set of TU-games with a priori unions and with player set $N$ will be denoted by $\mathcal{G}_N^U$, and the set of all TU-games with a priori unions by $\mathcal{G}^U$. A value for TU-games with a priori unions is a map $g$ that assigns to every $(N, v, P) \in \mathcal{G}^U$ a vector $g(N, v, P) = (g_i(N, v, P))_{i\in N} \in \mathbb{R}^N$ with $\sum_{i\in N} g_i(N, v, P) = v(N)$. The next definition provides the natural extension of the equal division value to TU-games with a priori unions.

**Definition 3.1.** The equal division value for TU-games with a priori unions $ED^U$ is defined by

$$ED_i^U(N, v, P) = \frac{v(N)}{mp_k}$$

for all $i \in N$ and all $(N, v, P) \in \mathcal{G}^U$ with $P = \{P_1, \dots, P_m\}$ and $i \in P_k$; $p_k$ denotes the cardinal of $P_k$.

Notice that the equal division value for TU-games with a priori unions has been used in the motivating example in Section 2 (see Table 3.2 and the corresponding

[2]Notice that we have included the efficiency in the definition of value. We could have considered it as one more property and then it would have appeared explicitly in the characterizations; nothing relevant would have changed in that case.

comments). Next we provide an axiomatic characterization of this value. We start giving some properties of a value $g$ for TU-games with a priori unions.

**Additivity (ADD).** A value $g$ for TU-games with a priori unions satisfies additivity if, for all $(N, v, P), (N, w, P) \in \mathcal{G}^U$, it holds that

$$g(N, v + w, P) = g(N, v, P) + g(N, w, P).$$

Take a TU-game $(N, v) \in \mathcal{G}^N$ and $i, j \in N$. We say that $i, j$ are indistinguishable in $v$ if $v(S \cup i) = v(S \cup j)$ for all $S \subseteq N \setminus \{i, j\}$.

**Symmetry within unions (SWU).** A value $g$ for TU-games with a priori unions satisfies symmetry within unions if, for all $(N, v, P) \in \mathcal{G}^U$, all $P_k \in P$, and all $i, j \in P_k$ indistinguishable in $v$, it holds that $g_i(N, v, P) = g_j(N, v, P)$.

Take $(N, v, P) \in \mathcal{G}^U$ with $P = \{P_1, \ldots, P_m\}$ and denote $M = \{1, \ldots, m\}$. The quotient game of $(N, v, P)$ is the TU-game $(M, v/P)$ where

$$(v/P)(R) = v(\cup_{r \in R} P_r) \quad \text{for all } R \subseteq M.$$

**Symmetry among unions (SAU).** A value $g$ for TU-games with a priori unions satisfies symmetry among unions if, for all $(N, v, P) \in \mathcal{G}^U$ and all $k, l \in M$ indistinguishable in $v/P$, it holds that $\sum_{i \in P_k} g_i(N, v, P) = \sum_{i \in P_l} g_i(N, v, P)$.

Take a TU-game $(N, v) \in \mathcal{G}^N$ and $i \in N$. We say that $i$ is a nullifying player in $v$ if $v(S \cup i) = 0$ for all $S \subseteq N$. In words, a player is nullifying when every coalition containing it receives zero according to the characteristic function.

**Nullifying player property (NPP).** A value $g$ for TU-games with a priori unions satisfies the nullifying player property if, for all $(N, v, P) \in \mathcal{G}^U$ and all $i \in N$ nullifying player in $v$, it holds that $g_i(N, v, P) = 0$.

An analogous to NPP above is used in van den Brink (2007) to characterize the equal division value for TU-games. In the next theorem, we extend van den Brink's result to TU-games with a priori unions.

**Theorem 3.1.** *$ED^U$ is the unique value for TU-games with a priori unions that satisfies ADD, SWU, SAU and NPP.*

*Proof.* It is immediate to check that $ED^U$ satisfies ADD, SWU, SAU and NPP. To prove the unicity, consider a value $g$ for TU-games with a priori unions that satisfies ADD, SWU, SAU and NPP. Fix $N$ and define for all $\alpha \in \mathbb{R}$ and all non-empty

$T \subseteq N$ the TU-game $(N, e_T^\alpha)$ given by $e_T^\alpha(S) = \alpha$ if $S = T$ and $e_T^\alpha(S) = 0$ if $S \neq T$. If $T = N$, since $g$ satisfies SWU and SAU, it is clear that $g_i(N, e_N^\alpha, P) = \frac{\alpha}{mp_k}$ for any $P = \{P_1, \ldots, P_m\}$ and all $i \in P_k \subseteq N$, because all players in $N$ are indistinguishable in $e_N^\alpha$ and all players in $M$ are indistinguishable in $e_N^\alpha/P$. If $T \subset N$ notice that all players in $N \setminus T$ are nullifying players in $e_T^\alpha$ and then, since $g$ satisfies NPP,

$$\sum_{i \in T} g_i(N, e_T^\alpha, P) = \sum_{i \in N} g_i(N, e_T^\alpha, P) = e_T^\alpha(N) = 0$$

for any $P$. Then, since $g$ satisfies SWU and SAU it is not difficult to check that $g(N, e_T^\alpha, P) = 0$. Finally, the additivity of $g$ and the fact that $v = \sum_{T \subseteq N} e_T^{v(T)}$ imply that

$$g_i(N, v, P) = \sum_{T \subseteq N} g_i(N, e_T^{v(T)}, P) = g_i(N, e_N^{v(N)}, P) = \frac{v(N)}{mp_k} = ED_i^U(N, v, P)$$

for any $P$ and all $i \in P_k \subseteq N$. $\square$

## 3.4 The equal surplus division value for TU-games with a priori unions

In this section we extend the equal surplus division value for TU-games to the more general setup of TU-games with a priori unions. To start with, remember that the equal surplus division value $ESD$ is defined for every $(N, v) \in \mathcal{G}$ and for all $i \in N$ by

$$ESD_i(N, v) = v(i) + \frac{v^0(N)}{n},$$

where $v^0(S) = v(S) - \sum_{i \in S} v(i)$ for all $S \subseteq N$. Notice that $ESD$ is a variant of $ED$ in which we first allocate $v(i)$ to each player $i \in N$, and then distribute $v^0(N)$ among the players using $ED$. $ESD$ is a reasonable alternative to $ED$ for situations where individual benefits and joint benefits are neatly separable. Let us illustrate this with the example of Section 2 (notice that it deals with costs instead of with benefits).

Consider again the three-storey building of Section 2. Assume now that the cost of the machine is 55 and that the owner of the third apartment of the third floor can get a discount of 5. Clearly, the cost of the machine is a joint cost, whereas the cost due to the works to be done on each floor should be paid by the owners of each floor. With respect to the costs of the hollow, assume that there is a fixed cost of 10 and an individual cost of 10 for the owners of the first floor that is incremented by 10 for the owners of the second floor and by an additional 10 for the owners of the third floor. According to this, the cost $c(i)$ in which each player is involved is:

- 55 (machine) + 10 (floor) + 40 (hollow) = 105, for the first and second players of the third floor,
- 50 (machine) + 10 (floor) + 40 (hollow) = 100, for the third player of the third floor,
- 55 (machine) + 10 (floor) + 30 (hollow) = 95, for the players of the second floor,
- 55 (machine) + 10 (floor) + 20 (hollow) = 85, for the players of the first floor.

Now we can compute the equal surplus division value for the game in which the players are the apartments and $c(N) = 120$ (this is what we call the ES-Dutch rule) and the equal surplus division value for the game in which the players are the quota units and $c(N) = 120$ (this is what we call the ES-Spanish rule). In the latter case, the discount achieved by the owner of the third apartment of the third floor is divided equally among its quota units, i.e., $c(i) = 105 - 5/60$ for each square meter $i$ in the third apartment of the third floor. Table 3.3 below displays the distributions of the cost among the apartments using both rules. Notice that these distributions are not satisfactory because they seem to penalize too much the apartments on the third floor, specially the ES-Spanish rule that even proposes that the apartment on the first floor is recompensed if the elevator is installed. The reason for this seems to be that the individual costs in this example actually belong to the floors instead of to the players; consequently it would be more reasonable to use a kind of two-step rule for the equal surplus division value analogous to the two-step rule for the equal division value introduced in Section 2. In other words, this example suggests that we should consider the structure of a priori unions given by the floors and distribute the costs using an extension of the equal surplus division value to TU-games with a priori unions.

| | Dutch rule | Spanish rule |
|---|---|---|
| 3rd floor | 27.5 27.5 22.5 | 613.6364 613.6364 608.6364 |
| 2nd floor | 17.5 17.5 | 22.7273 20.4545 |
| 1st floor | 7.5 | -1759.0910 |

Table 3.3: Distribution according to the ES-Dutch and ES-Spanish rules.

Next we propose three alternative ways for extending the equal surplus division value to TU-games with a priori unions. The first one divides the value of the grand coalition in the quotient game using the equal surplus division value and then divides the amount assigned to each union equally among its members. It maintains the spirit of $ESD$ but applies it to the quotient game in order to take into account the unions; then it divides equally within the unions.

**Definition 3.2.** The equal surplus division value (one) for TU-games with a priori unions $ESD1^U$ is defined by

$$ESD1_i^U(N,v,P) = \frac{(v/P)(k)}{p_k} + \frac{(v/P)^0(M)}{mp_k} = \frac{v(P_k)}{p_k} + \frac{v(N) - \sum_{l\in M} v(P_l)}{mp_k}$$

for all $i \in N$ and all $(N,v,P) \in \mathcal{G}^U$ with $P = \{P_1, \ldots, P_m\}$ and with $i \in P_k$.

The second extension divides again the value of the grand coalition in the quotient game using the equal surplus division value; then it distributes the amount $\frac{v(N)-\sum_{l\in M} v(P_l)}{m}$ equally among the players in each union, and the amount $v(P_k)$ giving $v(i)$ to each player $i \in P_k$ and dividing $v(P_k) - \sum_{j\in P_k} v(j)$ equally among the players in $P_k$. It maintains the spirit of $ESD$ and, in some sense, applies it twice: first to the quotient game and second to divide each $v(P_k)$ among its members.

**Definition 3.3.** The equal surplus division value (two) for TU-games with a priori unions $ESD2^U$ is defined by

$$ESD2_i^U(N,v,P) = v(i) + \frac{v(P_k) - \sum_{j\in P_k} v(j)}{p_k} + \frac{v(N) - \sum_{l\in M} v(P_l)}{mp_k}$$

for all $i \in N$ and all $(N,v,P) \in \mathcal{G}^U$ with $P = \{P_1, \ldots, P_m\}$ and with $i \in P_k$.

Finally, the third extension assigns $v(i)$ to each player $i$ and then divides $v^0(N)$ among the players using $ED^U$. It maintains the spirit of $ESD$ in the sense that it allocates $v(i)$ to each $i$; then it divides $v(N) - \sum_{j\in N} v(j)$ equally, first among the unions and then within the unions.

**Definition 3.4.** The equal surplus division value (three) for TU-games with a priori unions $ESD3^U$ is defined by

$$ESD3_i^U(N,v,P) = v(i) + ED^U(N,v^0,P) = v(i) + \frac{v(N) - \sum_{j\in N} v(j)}{mp_k}$$

for all $i \in N$ and all $(N,v,P) \in \mathcal{G}^U$ with $P = \{P_1, \ldots, P_m\}$ and with $i \in P_k$.

Now we can compute the equal surplus division values one, two and three for the game with a priori unions in which the players are the apartments, the unions are the floors and $c(N) = 120$ (they are what we call the $ESD1^U$, $ESD2^U$ and $ESD3^U$-Dutch rules) and the equal surplus division values one, two and three for the game with a priori unions in which the players are the quota units, the unions are the floors and $c(N) = 120$ (they are what we call the $ESD1^U$, $ESD2^U$ and $ESD3^U$-Spanish rules). Tables 3.4, 3.5 and 3.6 below display the distributions of the cost among the apartments using these rules. The results in Tables 3.4 and 3.5 seem to be more reasonable than those in Table 3.3; notice that they slightly penalize the higher floors in comparison with the results in Table 3.2 (except for the elevator rule). The results in Table 3.6 are not satisfactory since they penalize too much the apartments on the third floor and allocate a negative cost to the apartment on the first floor; thus, Table 3.6 shows that $ESD3^U$ is not appropriate for this example. The reason for this can be that $ESD3^U$ does not satisfy the quotient game property (QGP).[3] We believe that QGP is especially relevant in this example because, as we have already mentioned, the individual costs here belong to the floors (the unions) more than to the players.

| | Dutch rule | Spanish rule |
|---|---|---|
| 3rd floor | 15.5556 15.5556 15.5556 | 15.5556 15.5556 15.5556 |
| 2nd floor | 20.8333 20.8333 | 21.9298 19.7368 |
| 1st floor | 31.6667 | 31.6667 |

Table 3.4: Distribution according to $ESD1^U$.

| | Dutch rule | Spanish rule |
|---|---|---|
| 3rd floor | 17.2222 17.2222 12.2222 | 17.2222 17.2222 12.2222 |
| 2nd floor | 20.8333 20.8333 | 21.9298 19.7368 |
| 1st floor | 31.6667 | 31.6667 |

Table 3.5: Distribution according to $ESD2^U$.

| | Dutch rule | Spanish rule |
|---|---|---|
| 3rd floor | 53.3333 53.3333 48.3333 | 508.3333 508.3333 503.3333 |
| 2nd floor | 17.5 17.5 | 355.2632 319.7368 |
| 1st floor | -70 | -2075 |

Table 3.6: Distribution according to $ESD3^U$.

[3] A value $g$ for TU-games with a priori unions satisfies the quotient game property if, for all $(N, v, P) \in \mathcal{G}^U$ with $P = \{P_1, \ldots, P_m\}$ and for its quotient game $(M, v/P)$, it holds that $\sum_{i \in P_k} g_i(N, v, P) = g_k(M, v/P, P^m)$ for all $P_k \in P$, where $P^m = \{\{1\}, \{2\}, \ldots, \{m\}\}$.

In the remainder of this section we study $ESD1^U$, $ESD2^U$ and $ESD3^U$ from the point of view of their properties; in particular, we provide axiomatic characterizations of these values. We start by introducing new properties of a value $g$ for TU-games with a priori unions. Take $(N, v) \in \mathcal{G}$ and $i \in N$. We say that $i$ is a dummifying player in $v$ if $v(S \cup i) = \sum_{j \in S \cup i} v(j)$ for all $S \subseteq N$. In words, a player is dummifying when every coalition containing it is inessential according to the characteristic function, in the sense that its value equals the sum of the individual values of its members. Take now a TU-game with a priori unions $(N, v, P) \in \mathcal{G}^U$ where $P = \{P_1, \ldots, P_m\}$. We say that $P_k$ is a dummifying union in $(v, P)$ if $k$ is a dummifying player in $v/P$. Dummifying players and dummifying unions should play a relevant role in the characterizations of $ESD1^U$, $ESD2^U$ and $ESD3^U$ since a property on dummifying players is used in Casajus and Hüttner (2014) for characterizing $ESD$. In fact they use the following property (for $\mathcal{G}$ instead of $\mathcal{G}^U$).

**Dummifying player property (DPP).** A value $g$ for TU-games with a priori unions satisfies the dummifying player property if, for all $(N, v, P) \in \mathcal{G}^U$ and all $i \in N$ dummifying player in $v$, it holds that $g_i(N, v, P) = v(i)$.

Notice that $ESD3^U$ satisfies DPP, but neither $ESD1^U$ nor $ESD2^U$ satisfy it. In the search of properties that $ESD1^U$ or $ESD2^U$ might satisfy, we propose the following variations of DPP and NPP.

**Dummifying union/player property (DUPP).** A value $g$ for TU-games with a priori unions satisfies the dummifying union/player property if, for all $(N, v, P) \in \mathcal{G}^U$ and all $P_k \in P$ dummifying union in $(v, P)$ with $i \in P_k$ being a dummifying player in $v_{P_k}$,[4] it holds that $g_i(N, v, P) = v(i)$.

**Dummifying union/nullifying player property (DUNPP).** A value $g$ for TU-games with a priori unions satisfies the dummifying union/nullifying player property if, for all $(N, v, P) \in \mathcal{G}^U$ and all $P_k \in P$ dummifying union in $(v, P)$ with $i \in P_k$ being a nullifying player in $v_{P_k}$, it holds that $g_i(N, v, P) = 0$.

Now we give parallel characterizations of the three extensions of $ESD$ using the properties we have introduced above.

**Theorem 3.2.** *$ESD1^U$ is the unique value for TU-games with a priori unions that satisfies ADD, SWU, SAU and DUNPP.*

*Proof.* It is immediate to check that $ESD1^U$ satisfies ADD, SWU, SAU and DUNPP. To prove the unicity, consider a value $g$ for TU-games with a priori

[4] $v_{P_k}$ denotes the characteristic function of the TU-game $(P_k, v_{P_k})$, where $v_{P_k}(S) = v(S)$ for all $S \subseteq P_k$.

unions that satisfies ADD, SWU, SAU and DUNPP. Take $(N, v, P) \in \mathcal{G}^U$ with $P = \{P_1, \dots, P_m\}$ and define the TU-game $(N, v^1)$ given by

$$v^1(S) = \sum_{P_l \subseteq S} v(P_l) = \sum_{l=1}^{m} v^{P_l}(S)$$

for all $S \subseteq N$, where $v^{P_l}(S) = v(P_l)$ if $P_l \subseteq S$ and $v^{P_l}(S) = 0$ otherwise.

Take $P_k \in P$. Since $g$ is a value for TU-games with a priori unions, then

$$\sum_{i \in N} g_i(N, v^{P_k}, P) = v^{P_k}(N) = v(P_k).$$

All unions $P_l \in P$ are dummifying unions in $(v^{P_k}, P)$ and all players $i \in P_l$, with $l \neq k$, are nullifying players in $(v^{P_k})_{P_l}$. By DUNPP, $g_i(N, v^{P_k}, P) = 0$ for all $i \notin P_k$. And since all players in $P_k$ are indistinguishable in $v^{P_k}$, then SWU implies that, for all $i \in P_k$, $g_i(N, v^{P_k}, P) = \frac{v(P_k)}{p_k}$. Using the additivity of $g$, for all $i \in P_k$,

$$g_i(N, v^1, P) = \frac{v(P_k)}{p_k}. \tag{3.1}$$

Define now $v^2 = v - v^1$ and, for all $\alpha \in \mathbb{R}$ and all non-empty $T \subseteq N$, $e_T^\alpha$ by $e_T^\alpha(S) = \alpha$ if $S = T$ and $e_T^\alpha(S) = 0$ if $S \neq T$. It is clear that $v^2 = \sum_{T \subseteq N} e_T^{v^2(T)}$. If $T = N$, since all players in $N$ are indistinguishable in $e_N^{v^2(N)}$ and all players in $M$ are indistinguishable in $e_N^{v^2(N)}/P$, SWU and SAU imply that, for all $i \in P_k$,

$$g_i(N, e_N^{v^2(N)}, P) = \frac{v^2(N)}{mp_k} = \frac{v(N) - \sum_{l \in M} v(P_l)}{mp_k}.$$

If $T \subset N$, consider two cases:

- Take $T = \cup_{l \in L} P_l$, with $\emptyset \subset L \subset M$. For all $P_u \in P$, if $T \neq P_u$ then $e_T^{v^2(T)}(P_u) = 0$ and if $T = P_u$ then $e_T^{v^2(T)}(P_u) = v^2(P_u) = 0$. Hence, it is easy to see that all the unions in $M \setminus L$ are dummifying unions in $(e_T^{v^2(T)}, P)$. Also, since all players in $N \setminus T$ are nullifying players in $e_T^{v^2(T)}$, DUNPP implies that $g_i(N, e_T^{v^2(T)}, P) = 0$ for all $i \notin T$. Notice that since all unions in $L$ are indistinguishable in $e_T^{v^2(T)}$, then by SAU $\sum_{i \in P_k} g_i(N, e_T^{v^2(T)}, P) = \sum_{i \in P_l} g_i(N, e_T^{v^2(T)}, P)$ for all $k, l \in L$; notice also that since

$$\sum_{i \in T} g_i(N, e_T^{v^2(T)}, P) = \sum_{i \in N} g_i(N, e_T^{v^2(T)}, P) = e_T^{v^2(T)}(N) = 0$$

  then $\sum_{i \in P_k} g_i(N, e_T^{v^2(T)}, P) = 0$ for all $k \in L$. To conclude, SWU implies that $g_i(N, e_T^{v^2(T)}, P) = 0$ for all $i \in P_k$, with $k \in L$, and therefore for all $i \in N$.

- For any other $T \subset N$ that is not in the previous case, the quotient game $(M, e_T^{v^2(T)}/P)$ satisfies that $(e_T^{v^2(T)}/P)(R) = 0$ for all $R \subseteq M$ and, thus, all the unions in $P$ are indistinguishable and dummifying unions in $(e_T^{v^2(T)}, P)$. If $i \notin T$, then $i$ is a nullifying player in $e_T^{v^2(T)}$ and DUNPP implies that $g_i(N, e_T^{v^2(T)}, P) = 0$. Analogously as in the previous case, SAU and SWU imply that $g_i(N, e_T^{v^2(T)}, P) = 0$ for all $i \in T$.

Now ADD implies that, for all $i \in P_k$ with $P_k \in P$,

$$g_i(N, v^2, P) = \sum_{T \subseteq N} g_i(N, e_T^{v^2(T)}, P) = \frac{v^2(N)}{mp_k}. \tag{3.2}$$

Finally, from (3.1), (3.2), ADD and $v = v^1 + v^2$ it is clear that

$$g(N, v, P) = ESD1^U(N, v, P).$$

□

**Theorem 3.3.** *$ESD2^U$ is the unique value for TU-games with a priori unions that satisfies ADD, SWU, SAU and DUPP.*

*Proof.* It is immediate to check that $ESD2^U$ satisfies ADD, SWU, SAU and DUPP. To prove the unicity, consider a value $g$ for TU-games with a priori unions that satisfies ADD, SWU, SAU and DUPP. Take $(N, v, P) \in \mathcal{G}^U$ with $P = \{P_1, \ldots, P_m\}$ and define $v^a$, $v^{01}$ and $v^{02}$ by:

- $v^a(S) = \sum_{i \in S} v(i)$,
- $v^{01}(S) = \sum_{P_l \subseteq S} v^0(P_l) = \sum_{l=1}^{m} v^{0P_l}(\mathrm{S})$,
- $v^{02}(S) = v^0(S) - \sum_{P_l \subseteq S} v^0(P_l)$,

for all $S \subseteq N$, where $v^{0P_l}(S) = v^0(P_l)$ if $P_l \subseteq S$ and $v^{0P_l}(S) = 0$ otherwise.

Since all unions are dummifying in $(v^a, P)$ and all players are dummifying in $v^a$, then DUPP implies that, for all $i \in N$,

$$g_i(N, v^a, P) = v^a(i) = v(i). \tag{3.3}$$

Take $P_k \in P$. Since $g$ is a value for TU-games with a priori unions, then

$$\sum_{i \in N} g_i(N, v^{0P_k}, P) = v^{0P_k}(N) = v^0(P_k).$$

All unions $P_l \in P$ are dummifying unions in $(v^{0P_k}, P)$ and all players $i \in P_l$, with $l \neq k$, are dummifying players in $(v^{0P_k})_{P_l}$. By DUPP, $g_i(N, v^{0P_k}, P) = v^{0P_k}(i) = 0$ for all $i \notin P_k$. And since all players in $P_k$ are indistinguishable in $v^{0P_k}$, then SWU implies that, for all $i \in P_k$, $g_i(N, v^{0P_k}, P) = \frac{v^0(P_k)}{p_k}$. Using ADD, for all $i \in P_k$,

$$g_i(N, v^{01}, P) = \frac{v^0(P_k)}{p_k}. \tag{3.4}$$

Take now into account that $v^{02} = \sum_{T \subseteq N} e_T^{v^{02}(T)}$. If $T = N$, since all players in $N$ are indistinguishable in $e_N^{v^{02}(N)}$ and all players in $M$ are indistinguishable in $e_N^{v^{02}(N)}/P$, SWU and SAU imply that, for all $i \in P_k$,

$$g_i(N, e_N^{v^{02}(N)}, P) = \frac{v^{02}(N)}{mp_k}.$$

If $T \subset N$, consider two cases:

- Take $T = \cup_{l \in L} P_l$, with $\emptyset \subset L \subset M$. Since $e_T^{v^{02}(T)}(P_u) = 0$ for all $P_u \in P$ and $(e_T^{v^{02}(T)}/P)(R) = 0$ for all $R \subseteq M$ with $R \cap (M \setminus L) \neq \emptyset$, all the unions in $M \setminus L$ are dummifying unions in $(e_T^{v^{02}(T)}, P)$. Also, since all players in $N \setminus T$ are dummifying players in $e_T^{v^{02}(T)}$, DUPP implies that $g_i(N, e_T^{v^{02}(T)}, P) = e_T^{v^{02}(T)}(i) = 0$ for all $i \notin T$. Notice that since all unions in $L$ are indistinguishable in $e_T^{v^{02}(T)}$, then by SAU $\sum_{i \in P_k} g_i(N, e_T^{v^{02}(T)}, P) = \sum_{i \in P_l} g_i(N, e_T^{v^{02}(T)}, P)$ for all $k, l \in L$, and notice that since

$$\sum_{i \in T} g_i(N, e_T^{v^{02}(T)}, P) = \sum_{i \in N} g_i(N, e_T^{v^{02}(T)}, P) = e_T^{v^{02}(T)}(N) = 0$$

  then $\sum_{i \in P_k} g_i(N, e_T^{v^{02}(T)}, P) = 0$ for all $k \in L$. Hence, SWU implies that $g_i(N, e_T^{v^{02}(T)}, P) = 0$ for all $i \in T$.

- For any other $T \subset N$ that is not in the previous case, the quotient game $(M, e_T^{v^{02}(T)}/P)$ satisfies that $(e_T^{v^{02}(T)}/P)(R) = 0$ for all $R \subseteq M$ and, thus, all the unions in $P$ are indistinguishable and dummifying unions in $(e_T^{v^{02}(T)}, P)$. If $i \notin T$, then $i$ is a dummifying player in $e_T^{v^{02}(T)}$ and DUPP implies that $g_i(N, e_T^{v^{02}(T)}, P) = e_T^{v^{02}(T)}(i) = 0$. Analogously as in the previous case, SAU and SWU imply that $g_i(N, e_T^{v^{02}(T)}, P) = 0$ for all $i \in T$.

Now ADD implies that, for all $i \in P_k$ with $P_k \in P$,

$$g_i(N, v^{02}, P) = \sum_{T \subseteq N} g_i(N, e_T^{v^{02}(T)}, P) = \frac{v^{02}(N)}{mp_k}. \tag{3.5}$$

Finally, from (3.3), (3.4), (3.5), ADD and $v = v^a + v^{01} + v^{02}$ it is clear that

$$g(N, v, P) = ESD2^U(N, v, P).$$

□

Now we provide a characterization of $ESD3^U$. In order to do it we introduce a new property that is a weaker version of SAU.

**Weak symmetry among unions (WSAU).** A value $g$ for TU-games with a priori unions satisfies weak symmetry among unions if, for all $(N, v, P) \in \mathcal{G}^U$ with $v(j) = 0$ for all $j \in N$, and for all $k, l \in M$ indistinguishable in $v/P$, it holds that $\sum_{i \in P_k} g_i(N, v, P) = \sum_{i \in P_l} g_i(N, v, P)$.

**Theorem 3.4.** *$ESD3^U$ is the unique value for TU-games with a priori unions that satisfies ADD, SWU, WSAU and DPP.*

*Proof.* It is immediate to check that $ESD3^U$ satisfies ADD, SWU, WSAU and DPP. To prove the unicity, consider a value $g$ for TU-games with a priori unions that satisfies ADD, SWU, WSAU and DPP. Take now $(N, v, P) \in \mathcal{G}^U$ and $i \in P_k$ with $P_k \in P$, and define $v^a = v - v^0$. ADD implies that

$$g_i(N, v, P) = g_i(N, v^a, P) + g_i(N, v^0, P). \tag{3.6}$$

Since all players are dummifying in $v^a$, then DPP implies that

$$g_i(N, v^a, P) = v^a(i) = v(i). \tag{3.7}$$

Now, using for $(N, v^0)$ analogous arguments as those used in the proof of Theorem 3.1, it is clear that ADD, SWU, WSAU and DPP imply that

$$g_i(N, v^0, P) = ED_i(N, v^0, P). \tag{3.8}$$

Finally, from (3.6), (3.7) and (3.8) it is clear that

$$g(N, v, P) = ESD3^U(N, v, P).$$

□

It is immediate to prove that $ESD3^U$ does not satisfy SAU. Since WSAU is a weaker version of SAU, and $ESD3^U$ is characterized with ADD, SWU, WSAU and DPP, we conclude that there does not exist a value for TU-games with a priori unions satisfying ADD, SWU, SAU and DPP.

To conclude, we indicate that the properties in the theorems of this paper are independent. We prove this in a separate appendix.

# References

Algaba E, Fragnelli V, Sánchez-Soriano J (2019). Handbook of the Shapley value. CRC Press.

Alonso-Meijide JM, Casas-Méndez B, Fiestras-Janeiro G, Holler MJ (2011). The Deegan-Packel index for simple games with a priori unions. Quality & Quantity 45, 425-439.

Alonso-Meijide JM, Casas-Méndez B, González-Rueda A, Lorenzo-Freire S (2014). Axiomatic of the Shapley value of a game with a priori unions. Top 22, 749-770.

Alonso-Meijide JM, Fiestras-Janeiro G (2002). Modification of the Banzhaf value for games with a coalition structure. Annals of Operations Research 109, 213-227.

Arín J, Kuipers J, Vermeulen D (2003). Some characterizations of egalitarian solutions on classes of TU-games. Mathematical Social Sciences 46, 327-345.

Aumann RJ, Drèze J (1974). Cooperative games with coalition structures. International Journal of Game Theory 3, 217-237.

Béal S, Rémila E, Solal P (2019). Coalitional desirability and the equal division value. Theory and Decision 86, 95-106.

Casajus A, Hüttner F (2014). Null, nullifying, or dummifying players: The difference between the Shapley value, the equal division value, and the equal surplus division value. Economics Letters 122, 167-169.

Casas-Méndez B, García-Jurado I, van den Nouweland A, Vázquez-Brage M (2003). An extension of the $\tau$-value to games with coalition structures. European Journal of Operational Research 148, 494-513.

Chun Y, Park B (2012). Population solidarity, population fair-ranking and the egalitarian value. International Journal of Game Theory 41, 255-270.

Costa J (2016). A polynomial expression of the Owen value in the maintenance cost game. Optimization 65, 797-809.

Crettez B, Deloche R (2019). A law-and-economics perspective on cost-sharing rules for a condo elevator. To appear in Review of Law & Economics. doi: 10.1515/rle-2016-0001.

Dietzenbacher B, Borm P, Hendrickx R (2017). The procedural egalitarian solution. Games and Economic Behavior 106, 179-187.

Driessen TSH, Funaki Y (1991). Coincidence of and collinearity between game theoretic solutions. OR Spectrum 13, 15-30.

Dutta B (1990). The egalitarian solution and reduced game properties in convex games. International Journal of Game Theory 19, 153-169.

Dutta B, Ray D (1989). A concept of egalitarianism under participation constraints. Econometrica 57, 615-635.

Ferrières S (2017). Nullified equal loss property and equal division values. Theory and Decision 83, 385-406.

Hu XF (2019). The weighted Shapley-egalitarian value for cooperative games with a coalition structure. To appear in Top. https://doi.org/10.1007/s11750-019-00530-4.

Hu XF, Xu GJ, Li DF (2019). The egalitarian efficient extension of the Aumann-Drèze value. Journal of Optimization Theory and Applications 181, 1033-1052.

Ju Y, Borm P, Ruys P (2007). The consensus value: A new solution concept for cooperative games. Social Choice and Welfare 28, 685-703.

Klijn F, Slikker M, Tijs S, Zarzuelo J (2000). The egalitarian solution for convex games: some characterizations. Mathematical Social Sciences 40, 111-121.

Lorenzo-Freire S (2016). On new characterizations of the Owen value. Operations Research Letters 44, 491-494.

Moretti S, Patrone F (2008). Transversality of the Shapley value. Top 16, 1-41.

Owen G (1977) Values of games with a priori unions. In: Mathematical Economics and Game Theory (R Henn, O Moeschlin, eds.), Springer, 76-88.

Saavedra-Nieves A, García-Jurado I, Fiestras-Janeiro G (2018). Estimation of the Owen value based on sampling. In: The Mathematics of the Uncertain: A Tribute to Pedro Gil (E Gil, E Gil, J Gil, MA Gil, eds.), Springer, 347-356.

Selten R (1972). Equal share analysis of characteristic function experiments. In: Contributions to Experimental Economics III. (Sauermann H, ed.), Mohr Siebeck, 130-165.

Shapley LS (1953). A value for n-person games. In: Contributions to the Theory of Games II (HW Kuhn, AW Tucker, eds.), Princeton University Press, 307-317.

van den Brink R (2007). Null or nullifying players: the difference between the Shapley value and equal division solutions. Journal of Economic Theory 136, 767-775.

van den Brink R, Funaki Y (2009) Axiomatizations of a class of equal surplus sharing solutions for TU-games. Theory and Decision 67, 303-340.

van den Brink R, Chun Y, Funaki Y, Park B (2016). Consistency, population solidarity, and egalitarian solutions for TU-games. Theory and Decision 81, 427-

447.

van den Brink R, van der Laan G, Moes N (2015). Values for transferable utility games with coalition and graph structure. Top 23, 77-99.

Vázquez-Brage M, García-Jurado I, Carreras F (1996). The Owen value applied to games with graph-restricted communication. Games and Economic Behavior 12, 42-53.

# Appendix

a) Independence of the properties of Theorem 3.1:

- $\varphi_i = \frac{v(P_k)}{p_k} + \frac{v(N)-\sum_{l\in M} v(P_l)}{mp_k}$ satisfies ADD, SWU and SAU, but not NPP.
- $\varphi_i = \frac{v(N)}{n}$ satisfies ADD, SWU and NPP, but not SAU.
- $\varphi_i = \frac{2v(N)}{mp_k}$ if $i = \min\limits_{j\in P_k} j$ or $\varphi_i = \frac{(p_k-2)v(N)}{mp_k(p_k-1)}$ if $i \in P_k$ and $i \neq \min_{j\in P_k} j$, satisfies ADD, SAU and NPP, but not SWU.
- $\varphi_i = \frac{2v(N)}{mp_k|Z_k|}$ if $i \in Z_k = \{j \in P_k / v(j) = \min\limits_{z\in P_k} v(z)\}$, $\varphi_i = \frac{(p_k-2)v(N)}{mp_k(p_k-|Z_k|)}$ if $i \in P_k \backslash Z_k$, satisfies SWU, SAU and NPP, but not ADD.

b) Independence of the properties of Theorem 3.2:

- $\varphi_i = \frac{v(P_k)}{p_k} + \frac{v(N)-\sum_{l\in M} v(P_l)}{n}$ satisfies ADD, SWU, DUNPP, but not SAU.
- $\varphi_i = \frac{v(P_k)}{p_k} + \frac{2(v(N)-\sum_{l\in M} v(P_l))}{mp_k}$ if $i = \min\limits_{j\in P_k} j$ or $\varphi_i = \frac{v(P_k)}{p_k} + \frac{(p_k-2)(v(N)-\sum_{l\in M} v(P_l))}{mp_k(p_k-1)}$ if $i \in P_k$ and $i \neq \min_{j\in P_k} j$, satisfies ADD, SAU and DUNPP, but not SWU.
- $\varphi_i = \frac{v(P_k)}{p_k} + \frac{2(v(N)-\sum_{l\in M} v(P_l))}{mp_k|Z_k|}$ if $i \in Z_k = \{j \in P_k / v(j) = \min\limits_{z\in P_k} v(z)\}$, $\varphi_i = \frac{v(P_k)}{p_k} + \frac{(p_k-2)(v(N)-\sum_{l\in M} v(P_l))}{mp_k(p_k-|Z_k|)}$ if $i \in P_k \backslash Z_k$, satisfies SWU, SAU and DUNPP, but not ADD.
- $\varphi_i = v(i) + \frac{v(P_k)-\sum_{j\in P_k} v(j)}{p_k} + \frac{v(N)-\sum_{l\in M} v(P_l)}{mp_k}$ satisfies ADD, SWU and SAU, but not DUNPP.

c) Independence of the properties of Theorem 3.3:

- $\varphi_i = v(i) + \frac{v(P_k)-\sum_{j\in P_k} v(j)}{p_k} + \frac{v(N)-\sum_{l\in M} v(P_l)}{n}$ satisfies ADD, SWU and DUPP, but not SAU.

- $\varphi_i = v(i) + \frac{v(P_k)-\sum_{j\in P_k} v(j)}{p_k} + \frac{2(v(N)-\sum_{l\in M} v(P_l))}{mp_k}$ if $i = \min\limits_{j\in P_k} j$ or $\varphi_i = v(i) + \frac{v(P_k)-\sum_{j\in P_k} v(j)}{p_k} + \frac{(p_k-2)(v(N)-\sum_{l\in M} v(P_l))}{mp_k(p_k-1)}$ if $i \in P_k$ and $i \neq \min_{j\in P_k} j$, satisfies ADD, SAU and DUPP, but not SWU.

- $\varphi_i = v(i) + \frac{v(P_k)-\sum_{j\in P_k} v(j)}{p_k} + \frac{2(v(N)-\sum_{l\in M} v(P_l))}{mp_k|Z_k|}$ if $i \in Z_k = \{j \in P_k / v(j) = \min\limits_{z\in P_k} v(z)\}$, $\varphi_i = v(i) + \frac{v(P_k)-\sum_{j\in P_k} v(j)}{p_k} + \frac{(p_k-2)(v(N)-\sum_{l\in M} v(P_l))}{mp_k(p_k-|Z_k|)}$ if $i \in P_k \backslash Z_k$, satisfies SWU, SAU and DUNPP, but not ADD.

- $\varphi_i = \frac{v(P_k)}{p_k} + \frac{v(N)-\sum_{l\in M} v(P_l)}{mp_k}$ satisfies ADD, SWU and SAU, but not DUPP.

d) Independence of the properties of Theorem 3.4:

- $\varphi_i = v(i) + \frac{v(N)-\sum_{j\in N} v(j)}{n}$ satisfies ADD, SWU and DPP, but not WSAU.
- $\varphi_i = v(i) + \frac{2(v(N)-\sum_{j\in N} v(j))}{mp_k}$ if $i = \min\limits_{j\in P_k} j$ or $\varphi_i = v(i) + \frac{(p_k-2)(v(N)-\sum_{j\in N} v(j))}{mp_k(p_k-1)}$ if $i \in P_k$ and $i \neq \min_{j\in P_k} j$, satisfies ADD, WSAU and DPP, but not SWU.
- $\varphi_i = v(i) + \frac{2(v(N)-\sum_{j\in N} v(j))}{mp_k|Z_k|}$ if $i \in Z_k = \{j \in P_k / v(j) = \min\limits_{z\in P_k} v(z)\}$, $\varphi_i = v(i) + \frac{(p_k-2)(v(N)-\sum_{j\in N} v(j))}{mp_k(p_k-|Z_k|)}$ if $i \in P_k \backslash Z_k$, satisfies SWU, WSAU and DPP, but not ADD.
- $\varphi_i = \frac{v(P_k)}{p_k} + \frac{v(N)-\sum_{l\in M} v(P_l)}{mp_k}$ satisfies ADD, SWU, WSAU but not DPP.

# Chapter 4

# New results on egalitarian values for games with a priori unions

JM Alonso-Meijide, JC Gonçalves-Dosantos

Preprint

## 4.1 Introduction

In cooperative games, the egalitarian values are rules for distributing the cooperative benefits between the agents on the basis of a kind of egalitarian principle. These values are some of the most studied in the literature, as well as the Shapley value (Shapley 1953). For instance, van den Brink (2007) compare the equal division value and the Shapley value, Casajus and Hüttner (2014) also study the equal surplus division value proposed in Driessen and Funaki (1991). These values satisfy several good properties and then, van den Brink and Funaki (2009), Chun and Park (2012), van den Brink et al. (2016), Ferrières (2017) and Bèal et al. (2019), among others, provide many axiomatic characterizations.

In Owen (1977) cooperative games with a priori unions are considered. In these games there is a partition of the set of agents, that affects to the negotiation among the different agents. In addition, Owen (1977) extends the Shapley value to cooperative games with a priori unions, known as the Owen value, and characterises it. Numerous papers study the Owen value providing new axiomatic characterizations as Lorenzo-Freire (2016), applications as Costa (2016), and a sampling procedure to approximate it in Saavedra-Nieves et al. (2018). Another values are extended to cooperative games with a priori unions. In the context that concerns us, Alonso-Meijide et al. (2020) extend and characterize the equal division value and the equal surplus division value. For the second value, three alternative ways to adapt it to a priori unions are proposed.

In this paper we provide new axiomatic characterizations of these coalitional extensions of the equal division and equal surplus division values. Besides we obtain new solutions using the Owen method and balanced contributions. In Section 4.2 we introduce some preliminary concepts about game theory. In Section 4.3 we characterise the coalitional equal division value using the concept of coalitional value. We also prove that this solution coincides with the one obtained by the Owen method. In Section 4.4 we characterise the coalitional equal surplus division values in the same context as the previous section. Finally, in Section 4.5 we obtain a new solution with the Owen method. With the concept of balanced contributions and quotient game property, we obtain a new coalitional extension of the equal surplus division value.

## 4.2 Preliminaries

### 4.2.1 TU-games and values

A transferable utility cooperative game (from now on a TU-game) is a pair $(N, v)$ where $N$ is a finite set of $n$ players, and $v$ is a map from $2^N$ to $\mathbb{R}$ with $v(\emptyset) = 0$, that is called the characteristic function of the game. In the sequel, $\mathcal{G}_N$ will denote the family of all TU-games with player set $N$ and $\mathcal{G}$ the family of all TU-games. A value for TU-games is a map $f$ that assigns to every TU-game $(N, v) \in \mathcal{G}$ a vector $f(N, v) = (f_i(N, v))_{i \in N} \in \mathbb{R}^N$.

Well-known values for TU-games are the egalitarian values. The equal division value $ED$ distributes $v(N)$ equally among the players in $N$. Formally, the equal division value $ED$ is defined for every $(N, v) \in \mathcal{G}$, and every $i \in N$ by

$$ED_i(N, v) = \frac{v(N)}{n}.$$

The equal surplus division value $ESD$ is defined for every $(N, v) \in \mathcal{G}$, and every $i \in N$ by

$$ESD_i(N, v) = v(i) + \frac{v^0(N)}{n}$$

where $v^0(N) = v(N) - \sum_{i \in N} v(i)$. Notice that $ESD$ is a variant of $ED$ in which we first allocate $v(i)$ to each player $i$, and then distribute $v^0(N)$ among the players using $ED$. $ESD$ is a reasonable alternative to $ED$ for situations where individual benefits and joint benefits are neatly separable.

Alternative values for TU-games are the Shapley (Shapley 1953) and Banzhaf value (Banzhaf 1964).

### 4.2.2 Games with a priori unions

We denote by $P(N)$ the set of all partitions of $N$. Then, a TU-game with a priori unions is a triplet $(N,v,P)$ where $(N,v) \in \mathcal{G}$, $P = \{P_1,\dots,P_m\} \in P(N)$ and $P_k \in P$ is called a priori union for all $k \in M$ with $M = \{1,...,m\}$. The set of TU-games with a priori unions and with player set $N$ will be denoted by $\mathcal{G}_N^U$, and the set of all TU-games with a priori unions by $\mathcal{G}^U$. A value for TU-games with a priori unions is a map $g$ that assigns to every $(N,v,P) \in \mathcal{G}^U$ a vector $g(N,v,P) = (g_i(N,v,P))_{i\in N} \in \mathbb{R}^N$.

Two examples of values for TU-games with a priori unions are the Owen value (Owen 1977), and the Banzhaf-Owen value (Owen 1981).

Given $(N,v,P) \in \mathcal{G}^U$ with $P = \{P_1,\dots,P_m\} \in P(N)$, the quotient game of $(N,v,P)$ is the TU-game $(M,v/P)$ where

$$(v/P)(R) = v\left(\cup_{r\in R} P_r\right) \text{ for all } R \subseteq M.$$

We say that a value for TU-games with a priori unions $g$ is a *coalitional equal division value* if, for any TU-game $(N,v) \in \mathcal{G}$, it holds that

$$g(N,v,P^n) = ED(N,v),$$

where $P^n$ denotes the trivial partition $\{\{1\},\dots,\{n\}\}$. Using similar concepts, the Owen value is a coalitional Shapley value, and Banzhaf-Owen value is a coalitional Banzhaf value.

### 4.2.3 Coalitional values in two steps

Given $(N,v,P) \in \mathcal{G}_N^U$ with $P = \{P_1,\dots,P_m\}$ and a coalition $S \subseteq P_r$, the modified game of $(N,v,P)$ is defined as $(M,u_{r,S})$ where

$$u_{r,S}(H) = \begin{cases} v\left(\cup_{k\in H} P_k\right) & \text{if } r \notin H \\ v\left(\cup_{k\in H\setminus r} P_k \cup S\right) & \text{if } r \in H \end{cases} \tag{4.1}$$

for all $H \subseteq M$. That is, the modified game $(M,u_{r,S})$, is defined based on the game $(M,v,P)$ where each player $k$ with $k \neq r$ is the union $P_k$, and the player $r$ is the coalition $S$.

Using the modified game $(M,u_{r,S})$ and a value $f$ for TU-games, the reduced game $(P_r,w_r)$ is a TU-game with set of players $P_r$ and characteristic function

$$w_r(S) = f_r(M, u_{r,S}) \tag{4.2}$$

for any $S \subseteq P_r$.

Finally, a value $g$ for TU-games with a priori unions is obtained reapplying the value $f$ over $(P_r, w_r)$. That is

$$g_i(N, v, P) = f_i(P_r, w_r) \tag{4.3}$$

for all $i \in P_r$.

We call Owen procedure to that described in (4.2) and (4.3) to obtain a coalitional value g for TU games with a priori unions using a value f for TU games.

The Owen value is the result of applying the Owen procedure using the Shapley value (Owen 1977).

The Banzhaf-Owen value is the result of applying the Owen procedure using the Banzhaf value (Owen 1981).

## 4.3 The equal division value for TU-games with a priori unions

Alonso-Meijide et al. (2020) provide an extension of the equal division value to TU-games with a priori unions and an axiomatic characterization of this new value.

**Definition 4.1.** (Alonso-Meijide et al. 2020) The equal division value for TU-games with a priori unions $ED^U$ is defined by

$$ED_i^U(N, v, P) = \frac{v(N)}{mp_k}$$

for all $(N, v, P) \in \mathcal{G}^U$ with $P = \{P_1, \ldots, P_m\}$ and $i \in P_k$ where $p_k$ denotes the cardinal of $P_k$.

Now we provide a second axiomatic characterization of the equal division value for TU-games with a priori unions in the same spirit as an axiomatic characterization of the Owen value given in Vázquez-Brage et al. (1997). Let us first introduce two additional properties.

**Quotient Game Property (QGP).** A value $g$ for TU-games with a priori unions satisfies the quotient game property if, for all $(N, v, P) \in \mathcal{G}_N^U$ with $P = \{P_1, \ldots, P_m\}$, it holds that

$$\sum_{i \in P_k} g_i(N, v, P) = g_k(M, v/P, P^m)$$

for all $P_k \in P$, where $(M, v/P)$ is the quotient game of $(N, v, P)$.

**Balanced Contributions in the Unions (BCU).** A value $g$ for TU-games with a priori unions satisfies balanced contributions in the unions if, for all $(N, v, P) \in \mathcal{G}_N^U$ and all $i, j \in P_k$ with $P_k \in P$, it holds that

$$g_i(N, v, P) - g_i(N, v, P_{-j}) = g_j(N, v, P) - g_j(N, v, P_{-i})$$

where $P_{-l}$ denotes the partition $\{P_1, \ldots, P_{k-1}, P_k \setminus \{l\}, \{l\}, P_{k+1}, \ldots, P_m\}$ for all $l \in P_k$.

**Theorem 4.1.** *$ED^U$ is the unique coalitional equal division value satisfying QGP and BCU.*

*Proof.* Take a TU-game with a priori unions $(N, v, P) \in \mathcal{G}^U$ such that $P = \{P_1, ..., P_m\}$ and denote $M = \{1, ..., m\}$. Let us check that $ED^U$ satisfies QGP. For all $k \in M$, we have that

$$\sum_{i \in P_k} ED_i^U(N, v, P) = \sum_{i \in P_k} \frac{v(N)}{mp_k} = \frac{v(N)}{m}$$

and

$$ED_k^U(M, v/P, P^m) = \frac{(v/P)(M)}{m} = \frac{v(N)}{m}.$$

Let us check that $ED^U$ satisfies BCU. For all $i, j \in P_k$, we have that

$$ED_i^U(N, v, P) - ED_i^U(N, v, P_{-j}) = \frac{v(N)}{mp_k} - \frac{v(N)}{(m+1)(p_k - 1)}$$

and

$$ED_j^U(N, v, P) - ED_j^U(N, v, P_{-i}) = \frac{v(N)}{mp_k} - \frac{v(N)}{(m+1)(p_k - 1)}.$$

Finally, the uniqueness is proven in an analogous way as the uniqueness in Theorem 2 of Vázquez-Brage et al. (1997). □

The $ED^U$ value is an extension of $ED$ for TU games with a priori unions quite intuitive and natural. Moreover, let us check that is the value obtained by the procedure to obtain coalitional values in two steps proposed in Owen (1977) described in Subsection 4.2.3.

**Theorem 4.2.** *The equal division value with a priori unions $ED^U$ is the result of applying the Owen procedure using the equal division value $ED$.*

*Proof.* Take a TU-game with a priori unions $(N, v, P) \in \mathcal{G}^U$ such that $P = \{P_1, ..., P_m\}$ and denote $M = \{1, ..., m\}$. Given a coalition $S \subseteq P_r$, we can obtained the reduced game (4.2) applying $ED$ to the modified game (4.1),

$$w_r(S) = ED_r(M, u_{r,S}) = \frac{u_{r,S}(M)}{m} = \frac{v\left(\cup_{k \in H \setminus r} P_k \cup S\right)}{m}$$

Again, if we reapply $ED$ to the reduced game (4.2) as (4.3), for all player $i \in P_r$ we obtain

$$ED_i(P_r, w_r) = \frac{w_r(P_r)}{p_r} = \frac{v\left(\cup_{k \in H \setminus r} P_k \cup P_r\right)/m}{p_r} = \frac{v(N)}{mp_r} = ED_i^U(N, v, P).$$

□

## 4.4 Three equal surplus division values for TU-games with a priori unions

In Alonso-Meijide et al. (2020), three alternative ways for extending the equal surplus division value to TU-games with a priori unions are proposed. In this section we provide new characterizations of these coalitional values.

### 4.4.1 The equal surplus division value 1

The equal surplus division value 1 divides the value of the grand coalition in the quotient game using the equal surplus division value and then divides the amount assigned to each union equally among its members.

**Definition 4.2.** (Alonso-Meijide et al. 2020) The equal surplus division value (one) for TU-games with a priori unions $ESD1^U$ is defined by

$$ESD1_i^U(N, v, P) = \frac{(v/P)(k)}{p_k} + \frac{(v/P)^0(M)}{mp_k} = \frac{v(P_k)}{p_k} + \frac{v(N) - \sum_{l \in M} v(P_l)}{mp_k}$$

for all $(N, v, P) \in \mathcal{G}^U$ with $P = \{P_1, \ldots, P_m\}$ and $i \in P_k$.

Notice that it is easy to check that $ESD1^U$ is a *coalitional equal surplus division value*, in the sense that

$$ESD1^U(N, v, P^n) = ESD(N, v)$$

for all $(N, v) \in \mathcal{G}$.

We use this feature to provide a new characterization of $ESD1^U$ in the remainder of this subsection. First let us define a new property.

**Equality Inside Unions (EIU).** A value $g$ for TU-games with a priori unions satisfies equality inside unions if, for all $(N, v, P) \in \mathcal{G}^U$, all $P_k \in P$, and all $i, j \in P_k$, it holds that $g_i(N, v, P) - g_j(N, v, P) = 0$.

**Theorem 4.3.** *$ESD1^U$ is the unique coalitional equal surplus division value for TU-games with a priori unions satisfying QGP and EIU.*

*Proof.* Take a TU-game with a priori unions $(N, v, P) \in \mathcal{G}^U$ such that $P = \{P_1, ..., P_m\}$ and denote $M = \{1, ..., m\}$. Let us check that $ESD1^U$ satisfies QGP. For all $k \in M$, we have that

$$\sum_{i \in P_k} ESD1_i^U(N, v, P) = \sum_{i \in P_k} \left( \frac{v(P_k)}{p_k} + \frac{v(N) - \sum_{l \in M} v(P_l)}{mp_k} \right)$$
$$= v(P_k) + \frac{v(N) - \sum_{l \in M} v(P_l)}{m}$$

and

$$ESD1_k^U(M, v/P, P^m) = \frac{(v/P)(k)}{1} + \frac{(v/P)(M) - \sum_{l \in M}(v/P)(l)}{m}$$
$$= v(P_k) + \frac{v(N) - \sum_{l \in M} v(P_l)}{m}.$$

Let us check that $ESD1^U$ satisfies EIU. For all $i, j \in P_k$, we have that

$$ESD1_i^U(N, v, P) - ESD1_j^U(N, v, P) = 0.$$

Finally, the uniqueness is proven in an analogous way as the uniqueness in Theorem 2 of Vázquez-Brage et al. (1997). □

### 4.4.2 The equal surplus division value 2

The equal surplus division value 2 divides again the value of the grand coalition in the quotient game using the equal surplus division value; then it distributes the amount $v(P_k)$ assigned to each union $P_k$ giving $v(i)$ to each player $i \in P_k$ and dividing $v(P_k) - \sum_{j \in P_k} v(j)$ equally among the players in $P_k$.

**Definition 4.3.** (Alonso-Meijide et al. 2020) The equal surplus division value (two) for TU-games with a priori unions $ESD2^U$ is defined by

$$ESD2_i^U(N,v,P) = v(i) + \frac{v(P_k) - \sum_{j \in P_k} v(j)}{p_k} + \frac{v(N) - \sum_{l \in M} v(P_l)}{mp_k}$$

for all $(N,v,P) \in \mathcal{G}^U$ with $P = \{P_1, \ldots, P_m\}$ and $i \in P_k$.

Notice that it is easy to check that $ESD2^U$ is a *coalitional equal surplus division value*, in the sense that

$$ESD2^U(N,v,P^n) = ESD(N,v)$$

for all $(N,v) \in \mathcal{G}$.

We use this feature to provide a new characterization of $ESD2^U$ in the remainder of this subsection. First let us define a new property.

**Difference Maintenance of Individual Values Inside Unions (DMIVIU).** A value $g$ for TU-games with a priori unions satisfies difference maintenance of individual values inside unions if, for all $(N,v,P) \in \mathcal{G}^U$, all $P_k \in P$, and all $i,j \in P_k$, it holds that $g_i(N,v,P) - g_j(N,v,P) = v(i) - v(j)$.

**Theorem 4.4.** *$ESD2^U$ is the unique coalitional equal surplus division value for TU-games with a priori unions satisfying QGP and DMIVIU.*

*Proof.* Take a TU-game with a priori unions $(N,v,P) \in \mathcal{G}^U$ such that $P = \{P_1, ..., P_m\}$ and denote $M = \{1, ..., m\}$. Let us check that $ESD2^U$ satisfies QGP. For all $k \in M$, we have that

$$\begin{aligned}
\sum_{i \in P_k} ESD2_i^U(N,v,P) &= \sum_{i \in P_k} \left( v(i) + \frac{v(P_k) - \sum_{j \in P_k} v(j)}{p_k} + \frac{v(N) - \sum_{l \in M} v(P_l)}{mp_k} \right) \\
&= \sum_{i \in P_k} v(i) + v(P_k) - \sum_{j \in P_k} v(j) + \frac{v(N) - \sum_{l \in M} v(P_l)}{m} \\
&= v(P_k) + \frac{v(N) - \sum_{l \in M} v(P_l)}{m}
\end{aligned}$$

and

$$\begin{aligned}
ESD2_k^U(M, v/P, P^m) &= (v/P)(k) + \frac{(v/P)(k) - (v/P)(k)}{1} \\
&\quad + \frac{(v/P)(M) - \sum_{l \in M} (v/P)(l)}{m} \\
&= v(P_k) + \frac{v(N) - \sum_{l \in M} v(P_l)}{m}.
\end{aligned}$$

Let us check that $ESD2^U$ satisfies DMIVIU. For all $i, j \in P_k$, we have that

$$ESD2^U_i(N, v, P) - ESD2^U_j(N, v, P) = v(i) - v(j).$$

Finally, the uniqueness is proven in an analogous way as the uniqueness in Theorem 2 of Vázquez-Brage et al. (1997). □

### 4.4.3 The equal surplus division value 3

Finally, the equal surplus division value 3 assigns $v(i)$ to each player $i$ and then divides $v^0(N)$ among the players using $ED^U$.

**Definition 4.4.** (Alonso-Meijide et al. 2020) The equal surplus division value (three) for TU-games with a priori unions $ESD3^U$ is defined by

$$ESD3^U_i(N, v, P) = v(i) + ED^U(N, v^0, P) = v(i) + \frac{v(N) - \sum_{j \in N} v(j)}{mp_k}$$

for all $(N, v, P) \in \mathcal{G}^U$ with $P = \{P_1, \ldots, P_m\}$ and $i \in P_k$.

Notice that it is easy to check that $ESD3^U$ is a *coalitional equal surplus division value*, in the sense that

$$ESD3^U(N, v, P^n) = ESD(N, v)$$

for all $(N, v) \in \mathcal{G}$.

We use this feature to provide a new characterization of $ESD3^U$ in the remainder of this section. Nevertheless $ESD3^U$ does not satisfy QGP, so in order to do it we introduce a new modified of quotient game.

Take $(N, v, P) \in \mathcal{G}^U$ with $P = \{P_1, ..., P_m\}$ and denote $M = \{1, ..., m\}$. The quotient* game of $(N, v, P)$ is the TU-game $(M, \bar{v}/P)$ where

$$(\bar{v}/P)(R) = \begin{cases} \displaystyle\sum_{k \in R} \sum_{i \in P_k} v(i) & \text{if } R \subset M \\ v(N) & \text{if } R = M \end{cases}$$

**Quotient* Game Property (Q*GP).** A value $g$ for TU-games with a priori unions satisfies the quotient* game property if, for all $(N, v, P) \in \mathcal{G}^U$ with $P = \{P_1, ..., P_m\}$, it holds that

$$\sum_{i \in P_k} g_i(N, v, P) = g_k(M, \bar{v}/P, P^m)$$

for all $P_k \in P$, where $(M, \bar{v}/P)$ is the quotient* game of $(N, v, P)$.

**Theorem 4.5.** *$ESD3^U$ is the unique coalitional equal surplus division value for TU-games with a priori unions satisfying Q*GP and BCU.*

*Proof.* Take a TU-game with a priori unions $(N, v, P) \in \mathcal{G}^U$ such that $P = \{P_1, ..., P_m\}$ and denote $M = \{1, ..., m\}$. Let us check that $ESD3^U$ satisfies Q*GP. For all $k \in M$, we have that

$$\sum_{i \in P_k} ESD3_i^U(N, v, P) = \sum_{i \in P_k} \left( v(i) + \frac{v(N) - \sum_{j \in N} v(j)}{m p_k} \right)$$
$$= \sum_{i \in P_k} v(i) + \frac{v(N) - \sum_{j \in N} v(j)}{m}$$

and

$$ESD3_k^U(M, \bar{v}/P, P^m) = (\bar{v}/P)(k) + \frac{(\bar{v}/P)(M) - \sum_{l \in M} (\bar{v}/P)(l)}{m}$$
$$= \sum_{i \in P_k} v(i) + \frac{v(N) - \sum_{k \in M} \sum_{j \in P_k} v(j)}{m}$$
$$= \sum_{i \in P_k} v(i) + \frac{v(N) - \sum_{j \in N} v(j)}{m}.$$

Let us check that $ESD3^U$ satisfies BCU. For all $i, j \in P_k$, we have that

$$ESD3_i^U(N, v, P) - ESD3_i^U(N, v, P_{-j}) = \frac{v(N) - \sum_{j \in N} v(j)}{m p_k} - \frac{v(N) - \sum_{j \in N} v(j)}{(m+1)(p_k - 1)}$$

and

$$ESD3_j^U(N, v, P) - ESD3_j^U(N, v, P_{-i}) = \frac{v(N) - \sum_{j \in N} v(j)}{m p_k} - \frac{v(N) - \sum_{j \in N} v(j)}{(m+1)(p_k - 1)}.$$

Finally, the uniqueness is proven in an similar way as the uniqueness in Theorem 2 of Vázquez-Brage et al. (1997). □

## 4.5 Two new extensions of the equal surplus division value

In this section, we introduce two new extensions of the equal surplus division value for TU-games with a priori unions. One of them is the value obtained applying the Owen procedure using the equal surplus division value. The other value is an extension of the equal surplus division value that satisfies the quotient game and balanced contributions in the unions properties.

### 4.5.1 Coalitional value using equal surplus division value in two steps

The first new extension is obtained applying the procedure in two steps using the equal surplus division value.

**Definition 4.5.** The equal surplus division value (four) for TU-games with a priori unions $ESD4^U$ is defined by

$$ESD4_i^U(N,v,P) = v(i) + \frac{v(P_k) - \sum_{j\in P_k} v(j)}{p_k} + \frac{v(N) - \sum_{l\in M} v(P_l)}{mp_k} +$$
$$\frac{1}{m}\left(\sum_{t\in P_k} \frac{v(t)}{p_k} - v(i)\right) + \frac{1}{m}\left(v\left(\cup_{r\in M\setminus k} P_r \cup i\right) - \sum_{t\in P_k} \frac{v\left(\cup_{r\in M\setminus k} P_r \cup t\right)}{p_k}\right)$$

for all $(N,v,P) \in \mathcal{G}^U$ with $P = \{P_1,\ldots,P_m\}$ and $i \in P_k$.

Using again the procedure proposed by Owen (1977) shown in Subsection 4.2.3 where now $ESD$ is used in the games (4.2) and (4.3), we can obtain the solution $g = ESD4^U$. Let us see this in the next theorem.

**Theorem 4.6.** *The equal surplus division value with a priori unions $ESD4^U$ is the result of applying the Owen procedure using the equal surplus division value $ESD$.*

*Proof.* Take a TU-game with a priori unions $(N,v,P) \in \mathcal{G}^U$ such that $P = \{P_1,...,P_m\}$ and denote $M = \{1,...,m\}$. First at all, $ESD$ is applied to the modified game (4.1), to obtained the reduced game (4.2); where note that only the individual pay-offs $u_{r,S}(i)$ and the total pay-off $u_{r,S}(M)$ are necessary. Therefore, for all union $P_r$

$$w_r(S) = ESD_r(M,u_{r,S}) = u_{r,S}(r) + \frac{u_{r,S}(M) - \sum_{l\in M} u_{r,S}(l)}{m} =$$
$$v(S) + \frac{v\left(\cup_{k\in M\setminus r} P_k \cup S\right) - \sum_{l\in M\setminus r} v(P_l) - v(S)}{m}.$$

Taking again $ESD$, and applying it to the reduced game (4.2) as (4.3), for all player $i \in P_r$,

$$\begin{aligned}
ESD_i\left(P_r, w_r\right) &= w_r(i) + \frac{w_r(P_r) - \sum_{t\in P_r} w_r(t)}{p_r} \\
&= v\left(i\right) + \frac{v\left(\cup_{k\in M\setminus r} P_k \cup i\right) - \sum_{l\in M\setminus r} v\left(P_l\right) - v\left(i\right)}{m} \\
&+ \frac{v\left(P_r\right) + \frac{v\left(\cup_{k\in M\setminus r} P_k \cup P_r\right) - \sum_{l\in M\setminus r} v(P_l) - v(P_r)}{m}}{p_r} \\
&- \frac{\sum_{t\in P_r} (v\left(t\right) + \frac{v\left(\cup_{k\in M\setminus r} P_k \cup t\right) - \sum_{l\in M\setminus r} v(P_l) - v(t)}{m})}{p_r}.
\end{aligned}$$

As

$$v\left(\cup_{k\in M\setminus r} P_k \cup P_r\right) - \sum_{l\in M\setminus r} v\left(P_l\right) - v\left(P_r\right) = v\left(N\right) - \sum_{l\in M} v\left(P_l\right)$$

we have that

$$\begin{aligned}
ESD_i\left(P_r, w_r\right) =& v\left(i\right) + \frac{v\left(\cup_{k\in M\setminus r} P_k \cup i\right)}{m} - \frac{v\left(i\right)}{m} - \frac{\sum_{l\in M\setminus r} v\left(P_l\right)}{m} + \frac{v\left(P_r\right)}{p_r} + \\
& \frac{v\left(N\right)}{mp_r} - \frac{\sum_{l\in M} v\left(P_l\right)}{mp_r} - \sum_{t\in P_r} \frac{v\left(t\right)}{p_r} - \sum_{t\in P_r} \frac{v\left(\cup_{k\in M\setminus r} P_k \cup t\right)}{mp_r} + \\
& \sum_{t\in P_r} \sum_{l\in M\setminus r} \frac{v\left(P_l\right)}{mp_r} + \sum_{t\in P_r} \frac{v\left(t\right)}{mp_r}.
\end{aligned}$$

The second last term can be written as

$$\sum_{t\in P_r} \sum_{l\in M\setminus r} \frac{v\left(P_l\right)}{mp_r} = \sum_{l\in M\setminus r} p_r \frac{v\left(P_l\right)}{mp_r} = \sum_{l\in M\setminus r} \frac{v\left(P_l\right)}{m}$$

and then, we obtain that

$$\begin{aligned}
& ESD_i\left(P_r, w_r\right) = v\left(i\right) + \frac{v\left(\cup_{k\in M\setminus r} P_k \cup i\right)}{m} - \frac{v\left(i\right)}{m} + \frac{v\left(P_r\right)}{p_r} + \\
& \frac{v\left(N\right)}{mp_r} - \frac{\sum_{l\in M} v\left(P_l\right)}{mp_r} - \sum_{t\in P_r} \frac{v\left(t\right)}{p_r} - \sum_{t\in P_r} \frac{v\left(\cup_{k\in M\setminus r} P_k \cup t\right)}{mp_r} + \sum_{t\in P_r} \frac{v\left(t\right)}{mp_r}.
\end{aligned}$$

Reordering terms, we have that

$$ESD_i(P_r, w_r) = v(i) + \frac{1}{m}\left(\sum_{t\in P_r}\frac{v(t)}{p_r} - v(i)\right) + \frac{1}{p_r}\frac{v(N) - \sum_{l\in M} v(P_l)}{m} +$$
$$\frac{1}{p_r}\left(v(P_r) - \sum_{t\in P_r} v(t)\right) + \frac{1}{m}\left(v\left(\cup_{k\in M\backslash r} P_k \cup i\right) - \sum_{t\in P_r}\frac{v\left(\cup_{k\in M\backslash r} P_k \cup t\right)}{p_r}\right)$$
$$= ESD4_i^U(N, v, P).$$

□

**Remark 4.1.** Note that $ESD4$ can be written in terms of $ESD2$

$$ESD4_i^U(N, v, P) = ESD2_i^U(N, v, P) + \frac{1}{m}\left(\sum_{t\in P_r}\frac{v(t)}{p_r} - v(i)\right) +$$
$$\frac{1}{m}\left(v\left(\cup_{k\in M\backslash r} P_k \cup i\right) - \sum_{t\in P_r}\frac{v\left(\cup_{k\in M\backslash r} P_k \cup t\right)}{p_r}\right)$$

for all $(N, v, P) \in \mathcal{G}^U$ with $P = \{P_1, \ldots, P_m\}$ and $i \in P_k$.

It is easy to check that

$$ESD4_i^U(N, v, P^n) = ESD_i(N, v)$$

and

$$\sum_{i\in P_r} ESD4_i^U(N, v, P) = ESD_r(M, v/P),$$

that is $ESD4^U$ is a coalitional equal surplus division value and it satisfies the quotient game property.

The value $ESD4$ is a coalitional equal surplus division value that satisfies quotient game. To characterize $ESD4^U$ in the proposed context, let us define the following property.

**Balanced contributions due to the players abandonment in the union (BCPA).** A value $g$ for TU-games with a priori unions satisfies BCPA if, for all $(N, v, P) \in \mathcal{G}^U$ and all $i, j \in P_k$ with $P_k \in P$, it holds that

$$g_i(N, v, P) - g_i\left(N\backslash P_k \cup i, v_{N\backslash P_k \cup i}, P\backslash P_k \cup \{i\}\right) =$$
$$g_j(N, v, P) - g_j\left(N\backslash P_k \cup j, v_{N\backslash P_k \cup j}, P\backslash P_k \cup \{j\}\right)$$

where the game $\left(N\backslash P_k \cup i, v_{N\backslash P_k\cup i}, P\backslash P_k \cup \{i\}\right)$ is defined as $v_{N\backslash P_k\cup i}(S) = v(S)$ for all $S \subseteq N\backslash P_k \cup i$.

This new property says that given two players in the same union, they get the same difference between the pay-off of the original game and the pay-off of the game where all the players of the union leave. This property has a similar interpretation to the property of balanced contributions.

**Theorem 4.7.** *$ESD4^U$ is the unique coalitional equal surplus division value for TU-games with a priori unions satisfying QGP and BCPA.*

*Proof.* Take a TU-game with a priori unions $(N, v, P) \in \mathcal{G}^U$ such that $P = \{P_1, ..., P_m\}$ and denote $M = \{1, ..., m\}$. Let us check that $ESD4^U$ satisfies BCPA. For all $l \in M$ and all $i, j \in P_l$,

$$ESD4_i^U(N,v,P) - ESD4_j^U(N,v,P) = ESD2_i^U(N,v,P) - ESD2_j^U(N,v,P) + \frac{1}{m}\left(\sum_{t\in P_l}\frac{v(t)}{p_l} - v(i)\right) + \frac{1}{m}\left(v\left(\cup_{k\in M\backslash l}P_k \cup i\right) - \sum_{t\in P_l}\frac{v\left(\cup_{k\in M\backslash l}P_k \cup t\right)}{p_l}\right) - \frac{1}{m}\left(\sum_{t\in P_l}\frac{v(t)}{p_l} - v(j)\right) - \frac{1}{m}\left(v\left(\cup_{k\in M\backslash l}P_k \cup j\right) - \sum_{t\in P_l}\frac{v\left(\cup_{k\in M\backslash l}P_k \cup t\right)}{p_l}\right).$$

By the property DMIVIU that satisfies $ESD2^U$ we have that $ESD2_i - ESD2_j = v(i) - v(j)$. Then we have that

$$ESD4_i^U(N,v,P) - ESD4_j^U(N,v,P) = v(i) - v(j) - \frac{v(i)}{m} + \frac{v(j)}{m} + \frac{v\left(\cup_{k\in M\backslash l}P_k \cup i\right)}{m} - \frac{v\left(\cup_{k\in M\backslash l}P_k \cup j\right)}{m}.$$

On the other hand

$$ESD4_i^U\left(N\backslash P_l \cup i, v_{N\backslash P_l\cup i}, P\backslash P_l \cup \{i\}\right) - ESD4_j^U\left(N\backslash P_l \cup j, v_{N\backslash P_l\cup j}, P\backslash P_l \cup \{j\}\right) = ESD2_i^U\left(N\backslash P_l \cup i, v_{N\backslash P_l\cup i}, P\backslash P_l \cup \{i\}\right) + \frac{1}{m}\left(\frac{v_{N\backslash P_l\cup i}(i)}{1} - v_{N\backslash P_l\cup i}(i)\right) + \frac{1}{m}\left(v_{N\backslash P_l\cup i}\left(\cup_{P_k\in P\backslash P_l}P_k \cup i\right) - \frac{v_{N\backslash P_l\cup i}\left(\cup_{P_k\in P\backslash P_l}P_k \cup i\right)}{1}\right) - ESD2_j^U\left(N\backslash P_l \cup j, v_{N\backslash P_l\cup j}, P\backslash P_l \cup \{j\}\right) - \frac{1}{m}\left(\frac{v_{N\backslash P_l\cup j}(j)}{1} - v_{N\backslash P_l\cup j}(j)\right) - \frac{1}{m}\left(v_{N\backslash P_l\cup j}\left(\cup_{P_k\in P\backslash P_l}P_k \cup j\right) - \frac{v_{N\backslash P_k\cup j}\left(\cup_{P_k\in P\backslash P_l}P_k \cup j\right)}{1}\right).$$

We have that

$$ESD4_i^U\left(N\backslash P_l\cup i, v_{N\backslash P_l\cup i}, P\backslash P_l\cup\{i\}\right) - ESD4_j^U\left(N\backslash P_l\cup j, v_{N\backslash P_l\cup j}, P\backslash P_l\cup\{j\}\right) =$$
$$ESD2_i^U\left(N\backslash P_l\cup i, v_{N\backslash P_l\cup i}, P\backslash P_l\cup\{i\}\right) - ESD2_j^U\left(N\backslash P_l\cup j, v_{N\backslash P_l\cup j}, P\backslash P_l\cup\{j\}\right)$$

and then

$$ESD2_i^U\left(N\backslash P_l\cup i, v_{N\backslash P_l\cup i}, P\backslash P_l\cup\{i\}\right) - ESD2_j^U\left(N\backslash P_l\cup j, v_{N\backslash P_l\cup j}, P\backslash P_l\cup\{j\}\right) =$$
$$v_{N\backslash P_l\cup i}(i) + \frac{v_{N\backslash P_l\cup i}(i) - v_{N\backslash P_l\cup i}(i)}{1} +$$
$$\frac{v_{N\backslash P_l\cup i}(N\backslash P_l\cup i) - (\sum_{P_k\in P\backslash P_l\cup i} v_{N\backslash P_l\cup i}(P_k))}{m} - v_{N\backslash P_l\cup j}(j) -$$
$$\frac{v_{N\backslash P_l\cup j}(j) - v_{N\backslash P_l\cup j}(j)}{1} - \frac{v_{N\backslash P_l\cup j}(N\backslash P_l\cup j) - (\sum_{P_k\in P\backslash P_l\cup j} v_{N\backslash P_l\cup j}(P_k))}{m} =$$
$$v_{N\backslash P_l\cup i}(i) - v_{N\backslash P_l\cup j}(j) + \frac{v_{N\backslash P_l\cup i}(N\backslash P_l\cup i)}{m} - \frac{v_{N\backslash P_l\cup j}(N\backslash P_l\cup j)}{m} -$$
$$\frac{\sum_{P_k\in P\backslash P_l\cup i} v_{N\backslash P_l\cup i}(P_k)}{m} + \frac{\sum_{P_k\in P\backslash P_l\cup j} v_{N\backslash P_l\cup j}(P_k)}{m} =$$
$$v(i) - v(j) + \frac{v\left(\cup_{k\in M\backslash l} P_k\cup i\right)}{m} - \frac{v\left(\cup_{k\in M\backslash l} P_k\cup j\right)}{m} - \frac{v(i)}{m} + \frac{v(j)}{m}.$$

Finally, the uniqueness is proven in an similar way as the uniqueness in Theorem 2 of Vázquez-Brage et al. (1997). □

### 4.5.2 Coalitional equal surplus division value satisfying balanced contributions and quotient game

In this section we define a value for TU-games with a priori unions that extends the equal surplus division value and satisfies the quotient game property and balanced contributions in the unions.

**Definition 4.6.** The equal surplus division value (five) for TU-games with a priori unions $ESD5^U$ is defined by

$$\begin{aligned} ESD5_i^U(N,v,P) = &\frac{v(P_k)}{p_k} + \frac{v(N) - \sum_{l\in M} v(P_l)}{mp_k} + \\ &\sum_{\substack{T\subset P_k\\ i\in T}} \frac{P^{m,p_k,t}}{t} v(T) - \sum_{\substack{T\subset P_k\\ i\notin T}} \frac{P^{m,p_k,t}}{p_k - t} v(T) \end{aligned}$$

for all $(N,v,P) \in \mathcal{G}^U$ with $P = \{P_1, \ldots, P_m\}$ and $i \in P_k$; where $t = |T|$ and

$$P^{m,p_k,t} = \frac{1}{2} \qquad \text{if } p_k = 2 \text{ and } t = 1,$$

$$P^{m,p_k,t} = \frac{1}{p_k}\left(1 + \sum_{j=1}^{p_k-2} \frac{1}{m+j}\right) \qquad \text{if } p_k > 2 \text{ and } t = 1,$$

$$P^{m,p_k,t} = \frac{m}{(m+1)p_k} \qquad \text{if } p_k > 2 \text{ and } t = p_k - 1,$$

$$P^{m,p_k,t} = \frac{m+(z-1)}{(p_k-(z-1))(m+z)}\left(\sum_{j=0}^{z-2} \frac{p_k-j-t}{p_k-j}\right)$$

if $p_k > 3$ and $t = (p_k - z)$ such that $z \in \{2, ..., p_k - 2\}$.

**Remark 4.2.** Note that $ESD5$ can be written in terms of $ESD1$

$$ESD5_i^U(N,v,P) = ESD1_i^U(N,v,P) + \sum_{\substack{T \subset P_k \\ i \in T}} \frac{P^{m,p_k,t}}{t} v(T) - \sum_{\substack{T \subset P_k \\ i \notin T}} \frac{P^{m,p_k,t}}{p_k - t} v(T).$$

for all $(N,v,P) \in \mathcal{G}^U$ with $P = \{P_1, \ldots, P_m\}$ and $i \in P_k$.

In the last result of this work, we characterized the $ESD5$ value in the same spirit as an axiomatic characterization of the Owen value given in Vázquez-Brage et al. (1997).

**Theorem 4.8.** *$ESD5^U$ is the unique coalitional equal surplus division value satisfying QGP and BCU.*

*Proof.* Take a TU-game with a priori unions $(N,v,P) \in \mathcal{G}^U$ such that $P = \{P_1, ..., P_m\}$ and denote $M = \{1, ..., m\}$. Let us check that $ESD5^U$ satisfies QGP. For all $k \in M$, we have that

$$\sum_{i \in P_k} ESD5_i^U(N,v,P) = \sum_{i \in P_k}\left(\frac{v(P_k)}{p_k} + \frac{v(N) - \sum_{l \in M} v(P_l)}{mp_k}\right) +$$

$$\sum_{i \in P_k}\sum_{\substack{T \subset P_k \\ i \in T}} \frac{P^{m,p_k,t}}{t} v(T) - \sum_{i \in P_k}\sum_{\substack{T \subset P_k \\ i \notin T}} \frac{P^{m,p_k,t}}{p_k - t} v(T) =$$

$$v(P_k) + \frac{v(N) - \sum_{l \in M} v(P_l)}{m} + t\sum_{T \subset P_k} \frac{P^{m,p_k,t}}{t} v(T) - (p_k - t)\sum_{T \subset P_k} \frac{P^{m,p_k,t}}{p_k - t} v(T) =$$

$$v(P_k) + \frac{v(N) - \sum_{l \in M} v(P_l)}{m}.$$

It is immediate that

$$ESD5_k^U\left(M, v/P, P^m\right) = v(P_k) + \frac{v(N) - \sum_{l\in M} v(P_l)}{m}.$$

Let us check that $ESD5^U$ satisfies BCU. For all $k \in M$ and all $i, j \in P_k$,

$$\begin{aligned}
&ESD5_i^U(N,v,P) - ESD5_j^U(N,v,P) = \sum_{\substack{T\subset P_k\\ i\in T}} \frac{P^{m,p_k,t}}{t} v(T) - \\
&\sum_{\substack{T\subset P_k\\ i\notin T}} \frac{P^{m,p_k,t}}{p_k - t} v(T) - \sum_{\substack{T\subset P_k\\ j\in T}} \frac{P^{m,p_k,t}}{t} v(T) + \sum_{\substack{T\subset P_k\\ j\notin T}} \frac{P^{m,p_k,t}}{p_k - t} v(T) = \\
&\sum_{\substack{T\subseteq P_k\backslash j\\ i\in T}} \frac{P^{m,p_k,t}}{t} v(T) - \sum_{\substack{T\subseteq P_k\backslash i\\ j\in T}} \frac{P^{m,p_k,t}}{t} v(T) - \sum_{\substack{T\subset P_k\\ i\notin T\\ j\in T}} \frac{P^{m,p_k,t}}{p_k - t} v(T) + \\
&+ \sum_{\substack{T\subset P_k\\ j\notin T\\ i\in T}} \frac{P^{m,p_k,t}}{p_k - t} v(T) = p_k \sum_{\substack{T\subseteq P_k\backslash j\\ i\in T}} \frac{P^{m,p_k,t}}{(p_k - t)t} v(T) - \sum_{\substack{T\subseteq P_k\backslash i\\ j\in T}} \frac{P^{m,p_k,t}}{(p_k - t)t} v(T).
\end{aligned}$$

On the other hand,

$$\begin{aligned}
&ESD5_i^U\left(N, v, P_{-j}\right) - ESD5_j^U\left(N, v, P_{-i}\right) = \frac{v(P_k\backslash j)}{p_k - 1} - \frac{v(P_k\backslash i)}{p_k - 1} + \\
&\frac{v(N) - \sum_{P_l\in P_{-j}} v(P_l)}{(m+1)(p_k - 1)} - \frac{v(N) - \sum_{P_l\in P_{-i}} v(P_l)}{(m+1)(p_k - 1)} + \sum_{\substack{T\subset P_k\backslash j\\ i\in T}} \frac{P^{m+1,p_k-1,t}}{t} v(T) - \\
&\sum_{\substack{T\subset P_k\backslash j\\ i\notin T}} \frac{P^{m+1,p_k-1,t}}{p_k - t} v(T) - \sum_{\substack{T\subset P_k\backslash i\\ j\in T}} \frac{P^{m+1,p_k-1,t}}{t} v(T) + \sum_{\substack{T\subset P_k\backslash i\\ j\notin T}} \frac{P^{m+1,p_k-1,t}}{p_k - t} v(T) = \\
&\frac{v(P_k\backslash j)}{p_k - 1} - \frac{v(P_k\backslash i)}{p_k - 1} - \frac{v(P_k\backslash j) + v(j)}{(m+1)(p_k - 1)} + \frac{v(P_k\backslash i) + v(i)}{(m+1)(p_k - 1)} + \sum_{\substack{T\subset P_k\backslash j\\ i\in T}} \frac{P^{m+1,p_k-1,t}}{t} v(T) \\
&- \sum_{\substack{T\subset P_k\backslash j\\ i\notin T}} \frac{P^{m+1,p_k-1,t}}{p_k - t} v(T) - \sum_{\substack{T\subset P_k\backslash i\\ j\in T}} \frac{P^{m+1,p_k-1,t}}{t} v(T) + \sum_{\substack{T\subset P_k\backslash i\\ j\notin T}} \frac{P^{m+1,p_k-1,t}}{p_k - t} v(T) = \\
&\frac{m\cdot v(P_k\backslash j)}{(m+1)(p_k - 1)} - \frac{m\cdot v(P_k\backslash i)}{(m+1)(p_k - 1)} + \frac{v(i)}{(m+1)(p_k - 1)} - \frac{v(j)}{(m+1)(p_k - 1)} +
\end{aligned}$$

$$\sum_{\substack{T\subset P_k\backslash j\\ i\in T}} \frac{P^{m+1,p_k-1,t}}{t} v\left(T\right) - \sum_{\substack{T\subset P_k\backslash j\\ i\notin T}} \frac{P^{m+1,p_k-1,t}}{p_k-t} v\left(T\right) - \sum_{\substack{T\subset P_k\backslash i\\ j\in T}} \frac{P^{m+1,p_k-1,t}}{t} v\left(T\right) +$$

$$\sum_{\substack{T\subset P_k\backslash i\\ j\notin T}} \frac{P^{m+1,p_k-1,t}}{p_k-t} v\left(T\right) = \frac{m\cdot v(P_k\backslash j)}{(m+1)(p_k-1)} - \frac{m\cdot v(P_k\backslash i)}{(m+1)(p_k-1)} + \frac{v(i)}{(m+1)(p_k-1)} -$$

$$\frac{v(j)}{(m+1)(p_k-1)} + \sum_{\substack{T\subset P_k\backslash j\\ i\in T}} \frac{P^{m+1,p_k-1,t}}{t} v\left(T\right) - \sum_{\substack{T\subset P_k\backslash i\\ j\in T}} \frac{P^{m+1,p_k-1,t}}{t} v\left(T\right).$$

Let us see that the two equations are the same. We only need to see that for a player $i$ and any coalition $T\subset P_k\backslash j$ such that $i\in T$, the weights coincide. Consider the all different cases. Noticed that as $i,j\in P_k$, then $p_k\geq 2$.

**Case i)** $p_k=2$ then

$$p_k \sum_{\substack{T\subseteq P_k\backslash j\\ i\in T}} \frac{P^{m,p_k,t}}{(p_k-t)t} v\left(T\right) = \frac{p_k}{2(p_k-1)} v(i) = v(i)$$

and on the other side

$$\frac{m\cdot v(P_k\backslash j)}{(m+1)(p_k-1)} + \frac{v(i)}{(m+1)(p_k-1)} = \frac{(m+1)v(i)}{(m+1)(p_k-1)} = v(i).$$

**Case ii)** $p_k>2$ and $|T|=1$ ($T=i$) then

$$\frac{p_k}{(p_k-t)t} P^{m,p_k,t} v\left(i\right) = \frac{p_k}{(p_k-1)} P^{m,p_k,1} v\left(i\right) = \frac{p_k}{(p_k-1)} \frac{1}{p_k} \left(1+\sum_{j=1}^{p_k-2} \frac{1}{m+j}\right) v(i)$$

and on the other side

$$\frac{v(i)}{(m+1)(p_k-1)} + \frac{P^{m+1,p_k-1,t}}{t} v\left(i\right) = \frac{v(i)}{(m+1)(p_k-1)} +$$

$$\frac{1}{p_k-1}\left(1+\sum_{j=1}^{p_k-3}\frac{1}{m+1+j}\right)v(i) = \frac{1}{p_k-1}\left(1+\sum_{j=0}^{p_k-3}\frac{1}{m+1+j}\right)v(i) =$$

$$\frac{1}{p_k-1}\left(1+\sum_{j=1}^{p_k-2}\frac{1}{m+j}\right)v(i).$$

**Case ii)** $p_k > 2$ and $|T| = p_k - 1$ ($T = P_k \backslash j$) then

$$p_k \frac{P^{m,p_k,t}}{(p_k - t)t} v\left(P_k \backslash j\right) = \frac{p_k}{p_k - 1} \frac{m}{(m+1)p_k} v\left(P_k \backslash j\right) = \frac{m}{(m+1)(p_k - 1)} v\left(P_k \backslash j\right)$$

and on the other side

$$\frac{m \cdot v(P_k \backslash j)}{(m+1)(p_k - 1)}.$$

**Case iii)** $p_k > 2$ and $|T| = p_k - 2$ then

$$p_k \frac{P^{m,p_k,t}}{2(p_k - 2)} v\left(T\right) = \frac{p_k}{2(p_k - 2)} \frac{m+1}{m+2} \frac{1}{p_k - 1} \frac{2}{p_k} v\left(T\right) = \frac{1}{(p_k - 2)} \frac{m+1}{m+2} \frac{1}{p_k - 1} v\left(T\right)$$

and on the other side

$$\frac{P^{m+1,p_k-1,t}}{t} v\left(T\right) = \frac{1}{p_k - 2} \frac{m+1}{m+2} \frac{1}{p_k - 1} v\left(T\right).$$

**Case iv)** $p_k > 2$ and $|T| = p_k - z$ where $z \in \{3, ..., p_k - 2\}$

$$p_k \frac{P^{m,p_k,t}}{(p_k - t)t} v\left(T\right) = \frac{p_k}{z(p_k - z)} \frac{m + (z-1)}{(p_k - (z-1))(m+z)} \left( \sum_{l=0}^{z-2} \frac{p_k - l - t}{p_k - l} \right) v\left(T\right)$$

$$= \frac{1}{(p_k - z)} \frac{m + (z-1)}{(p_k - (z-1))(m+z)} \left( \sum_{l=1}^{z-2} \frac{p_k - l - t}{p_k - l} \right) v\left(T\right)$$

and on the other side

$$\frac{P^{m+1,p_k-1,t}}{t} v\left(T\right) = \frac{1}{p_k - z} \frac{m + 1 + (z' - 1)}{(p_k - 1 - (z' - 1))(m + 1 + z')} \left( \sum_{j=0}^{z'-2} \frac{p_k - 1 - j - t}{p_k - 1 - j} \right)$$

where $z' = z - 1$ because $P^{m+1,p_k-1,t}$ depends on $P_k \backslash j$.

Finally, the uniqueness is proven in an analogous way as the uniqueness in Theorem 2 of Vázquez-Brage et al. (1997). □

## 4.6 Example

In this section, we apply the extensions of the egalitarian values to an example. In this way, we can check the behaviour of these solutions when a real problem is faced. In addition, we can compare these solutions to each other.

This problem is motivated by a building where an elevator wants to be installed. The building has three floors, where on the first floor there is one apartment, the second floor has two apartments, and the third, and last, floor there are 3 apartments. However, access to an apartment on the second and third floor is in a parallel corridor to the other apartments.

The cost of installing the elevator is as follows: 50 (in thousands of euros) the cost of the machine, 40 to the works to make the hollow, that is a fixed cost of 10 and an individual cost of 10 for the owners of the first floor that is incremented by 10 for the owners of the second floor and by another 10 for the owners of the third floor. Finally, 10 on each floor to allow access to the elevator (total 30). The total costs are 120 thousands of euros, but there is an increment if the apartments in the parallel corridor want direct access to the elevator. That is, an increment of 10 in the elevator to buy it with double door; and an increment of 5 to the second and third floor to allow access to the elevator. However, if both apartments decide to allow access to the elevator, 2 will be discounted in each one. Therefore the total costs are 136. According to this, the individual costs are

- 60 (machine with double door) + 15 (floor) + 40 (hollow) = 115, for the first player of the third floor,
- 50 (machine) + 10 (floor) + 40 (hollow) = 100, for the second and third players of the third floor,
- 60 (machine with double door) + 15 (floor) + 30 (hollow) = 105, for the first player of the second floor,
- 50 (machine) + 10 (floor) + 30 (hollow) = 90, for the second player of the second floor,
- 50 (machine) + 10 (floor) + 20 (hollow) = 80, for the players of the first floor.

| | $ED^U$ | | |
|---|---|---|---|
| 3rd floor | 15.11 | 15.11 | 15.11 |
| 2nd floor | 22.67 | 22.67 | |
| 1st floor | 45.33 | | |

Table 4.1: Distribution according to $ED^U$.

Table 4.1 displays the distribution proposed by $ED^U$, Table 4.2 displays the distribution proposed by $ESD1^U$, $ESD2^U$ and $ESD3^U$, and Table 4.3 the distribution of $ESD4^U$ and $ESD5^U$. Since $ESD3^U$ does not satisfy the QGP property,

| | $ESD1^U$ | $ESD2^U$ | $ESD3^U$ |
|---|---|---|---|
| 3rd floor | 20.11 20.11 20.11 | 30.11 15.11 15.11 | 64.56 49.56 49.56 |
| 2nd floor | 25.17 25.17 | 32.67 17.67 | 29.33 14.33 |
| 1st floor | 25.33 | 25.33 | -71.33 |

Table 4.2: Distribution according to $ESD1^U$, $ESD2^U$, $ESD3^U$.

| | $ESD4^U$ | $ESD5^U$ |
|---|---|---|
| 3rd floor | 27.00 16.67 16.67 | 30.11 15.11 15.11 |
| 2nd floor | 30.33 20.00 | 32.67 17.67 |
| 1st floor | 25.33 | 25.33 |

Table 4.3: Distribution according to $ESD4^U$, $ESD5^U$.

the results of this solution are not reasonable. $ESD1^U$ allocates the same value for the apartments on the same floor, and $ESD2^U$ takes into account the individual costs of the apartments to do the distribution. However, $ESD2^U$ does not take into account the relationships between unions and doest not consider the discounts. $ESD4^U$ makes up for this shortfall. Finally, $ESD5^U$ is equal to $ESD2^U$. This is because when $|P_k| = 1$ then $ESD1^U = ESD2^U = ESD4^U = ESD5^U$; if $|P_k| = 2$ then $ESD2^U = ESD5^U$; and for $|P_k| > 2$ there are differences between these two values, but the fact of the symmetry between the apartments causes the equality of these two values.

# References

Alonso-Meijide JM, Costa J, García-Jurado I, Gonçalves-Dosantos JC (2020). On egalitarian values for cooperative games with a priori unions. TOP. DOI: 10.1007/s11750-020-00553-2.

Banzhaf III, JF (1964). Weighted voting doesn't work: A mathematical analysis. Rutgers L. Rev., 19, 317.

Béal S, Rémila E, Solal P (2019). Coalitional desirability and the equal division value. Theory and Decision 86, 95-106.

Casajus A, Hüttner F (2014). Null, nullifying, or dummifying players: The difference between the Shapley value, the equal division value, and the equal surplus division value. Economics Letters 122, 167-169.

Chun Y, Park B (2012). Population solidarity, population fair-ranking and the egalitarian value. International Journal of Game Theory 41, 255-270.

Costa J (2016). A polynomial expression of the Owen value in the maintenance cost game. Optimization 65, 797-809.

Driessen TSH, Funaki Y (1991). Coincidence of and collinearity between game theoretic solutions. OR Spectrum 13, 15-30.

Ferrières S (2017). Nullified equal loss property and equal division values. Theory and Decision 83, 385-406.

Lorenzo-Freire S (2016). On new characterizations of the Owen value. Operations Research Letters 44, 491-494.

Owen G (1977) Values of games with a priori unions. In: Mathematical Economics and Game Theory (R Henn, O Moeschlin, eds.), Springer, 76-88.

Owen G (1981) Modification of the Banzhaf-Coleman index for games with a priori unions. In: Power, voting, and voting power. Physica, Heidelberg, 232-238.

Saavedra-Nieves A, García-Jurado I, Fiestras-Janeiro G (2018). Estimation of the Owen value based on sampling. In: The Mathematics of the Uncertain: A Tribute to Pedro Gil (E Gil, E Gil, J Gil, MA Gil, eds.), Springer, 347-356.

Shapley LS (1953). A value for n-person games. In: Contributions to the Theory of Games II (HW Kuhn, AW Tucker, eds.), Princeton University Press, 307-317.

van den Brink R (2007). Null or nullifying players: the difference between the Shapley value and equal division solutions. Journal of Economic Theory 136, 767-775.

van den Brink R, Funaki Y (2009) Axiomatizations of a class of equal surplus sharing solutions for TU-games. Theory and Decision 67, 303-340.

van den Brink R, Chun Y, Funaki Y, Park B (2016). Consistency, population solidarity, and egalitarian solutions for TU-games. Theory and Decision 81, 427-447.

Vázquez-Brage M, van den Nouweland A, García-Jurado I (1997). Owen's coalitional value and aircraft landing fees. Mathematical Social Sciences 34, 273-286.

# Chapter 5

# Necessary players and values

JC Gonçalves-Dosantos, I García-Jurado, J Costa, JM Alonso-Meijide

In second revision.

## 5.1 Introduction

The Shapley value, introduced in Shapley (1953), is a rule for distributing the benefits that a set of agents $N$ can generate, taking into account the marginal contributions of each agent to all possible coalitions of which that agent may be part, and the sizes of such coalitions. The Shapley value is one of the most important solutions of cooperative game theory and it has many applications in a wide variety of fields. For instance, in recent years the Shapley value has been applied to cancer research (see Albino et al., 2008), to machine learning (see Strumbelj and Kononenko, 2010), to data envelopment analysis (see Yang and Zhang, 2015), to image classification (see Gurram et al., 2016), to project management (see Bergantiños et al., 2018, and Gonçalves-Dosantos et al., 2020), etc. Moretti and Patrone (2008) is a survey explaining the transversality of the Shapley value.

The game theory literature provides many alternatives to the Shapley value, such as the nucleolus (Schmeidler, 1969), the Banzhaf value (Owen, 1975), the $\tau$-value (Tijs, 1981), the equal-surplus division value (Driessen and Funaki, 1991) or, more recently, the consensus value (Ju et al., 2007) and the ie-Banzhaf value (Alonso-Meijide et al., 2019b). All those alternative values have appealing properties and could be used instead of the Shapley value. In order to decide what is the most appropriate value for a particular problem it is helpful to know the properties that are essentially connected to each value. This is why game theory is interested in the so-called characterizations: to characterize a value in a class of games is to find a set of properties so that it is the only value that fulfills them in that class. For instance, in Luchetti et al. (2010), two relevance indexes for genes are compared, one based on the Shapley value and the other based on the Banzhaf value, and for that purpose they are characterized in the corresponding class of games, the so-called microarray games.

In this article we introduce a value for cooperative games that results from proposing a new property for so-called necessary players that, in a way, corrects the properties for such players met by the Shapley and Banzhaf values. Informally, necessary players are those without whom the characteristic function of the game would be zero. These players have attracted the attention of game theorists for axiomatic studies in the last years. For instance, Alonso-Meijide et al. (2019a) and Béal and Navarro (2020) are two recent papers dealing with necessary players and characterizations. Apart from introducing a new value, in this paper we provide an axiomatic characterization of it, which allows to compare the new value with other solution concepts for cooperative games. Furthermore, we extend and characterize the new value for cooperative games with a coalition structure. A cooperative game with a coalition structure models those situations where the agents in a set $N$ aim to distribute the benefits they generate taking into account the contributions of each agent and of each possible subset of $N$, as well as a coalition structure (a partition of $N$) that conditions the distribution, in the sense that distribution among the classes of the partition is made first and, then, a distribution within those classes is performed. Cooperative games with a coalition structure have been applied in several fields like political analysis (see, for instance, Carreras and Puente, 2015), infrastructure management (see Costa, 2016), cost allocation (see Fragnelli and Iandolino, 2004), etc. The Owen value (Owen, 1977) and the Banzhaf-Owen value (Owen, 1982) are, respectively, the variations of the Shapley value and the Banzhaf value for cooperative games with a coalition structure. In this paper we also provide new characterizations of the Owen and the Banzhaf-Owen values using properties involving necessary players.

The structure of this paper is as follows. In Section 5.2 we introduce the $\Gamma$ value, a new value for cooperative games. We also provide an axiomatic characterization of the $\Gamma$ value and illustrate its behaviour in a practical example that arises in a problem of sharing the costs of installing an elevator. In Section 5.3 we provide new characterizations of the Owen and Banzhaf-Owen values and introduce and characterize an extension of the $\Gamma$ value for cooperative games with a coalitional structure. We finish the paper with a section of concluding remarks.

## 5.2 Values and necessary players

A cooperative game is a pair $(N, v)$ given by a finite set of players $N$ and a characteristic function $v : 2^N \to \mathbb{R}$, that assigns to each coalition $S \subseteq N$ a real number $v(S)$ that indicates the benefits that coalition $S$ is able to generate; by definition $v(\emptyset) = 0$. We denote by $\mathcal{G}_N$ the family of all cooperative games with player set $N$.

A *value* for cooperative games is a map $f$ that assigns to every game $(N, v) \in \mathcal{G}_N$ a vector $f(N, v) \in \mathbb{R}^N$. Two of the most important values for cooperative games are the Shapley value (Shapley, 1953) and the Banzhaf value (Owen, 1975). A number of characterizations of these two values can be found on the literature. For example, Alonso-Meijide et al. (2019a) provides characterizations of those values using only three properties for each of them: two common properties and one extra property concerning the so-called *necessary players* that differs for the Shapley and the Banzhaf values. In this paper we concentrate on characterizations of values involving necessary players. Let us first remember the formal definition of a necessary player.

**Definition 5.1.** A player $i \in N$ is said to be necessary in the cooperative game $(N, v)$ if $v(S) = 0$ for all $S \subseteq N \setminus \{i\}$.

In words, a necessary player is one without whom cooperation does not produce any results. In fact, notice that if $i$ is necessary in $(N, v)$, then $v(S) = \sum_{j \in S} v(\{j\}) = 0$ for all $S \subseteq N \setminus \{i\}$; hence, the game resulting after the elimination of $i$ is additive and null. Necessary players often arise in real situations. Take, for instance, the following example.

**Example 5.1.** Consider a council formed by three entities with 24, 15 and 9 votes, respectively. Any proposal must receive at least 25 votes to be approved. In the resulting voting game, it is easy to see that the entity with 24 votes is a necessary player because, without it, the other two entities cannot get any proposals approved.

In some specific problems, as in the example above, the necessary players arise in a natural way and, therefore, a characterization based on such players can be relevant in deciding what value to use in those problems. We start by remembering the characterizations of the Shapley and Banzhaf values in Alonso-Meijide et al. (2019a) and some other preliminary material. The Shapley value $\varphi$ is defined as

$$\varphi_i\left(N, v\right) = \frac{1}{n} \sum_{S \subseteq N \setminus \{i\}} \frac{1}{\binom{n-1}{s}} \left(v\left(S \cup \{i\}\right) - v\left(S\right)\right)$$

for all $(N, v) \in \mathcal{G}_N$ and all $i \in N$; $n$ and $s$ denote the cardinalities of $N$ and $S$, respectively. The Banzhaf value $\beta$ is defined as

$$\beta_i\left(N, v\right) = \frac{1}{2^{n-1}} \sum_{S \subseteq N \setminus \{i\}} \left(v\left(S \cup \{i\}\right) - v\left(S\right)\right)$$

for all $(N, v) \in \mathcal{G}_N$ and all $i \in N$. Both the Shapley and the Banzhaf value are additive. This means that they satisfy the following condition.

**Additivity.** A value for cooperative games $f$ satisfies the property of additivity if for each pair of cooperative games $(N, v), (N, w)$ it holds that

$$f(N, v + w) = f(N, v) + f(N, w).$$

Additivity is a good property because, at the same time that it is natural and easily interpretable, it greatly facilitates the mathematical analysis of the values that comply with it and the calculation of such values; for instance, Benati et al. (2019) provides a method to approximate additive values in cooperative games that is useful when the number of players is large.

Another reasonable property that is satisfied by the Shapley and Banzhaf value concerns null players. Remember that a null player of $(N, v)$ is an $i \in N$ such that $v(S) = v(S \cup \{i\})$ for all $S \subseteq N \setminus \{i\}$.

**Null Player.** A value for cooperative games $f$ satisfies the property of null player if for each cooperative game $(N, v)$ and for each $i \in N$ null player of $(N, v)$, it holds that $f_i(N, v) = 0$.

Now let us see two alternative properties for necessary players introduced in Alonso-Meijide et al. (2019a) and the main result concerning them.

**Necessary Players Get the Weighted Mean.** A value for cooperative games $f$ satisfies the property of necessary players get the weighted mean if, for each cooperative game $(N, v)$ and for each $i \in N$ necessary player in $(N, v)$, it holds that

$$f_i(N, v) = \frac{1}{n} \sum_{S \subseteq N, i \in S} \frac{1}{\binom{n-1}{s-1}} v(S).$$

**Necessary Players Get the Mean.** A value for cooperative games $f$ satisfies the property of necessary players get the mean if, for each cooperative game $(N, v)$ and for each $i \in N$ necessary player in $(N, v)$, it holds that

$$f_i(N, v) = \frac{1}{2^{n-1}} \sum_{S \subseteq N, i \in S} v(S).$$

Observe that the two properties above are similar. Both establish that a necessary player must receive the average of the values of the coalitions to which that player belongs, although the former takes into account the size of such coalitions and the latter does not.

**Theorem 5.1.** *(Alonso-Meijide et al., 2019a).*
*1. The Shapley value is the unique value for cooperative games that satisfies the properties of additivity, null player and necessary players get the weighted mean.*
*2. The Banzhaf value is the unique value for cooperative games that satisfies the properties of additivity, null player and necessary players get the mean.*

Now, we remember two widely known properties for values that will be relevant in the subsequent discussion.

**Efficiency.** A value for cooperative games $f$ satisfies the property of efficiency if for each cooperative game $(N, v)$, it holds that

$$\sum_{i \in N} f_i (N, v) = v (N) .$$

We say that players $i, j \in N$ are symmetric in $(N, v) \in \mathcal{G}_N$ if $v(S \cup \{i\}) = v(S \cup \{j\})$ for every $S \subseteq N \setminus \{i, j\}$.

**Symmetry.** A value for cooperative games $f$ satisfies the property of symmetry if for each cooperative game $(N, v)$ and for all $i, j \in N$ symmetric players in $(N, v)$, it holds that

$$f_i (N, v) = f_j (N, v) .$$

It is well-known that the Shapley and Banzhaf values satisfy the symmetry property. However, only the Shapley value is efficient. In some problems, efficiency is not an essential property for a value, see for example microarray games in Lucchetti et al. (2010). In many cases, however, efficiency will be required for a value to make sense; this happens, for example, when we are faced with cost allocation problems. One question we can ask is whether there is a value that fulfills the necessary players get the mean property and the efficiency property. The answer is negative because those properties are incompatible. Indeed, assume that a value for cooperative games $f$ satisfies both properties and for every non-empty $S \subseteq N$ denote by $(N, e_S)$ the cooperative game in $\mathcal{G}_N$ given, for every $T \subseteq N$, by:

$$e_S(T) = \begin{cases} 1 & \text{if } T = S, \\ 0 & \text{otherwise.} \end{cases} \tag{5.1}$$

Since $f$ satisfies efficiency, it holds that

$$\sum_{i \in N} f_i (N, e_N) = 1. \tag{5.2}$$

Notice now that every $i \in N$ is necessary in $(N, e_N)$ and then, since $f$ satisfies the necessary players get the mean property, it holds that

$$\sum_{i \in N} f_i (N, e_N) = \sum_{i \in N} \frac{1}{2^{n-1}} = \frac{n}{2^{n-1}}. \tag{5.3}$$

Observe that (5.2) and (5.3) are incompatible for $n > 2$, which implies that necessary players get the mean and efficiency are incompatible properties. Such incompatibility vanishes when we consider the next weak version of the former property.

**(Weak) Necessary Players Get the Mean.** A value for cooperative games $f$ satisfies the (weak) necessary players get the mean property if, for each cooperative game $(N, v)$ with $v(N) = 0$ and for each $i \in N$ necessary player in $(N, v)$, it holds that

$$f_i\,(N,v) = \frac{1}{2^{n-1}} \sum_{S \subseteq N, i \in S} v(S).$$

With this new property we can prove the following theorem.

**Theorem 5.2.** *There exists a unique value for cooperative games that satisfies the properties of additivity, (weak) necessary players get the mean, efficiency and symmetry. This value that we denote by $G$ is given, for all $(N, v) \in \mathcal{G}_N$ and all $i \in N$, by:*

$$G_i\,(N,v) = \frac{1}{2^{n-1}} \left( \sum_{S \subset N, i \in S} v(S) - \sum_{S \subset N, i \notin S} \frac{s}{n-s} v(S) \right) + \frac{v(N)}{n}. \tag{5.4}$$

*Proof.* (Existence). It is clear that $G$ satisfies additivity. To check that it satisfies the (weak) necessary players get the mean property, take a cooperative game $(N, v)$ with $v(N) = 0$ and such that $i \in N$ is a necessary player in $(N, v)$. Then expression (5.4) reduces to

$$G_i(N,v) = \frac{1}{2^{n-1}} \sum_{S \subset N, i \in S} v(S) = \frac{1}{2^{n-1}} \sum_{S \subseteq N, i \in S} v(S).$$

To check that $G$ satisfies efficiency notice that, for every cooperative game $(N, v)$,

$$\begin{aligned}
\sum_{i \in N} G_i(N,v) &= \frac{1}{2^{n-1}} \sum_{i \in N} \left( \sum_{S \subset N, i \in S} v(S) - \sum_{S \subset N, i \notin S} \frac{s}{n-s} v(S) \right) + v(N) \\
&= \frac{1}{2^{n-1}} \left( \sum_{S \subset N} s v(S) - \sum_{S \subset N} (n-s) \frac{s}{n-s} v(S) \right) + v(N) \\
&= v(N).
\end{aligned}$$

To check that $G$ satisfies symmetry take a cooperative game $(N, v)$ and a pair of symmetric players in $(N, v)$ $i, j \in N$. Notice that

$$\begin{aligned}
\sum_{S \subset N, i \in S} v(S) - \sum_{S \subset N, i \notin S} \frac{s}{n-s} v(S) &= \sum_{S \subseteq N \setminus \{i,j\}} (v(S \cup \{i\})) + \sum_{S \subset N \setminus \{i,j\}} (v(S \cup \{i,j\})) \\
&\quad - \sum_{S \subseteq N \setminus \{i,j\}} \left( \frac{s}{n-s} v(S) + \frac{s+1}{n-s-1} v(S \cup \{j\}) \right).
\end{aligned}$$

Now, since $i, j$ are symmetric in $(N, v)$, the last expression is equal to

$$\sum_{S\subseteq N\setminus\{i,j\}} (v(S\cup\{j\})) + \sum_{S\subset N\setminus\{i,j\}} (v(S\cup\{i,j\})) - \sum_{S\subseteq N\setminus\{i,j\}} \left(\frac{s}{n-s}v(S) + \frac{s+1}{n-s-1}v(S\cup\{i\})\right)$$

and then it is clear that $G_i(N, v) = G_j(N, v)$.

(Uniqueness). Take $f$, a value for cooperative games that satisfies efficiency, symmetry, (weak) necessary players get the mean and additivity and take a cooperative game $(N, v)$. We prove now that $f(N, v) = G(N, v)$. Indeed, consider the canonical basis of the vector space of characteristic functions of cooperative games with set of players $N$: $\{e_S\}_{S\in 2^N\setminus\emptyset}$ (see expression (5.1)). Observe that $v$ can be written in a unique way as a linear combination of the elements of the canonical basis: $v = \sum_{S\in 2^N\setminus\emptyset} v(S)e_S$. Since $f$ satisfies additivity,

$$f(N, v) = \sum_{S\in 2^N\setminus\emptyset} f(N, v(S)e_S).$$

Note that efficiency, symmetry and (weak) necessary players get the mean characterize a unique value in the class $\{(N, v(S)e_S) \mid S \subset N, S \neq \emptyset\}$. Besides, efficiency and symmetry characterize a unique value for $(N, v(N)e_N)$. Hence $f(N, v) = G(N, v)$. $\square$

Surprisingly enough, the new value $G$ introduced in Proposition 5.2 looks a lot like the e-Banzhaf value defined in Alonso-Meijide et al. (2019b) but it is not the same, because $\frac{n-s}{s}$ is changed by $\frac{s}{n-s}$ and, moreover, those two parameters do not multiply the same summands in the expressions of $G$ and of the e-Banzhaf value. Indeed, such parameters do not seem to have a clear interpretation from the point of view of fairness, which leads us to think that perhaps the (weak) necessary players get the mean property should be reformulated. In fact, it is more reasonable to ask that a necessary player be entitled to the average of the per capita values of the coalitions that contain it rather than the average of the values of those coalitions; in fact, such players are necessary for coalitions to have a value other than zero, but they require the other coalition members to generate such a value. Thus we propose the new property formulated below.

**Necessary Players Get the Per Capita Mean.** A value for cooperative games $f$ satisfies the necessary players get the per capita mean property if, for each cooperative game $(N, v)$ with $v(N) = 0$ and for each $i \in N$ necessary player in $(N, v)$, it holds that

$$f_i(N, v) = \frac{1}{2^{n-1}} \sum_{S\subseteq N, i\in S} \frac{v(S)}{s}.$$

The next result introduces and characterizes a new value for cooperative games.

**Theorem 5.3.** *There exists a unique value for cooperative games that satisfies the properties of additivity, necessary players get the per capita mean, efficiency and symmetry. This value that we denote by* $\gamma$ *is given, for all* $(N, v) \in \mathcal{G}_N$ *and all* $i \in N$*, by:*

$$\gamma_i(N, v) = \frac{1}{2^{n-1}} \left( \sum_{S \subset N, i \in S} \frac{v(S)}{s} - \sum_{S \subset N, i \notin S} \frac{v(S)}{n-s} \right) + \frac{v(N)}{n}. \tag{5.5}$$

*Proof.* (Existence). It is clear that $\gamma$ satisfies additivity. To check that it satisfies the necessary players get the per capita mean property take a cooperative game $(N, v)$ with $v(N) = 0$ and such that $i \in N$ is a necessary player in $(N, v)$. Then expression (5.5) reduces to

$$\gamma_i(N, v) = \frac{1}{2^{n-1}} \sum_{S \subset N, i \in S} \frac{v(S)}{s} = \frac{1}{2^{n-1}} \sum_{S \subseteq N, i \in S} \frac{v(S)}{s}.$$

To check that $\gamma$ satisfies efficiency notice that, for every cooperative game $(N, v)$,

$$\begin{aligned} \sum_{i \in N} \gamma_i(N, v) &= \frac{1}{2^{n-1}} \sum_{i \in N} \left( \sum_{S \subset N, i \in S} \frac{v(S)}{s} - \sum_{S \subset N, i \notin S} \frac{v(S)}{n-s} \right) + v(N) \\ &= \frac{1}{2^{n-1}} \left( \sum_{S \subset N} s \frac{v(S)}{s} - \sum_{S \subset N} (n-s) \frac{v(S)}{n-s} \right) + v(N) \\ &= v(N). \end{aligned}$$

To check that $\gamma$ satisfies symmetry take a cooperative game $(N, v)$ and a pair of symmetric players in $(N, v)$ $i, j \in N$. Notice that

$$\sum_{S \subset N, i \in S} \frac{v(S)}{s} - \sum_{S \subset N, i \notin S} \frac{v(S)}{n-s} =$$

$$\sum_{S \subseteq N \backslash \{i,j\}} \frac{v(S \cup \{i\})}{s+1} + \sum_{S \subset N \backslash \{i,j\}} \frac{v(S \cup \{i,j\})}{s+2} - \sum_{S \subseteq N \backslash \{i,j\}} \left( \frac{v(S)}{n-s} + \frac{v(S \cup \{j\})}{n-s-1} \right).$$

Now, since $i, j$ are symmetric in $(N, v)$, the last expression is equal to

$$\sum_{S \subseteq N \backslash \{i,j\}} \frac{v(S \cup \{j\})}{s+1} + \sum_{S \subset N \backslash \{i,j\}} \frac{v(S \cup \{i,j\})}{s+2} - \sum_{S \subseteq N \backslash \{i,j\}} \left( \frac{v(S)}{n-s} + \frac{v(S \cup \{i\})}{n-s-1} \right)$$

and then it is clear that $\gamma_i(N, v) = \gamma_j(N, v)$.

(Uniqueness). Take $f$, a value for cooperative games that satisfies efficiency, symmetry, necessary players get the per capita mean and additivity and take a cooperative game $(N, v)$. We prove now that $f(N, v) = \gamma(N, v)$. Indeed, consider the basis of the vector space of characteristic functions of cooperative games with set of players $N$: $\{e_S\}_{S\in 2^N\setminus\emptyset}$ (see expression (5.1)). Observe that $v$ can be written in a unique way as a linear combination of the elements of the basis: $v = \sum_{S\in 2^N\setminus\emptyset} v(S)e_S$. Since $f$ satisfies additivity,

$$f(N, v) = \sum_{S\in 2^N\setminus\emptyset} f(N, v(S)e_S).$$

Notice that efficiency, symmetry and necessary players get the per capita mean characterize a unique value in the class of games $\{(N, v(S)e_S) \mid S \subset N, S \neq \emptyset\}$. Besides, efficiency and symmetry characterize a unique value for $(N, v(N)e_N)$. Hence $f(N, v) = \gamma(N, v)$. □

A very desirable property for values for cooperative games is the invariance to S-equivalence, which we remember below. Two cooperative games with the same sets of players $(N, v)$ and $(N, w)$ are said to be $S$-equivalent if there exist $a \in \mathbb{R}$ with $a > 0$ and $b \in \mathbb{R}^N$ such that, for every $T \subseteq N$, it holds that

$$w(T) = av(T) + \sum_{j\in T} b_j.$$

When $(N, v)$ and $(N, w)$ are $S$-equivalent we can transform $v$ into $w$ simply by changing the scale and translating the players' utilities. In these conditions it seems reasonable to ask a value for cooperative games $f$ that $f(N, v)$ is transformed into $f(N, w)$ by doing the corresponding change of scale and translations.

**Invariance to $S$-equivalence (INV).** A value for cooperative games $f$ satisfies invariance to S-equivalence if for each pair of S-equivalent cooperative games $(N, v)$ and $(N, w)$ such that $w(T) = av(T) + \sum_{j\in T} b_j$ for all $T \subseteq N$ (with $a \in \mathbb{R}$, $a > 0$ and $b \in \mathbb{R}^N$) it holds that, for every $i \in N$,

$$f_i(N, w) = af_i(N, v) + b_i.$$

Unfortunately, the value $\gamma$ defined by (5.5) is not invariant to $S$-equivalence. Then, we make an adjustment of $\gamma$ that leads us to the $\Gamma$ value for cooperative games that we define below.

**Definition 5.2.** The $\Gamma$ value for cooperative games is given for every $(N, v) \in \mathcal{G}_N$ and every $i \in N$ by:

$$\Gamma_i(N, v) = v(\{i\}) + \gamma_i(N, v^0), \tag{5.6}$$

where $v^0(S) = v(S) - \sum_{j\in S} v(\{j\})$ for all $S \subseteq N$.

It is easy to check that $\Gamma$ satisfies the invariance to $S$-equivalence. In order to characterize it, we introduce below a new property concerning the necessary players.

**Necessary Players Get the 0-Normalized Per Capita Mean.** A value for cooperative games $f$ satisfies the necessary players get the 0-normalized per capita mean property if, for each cooperative game $(N, v)$ with $v(N) = \sum_{j\in N} v(\{j\})$ and for each $i \in N$ necessary player in $(N, v)$, it holds that

$$f_i\,(N, v) = v(\{i\}) + \frac{1}{2^{n-1}} \sum_{S\subseteq N, i\in S} \frac{v^0(S)}{s}.$$

**Theorem 5.4.** $\Gamma$ *is the unique value for cooperative games that satisfies the properties of additivity, necessary players get the* 0*-normalized per capita mean, efficiency and symmetry.*

*Proof.* (Existence). Since $\gamma$ satisfies additivity, efficiency and symmetry, it is clear that $\Gamma$ also satisfies those properties. To check that it fulfils the necessary players get the 0-normalized per capita mean property take a cooperative game $(N, v)$ with $v(N) = \sum_{j\in N} v(\{j\})$ and such that $i \in N$ is a necessary player in $(N, v)$. Then expression (5.6) reduces to

$$\Gamma_i(N, v) = v(\{i\}) + \frac{1}{2^{n-1}} \sum_{S\subset N, i\in S} \frac{v^0(S)}{s} = v(\{i\}) + \frac{1}{2^{n-1}} \sum_{S\subseteq N, i\in S} \frac{v^0(S)}{s}.$$

(Uniqueness). Take $f$ a value for cooperative games that satisfies efficiency, symmetry, necessary players get the 0-normalized per capita mean and additivity and take a cooperative game $(N, v)$. We prove now that $f(N, v) = \Gamma(N, v)$. Indeed, consider the basis of the vector space of characteristic functions of cooperative games with set of players $N$ given by:

$$\{e_{\{i\}} + e_N \mid i \in N\} \cup \{e_S \mid S \in 2^N, |S| \geq 2\}.$$

Observe that $v$ can be written in a unique way as a linear combination of the elements of this basis. Since $f$ satisfies additivity and, moreover, the properties of efficiency, symmetry and necessary players get the 0-normalized per capita mean characterize a unique value in the games of the basis, the proof is concluded. $\square$

Now we analyse an example in order to make some comments on the $\Gamma$ value. It is based on a similar example in Alonso-Meijide et al. (2020).

**Example 5.2.** Consider a three-story building with one apartment on each floor, the three apartments having the same surface. The three corresponding owners have agreed to install an elevator and share the corresponding cost. Such a cost is 120 (in thousands of euros), 50 of which correspond to the machine, 40 to the works to make the hollow of the elevator (a fixed cost of 10 plus a cost of 10 for the owner of the apartment in the first floor that is incremented by 10 for the owner of the apartment in the second floor and by an additional 10 for the owner of the apartment in the third floor), and 30 to the works to be done on each floor to allow access to the elevator (10 in each of them). According to this, the cost $c(i)$ in which each player is involved is:

- 50 (machine) + 10 (floor) + 20 (hollow) = 80 for $i = 1$, the player of the first floor,
- 50 (machine) + 10 (floor) + 30 (hollow) = 90 for $i = 2$, the player of the second floor,
- 50 (machine) + 10 (floor) + 40 (hollow) = 100 for $i = 3$, the player of the third floor.

The rest of the corresponding cost game is given by: $c(\{1,2\}) = 100$, $c(\{1,3\}) = c(\{2,3\}) = 110$, $c(N) = 120$. Table 5.1 below shows the distribution of costs for each of the apartments according to the Egalitarian value, the Shapley value and $\Gamma$. In European city centres it is common to find buildings coping with situations like the one described in this example. It is not uncommon for the owners of the lower floors to be less favourable to installing an elevator because of the costs involved. According to Spanish legislation, when owners decide to make an investment in the common elements of a building, the corresponding costs will be distributed in proportion to the owners' shares (which, in turn, sometimes depend only on the surface areas of the apartments). Therefore, the distribution due to the Egalitarian value will be the one proposed by the legislation in some occasions. Note that the proposed Shapley value and $\Gamma$ distributions tend to favour the owners of the lower floors. In short, $\Gamma$ seems to be the least controversial distribution in view of the usual dynamics of homeowners' communities, because it tends to favour the owners of the lowest floor, who are usually the most reluctant to bear the costs of installing an elevator.

| | Egalitarian | Shapley | $\Gamma$ |
|---|---|---|---|
| 1 | 40 | 33.3333 | 32.5 |
| 2 | 40 | 38.3333 | 38.75 |
| 3 | 40 | 48.3333 | 48.75 |

Table 5.1: The Egalitarian value, the Shapley value and $\Gamma$ for $(N, c)$.

It is not uncommon that in real situations such as those described in this example not all the owners are in favour of the elevator. When this occurs, sometimes the elevator will not be installed immediately even if the owners in favour of it have a majority. The reason for this is that the unfavourable owners (generally those on the lower floors) may refuse to pay the financial amounts due to them and the owners' community can only force them to do so by initiating legal proceedings which may be long, economically costly and which, moreover, may profoundly damage coexistence in the building. The practical consequence of this is that negotiations often take place within the owners' community to try to ensure that the installation of the elevator is possible without damaging coexistence in the building. One possible solution is that the owners not in favour of the elevator give up its service; this means that the elevator will not have stops on the corresponding floors, so that the works to give access to the elevator on those floors will not be necessary and the total cost of the installation will be lower. Assume, for instance that in the three-story building in this example the owner of the apartment in the first floor is not in favour to install the elevator and, moreover, declares that he will not pay any costs unless a court decision obliges him to do so. Negotiation in the community may propose that the elevator does not serve the first floor. In that case, the cost $d(i)$ in which each player is involved is:

- 0 for $i = 1$, the player of the first floor,
- 50 (machine) + 10 (floor) + 30 (hollow) = 90 for $i = 2$, the player of the second floor,
- 50 (machine) + 10 (floor) + 40 (hollow) = 100 for $i = 3$, the player of the third floor.

The rest of the corresponding cost game is given by: $d(\{1,2\}) = 90$, $d(\{1,3\}) = 100$, $d(\{2,3\}) = 110$, $d(N) = 110$. Table 5.2 below shows the distribution of costs for each of the apartments according to the Egalitarian value, the Shapley value and $\Gamma$. Note that the distribution given by the Egalitarian rule does not seem to facilitate the agreement on the installation of the elevator because the owner of the first floor will continue to pay a considerable amount and, in addition, will give up the service of the elevator. The distributions given by the Shapley value and by $\Gamma$, however, do seem to facilitate a final settlement. According to the Shapley value, the owner of the first floor will waive elevator service but pay nothing in return. According to $\Gamma$, the owner of the first floor will even receive a small compensation for the inconvenience caused to him by the works and the installation.

| | Egalitarian | Shapley | $\Gamma$ |
|---|---|---|---|
| 1 | 36.6666 | 0 | -6.6666 |
| 2 | 36.6666 | 50 | 53.3333 |
| 3 | 36.6666 | 60 | 63.3333 |

Table 5.2: The Egalitarian value, the Shapley value and $\Gamma$ for $(N, d)$.

## 5.3 Coalitional values and necessary players

In this section we extend the $\Gamma$ value to cooperative games with a coalition structure. We start by remembering the main features concerning that model.

We denote by $P(N)$ the set of all partitions of a finite set $N$. Each $P \in P(N)$, of the form $P = \{P_1, \ldots, P_m\}$, is called a *coalition structure* on $N$. We call unions of $P$ to its elements $P_1, \ldots, P_m$. We denote by $M$ the set $\{1, ..., m\}$.

A *cooperative game with a coalition structure* is a triple $(N, v, P)$ where $(N, v) \in \mathcal{G}_N$ and $P \in P(N)$. $\mathcal{G}_N^{cs}$ denotes the family of all cooperative games with a coalition structure and with player set $N$. Note that the first two elements of a cooperative game with a coalition structure, $(N, v)$, characterize a cooperative game.

By a *coalitional value* we mean a map $g$ that assigns to every game with a coalition structure $(N, v, P)$ a vector $g(N, v, P) \in \mathbb{R}^N$ with components $g_i(N, v, P)$, $i \in N$. Two of the most important coalitional values are the Owen value (Owen, 1977) and the Banzhaf-Owen value (Owen, 1982). In a similar way to the Shapley and Banzhaf values, the value of a particular player is a weighted sum of his contributions. In the case of the Shapley and Banzhaf values all possible contributions are taken into account, but for the coalitional values only the contributions to some coalitions are used to compute the values.

The Owen value $\Phi$ is the coalitional value defined by:

$$\Phi_i(N, v, P) = \frac{1}{m}\frac{1}{p_k} \sum_{R \subseteq M \setminus \{k\}} \sum_{T \subseteq P_k \setminus \{i\}} \frac{1}{\binom{m-1}{r}} \frac{1}{\binom{p_k-1}{t}} \Big[ v(\bigcup_{r \in R} P_r \cup T \cup \{i\}) - v(\bigcup_{r \in R} P_r \cup T) \Big]$$

for all $(N, v, P) \in \mathcal{G}_N^{cs}$ and all $i \in N$, where $P_k \in P$ is the union such that $i \in P_k$; $m$, $p_k$, $r$ and $t$ are the cardinalities of $M$, $P_k$, $R$ and $T$, respectively.

The *Banzhaf–Owen value* $\Psi$ is the coalitional value defined as

$$\Psi_i(N, v, P) = \frac{1}{2^{m-1}}\frac{1}{2^{p_k-1}} \sum_{R \subseteq M \setminus \{k\}} \sum_{T \subseteq P_k \setminus \{i\}} \Big[ v(\bigcup_{r \in R} P_r \cup T \cup \{i\}) - v(\bigcup_{r \in R} P_r \cup T) \Big]$$

for all $(N, v, P) \in \mathcal{G}_N^{cs}$ and all $i \in N$, where $P_k \in P$ is the union such that $i \in P_k$; $m$, $p_k$, $r$ and $t$ are the cardinalities of $M$, $P_k$, $R$ and $T$, respectively.

In the literature, we can find several characterizations of the Owen and the Banzhaf-Owen coalitional values; see for example Vázquez et al. (1997), Amer et al. (2002), Khmelnitskaya and Yanovskaya (2007), Alonso-Meijide et al. (2007), Casajus (2010) and Lorenzo-Freire (2016). We contribute to this research line providing a new characterization of these two coalitional values using necessary players. Only three properties are used in our results and the difference between them is the assigned payoff to necessary players.

**Necessary Players Get the Weighted Coalitional Mean**. A coalitional value $g$ satisfies the property of necessary players get the weighted coalitional mean if for each coalitional game $(N, v, P)$ and for each necessary player $i \in P_k$ in $(N, v)$, it holds that

$$g_i\left(N, v, P\right) = \frac{1}{m}\frac{1}{p_k}\sum_{R\subseteq M\setminus\{k\}}\sum_{T\subseteq P_k}\frac{1}{\binom{m-1}{r}}\frac{1}{\binom{p_k-1}{t-1}}v(\bigcup_{r\in R} P_r \cup T).$$

**Necessary Players Get the Coalitional Mean**. A coalitional value $g$ satisfies the property of necessary players get the coalitional mean if for each coalitional game $(N, v, P)$ and for each necessary player $i \in P_k$ in $(N, v)$, it holds that

$$g_i\left(N, v, P\right) = \frac{1}{2^{m-1}}\frac{1}{2^{p_k-1}}\sum_{R\subseteq M\setminus\{k\}}\sum_{T\subseteq P_k} v(\bigcup_{r\in R} P_r \cup T).$$

Both properties propose that a necessary player must receive the average worth over all coalitions that are compatible with the partitions (i.e., those that are formed by some complete unions and a subset of another union), but the first one takes into account the size of the coalitions while the second one assigns the same weight to all compatible coalitions.

Based on these two properties we will now state and prove two results characterizing the Banzhaf-Owen and the Owen values, respectively. But first we must make the following clarification. In this section, the properties of additivity, null player and efficiency in general refer to coalitional values for coalitional games with a coalition structure instead of to values for cooperative games (as in the previous section). The extensions of the null player and efficiency properties to coalitional values are immediate and we will not write them down formally. The additivity is extended in the following way: a coalitional value $g$ satisfies the property of additivity if for each pair of cooperative games with a coalition structure $(N, v, P), (N, w, P)$ it holds that

$$g(N, v + w, P) = g(N, v, P) + g(N, w, P).$$

**Theorem 5.5.** *The Banzhaf-Owen value is the unique coalitional value that satisfies the properties of additivity, null player and necessary players get the coalitional mean.*

*Proof.* (Existence). It is known that the Banzhaf-Owen value satisfies additivity and null player. Now let us see that it satisfies the property of necessary players get the coalitional mean. Take a cooperative game with a coalition structure $(N, v, P)$ and take $i \in P_k$ a necessary player in $(N, v)$. Then the Banzhaf-Owen value is reduced to

$$\begin{aligned}\Psi_i(N,v,P) &= \frac{1}{2^{m-1}}\frac{1}{2^{p_k-1}}\sum_{R\subseteq M\backslash\{k\}}\sum_{T\subseteq P_k\backslash\{i\}}\Big[v(\bigcup_{r\in R}P_r\cup T\cup\{i\})-v(\bigcup_{r\in R}P_r\cup T)\Big]\\ &= \frac{1}{2^{m-1}}\frac{1}{2^{p_k-1}}\sum_{R\subseteq M\backslash\{k\}}\sum_{T\subseteq P_k\backslash\{i\}}v(\bigcup_{r\in R}P_r\cup T\cup\{i\})\\ &= \frac{1}{2^{m-1}}\frac{1}{2^{p_k-1}}\sum_{R\subseteq M\backslash\{k\}}\sum_{T\subseteq P_k}v(\bigcup_{r\in R}P_r\cup T).\end{aligned}$$

(Uniqueness). For every $S \subseteq N$, $S \neq \emptyset$, the unanimity game $(N, u_S)$ is given, for every $T \subseteq N$, by:

$$u_S(T) = \begin{cases} 1 & \text{if } S \subseteq T, \\ 0 & \text{otherwise.} \end{cases} \tag{5.7}$$

Take a coalitional value $g$ that satisfies additivity, null player and necessary players get the coalitional mean and take a cooperative game with a coalition structure $(N, v, P)$. We prove now that $g(N, v, P) = \Psi(N, v, P)$. Given $S \subseteq N$, in the unanimity game $(N, u_S)$ every $i \in S$ is a necessary player and every $i \in N\backslash S$ is a null player. Let us fix $P$, a finite set $S \subseteq N$ and $c \in \mathbb{R}$. By additivity it is sufficient to prove that for all $i \in N$, $g_i(N, cu_S, P) = \Psi_i(N, cu_S, P)$. If $i \in N\backslash S$, applying the null player property $g_i(N, cu_S, P) = \Psi_i(N, cu_S, P) = 0$. If $i \in S$, applying neccesary players get the coalitional mean, we have that

$$g_i(N,cu_S,P)=\Psi_i(N,cu_S,P)=c\frac{1}{2^{m-1}}\frac{1}{2^{p_k-1}}\sum_{R\subseteq M\backslash k}\sum_{T\subseteq P_k}u_S(\bigcup_{r\in R}P_r\cup T),$$

where $P_k$ is the union such that $i \in P_k$. $\square$

**Theorem 5.6.** *The Owen value is the unique coalitional value that satisfies the properties of additivity, null player and necessary players get the weighted coalitional mean.*

*Proof.* (Existence). It is known that the Owen value satisfies additivity and null player. Let us see that it satisfies the property of necessary players get the weighted coalitional mean. Suppose that $i$ is a necessary player with $i \in P_k$; then the Owen value is

$$\begin{aligned}
&\Phi_i(N,v,P) = \\
&\frac{1}{m}\frac{1}{p_k}\sum_{R\subseteq M\backslash\{k\}}\sum_{T\subseteq P_k\backslash\{i\}}\frac{1}{\binom{m-1}{r}}\frac{1}{\binom{p_k-1}{t}}\Big[v(\bigcup_{r\in R}P_r\cup T\cup\{i\}) - v(\bigcup_{r\in R}P_r\cup T)\Big] = \\
&\frac{1}{m}\frac{1}{p_k}\sum_{R\subseteq M\backslash\{k\}}\sum_{T\subseteq P_k\backslash\{i\}}\frac{1}{\binom{m-1}{r}}\frac{1}{\binom{p_k-1}{t}}v(\bigcup_{r\in R}P_r\cup T\cup\{i\}) = \\
&\frac{1}{m}\frac{1}{p_k}\sum_{R\subseteq M\backslash\{k\}}\sum_{T\subseteq P_k}\frac{1}{\binom{m-1}{r}}\frac{1}{\binom{p_k-1}{t-1}}v(\bigcup_{r\in R}P_r\cup T).
\end{aligned} \tag{5.8}$$

(Uniqueness) Take a coalitional value $g$ that satisfies additivity, null player and necessary players get the weighted coalitional mean and take a cooperative game with a coalition structure $(N,v,P)$. We prove now that $g(N,v,P) = \Phi(N,v,P)$. Given $S \subseteq N$, in the unanimity game $(N,u_S)$ every $i \in S$ is a necessary player and every $i \in N\backslash S$ is a null player. Let us fix $P$, a finite set $S \subseteq N$ and $c \in \mathbb{R}$. By additivity it is sufficient to prove that for all $i \in N$, $g_i(N,cu_S,P) = \Phi_i(N,cu_S,P)$. If $i \in N\backslash S$, applying the null player property

$$g_i(N,cu_S,P) = \Phi_i(N,cu_S,P) = 0.$$

If $i \in S$, applying necessary players get the weighted coalitional mean, we have that

$$g_i(N,cu_S,P) = \Phi_i(N,cu_S,P) = c\frac{1}{2^{m-1}}\frac{1}{2^{p_k-1}}\sum_{R\subseteq M\backslash k}\sum_{T\subseteq P_k}u_S(\bigcup_{r\in R}P_r\cup T),$$

where $P_k$ is the union such that $i \in P_k$. □

We are now willing to extend the $\Gamma$ value, defined in Section 5.2, to cooperative games with a coalition structure. We next remind some properties that are relevant for our aim.

**Symmetry Inside Unions.** A coalitional value $g$ satisfies the property of symmetry inside unions if for each cooperative game with a coalition structure $(N,v,P)$, it holds that

$$g_i\left(N,v,P\right) = g_j\left(N,v,P\right).$$

for all $i,j$ symmetric players in $(N,v)$ with $i,j \in P_k$, $P_k \in P$.

We say that unions $P_k, P_l \in P$ are symmetric in $(N, v, P) \in \mathcal{G}_N^{cs}$ if $v(S \cup P_k) = v(S \cup P_l)$, for every $S = \cup_{j\in R} P_j$ with $R \subseteq M \backslash \{k, l\}$.

**Symmetry Among Unions.** A coalitional value $g$ satisfies the property of symmetry among unions if for each cooperative game with a coalition structure $(N, v, P)$, it holds that

$$\sum_{i\in P_k} g_i (N, v, P) = \sum_{j\in P_r} g_j (N, v, P)$$

for all $P_k, P_r \in P$, symmetric unions in $(N, v, P)$.

Given the properties of efficiency, additivity, symmetry inside unions and symmetry among unions one can expect to extend the $\Gamma$ value to cooperative games with a coalition structure and to characterize the new value using a property for necessary players that somewhat adapts the necessary players get the 0-normalized per capita mean property. First at all, let us see how to extend the $\gamma$ value, since the $\Gamma$ value depends on it.

**Definition 5.3.** The $\gamma^C$ value for cooperative games with a coalition structure is given for every $(N, v, P) \in \mathcal{G}_N^{cs}$ and every $i \in P_k$ by:

$$\begin{aligned}
\gamma_i^C (N, v, P) = {} & \frac{1}{2^{m-1}} \frac{1}{2^{p_k-1}} \left( \sum_{R\subseteq M\backslash k} \sum_{T\subset P_k, i\in T} \frac{v(\bigcup\limits_{r\in R} P_r \cup T)}{t} \right. \\
& \left. - \sum_{R\subseteq M\backslash k} \sum_{T\subset P_k, i\notin T, T\neq\emptyset} \frac{v(\bigcup\limits_{r\in R} P_r \cup T)}{p_k - t} \right) \\
& + \frac{1}{2^{m-1}} \frac{1}{p_k} \left( \sum_{R\subset M, k\in R} \frac{v(\bigcup\limits_{r\in R} P_r)}{r} - \sum_{R\subseteq M\backslash k} \frac{v(\bigcup\limits_{r\in R} P_r)}{m - r} \right) + \frac{v(N)}{mp_k}.
\end{aligned} \tag{5.9}$$

Let us see that $\gamma^C$ is a reasonable extension of $\gamma$. To check it, we can see that $\gamma^C$ is a coalitional value of $\gamma$, that is $\gamma^C(N, v, P^n) = \gamma(N, v)$ for all $(N, v, P^n) \in \mathcal{G}_N^{cs}$ where $P^n = \{\{1\}, ..., \{n\}\}$. In fact

$$\begin{aligned}
\gamma_i^C (N, v, P^n) &= \frac{1}{2^{m-1}} \left( \sum_{R\subset M, k\in R} \frac{v(\bigcup\limits_{r\in R} P_r)}{r} - \sum_{R\subseteq M\backslash k} \frac{v(\bigcup\limits_{r\in R} P_r)}{m - r} \right) + \frac{v(N)}{mp_k} \\
&= \frac{1}{2^{n-1}} \left( \sum_{S\subset N, i\in S} \frac{v(S)}{s} - \sum_{S\subseteq N\backslash\{i\}} \frac{v(S)}{n - s} \right) + \frac{v(N)}{n} = \gamma_i (N, v).
\end{aligned}$$

The next lemma proves that $\gamma^C$ satisfies an interesting property for cooperative games with a coalition structure.

**Lemma 5.1.** The $\gamma^C$ value satisfies the quotient game property, i.e., that

$$\sum_{i\in P_k} \gamma_i^C (N, v, P) = \gamma_k^C \left(M, v^P, P^m\right)$$

for all $P_k \in P$, where $v^P(R) = v(\cup_{r\in R} P_r)$ for all $R \subseteq M$, and $P^m = \{\{1\}, \ldots, \{m\}\}$.

*Proof.* Take a cooperative game with a coalition structure $(N, v, P)$ and $P_k \in P$. Then

$$\sum_{i\in P_k} \gamma_i^C (N, v, P) =$$

$$\frac{1}{2^{m-1}} \frac{1}{2^{p_k-1}} \sum_{i\in P_k} \left( \sum_{R\subseteq M\setminus k} \sum_{\substack{T\subset P_k \\ i\in T}} \frac{v(\bigcup\limits_{r\in R} P_r \cup T)}{t} - \sum_{R\subseteq M\setminus k} \sum_{T\subset P_k, i\notin T, T\neq\emptyset} \frac{v(\bigcup\limits_{r\in R} P_r \cup T)}{p_k - t} \right)$$

$$+ \frac{1}{2^{m-1}} \frac{1}{p_k} \sum_{i\in P_k} \left( \sum_{R\subset M, k\in R} \frac{v(\bigcup\limits_{r\in R} P_r)}{r} - \sum_{R\subseteq M\setminus k} \frac{v(\bigcup\limits_{r\in R} P_r)}{m-r} \right) + \sum_{i\in P_k} \frac{v(N)}{m p_k} =$$

$$\frac{1}{2^{m-1}} \frac{1}{2^{p_k-1}} \left( \sum_{R\subseteq M\setminus k} \sum_{T\subset P_k} \frac{t v(\bigcup\limits_{r\in R} P_r \cup T)}{t} - \sum_{R\subseteq M\setminus k} \sum_{T\subset P_k} \frac{(p_k - t) v(\bigcup\limits_{r\in R} P_r \cup T)}{p_k - t} \right)$$

$$+ \frac{1}{2^{m-1}} \left( \sum_{R\subset M, k\in R} \frac{v(\bigcup\limits_{r\in R} P_r)}{r} - \sum_{R\subseteq M\setminus k} \frac{v(\bigcup\limits_{r\in R} P_r)}{m-r} \right) + \frac{v(N)}{m} =$$

$$\frac{1}{2^{m-1}} \left( \sum_{R\subset M, k\in R} \frac{v(\bigcup\limits_{r\in R} P_r)}{r} - \sum_{R\subseteq M\setminus k} \frac{v(\bigcup\limits_{r\in R} P_r)}{m-r} \right) + \frac{v(N)}{m} = \gamma_k^C \left(M, v^P, P^m\right).$$

□

The quotient game is an interesting property because it guarantees that the total worth obtained by the players of a union coincides with the worth obtained by the union in the game played by the unions with the trivial coalition structure. Note that the Banzhaf-Owen value does not satisfy this property; however, Alonso-Meijide and Fiestras-Janeiro (2002) introduces the so-called symmetric coalitional Banzhaf value, which is an extension of the Banzhaf value to cooperative games with a coalition structure that satisfies the quotient game property.

In order to characterize $\gamma^C$, we introduce a new property for necessary players.

**Necessary Players Get the Per Capita Coalitional Mean**. A coalitional value $g$ satisfies the property of necessary players get the per capita coalitional mean if for each coalitional game $(N, v, P)$ with $v(N) = 0$ and for each necessary player $i \in P_k$ in $(N, v)$, it holds that

$$g_i\,(N, v, P) = \frac{1}{2^{m-1}}\left[\frac{1}{2^{p_k-1}}\sum_{R\subseteq M\backslash k}\ \sum_{T\subset P_k, i\in T}\frac{v(\bigcup\limits_{r\in R} P_r \cup T)}{t} + \frac{1}{p_k}\sum_{R\subseteq M, k\in R}\frac{v(\bigcup\limits_{r\in R} P_r)}{r}\right]$$

where $t = |T|$ and $r = |R|$ for all $T \subset P_k$ and $R \subseteq M$.

**Theorem 5.7.** *The $\gamma^C$ value is the unique value for cooperative games with a coalition structure that satisfies the properties of additivity, necessary players get the per capita coalitional mean, efficiency, symmetry inside unions and symmetry among unions.*

*Proof.* (Existence). It is clear that $\gamma^C$ satisfies additivity. To check that it satisfies the necessary players get the per capita coalitional mean property take a cooperative game with a coalition structure $(N, v, P)$ with $v(N) = 0$ and such that $i \in P_k \subseteq N$ is a necessary player in $(N, v)$. Then expression (5.9) reduces to

$$\gamma_i^C(N, v, P) =$$

$$\frac{1}{2^{m-1}}\frac{1}{2^{p_k-1}}\left(\sum_{R\subseteq M\backslash k}\ \sum_{T\subset P_k, i\in T}\frac{v(\bigcup\limits_{r\in R} P_r \cup T)}{t}\right) + \frac{1}{2^{m-1}}\frac{1}{p_k}\left(\sum_{R\subset M, k\in R}\frac{v(\bigcup\limits_{r\in R} P_r)}{r}\right) =$$

$$\frac{1}{2^{m-1}}\frac{1}{2^{p_k-1}}\sum_{R\subseteq M\backslash k}\ \sum_{T\subset P_k}\frac{v(\bigcup\limits_{r\in R} P_r \cup T)}{t} + \frac{1}{2^{m-1}}\frac{1}{p_k}\sum_{R\subseteq M}\frac{v(\bigcup\limits_{r\in R} P_r)}{r}.$$

To check that $\gamma^C$ satisfies symmetry inside unions take a cooperative game with a coalition structure $(N, v, P)$ and a pair of symmetric players in $(N, v)$ $i, j \in P_k$ with $P_k \in P$. Notice that, for a fixed $R \subseteq M\backslash k$,

$$\sum_{T\subset P_k, i\in T}\frac{v(\bigcup\limits_{r\in R} P_r \cup T)}{t} - \sum_{T\subset P_k, i\notin T, T\neq\emptyset}\frac{v(\bigcup\limits_{r\in R} P_r \cup T)}{p_k - t} =$$

$$\sum_{T\subseteq P_k\backslash\{i,j\}}\frac{v(\bigcup\limits_{r\in R} P_r \cup T\cup\{i\})}{t+1} + \sum_{T\subset P_k\backslash\{i,j\}}\frac{v(\bigcup\limits_{r\in R} P_r \cup T\cup\{i,j\})}{t+2}$$

$$- \sum_{T\subseteq P_k\backslash\{i,j\}, T\neq\emptyset}\frac{v(\bigcup\limits_{r\in R} P_r \cup T)}{p_k - t} - \sum_{T\subseteq P_k\backslash\{i,j\}}\frac{v(\bigcup\limits_{r\in R} P_r \cup T\cup\{j\})}{p_k - t - 1}.$$

Now, since $i, j$ are symmetric in $(N, v)$, the last expression is equal to

$$\sum_{T\subseteq P_k\setminus\{i,j\}} \frac{v(\bigcup_{r\in R} P_r \cup T \cup \{j\})}{t+1} + \sum_{T\subset P_k\setminus\{i,j\}} \frac{v(\bigcup_{r\in R} P_r \cup T \cup \{i,j\})}{t+2}$$
$$- \sum_{T\subseteq P_k\setminus\{i,j\}, T\neq\emptyset} \frac{v(\bigcup_{r\in R} P_r \cup T)}{p_k - t} - \sum_{T\subseteq P_k\setminus\{i,j\}} \frac{v(\bigcup_{r\in R} P_r \cup T \cup \{i\})}{p_k - t - 1}$$

and then it is clear that $\gamma_i^C(N, v, P) = \gamma_j^C(N, v, P)$.

Since $\gamma^C$ satisfies the quotient game property and it is a coalitional value of $\gamma$, then

$$\sum_{i\in P_k} \gamma_i^C (N, v, P) = \gamma_k^C \left(M, v^P, P^m\right) = \gamma_k(M, v^P).$$

Now, the efficiency and the symmetry properties of $\gamma$ imply that $\gamma^C$ satisfies symmetry among unions and efficiency.

(Uniqueness). Take $g$, a value for cooperative games with a coalition structure that satisfies efficiency, symmetry inside unions, symmetry among unions, necessary players get the per capita coalitional mean and additivity, and take a cooperative game with a coalition structure $(N, v, P)$. We prove now that $g(N, v, P) = \gamma^C(N, v, P)$. Indeed, consider the basis of the vector space of characteristic functions of cooperative games with set of players $N$ given by: $\{e_S\}_{S\in 2^N\setminus\emptyset}$ (see expression (5.1)). Observe that $v$ can be written in a unique way as a linear combination of the elements of the basis: $v = \sum_{S\in 2^N\setminus\emptyset} v(S)e_S$. Since $g$ satisfies additivity,

$$g(N, v, P) = \sum_{S\in 2^N\setminus\emptyset} g(N, v(S)e_S, P).$$

Notice that efficiency, symmetry inside unions, symmetry among unions, and necessary players get the per capita coalitional mean characterize a unique value in the class of games $\{(N, v(S)e_S, P) \mid S \subset N, S \neq \emptyset\}$. Besides, efficiency, symmetry inside unions and symmetry among unions, characterize a unique value for $(N, v(N)e_N, P)$. Hence $g(N, v, P) = \gamma^C(N, v, P)$. □

Now, in an analogous way as we obtain $\Gamma$ from $\gamma$, we introduce the following value.

**Definition 5.4.** The $\Gamma^C$ value for cooperative games with a coalition structure is given for every $(N, v, P) \in \mathcal{G}_N^{cs}$ and every $i \in P_k$ by:

$$\Gamma_i^C(N, v, P) = v(\{i\}) + \frac{v(P_k) - \sum_{j\in P_k} v(\{j\})}{p_k} + \gamma_i^C(N, v^{0'}, P), \tag{5.10}$$

where $v^{0'}(S) = v(S) - \sum_{r\in R} v(P_r) - \sum_{j\in S\setminus(\cup_{r\in R}P_r)} v(\{j\})$ and $R = \{r \in M \mid P_r \subseteq S\}$ for all $S \subseteq N$.

As for $\gamma^C$, we check that $\Gamma^C$ is a coalitional value of $\Gamma$. Take the cooperative game with a coalition structure $(N, v, P^n)$. Then

$$\begin{aligned}
&\Gamma_i^C\left(N,v,P^n\right) = \\
&v(\{i\}) + \frac{v(P_k) - \sum_{j\in P_k} v(\{j\})}{p_k} \\
&+ \frac{1}{2^{m-1}}\left(\sum_{R\subset M, k\in R} \frac{v^{0'}(\bigcup\limits_{r\in R} P_r)}{r} - \sum_{R\subseteq M\setminus k} \frac{v^{0'}(\bigcup\limits_{r\in R} P_r)}{m-r}\right) + \frac{v^{0'}(N)}{mp_k} = \\
&v(\{i\}) + \frac{1}{2^{n-1}}\left(\sum_{S\subset N, i\in S} \frac{v^0(S)}{s} - \sum_{S\subseteq N\setminus\{i\}} \frac{v^0(S)}{n-s}\right) + \frac{v^0(N)}{n} = \\
&\Gamma_i\left(N,v\right).
\end{aligned}$$

Now we provide an axiomatic characterization of $\Gamma^C$. We start with a lemma concerning the quotient game property.

**Lemma 5.2.** The $\Gamma^C$ value satisfies the quotient game property, i.e., that

$$\sum_{i\in P_k} \Gamma_i^C\left(N,v,P\right) = \Gamma_k^C\left(M, v^P, P^m\right)$$

for all $P_k \in P$, where $v^P(R) = v(\cup_{r\in R} P_r)$ for all $R \subseteq M$, and $P^m = \{\{1\}, \ldots, \{m\}\}$.

*Proof.* Take a cooperative game with a coalition structure $(N, v, P)$ and $i \in N$ such that $P_k \in P$. Then

$$\begin{aligned}
&\sum_{i\in P_k} \Gamma_i^C\left(N,v,P\right) = \\
&v(P_k) + \frac{1}{2^{m-1}}\frac{1}{2^{p_k-1}}\sum_{i\in P_k}\left(\sum_{R\subseteq M\setminus k}\sum_{T\subset P_k, i\in T} \frac{v^{0'}(\bigcup\limits_{r\in R} P_r \cup T)}{t}\right. \\
&\left.-\sum_{R\subseteq M\setminus k}\sum_{T\subset P_k, i\notin T, T\neq\emptyset} \frac{v^{0'}(\bigcup\limits_{r\in R} P_r \cup T)}{p_k - t}\right) + \\
&\frac{1}{2^{m-1}}\frac{1}{p_k}\sum_{i\in P_k}\left(\sum_{R\subset M, k\in R} \frac{v^{0'}(\bigcup\limits_{r\in R} P_r)}{r} - \sum_{R\subseteq M\setminus k} \frac{v^{0'}(\bigcup\limits_{r\in R} P_r)}{m-r}\right) + \sum_{i\in P_k} \frac{v^{0'}(N)}{mp_k} =
\end{aligned}$$

$$\frac{1}{2^{m-1}}\frac{1}{2^{p_k-1}}\left(\sum_{R\subseteq M\setminus k}\sum_{T\subset P_k}\frac{tv^{0'}(\bigcup_{r\in R}P_r\cup T)}{t}-\sum_{R\subseteq M\setminus k}\sum_{T\subset P_k}\frac{(p_k-t)v^{0'}(\bigcup_{r\in R}P_r\cup T)}{p_k-t}\right)+$$

$$\frac{1}{2^{m-1}}\left(\sum_{R\subset M,k\in R}\frac{v^{0'}(\bigcup_{r\in R}P_r)}{r}-\sum_{R\subseteq M\setminus k}\frac{v^{0'}(\bigcup_{r\in R}P_r)}{m-r}\right)+\frac{v^{0'}(N)}{m}+v(P_k)=$$

$$v(P_k)+\frac{1}{2^{m-1}}\left(\sum_{R\subset M,k\in R}\frac{v^{0'}(\bigcup_{r\in R}P_r)}{r}-\sum_{R\subseteq M\setminus k}\frac{v^{0'}(\bigcup_{r\in R}P_r)}{m-r}\right)+\frac{v^{0'}(N)}{m}=$$

$$\Gamma_k^C\left(M,v^P,P^m\right).$$

□

In order to characterize $\Gamma^C$, we introduce a new property for necessary players.

**Necessary Players Get the 0-Normalized Per Capita Coalitional Mean**. A coalitional value $g$ satisfies the property of necessary players get the 0-normalized per capita coalitional mean if for each coalitional game $(N,v,P)$ with $v(N)=\sum_{r\in M}v(P_r)$ and for each necessary player $i\in P_k$ in $(N,v)$, it holds that

$$g_i(N,v,P)=v(\{i\})+\frac{v(P_k)-\sum_{j\in P_k}v(\{j\})}{p_k}+$$
$$\frac{1}{2^{m-1}}\frac{1}{2^{p_k-1}}\sum_{R\subseteq M\setminus k}\sum_{T\subset P_k}\frac{v^{0'}(\bigcup_{r\in R}P_r\cup T)}{t}+\frac{1}{2^{m-1}}\frac{1}{p_k}\sum_{R\subseteq M}\frac{v^{0'}(\bigcup_{r\in R}P_r)}{r},$$

where $t=|T|$ and $r=|R|$ for all $T\subset P_k$ and $R\subseteq M$.

**Theorem 5.8.** *$\Gamma^C$ is the unique coalitional value for cooperative games with a coalition structure that satisfies the properties of additivity, necessary players get the 0-normalized per capita coalitional mean, efficiency, symmetry inside unions and symmetry among unions.*

*Proof.* (Existence). Since $\gamma^C$ satisfies additivity and symmetry inside unions, it is clear that $\Gamma^C$ also satisfies those properties. To check that it fulfils the necessary players get the 0-normalized per capita coalitional mean take a cooperative game with a coalition structure $(N,v,P)$ with $v(N)=\sum_{r\in M}v(P_r)$ and such that $i\in N$, with $i\in P_k\in P$, is a necessary player in $(N,v)$. Then expression (5.10) reduces to

$$\Gamma_i^C(N,v,P)=v(\{i\})+\frac{v(P_k)-\sum_{j\in P_k}v(\{j\})}{p_k}+$$

$$\frac{1}{2^{m-1}}\frac{1}{2^{p_k-1}}\left(\sum_{R\subseteq M\setminus k}\sum_{T\subset P_k}\frac{v^{0'}(\bigcup_{r\in R}P_r\cup T)}{t}\right)+\frac{1}{2^{m-1}}\frac{1}{p_k}\left(\sum_{R\subset M}\frac{v^{0'}(\bigcup_{r\in R}P_r)}{r}\right).$$

Since this solution satisfies the quotient game property and it is a coalitional value of $\Gamma$, for a cooperative game with a coalition structure $(N, v, P)$ and $P_k \in P$ it holds that

$$\sum_{i\in P_k}\Gamma_i^C(N,v,P)=\Gamma_k^C\left(M,v^P,P^m\right)=\Gamma_k(M,v^P).$$

Then it is easy to check that $\Gamma^C$ satisfies symmetry among unions and efficiency taking into account that $\Gamma$ satisfies efficiency and symmetry.

(Uniqueness). Take $g$, a value for cooperative games with a coalition structure that satisfies efficiency, symmetry, necessary players get the 0-normalized per capita coalitional mean and additivity and take a cooperative game with a coalition structure $(N, v, P)$. We prove now that $g(N, v, P) = \Gamma^C(N, v, P)$. Indeed, consider the basis of the vector space of characteristic functions of cooperative games with set of players $N$ given by:

$$\{e_{P_r}+e_N \mid r\in M\}\cup\{e_S \mid S\in 2^N, S\neq P_r, r\in M\}.$$

Observe that $v$ can be written in a unique way as a linear combination of the elements of this basis. Since $g$ satisfies additivity and, moreover, the properties of efficiency, symmetry and necessary players get the 0-normalized per capita coalitional mean characterize a unique value in the games of the basis, the proof is concluded. □

## 5.4 Concluding Remarks

Notice that $G$, $\gamma$ and $\Gamma$, the three values introduced in Section 5.2, satisfy the properties of additivity, efficiency and symmetry and then they can be written using the formula provided in Ruiz et al. (1998). Moreover, it is clear that $G$ and $\gamma$ satisfy the property of coalitional monotonicity dealt with in Wang et al. (2019) and thus, in view of Theorem 3.2 in Wang et al. (2019), they belong to the family of ideal values. Moreover, it is not difficult to prove that $\Gamma$ also satisfies the property of coalitional monotonicity and then it is also an ideal value. We provide next such a proof; to start with, we remember the property of coalitional monotonicity.

**Coalitional Monotonicity.** A value for cooperative games $f$ satisfies the property of coalitional monotonicity if for each pair of cooperative games $(N, v)$ and

$(N,w)$ fulfilling that there exists $T \subseteq N$ with $v(T) > w(T)$ and $v(S) = w(S)$ for all $S \subseteq N$, $S \neq T$, it holds that

$$f_i(N,v) \geq f_i(N,w)$$

for all $i \in T$.

In view of expressions (5.4) and (5.5), it is clear that $G$ and $\gamma$ satisfy the property of coalitional monotonicity. With respect to $\Gamma$ notice that, in view of expressions (5.5) and (5.6), for every TU-game $(N,u)$ and for every $i \in N$, $\Gamma_i(N,u)$ can be written as:

$$\begin{aligned} \Gamma_i(N,u) \; &= \; \frac{1}{2^{n-1}} \sum_{S \subset N, i \in S} \frac{1}{s} \left( u(S) - \sum_{j \in S} u(j) \right) \\ & -\frac{1}{2^{n-1}} \sum_{S \subset N, i \notin S} \frac{1}{n-s} \left( u(S) - \sum_{j \in S} u(j) \right) \\ & +\frac{1}{n} \left( u(N) - \sum_{j \in N} u(j) \right) + u(i). \end{aligned} \tag{5.11}$$

Take now $(N,v)$, $(N,w)$ and $T \subseteq N$ as in the statement of coalitional monotonicity. Using (5.11) it is clear that if $T$ has two or more elements, then $\Gamma_i(N,v) \geq \Gamma_i(N,w)$ for all $i \in T$. Assume now that $T = \{i\}$ $(i \in N)$. According to (5.6), the coefficients of $v(i)$ and $w(i)$ in $\Gamma_i(N,v)$ and $\Gamma_i(N,w)$ are identical and given by:

$$\begin{aligned} & -\frac{1}{2^{n-1}} \sum_{S \subset N, i \in S, S \neq i} \frac{1}{s} - \frac{1}{n} + 1 \\ = \; & -\frac{1}{2^{n-1}} \sum_{s=2}^{n-1} \frac{1}{s} \binom{n-1}{s-1} - \frac{1}{n} + 1 \\ = \; & -\frac{1}{2^{n-1}} \sum_{s=2}^{n-1} \frac{1}{n} \binom{n}{s} - \frac{1}{n} + 1 \\ = \; & -\frac{1}{2^{n-1}} \frac{1}{n} (2^n - 1 - n - 1) - \frac{1}{n} + 1 = \frac{n-3}{n} + \frac{2+n}{2^{n-1}n}. \end{aligned}$$

It is easy to check that $\frac{n-3}{n} + \frac{2+n}{2^{n-1}n} > 0$ for all $n \geq 1$, which implies that $\Gamma_i(N,v) > \Gamma_i(N,w)$ and completes the proof.

We finish this paper with a remark on the relation between our new values and the equal division and the equal surplus division values, that we denote by $ED$ and $ESD$ (see, for instance Alonso-Meijide et al., 2020). It is clear that, for every $(N,v) \in \mathcal{G}_N$ and every $i \in N$,

$$\gamma_i\left(N,v\right)=ED_i(N,v)+\frac{1}{2^{n-1}}\left(\sum_{S\subset N,i\in S}\frac{v(S)}{s}-\sum_{S\subset N,i\notin S}\frac{v(S)}{n-s}\right),$$

$$\Gamma_i\left(N,v\right)=ESD_i(N,v)+\frac{1}{2^{n-1}}\left(\sum_{S\subset N,i\in S}\frac{v^0(S)}{s}-\sum_{S\subset N,i\notin S}\frac{v^0(S)}{n-s}\right).$$

Now, if $ED^U$ and $ESD2^U$ are the extensions of $ED$ and $ESD$ for cooperative games with a coalition structure introduced in Alonso-Meijide et al. (2020), then for every $(N,v,P)\in\mathcal{G}_N^{cs}$ and every $i\in N$ it holds that

$$\begin{aligned}\gamma_i^C\left(N,v,P\right)=&\,ED_i^U(N,v,P)+\\ &\frac{1}{2^{m-1}}\frac{1}{2^{p_k-1}}\left(\sum_{R\subseteq M\setminus k}\sum_{\substack{T\subset P_k\\ i\in T}}\frac{v(\bigcup\limits_{r\in R}P_r\cup T)}{t}-\sum_{R\subseteq M\setminus k}\sum_{T\subset P_k,i\notin T,T\neq\emptyset}\frac{v(\bigcup\limits_{r\in R}P_r\cup T)}{p_k-t}\right)\\ &+\frac{1}{2^{m-1}}\frac{1}{p_k}\left(\sum_{R\subset M,k\in R}\frac{v(\bigcup\limits_{r\in R}P_r)}{r}-\sum_{R\subseteq M\setminus k}\frac{v(\bigcup\limits_{r\in R}P_r)}{m-r}\right),\end{aligned}$$

$$\begin{aligned}\Gamma_i^C\left(N,v,P\right)=&\,ESD2_i^U(N,v,P)+\\ &\frac{1}{2^{m-1}}\frac{1}{2^{p_k-1}}\left(\sum_{R\subseteq M\setminus k}\sum_{\substack{T\subset P_k\\ i\in T}}\frac{v^{0'}(\bigcup\limits_{r\in R}P_r\cup T)}{t}-\sum_{R\subseteq M\setminus k}\sum_{T\subset P_k,i\notin T,T\neq\emptyset}\frac{v^{0'}(\bigcup\limits_{r\in R}P_r\cup T)}{p_k-t}\right)\\ &+\frac{1}{2^{m-1}}\frac{1}{p_k}\left(\sum_{R\subset M,k\in R}\frac{v^{0'}(\bigcup\limits_{r\in R}P_r)}{r}-\sum_{R\subseteq M\setminus k}\frac{v^{0'}(\bigcup\limits_{r\in R}P_r)}{m-r}\right).\end{aligned}$$

# References

Albino D, Scaruffi P, Moretti S, Coco S, Truini M, Di Cristofano C, Cavazzana A, Stigliani S, Bonassi S, Tonini GP (2008). Identification of low intratumoral gene expression heterogeneity in neuroblastic tumors by genome-wide expression analysis and game theory. Cancer 113, 1412-1422.

Alonso-Meijide JM, Fiestras-Janeiro MG (2002). Modification of the Banzhaf value for games with a coalition structure. Annals of Operations Research 109, 213-227.

Alonso-Meijide JM, Carreras F, Fiestras-Janeiro MG, Owen, G. (2007). A comparative axiomatic characterization of the Banzhaf-Owen coalitional value. Decision Support Systems 43, 701-712.

Alonso-Meijide JM, Costa J, García-Jurado I (2019a). Null, Nullifying, and Necessary Agents: Parallel Characterizations of the Banzhaf and Shapley Values. Journal of Optimization Theory and Applications 180, 1027-1035.

Alonso-Meijide JM, Costa J, García-Jurado I (2019b). Values, Nullifiers and Dummifiers. In: Handbook of the Shapley Value (E Algaba, V Fragnelli, J Sánchez-Soriano eds.), CRC Press, 75-92.

Alonso-Meijide JM, Costa J, García-Jurado I, Gonçalves-Dosantos JC (2020). On egalitarian values for cooperative games with a priori unions. TOP. DOI: 10.1007/s11750-020-00553-2

Amer R, Carreras F, Giménez JM (2002). The modified Banzhaf value for games with a coalition structure: an axiomatic characterization. Mathematical Social Sciences 43, 45-54.

Béal S, Navarro F (2020). Necessary versus equal players in axiomatic studies. Preprint

Benati S, López-Blázquez F, Puerto J (2019). A stochastic approach to approximate values in cooperative games. European Journal of Operational Research 279, 93-106.

Bergantiños G, Valencia-Toledo A, Vidal-Puga J (2018). Hart and Mas-Colell consistency in PERT problems. Discrete Applied Mathematics 243, 11-20.

Carreras F, Puente A (2015). Coalitional multinomial probabilistic values. European Journal of Operational Research 245, 236-246.

Casajus A (2010). Another characterization of the Owen value without the additivity axiom. Theory and Decision 69, 523-536.

Costa J (2016). A polynomial expression of the Owen value in the maintenance cost game. Optimization 65, 797-809.

Driessen TSH, Funaki Y (1991). Coincidence of and collinearity between game theoretic solutions. OR Spectrum 13, 15-30.

Fragnelli V, Iandolino A (2004). Cost allocation problems in urban solid wastes collection and disposal. Mathematical Methods of Operations Research 59, 447-463.

Gonçalves-Dosantos JC, García-Jurado I, Costa J (2020). Sharing delay costs in stochastic scheduling problems with delays. 4OR-A Quarterly Journal of Operations Research 18, 457-476.

Gurram P, Kwon H, Davidson C (2016). Coalition game theory-based feature subspace selection for hyperspectral classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 9, 2354-2364.

Ju Y, Borm P, Ruys P (2007). The consensus value: a new solution concept for cooperative games. Social Choice and Welfare 28, 685-703.

Khmelnitskaya AB, Yanovskaya EB (2007). Owen coalitional value without additivity axiom. Mathematical Methods of Operations Research 66, 255-261.

Lorenzo-Freire S (2016). On new characterizations of the Owen value. Operations Research Letters 44, 491-494.

Lucchetti R, Moretti S, Patrone F, Radrizzani P (2010). The Shapley and Banzhaf values in microarray games. Computers and Operations Research 37, 1406-1412.

Moretti S, Patrone F (2008). Transversality of the Shapley value. Top 16, 1-41.

Owen G (1975). Multilinear extensions and the Banzhaf value. Naval Research Logistics Quarterly 22, 741-750.

Owen G (1977). Values of games with a priori unions. In: Mathematical Economics and Game Theory (R Henn, O Moeschlin eds.), Springer, 76-88.

Owen G (1982). Modification of the Banzhaf-Coleman index for games with a priori unions. In: Power, Voting and Voting Power (MJ Holler ed.), 232-238.

Ruiz LM, Valenciano F, Zarzuelo JM (1998). The family of least square values for transferable utility games. Games and Economic Behavior 24, 109-130.

Schmeidler D (1969). The nucleolus of a characteristic function form game. SIAM Journal on Applied Mathematics 17, 1163-1170.

Shapley LS (1953). A value for n-person games. In: Contributions to the Theory of Games II (HW Kuhn, AW Tucker eds.), Princeton University Press, 307-317.

Strumbelj E, Kononenko I (2010). An efficient explanation of individual classifications using game theory. Journal of Machine Learning Research 11, 1-18.

Tijs S (1981). Bounds for the core and the $\tau$-value. In: Game Theory and Mathematical Economics (O Moeschlin, D Pallaschke eds.), North Holland, 123-132.

Vázquez M, van den Nouweland A, García-Jurado I (1997). Owen's coalitional value and aircraft landing fees. Mathematical Social Sciences 34, 273-286.

Wang W, Sun H, van den Brink R, Xu G (2019). The family of ideal values for cooperative games. Journal of Optimization Theory and Applications 180, 1065-1086.

Yang Z, Zhang Q (2015). Resource allocation based on DEA and modified Shapley value. Applied Mathematics and Computation 263, 280-286.

# Chapter 6

# A two-stage heuristic algorithm for a class of multi-compartment vehicle routing problems with stochastic demands

JC Gonçalves-Dosantos, BV Casas-Méndez

Preprint

## 6.1 Introduction

The vehicle routing problems (VRPs) contemplate the design of a set of minimum cost routes for a fleet of vehicles to serve the demand of a group of customers. Due to this more basic problem, other more complex problems arise, adding different types of restrictions, in such a way as to adapt as much as possible to reality. Among them, we find the multi-compartment vehicle routing problems (MC-VRPs). This kind of problem shares the same objective as VRPs, but there are different incompatible products that must be transported in independent vehicle compartments.

However, many of these problems, even given an imposing amount of restrictions, do not take into account an important aspect, since they do not consider the fact that the parameters that constitute the problem in real life are random. When any of the model parameters is random, we are faced with stochastic vehicle routing problems (SVRPs). In contrast to deterministic models, these have a rather dispersed and unorganized literature.

From existing works in the literature, one of the most studied problems is when we consider the demands as random variables. These are called vehicle routing problems with stochastic demands (VRPSD). As in the deterministic case, we want to determine a fixed set of routes of minimum total distance, which corresponds to the distance traveled corresponding to the fixed set of routes plus the expected value of the extra distance (we call it expected distance) that could be required by a particular realization of the random variables. These extra distances are due to the fact that the demand on the routes may occasionally exceed the capacity of the vehicle, since when demand is random this is known gradually as the vehicle completes its route, forcing it to return to the depot before continuing the route.

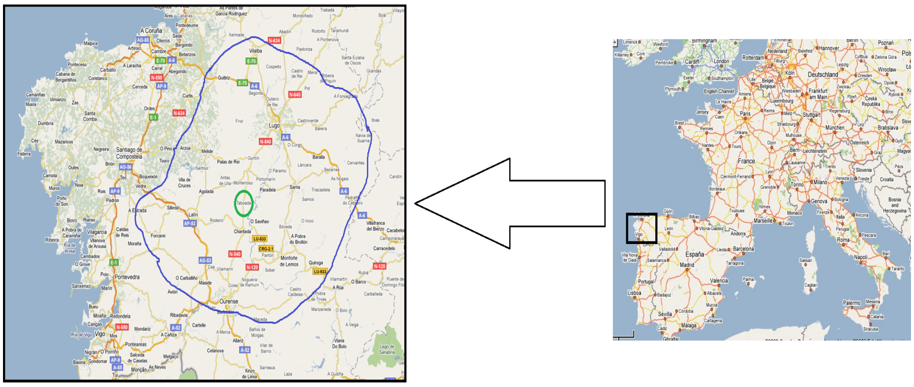

Figure 6.1: Area of influence of an agrarian cooperative in Spain.

This work combines the two models mentioned above, the multi-compartment vehicle routing problems and the vehicle routing problems with stochastic demands. We deal with that we call multi-compartment vehicle routing problem with stochastic demands (MC-VRPSD). There are not many works in the literature that analyze this double problem. In addition, we incorporate the restriction that in the same compartment we can not mix different products or products demanded by different customers, that there may be urgent orders (this is a kind of the so-called time window restriction), and that there may be problems of accessibility of the trucks to certain customers. All these ingredients make our model novel, as far as we know.

For the resolution of the problem, we designed a constructive algorithm extending the known Clarke and Wright algorithm, which provides an initial solution to the problem and that takes into account the restriction that prevents mixing products in the same hopper of a truck and the urgencies of the orders, when the saving associated with the insertion of a new customer in a route is defined. In addition, taking into account the stochastic nature of the problem and the possibility of large demands in relation to the capacity of the hoppers, we will allow that the demand of a customer can be divided and served in different hoppers of the same truck or of different trucks. The final solution is an improvement of this, obtained through the tabu search philosophy. It is also interesting to note that the algorithms designed pursue a double objective of maximizing the transported load along with a minimization of transport costs. We test the algorithm with real life data and we also created 14 testcases on which a computational study is carried out.

The motivation of the work is a problem taken from the real life. Certain types of agricultural cooperatives produce different types of food for farm animals that are distributed to a large number of farmers. Many times, the costumers are scattered over a wide area. Traditionally, they order different types of feed, and the cooperative has a specific delivery time depending on the order urgency. If the order is urgent this must be delivered in one day, but delivery costs are higher. This procedure can produce saturations in the system and it is not the most adequate for an efficient design of the routes.

We consider a cooperative in Galicia, a region of the northwest of Spain, where four types of feed are produced. In this cooperative, there are more than 1500 customers and each of them places at most two orders per month, and usually only one kind of feed. This cooperative has a fleet of different trucks. Each of these has several hoppers, compartments, with different capacities, and each hopper can only transport one type of feed. They are also restricted to a limited number of distance traveled per day and load. The truck's driver is paid for the distance traveled (which also can be expressed in terms of the time needed to travel it) and the cargo transported. For last, access to each farm is restricted to only a few vehicles. Figure 6.1 presents the area of influence of an agrarian cooperative in Spain.

The goal for this work is to provide the company with an anticipation tool, that is to say, a method to design their routes of future deliveries, which can improve the efficiency of distribution, reduce the associated costs and above all avoid saturations of the system. Knowing the number of days since the last delivery and the estimated daily consumption of a farm, we can know how urgently it needs to be visited. In addition, by knowing the past orders we can estimate a distribution for the customer's demand. Moreover, technological advances allow to know the inventory levels of the members of the cooperative. With this information, the company can make the selection of customers to be served every day according to the available fleet and they need a tool that is responsible for planning the routes of the vehicles so that they generate the minimum cost of transport. In order to address this problem in a general way, our model will consider a heterogeneous vehicle fleet both in capacity, as well as in the number and capacity of its compartments.

The paper is structured as follows. Section 2 deals with an overview about related problems of vehicle routes. Sections 3 and 4 present the methodology adopted in the paper and the mathematical formulation of the problem, respectively. In section 5 the algorithms designed to solve the problem are shown, which are illustrated in section 6 through examples taken from real life. Section 7 presents a computational study and section 8 summarizes the conclusions of the paper. Finally, the references that have been handled are presented.

## 6.2 Literature review

Next, we made a broad overview of papers related to the problem of our interest, focusing especially on problems of compartmentalized vehicle routes.

Dantzig and Ramser (1959), under the name “The truck dispatching problem”, propose several problems with a common objective, to design optimal routes for petrol delivery trucks between a terminal and a large number of stations. We can see the different variants of VRPs and solutions methods in Cordeau et al. (2007), Laporte (2009), and Pollaris et al. (2015) surveys.

Gendreau et al. (1996), Berhan et al. (2014), Dror (2016), and Oyola et al. (2016) present surveys on SVRPs variants and solution methods. From the point of view of the applications, Dror et al. (1985) consider the delivery of home heating oil where daily customer consumption is also random. Bertsimas and Simchi−Levi (1996) discuss many other applications like the distribution of beer, gasoline or pharmaceuticals.

A special case of VRPs is one in which the vehicles are divided into multi-compartments (MC-VRP). Brown and Graves (1981) and Brown et al. (1987) are considered seminal jobs about this type of problem. They consider separately the design of the routes and the allocation of products to compartments and make use of a collection of traveling salesman problems. van der Bruggen et al. (1995) is another of the first articles on MC-VRPs. In it, a problem of redistribution of products in a large oil company is modeled and a heuristic algorithm is proposed for its resolution. This algorithm applied to company data generates cost savings with respect to the company’s starting distribution.

Chajakis and Guignard (2003) propose a heuristic algorithm based on Lagrange relaxation for the supply of so-called convenience stores. Avella et al. (2004) propose a formulation of set partitioning. Based on it, they consider an exact branch-and-price algorithm and, finally, they make use of a packing/routing heuristic algorithm for large problems. El Fallahi et al. (2008) compared a constructed algorithm, a memetic algorithm and a tabu search, when the assignment of product types to compartments is fixed. They conclude that the tabu search provides slightly better results although it requires more computing time. Oppen and LØkketangen (2008) propose a tabu search algorithm for a problem dealing with transportation of animals, multi-compartment vehicles and inventory constraints.

Caramia and Guerriero (2010) investigate a multi-compartment vehicle routing problem where at most one type of product can be assigned to a compartment and the aditional constraint that some farms are small and inaccesible by large vehicles. A model that encompasses the whole problem is not provided. Instead, a heuristic in two stages is used. The first is to minimize the number of vehicles and the assignment of customers to vehicles. In the second part the cost of the routes is minimized. The restriction of inaccessibility in this case, is handled assuming the existence of vehicles formed by a truck and a trailer, so that some farms can not be visited by the trailer in which case they are visited only by the truck.

Muyldermans and Pang (2010) get a feasible initial solution built with Clarke and Wright's savings algorithm. Next, they perform a local search by means of movements taken from the literature and improve the quality of the solution obtained by means of a meta-heuristic of the type guided local search. They perform a sensitivity analysis on certain parameters and work with new instances and others taken from the literature. Derigs et al. (2011) consider a model involving a set of vehicles all of them with the same capacity and the same number of compartments, with the same capacity. They introduce a benchmark suite and a solver suite of heuristic components which covers a broad range of alternative approaches for construction, local search, large neighbourhood search and metaheuristics, with the aim of identifying effective algorithmic setups for achieving high solution quality. They say that "the following extensions present interesting avenues for further studies: optimized splitting of an order to allow delivering one order in several compartments and considering vehicles with different compartment setups (number of compartments and capacities of compartment) in one model."

Another recent paper is Coelho and Laporte (2015) that define and compare four categories of the multi-compartment fuel delivery problem that can involve both routing and inventory costs. They propose two formulations for each case and describe a branch-and-cut algorithm that solves single period and multi-period cases, which contain up to 50 and 20 customers, respectively. Archetti et al. (2016) study the problem of the delivery of various products and compare transport costs if vehicles are used for a single commodity or for more than one. Silvestrin and Ritt (2017) present a tabu search to solve an MC-VRP. Henke et al. (2018) consider a variant of the MC-VRP that occurs in the context of the recycling of glass waste. It is assumed that for the collection of the contents of glass containers, a homogeneous vehicle fleet is available. Individually for each vehicle, the capacity can be discretely separated in a limited number of compartments to which different types of glass waste are assigned. The objective of the problem is to minimize the total distance that the elimination vehicles must travel. To solve this problem optimally, they develop and implement a branch-and-cut algorithm. Ostermeier and Hübner (2018) consider vehicle slection for an MC-VRP with flexible compartments. The objective of the research is to show the benefits of considering both single-compartments vehicles and multicompartments for distribution fleets, taking into account the cost incurred by the use of the corresponding types of vehicles. The problem was solved with large neighborhood search.

According to Mendoza et al. (2010) "In contrast to the VRPSD, research to the MC-VRPSD is scarce. Tatarakis and Minis (2009) tackled a single-route variant of the problem in which the sequence of customers (route) is fixed beforehand. The problem consists of selecting the optimal restocking points along the route. The authors propose a set of dynamic programming algorithms and solve to optimality problems of up to 15 customers. Mendoza et al. (2011) proposed a set of construction heuristics comprising stochastic versions of the nearest neighbor, best insertion, and Clarke and Wright (Dror and Trudeau, 1986) heuristics, extended to the multi-compartment case. The latter proved to be the most competitive according to extensive computational experiments conducted on different types of instances. To the best of our knowledge,these are the few attempts made to solve the MC-VRPSD." Pandelis et al. (2012) consider the problem of designing the path of a single compartmentalized vehicle that delivers different demands of products that are random variables. They use a dynamic programming algorithm to meet customer demands with the expected total minimum cost. Huang 2015 proposes a mathematical programming model for an MC-SVRP. Two types of customers are considered, those who request a certain product or those to whom a product must be collected. They use two fleets of vehicles and form two sets of different routes. For the resolution of the model they use a tabu search, which starts from an initial solution created in a random way. The effectiveness of the algorithm is tested by means of different benchmarks. Goodson (2015) proposes a simulated annealing algorithm for an MC-SVRP.

The problems of the routes of the vehicles are of an NP-hard complexity. The different algorithms that provide an exact solution are interesting to test the model but, in general, they are useful only when a small number of clients are considered. Instead of exact solutions, solutions obtained by metaheuristic algorithms combined with heuristics that provide a good initial solution are generally considered. As we have seen in the previous panoramic, among the metaheuristics, one of the most used is the tabu algorithm (cf. Glover, 1989, Glover, 1990, and Xia et al. (2018) among others), which is characterized by its simplicity, speed, accuracy and robustness. Regarding heuristics, the pioneering works of MC-SVRP have designed adaptations of the Clarke and Wright (Clarke and Wright, 1964) algorithm.

To finish this review, with respect to the concrete problem of food distribution, recent works are for example Hsu et al. (2007), wich consider a VRPSD, Ambrosino and Sciomachen (2007) and Kuznietsov et al. (2016) that study a VRP and Frutos and Casas-Méndez (to appear) in the context of MC-VRP.

## 6.3 Methodological approach

We now explain, in summary form, the methodology followed to address our problem.

In the first place, we carry out the mathematical formulation of our problem of vehicle routes, with multi-compartments, stochastic demands and restrictions on the prohibition of mixing products, urgent customers and accessibility to the farms. Through this model we make explicit all the ingredients that constitute the problem, including deterministic and random parameters, decision variables and the different restrictions. For its resolution, we develop an initial solution by means of a new adaptation of the Clarke and Wright algorithm and from this we design a new tabu search algorithm. Note that this methodology is a consequence of what has been followed in the investigation of problems that can be seen as a particular case of the problem analyzed here The algorithms are programmed in the `R` language. First, we illustrate the model and its solution with a numerical example and a set of real data. Then, we perform a computational study through a set of simulated data and a benchmark created for the occasion.

## 6.4 Stochastic vehicle routing problem formulation

In this section, we are going to present a mathematical programming formulation of the model.

### 6.4.1 Indexes and sets

The main elements of our formulation are the following.

| | |
|---|---|
| $n, N$ | = Index and set of customers. |
| $\bar{N} := N \cup \{0\}$ | = Set of customers and the cooperative. |
| $t, T$ | = Index and set of trucks. |
| $h, H_t$ | = Index and set of hoppers of truck $t \in T$. |
| $f, F$ | = Index and set of livestock feeds. |
| $r, R_t$ | = Index and set of possible routes of truck $t \in T$ that are considered in the planning. |

### 6.4.2 Parameters

| | |
|---|---|
| $C^t := \{c_h^t\}_{h \in H_t}$ | = Vector of capacities of truck $t \in T$, where $c_h^t$ is the capacity in grams of hopper $h \in H_t$. |
| $\mathbf{O} := [\mathbf{O}_{n,f}]_{n \in N, f \in F}$ | = Random matrix of orders, where $\mathbf{O}_{n,f}$ is the random variable of the order of customer $n \in N$ and feed $f \in F$. |
| $D := [d_{n_1,n_2}]_{n_1,n_2 \in \bar{N}}$ | = Matrix of times in minutes, where $d_{n_1,n_2}$ is the time a truck needs to go from customer or cooperative $n_1 \in \bar{N}$ to customer or cooperative $n_2 \in \bar{N}$. |
| $E := [e_{n_1,n_2}]_{n_1,n_2 \in \bar{N}}$ | = Matrix of distances in kilometers, where $e_{n_1,n_2}$ is the distance a truck needs to go from customer or cooperative $n_1 \in \bar{N}$ to customer or cooperative $n_2 \in \bar{N}$. |
| $I := [i_{t,n}]_{t \in T, n \in N}$ | = Matrix that describes which customers can be visited by the truck $t \in T$, such that $i_{t,n}$ is equal 1 if the truck $t$ can visit client $n \in N$ and 0 otherwise. |
| $l_t$ | = Maximum allowable load of the truck $t \in T$. |
| $\mathbf{U} := \{\mathbf{U}_n\}_{n \in N}$ | = Random urgency vector where $\mathbf{U}_n$ is the random variable of the maximum number of days that stockbreeder $n \in N$ can wait to be served. |
| $M$ | = Estimated maximum distance that a truck can cover in a single day. |
| $\sigma$ | = Fixed cost of each unloading of a truck. |
| $\gamma(a)$ | = Variable cost per gram carried and kilometer traveled, where $a$ is the distance covered. |

### 6.4.3 Decision variables

The variables define the routes followed by the trucks of the fleet and the way in which their compartments are loaded.

$$x_{n_1,n_2}^{t,r} = \begin{cases} 1 & \text{if truck } t \in T \text{, on its route } r \in R_t, \text{ travels from } n_1 \in \bar{N} \text{ to} \\ & n_2 \in \bar{N}, \\ 0 & \text{otherwise.} \end{cases}$$

$y_{n,f,h}^{t,r}$ = Proportion ($\in [0,1]$) of the hopper $h \in H_t$ that is being used to carry feed $f \in F$ by truck $t \in T$ on its route $r \in R_t$ to serve the customer $n \in N$.

### 6.4.4 Constraints

The constraints of the mathematical program are classified in five groups.

1) Constraints describing the routes of trucks.

$$x^{t,r}_{n,n} = 0 \qquad \forall t \in T, \forall r \in R_t, \forall n \in \bar{N}. \tag{6.1}$$

$$\sum_{n\in N} x^{t,r}_{0,n} \leq 1 \qquad \forall t \in T, \forall r \in R_t. \tag{6.2}$$

$$x^{t,r}_{n_1,n_2} \leq \sum_{n\in N} x^{t,r}_{0,n} \qquad \forall t \in T, \forall r \in R_t, \forall n_1, n_2 \in N. \tag{6.3}$$

$$\sum_{n\in \bar{N}} x^{t,r}_{n,n_1} \leq \sum_{n\in \bar{N}} x^{t,r}_{n_1,n} \qquad \forall t \in T, \forall r \in R_t, \forall n_1 \in \bar{N}. \tag{6.4}$$

$$\sum_{n_1,n_2\in S} x^{t,r}_{n_1,n_2} \leq |S| - 1 \qquad \forall t \in T, \forall r \in R_t, \forall S \subseteq N \text{ and } |S| > 1. \tag{6.5}$$

Constraint (6.1) describes that no truck is allowed to go from a stockbreeder and back to the same stockbreeder. (6.2) indicates that each route of each truck departs from the cooperative headquarters at most once. (6.3) indicates that each planned route or each truck has to start from the cooperative. (6.4) is a flow conservation constraint, i.e., on each route if a truck arrives at a costumer it also has to leave from the same costumer. Finally, (6.5) eliminates possible subtours on each route of each truck.

2) Constraints that link the routes of trucks and their loading.

$$\frac{1}{|H_t|} \sum_{f\in F} \sum_{h\in H_t} y^{t,r}_{n_1,f,h} \leq \sum_{n\in \bar{N}} x^{t,r}_{n,n_1} \qquad \forall t \in T, \forall r \in R_t, \forall n_1 \in N. \tag{6.6}$$

$$\sum_{n\in \bar{N}} x^{t,r}_{n,n_1} \leq \sum_{f\in F} \sum_{h\in H_t} M' y^{t,r}_{n_1,f,h} \qquad \forall t \in T, \forall r \in R_t, \forall n_1 \in N. \tag{6.7}$$

Constraint (6.6) describes that if a truck is carrying feed for a stockbreeder, we ensure that the truck visits him. (6.7) indicates that if a truck visits a costumer on a route, it is indeed carrying feed for him. Here, $M'$ denotes a sufficiently large constant. In general, it will be enough to take $M' = c^t_h$ where $h \in H_t$.

3) Constraints modelling random urgent stockbreeder orders.

$$\mathbb{P}\left[\sum_{t\in T} \sum_{r\in R_t} \sum_{h\in H_t} c^t_h y^{t,r}_{n,f,h} \leq \mathbf{O}_{n,f}\right] \leq \alpha \qquad \forall f \in F, \forall n \in N, \text{if } \mathbb{P}\left[\mathbf{U}_n = 0\right] \geq \beta. \tag{6.8}$$

$$\mathbb{P}\left[\sum_{t\in T}\sum_{r\in R_t}\sum_{h\in H_t} c_h^t y_{n,f,h}^{t,r} \le \mathbf{O}_{n,f}\right] \ge 1-\alpha \qquad \forall f\in F, \forall n\in N, \text{if } \mathbb{P}\left[\mathbf{U}_n = 0\right] \le \beta. \tag{6.9}$$

Constraint (6.8) indicates that if a client's order is urgent with high probability, then it is unlikely that the client's demand will not be addressed. On the contrary, (6.9) indicates that when an order is unlikely to be urgent, it is more likely that the full order will not be served. In both constraints, we take $\beta = 0.95$. In the first restriction, we take $\alpha = 0.05$ while in the second we will take $\alpha$ equal to 0.2 or 0.3.

4) Constraints modelling accessibility, the capacities of trucks and limits on minutes and kilometers in the working day (capacity constraints).

$$\sum_{n\in\bar{N}} x_{n,n_1}^{t,r} \le i_{t,n_1} \qquad \forall t\in T, \forall r\in R_t, \forall n_1 \in N. \tag{6.10}$$

$$\sum_{f\in F} y_{n_1,f,h}^{t,r} \le i_{t,n_1} \qquad \forall t\in T, \forall r\in R_t, \forall h\in H_t, \forall n_1\in N. \tag{6.11}$$

$$\sum_{r\in R_t}\sum_{n_1\in\bar{N}}\sum_{n_2\in\bar{N}} (d_{n_1,n_2} + 2d_{0,n_2} p_{n_2,f,h}^{t,r}) x_{n_1,n_2}^{t,r} \le 540 \qquad \forall t\in T, \forall h\in H_t, \forall f\in F \tag{6.12}$$

where:

$$p_{n_2,f,h}^{t,r} := \begin{cases} 0 & \text{if } \mathbb{P}\left[\mathbf{U}_{n_2} = 0\right] \le \beta \text{ or } n_2 = 0, \\ \mathbb{P}\left[c_h^t y_{n_2,f,h}^{t,r} \le \mathbf{O}_{n_2,f}\right] & \text{otherwise.} \end{cases}$$

$$\sum_{r\in R_t}\sum_{n_1\in\bar{N}}\sum_{n_2\in\bar{N}} (e_{n_1,n_2} + 2e_{0,n_2} p_{n_2,f,h}^{t,r}) x_{n_1,n_2}^{t,r} \le M \qquad \forall t\in T, \forall h\in H_t, \forall f\in F. \tag{6.13}$$

$$\sum_{n\in N}\sum_{f\in F}\sum_{h\in H_t} c_h^t y_{n,f,h}^{t,r} \le l_t \qquad \forall t\in T, \forall r\in R_t. \tag{6.14}$$

(6.10) and (6.11) describe that impossible visits are not allowed. In (6.12), it is established that each route of each truck lasts a maximum of 540 minutes. Note that we do not consider unloading time. To properly formulate the restriction, we need to sum the expected time for a so-called failed route. This happens when a truck visits a costumer with urgent order and the truck does not carry enough cargo to meet the full demand. (6.13) is similar to (6.12), but instead minutes we use kilometers and a maximum of $M$ per route. Finally, (6.14) indicates that a truck can not transport more than its authorized legal cargo.

5) Constraints modelling technical restrictions on the loading procedure of the trucks.

$$y^{t,r}_{n_1,f_1,h} + y^{t,r}_{n_2,f_2,h} \leq \max\{y^{t,r}_{n_1,f_1,h}, y^{t,r}_{n_2,f_2,h}\} \qquad \forall t \in T, \forall r \in R_t, \forall h \in H_t, \forall n_1, n_2 \in N, n_1 \neq n_2, \forall f_1, f_2 \in F, f_1 \neq f_2. \tag{6.15}$$

$$y^{t,r}_{n_1,f,h} + y^{t,r}_{n_2,f,h} \leq \max\{y^{t,r}_{n_1,f,h}, y^{t,r}_{n_2,f,h}\} \qquad \forall t \in T, \forall r \in R_t, \forall h \in H_t, \forall n_1, n_2 \in N, n_1 \neq n_2, \forall f \in F. \tag{6.16}$$

$$y^{t,r}_{n,f_1,h} + y^{t,r}_{n,f_2,h} \leq \max\{y^{t,r}_{n,f_1,h}, y^{t,r}_{n,f_2,h}\} \qquad \forall t \in T, \forall r \in R_t, \forall h \in H_t, \forall n \in N, \forall f_1, f_2 \in F, f_1 \neq f_2. \tag{6.17}$$

(6.15) and (6.16) describe that each hopper of each truck cannot be used to serve more than one stockbreeder. (6.17) indicates that each hopper of each truck cannot be loaded with more than one type of feed. [1]

### 6.4.5 Objective function

In the main objective function, we maximize the amount of feed delivered in the current working day. In a second auxiliary objective function, we minimize the total transportation cost assumed by the cooperative.

$$\max \sum_{t \in T} \sum_{r \in R_t} \sum_{h \in H_t} \sum_{f \in F} \sum_{n \in N} c^t_h y^{t,r}_{n,f,h},$$

$$\min \sum_{t \in T} \sum_{r \in R_t} \sigma \sum_{n_1, n_2 \in \bar{N}} x^{t,r}_{n_1,n_2} + \sum_{t \in T} \sum_{r \in R_t} \gamma \left( \sum_{n_1 \in \bar{N}} \sum_{n_2 \in \bar{N}} (e_{n_1,n_2} + 2e_{0,n_2} p^{t,r}_{n_2,f,h}) x^{t,r}_{n_1,n_2} \right) \left( \sum_{n_1 \in \bar{N}} \sum_{n_2 \in \bar{N}} (e_{n_1,n_2} + 2e_{0,n_2} p^{t,r}_{n_2,f,h}) x^{t,r}_{n_1,n_2} \right) \sum_{h \in H_t} \sum_{f \in F} \sum_{n \in N} c^t_h y^{t,r}_{n,f,h}. \tag{6.18}$$

## 6.5 Heuristic algorithms

Constructive algorithms and tabu search algorithms have shown to be effective tools for solving many NP-hard combinatorial optimization problems. Constructive algorithms provide good initial solutions and tabu search algorithms are able

[1]Note that these restrictions are non-linear, however it would also be possible to express them linearly by incorporating new variables into the model.

to improve an initial solution and they require an initial solution, a neighborhood structure, and proceed by transiting from one solution to another using moves across a number of iterations trying to avoid local optima and revisit already explored solutions. In this section we provide our proposals of these two kinds of algorithms to solve the stochastic vehicle routing problem presentd in this paper.

### 6.5.1 Constructive heuristic

In this section, we present the constructive heuristic algorithm with which we will obtain an initial solution. It consists of a generalization of the Clarke and Wright algorithm (Clarke and Wright, 1964 and Dror and Trudeau, 1986) that adapts to the characteristics of our stochastic problem, with compartmentalized vehicles and restrictions on accessibility, urgency of some orders and mixing of different products. The general idea is to select a customer and consider the route that connects the deposit to that customer. In addition, a truck is selected to serve that customer, a hopper is assigned to it and the hopper must be loaded taking into account the customer's demand. Then, customers are inserted into the route, following the so-called criterion of maximum savings.

Keep in mind that a customer may have a very large demand so that if a single hopper is assigned it is very likely that if the order is urgent, upon arrival to the customer it is necessary to return to the warehouse to complete the order. For this reason, we will consider the so-called partner replicas, which we explain below. Given a customer $n \in N$, let $p_n = \mathbb{P}[\mathbf{U}_n = 0]$ be and $q_n$ the quantile of order $p_n$ of the random variable $\mathbf{O}_n$ corresponding to the demand of said customer. Let $c = \max\{c_h^t : t \in T, h \in H_t\}$ be the maximum size among the hoppers of all trucks and $r_n = \lceil \frac{q_n}{c} \rceil$.[2] The client $n$ will be replicated in $r_n$ clients with $r_n \geq 1$ and in such a way that the distance between a replica of $n$ and a replica of another client $n'$ will be equal than the distance between $n$ and $n'$. In addition, we assume that the distance between two replicas of the same client is equal to 0. On the other hand, the mean and the standard deviation of the demand of the replica of a client $n$ will be taken as the mean and the standard deviation, respectively, of the demand of $n$ divided by $r_n$.

Once the replicas of the clients are constructed by the procedure explained, the first step of the algorithm is to choose the first customer that forms the route. To do this, our goal is to minimize the distance expected for customers with large demands. To this end, for each agent, we consider both the expected distance traveled to attend it if it is urgent and the smallest hopper available is used and the expected distance if it is urgent and the largest hopper is used. Finally, we will choose the client that maximizes the difference between these distances. Formally, if $N^*$ is the set formed by the replicas of all the clients, defined as explained at

[2] $\lceil x \rceil$ denotes the smallest integer greater or equal than $x$.

the beginning of this section, and $n \in N^*$, we define:

$$Dif(n) = 2d_{0,n}\left(\max_{t\in T, h\in H_t}\{\mathbb{P}[\mathbf{O}_n > c_t^h]\} - \min_{t\in T, h\in H_t}\{\mathbb{P}[\mathbf{O}_n > c_t^h]\}\right)\mathbb{P}[\mathbf{U}_n = 0].$$

We will select the customer $n$ such that

$$n \in \{n^* \in N^* \,:\, Dif(n^*) = \max_{n'\in N^*} Dif(n')\}.$$

Once $n \in N^*$ has been selected, in a second step a route is created from the deposit to the client $n$ and from this it is returned to the deposit. To do this, we choose the vehicle $t_n$ and the hopper in it, $h_n$, more suitable, that is, such that:

$$\mathbb{P}[\mathbf{O}_n > c_{h_n}^{t_n}] = \min\{\mathbb{P}[\mathbf{O}_n > c_h^t] \,:\, t \in T,\, h \in H_t\}.$$

Regarding the filling process, the hopper $h_n \in H_{t_n}$ will be loaded with $q_n$ (quantil of order $p_n$ of $\mathbf{O}_n$, with $p_n = \mathbb{P}[\mathbf{U}_n = 0]$) provided that $q_n \leq c_{h_n}^{t_n}$. If $q_n > c_{h_n}^{t_n}$ the charge will be equal to $c_{h_n}^{t_n}$, that is, the hopper will be filled completely.

With this procedure, we take into account the two objectives of our optimization problem that are to maximize the amount of feed delivered and to minimize the total transport cost.

In the third stage, another customer replica is selected from which $n$ is replica. If there is no such replica, for each $n' \in N^*\backslash\{n\}$, we calculate the following amount, which is the saving obtained if the client $n'$ is inserted in the currently considered route:

$$S_{n,n'}^{h_n,t_n} = d_{n,0} + d_{0,n'} - d_{n,n'} + \lambda I_{n'}^{t_n,h_n}, \tag{6.19}$$

such that:

$$I_{n'}^{t_n,h_n} = 2d_{0,n'}\left(\min_{\substack{t'\in T\backslash\{t_n\}\\ h\in H_{t'}}}\{\mathbb{P}[\mathbf{O}_{n'} > c_h^{t'}]\} - \min_{h\in H_{t_n}\backslash\{h_n\}}\{\mathbb{P}[\mathbf{O}_{n'} > c_h^{t_n}]\}\right)\mathbb{P}[\mathbf{U}_{n'} = 0]. \tag{6.20}$$

Note that in (6.20), we evaluate the difference between the expected distance that is traveled by serving the client $n'$ in the largest hopper of a truck other than $t_n$ if it is urgent, and the expected distance when serving it in the largest free hopper of $t_n$ if it is urgent. In (6.19), this amount is weighted by a factor $\lambda \in [0, +\infty)$.

That is to say, when evaluating the saving obtained by inserting a new client in a route, we consider distances traveled by the truck and expected distances due to having to return to the warehouse, taking into account the restriction that prevents mixing products and the urgencies of the orders.

Then, the client that maximizes the saving as defined in (6.19) will be chosen. Once a new client $n'$ has been selected by one of the methods explained (that is, taking another replica of a client or taking the one that maximizes the saving) it will be inserted in the route to be visited after $n$. This new customer is served by assigning the load in a truck hopper following the same procedure as with the first customer.

To finish this stage, we have to verify that the so-called capacity restrictions are verified as well as the maximum distance that each vehicle can travel. If affirmative, the client $n'$ is definitely added to the truck route $t_n$. In another case, this stage is restarted eliminating $n'$ from the set of candidates to be inserted in said route.

The process of inserting customers on the route is repeated while the truck has free hoppers. Keep in mind that, in general, to insert a new client once we have two or more customers per route (and to simplify the process), the savings $S_{n',n}^{h_n,t_n}$ are considered, where $n$ represents the first client currently visited when leaving the deposit and the savings $S_{n,n'}^{h_n,t_n}$, where $n$ represents the last client currently visited, before arriving at said deposit. In both cases, $n'$ represents any client of $N^*$ who has not been assigned a route. Finally, once the route of a truck has been completely designed, the rest will be followed in a similar way.

Schematically, the algorithm proceeds as follows:

1. Make replicas of customers taking into account their demands.
   (a) While there are unattended customers:
      i. Select the customer $n \in N^*$ who maximize the value $Dif(n)$.
      ii. Create the route $r := (0, n, 0)$ for the vehicle $t_n \in T$ and allocate the load.
      iii. If there are unattended customers:
         A. Select a new replica of the same client or the best customer $n' \in N^*$ that maximizes $S_{n,n'}^{h_n,t_n}$ or $S_{n',n}^{h_n,t_n}$.
         B. If capacity and distance constraints are satisfied, insert $n'$ in the route and return to (iii).
         C. Otherwise, discard $n'$ and go back to (iii).
         D. End (iii) when the vehicle is full or there are no feasible insertions.

iv. Delete the vehicle $t_n$ and the allocated customers. Return to (a).

2. Conclude.

### 6.5.2 Tabu search

In this section, we propose an adaptation of the tabu search that allows obtaining a better solution than that provided by the constructive algorithm presented in the previous section.

We know that tabu search (Glover, 1989 and Glover, 1990) is a local search technique that manages to improve its performance by means of a memory structure consisting in that once a possible solution is obtained it is not considered again in the immediate iterations of the algorithm. As a local search technique, it makes use of a neighborhood structure that allows moving from one solution to another, until a certain stop rule is verified. There are different variants in the literature and we have adapted the variant proposed in Osman (1993).

Our tabu search algorithm starts with the selection of 3 parameters $\sigma$, $k$ and $m$. $\sigma$ is a number of random selected routes, $k$ the maximum number of possible exchanges between two routes and $m$ the maximum number of iterations. We also need an initial solution to our problem, which is obtained with the constructive algorithm of the previous section.

Note that the neighborhood structure that we are going to use is based on the so-called $k$-exchanges. More precisely, consider $r_1$ and $r_2$ two different routes, assigned to two respective trucks, of the same solution. A $k$-exchange consists of transferring at most $k$ clients from $r_1$ to $r_2$ and at most $k$ clients from $r_2$ to $r_1$. It must be taken into account that the number of clients that each route sends to the other is not necessarily the same, but both amounts must be less than $k$. Also one, and only one, of these routes may not send any clients.

Once the starting solution is selected, it is considered as the current solution and the algorithm proceeds as follows. In each iteration, $\sigma$ routes are arbitrarily chosen in the current solution. A solution neighboring the current solution is obtained by making a $k$-interchange on two routes belonging to the $\sigma$ routes chosen. All possible solutions neighboring the current one are considered and that they verify the restrictions of the problem. The one with the shortest total distance is chosen, which becomes the current solution and it is passed to a new iteration.

At the end of each iteration, information about the exchange made is stored in the tabu list. In particular, each client that has moved from one route to another is stored. This client is prevented from moving to another route (the original or any other route) for 3 iterations.

It must be borne in mind that an iteration can conclude taking as a current solution one that does not provide a total distance less than those obtained up to that moment. This type of exchanges are allowed to avoid local optimum, however, they can be done, at most, in 3 iterations.

Next, we present the algorithm schematically:

1. Start with an initial solution provided by the constructive algorithm as current solution and fixed $\sigma$, $k$ and $m$.
2. $iter = 0$ and $s.iter = 0$.
3. While $iter < m$:
   (a) $iter = iter + 1$.
   (b) Select $\sigma$ random routes of the current solution.
   (c) Make all possible $k$-exchanges.
   (d) Choose the feasible $k$-exchange that minimizes the total distance and update the current solution.
   (e) Add changed customers to the tabu list and if $iter \geq 4$ remove the first entry from the tabu list.
   (f) If the new solution does not improve the previous solutions, then $s.iter = s.iter + 1$ and if $s.iter = 3$ then conclude.
4. Conclude.
5. Select the best solution obtained in 3.

## 6.6 Examples

In this section, we illustrate the performance of our algorithms. To do this, we first consider a small fictional example and secondly a set of data taken from real life.

### 6.6.1 Fictitious example

Let's consider an example with five customers and a deposit. The distances in kilometers among the deposit and each customer, as well as among customers can be seen in the Table 6.1.

The average of orders of the different clients are $3.30, 2.95, 3.00, 3.02$, and $2.50$ tonnes, respectively. We are going to consider that the orders are random variables

| | Stockbreeders | | | | | |
|---|---|---|---|---|---|---|
| Stockbreeders | 0 | 1 | 2 | 3 | 4 | 5 |
| 0 | – | 28 | 69 | 64 | 27 | 17 |
| 1 | 28 | – | 67 | 62 | 20 | 20 |
| 2 | 69 | 67 | – | 7 | 74 | 58 |
| 3 | 64 | 62 | 7 | – | 69 | 53 |
| 4 | 27 | 20 | 74 | 69 | – | 25 |
| 5 | 17 | 20 | 58 | 53 | 25 | – |

Table 6.1: Travelling distances between pairs of stockbreeders and cooperative in the fictitious example.

with normal distribution. We will use different standard deviations to check the effect on the results. We assume that all the costumers have urgent orders of only one type of feed, then $P(\mathbf{U}_i = 0) = 0.95$ for each $i \in \{1, 2, 3, 4, 5\}$.

We take two similar trucks with five hoppers that can contain up to $3, 3.7, 3.8, 3.7$ and 3 tonnes of feed, respectively. The trucks cannot carry more than 11.6 tonnes, and each truck can deliver food to all the stockbreeders.

**Constructive heuristic**

We present below the results obtained for the constructive heuristic based on different values of the parameters of the model. In particular, we provide the total distance of the designed routes, which is the sum of the distance traveled by the truck when traveling the assigned route plus the expected distance for return to the deposit of those customers with urgent demands in the case where they are reached with an insufficient load.

The locally optimal solution that the constructive heuristic provides is, for a standard deviation equals to 0.1, where there is no necessary to make order divisions, as follows:

- The truck 1 goes from the deposit to costumer 1, then to 2, and then to 3, and it carries 3.46 tonnes of feed in hopper 4 to customer 1, 3.12 tonnes of feed in hopper 2 to customer 2 and 3.17 tonnes of feed in hopper 3 to customer 3.

- The truck 2 goes from the deposit to costumer 5 and then to 4, and it carries 2.66 tonnes of feed in hopper 2 to costumer 5 and 3.18 tonnes of feed in hopper 3 to customer 4.

With this solution, the total distance traveled is 255.5 km. In this case the solution does not vary although we use different values for the parameter $\lambda$ that appears in the saving definition used when designing the algorithm (see (6.19)).

In the following cases it is necessary to replicate customers given the greater variability in the demand. We will consider two situations, with and without replicas.

The locally optimal solution that the constructive heuristic provides is, for a standard deviation equals to 0.5 and $\lambda = 0$, without replicas, as follows:

- The truck 1 goes from the deposit to costumer 3, then to 2, and then to 1, and it carries 3.7 tonnes of feed in hopper 2 to 3, 3.77 tonnes of feed in hopper 3 to 2 and 3.7 tonnes of feed in hopper 4 to 1.
- The truck 2 goes from the deposit to costumer 5 and then to 4, and it carries 3.32 tonnes of feed in hopper 2 to 5 and 3.8 tonnes of feed in hopper 3 to 4.

With this solution, the total distance traveled is 269.07 km.

The locally optimal solution that the constructive heuristic provides is, for a standard deviation equals to 0.5 and $\lambda = 1$, without replicas, as follows:

- The truck 1 goes from the deposit to costumer 3, then to 2, and then to 5, and it carries 3.7 tonnes of feed in hopper 2 to 3, 3.77 tonnes of feed in hopper 3 to 2 and 3.32 tonnes of feed in hopper 4 to 5.
- The truck 2 goes from the deposit to costumer 4 and then to 1, and it carries 3.7 tonnes of feed in hopper 2 to 4 and 3.8 tonnes of feed in hopper 3 to 1.

The total distance traveled is 253.56 km.

For last, the locally optimal solution that the constructive heuristic provides is, for a standard deviation equals to 0.5 and any $\lambda$, with replicas, as follows:

- The truck 1 goes from the deposit to costumer 3, then to 2, and then to 1, and it carries 1.91 and 1.91 tonnes of feed in hoppers 4 and 2 to 3, 3.77 tonnes of feed in hopper 3 to 2 and 2.06 tonnes of feed in hopper 1 to 1.
- The truck 2 goes from the deposit to costumer 4 then to 1 and 5, and it carries 1.91 and 1.91 tonnes of feed in hoppers 1 and 4 to customer 4, 2.06 tonnes of feed in hopper 2 to 1 and 3.31 tonnes in hopper 3 to 5.

The total distance traveled is 270.5 km.

### Tabu search

In this section, the results of the tabu search algorithm applied to our so-called fictitious example are shown. With respect to the parameters of the algorithm, we take $\sigma = 2$, $k = 1$, and $m = 50$, because it is a case with few clients and trucks.

If we start from the solution obtained with the constructive heuristic, for the case in which the standard deviation of the demand variable is 0.1, and we apply the tabu search algorithm we get:

- The truck 1 goes from the deposit to costumer 1 and then to 4, and it carries 3.46 tonnes of feed in hopper 2 to customer 1 and 3.18 tonnes of feed in hopper 4 to customer 4.
- The truck 2 goes from the deposit to costumer 2, then to 3, and then to 5, and it carries 3.12 tonnes of feed in hopper 4 to customer 2, 3.17 tonnes of feed in hopper 2 to customer 3 and 2.66 tonnes of feed in hopper 5 to customer 5.

With this solution, the total distance is 241.49 km.

From the solution obtained for a standard deviation of 0.5, without replicas, the tabu search obtains the following solution:

- The truck 1 goes from the deposit to costumer 1 and then to 4, and it carries 3.8 tonnes of feed in hopper 3 to customer 1 and 3.7 tonnes of feed in hopper 4 to customer 4.
- The truck 2 goes from the deposit to costumer 2, then to 3 and then to 5, and it carries 3.7 tonnes of feed in hopper 4 to customer 2, 3.8 tonnes of feed in hopper 3 to customer 3 and 3.32 tonnes of feed in hopper 2 to customer 5.

In this solution, the total distance is 242.57 km.

Finally, in the case where the standard deviation is equal to 0.5 and replications of the clients are made, the tabu search provides the following solution:

- The truck 1 goes from the deposit to costumer 1 and then to 4, and it carries 2.06 and 2.06 tonnes of feed in hoppers 5 and 1 to costumer 1 and 1.91 and 1.91 tonnes of feed in hoppers 2 and 4 to customer 4.
- The truck 2 goes from the deposit to costumer 2, then to 3 and then to 5, and it carries 3.77 tonnes of feed in hopper 3 to customer 2, 1.91 and 1.91 tonnes of feed in hoppers 5 and 1 to customer 3 and 3.32 tonnes of feed in hopper 2 to customer 5.

In this case, the total distance is 241.50 km.

### Deterministic case

Guitián de Frutos and Casas-Méndez (to appear) have solved the deterministic problem when they use the mean of the random variables as the order of each customer respectively.

The locally optimal solution is as follows:

- The truck 1 goes from the deposit to costumer 5, then to 3, and then to customer 2, where it carries 2.50 tonnes of feed in hopper 2 to 5, 1.55 and 1.45 tonnes of feed in hoppers 3 and 4, respectively, to 3. Finally, 1.47 and 1.47 tonnes in hopper 1 and 5 to customer 2.
- The truck 2 goes from the deposit to costumer 4 and finally to 1. It carries 1.51 and 1.51 tonnes of feed in hoppers 1 and 5 to 4 and 3.30 tonnes to 1 in the hopper 3.

With this solution, the distance traveled is 221 km.

### Comparing solutions

| | Algorithm | | |
|---|---|---|---|
| SD | Constructive heuristic | Tabu search | Deterministic case |
| 0.1 | 255.5 km | 241.49 km | |
| 0.5 (no replicas) | 269.07 km ($\lambda = 0$)<br>253.56 km ($\lambda = 1$) | 242.57 km | 221 km |
| 0.5 (replicas) | 270.5 km | 241.50 km | |

Table 6.2: Different solutions obtained with the fictitious example (SD means standard desviation).

Table 6.2 shows a summary of the results obtained in this section. In particular, it reflects the total distance corresponding to the deterministic case and to the application of the two algorithms in the stochastic case and for different values in the standard deviation of the demand.

In the deterministic case, the total distance is 221 km. In the stochastic case with a standard deviation of 0.1, applying the constructive heuristic, the distance traveled is 235 km and the expected distance is 20.5 km (corresponding to possible returns to the deposit caused by the impossibility of fully satisfying the demands of a client). We observe an increase in the traveled distance of a 6.33%. When applying the improvement tabu algorithm, the distance traveled becomes 221 km, as in the deterministic case, and the expected distance is 20.49, which also means an improvement, although slight, with respect to the constructive heuristic.

For a standard deviation of 0.5 and the parameter $\lambda$ equal to 0 (if we do not consider replicas), the constructive heuristic obtains a traveled distance of 235 km and 34.07 km of expected distance. As in the previous case, we obtain a worsening of 6.33% with respect to the deterministic model. But this situation is rectified with the tabu search, because we get a distance traveled of 221 km and 21.57 km of expected distance. We are also getting a 36.7% improvement over the expected distance. If we consider $\lambda$ equal to 1, the constructive heuristic obtains a traveled distance of 221 km and an expected distance of 32.56 km. In this case, the tabu search only improves the expected distance, with respect to the constructive heuristic, by 33.75%. However, if client replicas are allowed, the solution obtained is the same for any $\lambda$ value. It is in this case when the worst results are obtained for the constructive heuristic. This is mainly due to the fact that both trucks visit the same client causing an unnecessary extra distance. The situation is corrected with the tabu search, and it even improves the solution obtained for the case without replicas. This is because when we replicate the customers, the probability of satisfying their demands is higher, decreasing the expected distances.

### 6.6.2 Real data

Now we are going to consider some real data provided by an agricultural cooperative. The information obtained corresponds to one working day of the company. The company would like to visit 19 customers scattered throughout Galicia from the depot. We do not know the demand of each client but, with data from past deliveries, we can estimate that demand. We know the mean and standard deviation of each demand and we are going to assume that it follows a normal distribution with these parameters. As we know the number of days since the last visit to each client we can obtain a probability urgency value between 0 and 1, such that 0 is not urgent and 1 is urgent. We will also assume that any value greater than 0.9 is an urgent order that must be placed on the same day.

For the different deliveries we have a fleet of 2 trucks. The first truck has five hoppers with a capacity of $4, 3, 1.7, 4.5$ and 3 tonnes and the second truck of $3, 3.7, 3.8, 3.7$ and 3 tonnes. In addition, the maximum load that can be transported by both trucks is 15.3 tonnes. Finally we know all the distances between customers and with the depot. Complete data can be obtained if requested from the authors.

First, we will solve our problem with the constructive heuristic. We will use different values for $\lambda$ and standard deviations to carry out a sensitivity analysis. The urgency values are 0.95 for all costumers. The results are shown in the Tables 6.3, 6.4, and 6.5.

The results corresponding to standard deviations of 1 or 3.04 do not have many differences between them, although with high values of $\lambda$ it is when the best results are obtained.

When working with a standard deviation of 0.5 is when more differences are observed in the results obtained, depending on the value of $\lambda$ considered. If we look at the total distance we can see differences of up to 13%, which is a value that can be considered relevant when dealing with a logistics problem.

It should be noted that with a value of $\lambda$ equal to 5 and a standard deviation of 1, we obtain a solution in which the distance traveled is 519 km., the expected distance is 113.24 km. and the total distance is 633.24 km. In this case, the amount delivered is 134.92 tonnes.

| | Results | | | |
|---|---|---|---|---|
| Parameters $\lambda = 0$ | Distance Traveled | Expected Distance | Total Distance | Amount Delivered |
| $sd = 0.5$ | 388 | 379.58 | 767.58 | 115.07 |
| $sd = 1$ | 507 | 158.58 | 665.58 | 130.68 |
| $sd = 3.04$ | 623 | 121.09 | 744.09 | 180.66 |

Table 6.3: Solutions obtained for the real data with the constructive heuristic and a value of urgency equal to 0.95 (a).

| | Results | | | |
|---|---|---|---|---|
| Parameters $\lambda = 1$ | Distance Traveled | Expected Distance | Total Distance | Amount Delivered |
| $sd = 0.5$ | 385 | 280.20 | 665.20 | 119.01 |
| $sd = 1$ | 526 | 117.90 | 643.90 | 134.10 |
| $sd = 3.04$ | 622 | 118.89 | 740.89 | 182.06 |

Table 6.4: Solutions obtained for the real data with the constructive heuristic and a value of urgency equal to 0.95 (b).

Now we are going to consider that the emergency variables follow a uniform distribution, that is to say, we will take for each variable a value generated with a uniform in $(0, 1)$. The new results are shown in the Tables 6.6, 6.7, and 6.8.

With the new distribution of the emergency variables, we can see that the expected distances are smaller than in the previous case. The reason is that there are fewer customers with urgent demands that are far from the deposit. This has

| | Results | | | |
|---|---|---|---|---|
| Parameters $\lambda = 2$ | Distance Traveled | Expected Distance | Total Distance | Amount Delivered |
| $sd = 0.5$ | 529 | 136.55 | 665.55 | 121.90 |
| $sd = 1$ | 539 | 116.53 | 655.53 | 134.99 |
| $sd = 3.04$ | 612 | 117.12 | 729.12 | 183.78 |

Table 6.5: Solutions obtained for the real data with the constructive heuristic and a value of urgency equal to 0.95 (c).

as a consequence that in each route we can visit more customers, that the amount delivered is less and also the distance traveled.

If the standard deviation is equal to 0.5, we obtain better results with $\lambda$ equal to 1, with an improvement of approximately 23% compared to the solution for $\lambda$ equal to 2. For a standard deviation of 1, the best solution is obtained when $\lambda$ is equal to 2, with an improvement of approximately 24% compared to the solution for $\lambda$ equal to 0. Finally, for a standard deviation of 3.04 we get the best solution for $\lambda$ equal to 0.

| | Results | | | |
|---|---|---|---|---|
| Parameters $\lambda = 0$ | Distance Traveled | Expected Distance | Total Distance | Amount Delivered |
| $sd = 0.5$ | 392 | 8.92 | 400.92 | 104.68 |
| $sd = 1$ | 492 | 3.72 | 495.72 | 111.03 |
| $sd = 3.04$ | 495 | 6.12 | 501.12 | 121.27 |

Table 6.6: Solutions obtained for the real data with the constructive heuristic and a uniform emergency (a).

We explain below the results obtained when applying tabu search. In the Table 6.9 we consider a value for the urgency of 0.95. If we compare with the results obtained when applying the constructive algorithm, for a standard deviation of 0.5, we obtain that the distance traveled may be higher in some cases, while the expected distance and the total distance always decrease. Comparing with the best result obtained with the constructive algorithm, the tabu search provides an improvement in the total distance of 11%. For a standard deviation of 1, the results are similar and in this case, comparing with the best result obtained with the constructive algorithm, the tabu search provides an improvement in the total distance of 3%. Finally, with a standard deviation of 3.04, the tabu search

| | Results | | | |
|---|---|---|---|---|
| Parameters $\lambda = 1$ | Distance Traveled | Expected Distance | Total Distance | Amount Delivered |
| $sd = 0.5$ | 389 | 2.10 | 391.10 | 105.28 |
| $sd = 1$ | 369 | 12.11 | 381.11 | 108.60 |
| $sd = 3.04$ | 515 | 7.36 | 522.36 | 121.02 |

Table 6.7: Solutions obtained for the real data with the constructive heuristic and a uniform emergency (b).

| | Results | | | |
|---|---|---|---|---|
| Parameters $\lambda = 2$ | Distance Traveled | Expected Distance | Total Distance | Amount Delivered |
| $sd = 0.5$ | 505 | 2.10 | 507.10 | 107.13 |
| $sd = 1$ | 375 | 3.06 | 378.06 | 108.27 |
| $sd = 3.04$ | 496 | 9.00 | 505.00 | 122.93 |

Table 6.8: Solutions obtained for the real data with the constructive heuristic and a uniform emergency (c).

always reduces the distance traveled, the expected distance and the total distance. Comparing with the best result obtained with the constructive algorithm, the tabu search provides an improvement in the total distance of 4%.

The results obtained when the urgency values are generated using a uniform distribution are shown in the Table 6.10. Comparing with the best results obtained when using the constructive algorithm, decreases in the total distance of 9%, 3% and 8% are observed, with standard deviations in the demands of 0.5, 1 and 3.04 respectively. Noteworthy are the decreases in the expected distance, which reach a 55% reduction when the standard deviation is 3.04.

## 6.7 Computational experiments

In this section, we create an MC-VRPSD benchmark suite for the research community, because to the best of our knowledge, no instances are publicly available for the characteristics of our problem. That is why we will modify the instances created by Christofides et al. (1979). There are 14 problems that contain between 50 and 199 customers. The 5 first instances no include a route length restriction.

| Parameters | Results | | | |
|---|---|---|---|---|
| | Distance Traveled | Expected Distance | Total Distance | Amount Delivered |
| $sd = 0.5$ | 483 | 108.97 | 591.97 | 122.24 |
| $sd = 1$ | 524 | 97.01 | 621.01 | 133.42 |
| $sd = 3.04$ | 606 | 92.77 | 698.77 | 184.70 |

Table 6.9: Solutions obtained for the real data with the tabu search and a value of urgency equal to 0.95.

| Parameters | Results | | | |
|---|---|---|---|---|
| | Distance Traveled | Expected Distance | Total Distance | Amount Delivered |
| $sd = 0.5$ | 352 | 2.10 | 354.10 | 105.86 |
| $sd = 1$ | 364 | 2.23 | 366.23 | 109.46 |
| $sd = 3.04$ | 461 | 2.76 | 463.76 | 116.50 |

Table 6.10: Solutions obtained for the real data with the tabu search a uniform emergency.

The code was created using the program `R` x64 3.5.0.0. and running in a PC with 3.6 Ghz Intel Core i7-9700K, 64 GB of RAM, and Windows 10 x64.

In order to adapt to the characteristics of our problem, we will consider that the orders of each customer have normal distribution. The mean is given by the exact value of each example, and the standard deviation is considered as 0.3 multiplied by the mean. For all problems, we will use the same vehicle with hoppers capacity equal to 50, 40, 40, 30, and 30 tonnes respectively. Each problem has its own maximum allowable load of the trucks. The complete data of these benchmarks can be requested from the authors.

In Table 6.11, we show the results corresponding to the resolution of all instances through the constructive heuristic when we considered all orders urgents. The different columns collect the number of clients of the problem, the distance traveled by the vehicle, the expected distance in case of return to the deposit and the total distance. Finally, we have the amount delivered and the time that we needed to get the solution in seconds. In brackets, we find the value of $\lambda$ with which the best solution is obtained. The solutions with or without route length restriction are the same, because each route only visits a maximum of 5 clients (one per hopper) so this restriction is never broken.

In Table 6.12, we have resolved the above problems using the tabu search

| Instance | $n$ | Results | | | | |
|---|---|---|---|---|---|---|
| | | Distance Traveled ($\lambda$) | Expected Distance | Total Distance | Amount Delivered | Time |
| vrpnc1 | 50 | 831.44 (2) | 121.25 | 952.69 | 1159.75 | 0.06 |
| vrpnc2 | 75 | 1258.96 (1) | 188.99 | 1447.95 | 2032.15 | 0.11 |
| vrpnc3 | 100 | 1560.47 (10) | 249.47 | 1809.94 | 2177.46 | 0.17 |
| vrpnc4 | 150 | 2113.23 (1) | 369.64 | 2482.87 | 3336.82 | 0.41 |
| vrpnc5 | 199 | 2664.90 (1) | 482.80 | 3147.70 | 4756.35 | 0.97 |
| vrpnc6 | 50 | 831.44 (2) | 121.25 | 952.69 | 1159.75 | 0.05 |
| vrpnc7 | 75 | 1258.96 (1) | 188.99 | 1447.95 | 2032.15 | 0.11 |
| vrpnc8 | 100 | 1560.47 (10) | 249.47 | 1809.94 | 2177.45 | 0.17 |
| vrpnc9 | 150 | 2113.23 (1) | 369.64 | 2482.87 | 3336.82 | 0.41 |
| vrpnc10 | 199 | 2664.90 (1) | 482.80 | 3147.70 | 4756.35 | 0.97 |
| vrpnc11 | 120 | 3064.89 (0) | 611.97 | 3676.86 | 2053.50 | 0.25 |
| vrpnc12 | 100 | 1590.32 (1) | 288.55 | 1878.87 | 2703.15 | 0.16 |
| vrpnc13 | 120 | 3064.89 (0) | 611.97 | 3676.86 | 2053.50 | 0.16 |
| vrpnc14 | 100 | 1590.32 (1) | 288.55 | 1878.87 | 2703.15 | 0.16 |

Table 6.11: Instances solved with the constructive heuristic.

algorithm. To this purpose, we have considered a limit of 5000 iterations and we only solve the problems that do not have route length restriction. We can observe in the column Total Distances, in brackets, the improvement achieved respect to the solution obtained by the constructive heuristic. Appendix 6.9 presents the solutions of the testcases in detail.

## 6.8 Conclusions and final remarks

In this paper, we introduce a general class of vehicle routing problems in which the fleet is heterogeneous with respect to the capacity and the vehicles are compartmentalized. The fleet is responsible for distributing various products among a set of customers, the demands are stochastic, each compartment can not contain different products or products of different customers, some vehicles can not access certain customers and, finally, the orders of some customers are considered urgent. We propose a tabu algorithm for the resolution that starts from a solution obtained through a constructive procedure.

In our opinion, the main contribution of this work is the proposal of a model that represents situations that appear in real life considering in a new way various

| Instance | $n$ | Results | | | | |
|---|---|---|---|---|---|---|
| | | Distance Traveled | Expected Distance | Total Distance (%) | Amount Delivered | Time |
| vrpnc1 | 50 | 752.70 | 120.03 | 872.73 (8.39%) | 1160.50 | 355.31 |
| vrpnc2 | 75 | 1129.98 | 181.84 | 1311.82 (9.40%) | 2036.75 | 200.47 |
| vrpnc3 | 100 | 1432.13 | 250.01 | 1682.14 (7.06%) | 2177.00 | 429.53 |
| vrpnc4 | 150 | 1977.12 | 368.45 | 2345.57 (5.53%) | 3337.50 | 469.11 |
| vrpnc5 | 199 | 2533.26 | 480.47 | 3013.73 (4.26%) | 4758.10 | 510.70 |
| vrpnc11 | 120 | 2786.79 | 613.50 | 3400.29 (7.52%) | 2053.00 | 442.46 |
| vrpnc12 | 100 | 1509.30 | 289.33 | 1798.63 (4.27%) | 2703.02 | 408.22 |

Table 6.12: Instances solved with the tabu search.

restrictions that are produced. In addition, the algorithm of Clarke and Wright is generalized and combined with a tabu search in a context of problems of routes with stochastic demands and vehicles with compartments, in which existing literature is scarce. In addition to illustrating the algorithms with small examples and see their operation in a real case, we have designed a collection of testcases for use in future research in this field.

These algorithms are programmed in `R` language so they can be integrated into a more complete tool to support the decision of the managers of the company, along with others able, for example, to forecast consumption on farms or generate reports of interest. The creation of simple interfaces that facilitate the task of the managers or allow simulations for advanced users is also viable through `R` libraries. Such tools suppose an advantage for the companies, since they allow to automate certain tasks in front of other methodologies "human based" less efficient or certain commercial tools less transparent and difficult to adapt.

The work is motivated from the tasks developed by an animal feed factory. This company is located in a Spanish region of about 13,000 km$^2$ in which there are more than 15 companies of this type. The model is applicable to other types of companies such as those that collect milk or distribute cereals or gasoline and the algorithms can generate savings in the transportation performed by these companies.

Regarding possible future lines of research, it may be interesting to adapt to the presented model another type of well-known metaheuristics such as simulated annealing or genetic algorithms. Regarding alternative models, it is worth considering the so-called "truck and trailer problem" (cf. Derigs et al., 2013) to deal with accessibility restrictions, when such vehicles are available to companies and can reduce transport costs.

# References


Ambrosino D, Sciomachen A (2007). A food distribution network problem: a case study. IMA Journal of Management Mathematics 18, 33-53.

Archetti C, Campbell AM, Speranza MG (2016). Multicommodity vs. single-commodity routing. Transportation Science 50, 461-472.

Avella P, Boccia M, Sforza A (2004). Solving a fuel delivery problem by heuristic and exact approaches. European Journal of Operational Research 152, 170-179.

Berhan E, Beshah B, Kitaw D, Abraham A (2014). Stochastic vehicle routing problem: A literature survey. Journal of Information and Knowledge Management 13, 1450022.

Bertsimas DJ, Simchi-Levi D (1996). A new generation of vehicle routing research: robust algorithms, addressing uncertainty. Operations Research 44, 286-303.

Brown GG, Ellis C, Graves GW (1987). Real-time, wide area dispatch of mobil tank trucks. Interfaces 17, 107-120.

Brown GG, Graves GW (1981). Real-time dispatch of petroleum tank trucks." Management Science 27, 19-32.

Caramia M, Guerriero F (2010). A milk collection problem with incompatibility constraints. Interfaces 40, 130-143.

Chajakis ED, Guignard M (2003). Scheduling deliveries in vehicles with multiple compartments. Journal of Global Optimization 26, 43-78.

Christofides N, Mingozzi A, Toth P, Sandi C (1979). Combinatorial Optimization, chapter 11. Chichester: John Wiley.

Clarke G, Wright J (1964). Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 12, 568-581.

Coelho LC, Laporte G (2015). Classification, models and exact algorithms for multicompartment delivery problems. European Journal of Operational Research 242, 854-864.

Cordeau JF, Laporte G, Savelsbergh MW, Vigo D (2007). Vehicle routing. In Handbooks in Operations Research and Management Science (Barnhart C, Laporte G, eds.) North-Holland: Elsevier 14, 367-428.

Dantzig GB, Ramser JH (1959). The truck dispatching problem. Management Science 6, 80-91.

Derigs U, Gottlieb J, Kalko J, Piesche M, Rothlauf F, Vogel U (2011). Vehicle routing with compartments: Applications, modelling and heuristics. OR Spectrum

33, 885-914.

Derigs U, Pullmann M, Vogel U (2013). Truck and trailer routing problems, heuristics and computational experience. Computers and Operations Research 40, 536-546.

Dror M, Ball MO, Golden BL (1985). Computational comparison of algorithms for inventory routing. Annals of Operations Research 4, 3-23.

Dror M (2016). Vehicle routing with stochastic demands: models and computational methods. International Series in Operations Research and Management Science 46, 625-649.

Dror M, Trudeau P (1986). Stochastic vehicle routing with modified savings algorithm. European Journal of Operational Research 23, 228-235.

El Fallahi A, Prins C, Calvo RW (2008). A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. Computers and Operations Research 35, 1725-1741.

Gendreau M, Laporte G, Séguin R (1996). Stochastic vehicle routing. European Journal of Operational Research 88, 3-12.

Glover F (1989). Tabu search-part I. ORSA Journal on Computing 1, 190-206.

Glover F (1990). Tabu search-part II. ORSA Journal on Computing 2, 4-32.

Goodson JC (2015). A priori policy evaluation and cyclic-order-based simulated annealing for the multicompartment vehicle routing problem with stochastic demands. European Journal of Operational Research 241, 361-369.

Guitián de Frutos RM, Casas-Méndez, BV To appear. Routing problems in agricultural cooperatives: A model for optimization of transport vehicle logistics. IMA Journal of Management Mathematics. DOI: 10.1093/imaman/dpy010.

Henke T, Speranza MG, Wäscher G (2018). A branch-and-cut algorithm for the multi-compartment vehicle routing problem with exible compartment sizes. Annals of Operations Research. DOI: 10.1007/s10479-018-2938-4.

Hsu C-I, Hung S-F, Li H-C (2007). Vehicle routing problem with time-windows for perishable food delivery. Journal of Food Engineering 80, 465-475.

Huang SH (2015). Solving the multi-compartment capacitated location routing problem with pickup-delivery routes and stochastic demands. Computers and Industrial Engineering 87, 104-113.

Kuznietsov KA, Gromov VA, Skorohod, VA (2016). Cluster-based supply chain logistics: a case study of a Ukrainian food distributor. IMA Journal Managment Mathematics 28, 553-578.

Laporte G (2009). Fifty years of vehicle routing. Transportation Science 43, 408-416

Mendoza JE, Castanier B, Guéret C, Medaglia AL, Velasco N (2010). A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. Computers and Operations Research 37, 1886-1898.

Mendoza JE, Castanier B, Guéret C, Medaglia AL, Velasco N (2011). Constructive heuristics for the multicompartment vehicle routing problem with stochastic demands. Transportation Science 45, 346-363.

Muyldermans L, Pang G (2010). On the benefits of co-collection: Experiments with a multi-compartment vehicle routing algorithm. European Journal of Operational Research 206, 93-103.

Oppen, J. and L∅kketangen, A (2008). A tabu search approach for the livestock collection problem. Computers and Operations Research 35, 3213-3229.

Osman I (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. Annals of Operations Research 41, 421-451.

Ostermeier M, Hübner A (2018). Vehicle selection for a multi-compartment vehicle routing problem. European Journal of Operational Research 269, 682-694.

Oyola J, Arntzen H, Woodru D (2016). The stochastic vehicle routing problem, a literature review. www.optimizationonline.org/DB FILE/2016/01/5299.pdf.

Pandelis DG, Kyriakidis EG, Dimitrakos TD (2012). Single vehicle routing problems with a predefined customer sequence, compartmentalized load and stochastic demands. European Journal of Operational Research 217, 324-332.

Pollaris H, Braekers K, Caris A, Janssens G, Limbourg S (2015). Vehicle routing problems with loading constraints: State-of-the-art and future directions. OR Spectrum 37, 297-330.

Silvestrin PV, Ritt M (2017). An iterated tabu search for the multi-compartment vehicle routing problem.” Computers and Operations Research 81, 192-202.

Tatarakis A, Minis I (2009). Stochastic single vehicle routing with a predefined customer sequence and multiple depot returns. European Journal of Operational Research 197, 557-571.

van der Bruggen R, Gruson L, Salomon M (1995). Reconsidering the distribution structure of gasoline products for a large oil company. European Journal of Operational Research 81, 460-473.

Xia Y, Fu Z, Pan L, Duan F (2018). Tabu search algorithm for the distance-constrained vehicle routing problem with split deliveries by order. Plos One 13, e0195457. DOI: 10.1371/journal.pone.0195457.

## 6.9 Appendix

This appendix contain the real solutions produced in this study for the test problems. The input data are not included due to the limited space, but can be obtained from the authoros. For every problem we use the following abbreviations:

| | |
|---|---|
| $n$ | = Index of customers. |
| $Q$ | = Total amount delivered. |
| $T.D$ | = Total distance traveled. |
| $D.T$ | = Distance traveled. |
| $E.D$ | = Expected distance traveled. |

In Tables 6.13, 6.14, 6.15, 6.16, 6.17, 6.18 and 6.19 we present the results obtained with the constructive algorithm. In Tables 6.20, 6.21, 6.22, 6.23, 6.24, 6.25 and 6.26 we present the results obtained with the tabu search algorithm. All routes start and end at the depot (0).

| | Results | | | | |
|---|---|---|---|---|---|
| $i$ | Distance Traveled | Expected Distance | Total Distance | Amount Delivered | Route |
| 1 | 61.55 | 8.54 | 70.09 | 98.34 | $4-17-15-37-12$ |
| 2 | 63.69 | 5.76 | 69.45 | 96.34 | $6-23-1-27-46$ |
| 3 | 89.38 | 12.73 | 102.11 | 127.34 | $14-25-24-43-7$ |
| 4 | 83.89 | 12.69 | 96.58 | 113.34 | $16-9-34-10-49$ |
| 5 | 76.10 | 11.66 | 87.76 | 104.34 | $22-31-26-8-48$ |
| 6 | 97.14 | 17.83 | 114.97 | 119.34 | $28-3-36-35-20$ |
| 7 | 86.17 | 13.92 | 100.09 | 111.34 | $30-50-21-29-2$ |
| 8 | 54.75 | 6.14 | 60.89 | 130.34 | $32-11-38-5-47$ |
| 9 | 120.09 | 17.04 | 137.13 | 127.67 | $39-33-45-44-41$ |
| 10 | 98.68 | 14.93 | 113.61 | 131.34 | $42-19-40-13-18$ |

Table 6.13: **Instance 1:** $n = 50$, $Q = 1159.75$, $T.D = 952.69$, $D.T = 831.44$, $E.D = 121.25$, $\lambda = 1$.

| $i$ | Results: Distance Traveled | Expected Distance | Total Distance | Amount Delivered | Route |
|---|---|---|---|---|---|
| 1 | 99.92 | 14.95 | 114.87 | 132.87 | $1 - 63 - 23 - 56 - 24$ |
| 2 | 80.91 | 11.20 | 92.11 | 139.87 | $5 - 47 - 74 - 62 - 33$ |
| 3 | 49.93 | 4.80 | 54.73 | 139.87 | $6 - 68 - 27 - 34 - 67$ |
| 4 | 69.03 | 8.23 | 77.26 | 137.87 | $7 - 8 - 54 - 46 - 52$ |
| 5 | 92.06 | 9.55 | 101.61 | 139.87 | $19 - 13 - 30 - 48 - 2$ |
| 6 | 93.99 | 18.61 | 112.60 | 130.90 | $37 - 20 - 70 - 36 - 21$ |
| 7 | 93.62 | 17.76 | 111.38 | 128.87 | $43 - 41 - 42 - 64 - 22$ |
| 8 | 70.26 | 10.47 | 80.73 | 138.87 | $49 - 3 - 44 - 32 - 40$ |
| 9 | 99.23 | 17.65 | 116.88 | 128.90 | $50 - 18 - 25 - 55 - 9$ |
| 10 | 55.18 | 5.67 | 60.85 | 129.87 | $51 - 16 - 17 - 12 - 26$ |
| 11 | 69.23 | 9.77 | 79.00 | 135.90 | $53 - 11 - 14 - 35$ |
| 12 | 61.23 | 9.50 | 70.73 | 130.87 | $57 - 15 - 29 - 45 - 4$ |
| 13 | 112.22 | 12.61 | 124.83 | 138.87 | $58 - 10 - 31 - 39 - 73$ |
| 14 | 106.13 | 17.92 | 124.05 | 112.87 | $60 - 71 - 69 - 61 - 28$ |
| 15 | 100 | 19.99 | 119.99 | 136.89 | $72 - 38 - 65 - 66 - 59$ |
| 16 | 6 | 0.3 | 6.3 | 28.97 | 75 |

Table 6.14: **Instance 2:** $n = 75$, $Q = 2032.15$, $T.D = 1447.95$, $D.T = 1258.96$, $E.D = 188.99$, $\lambda = 1$.

| | Results | | | | |
|---|---|---|---|---|---|
| $i$ | Distance Traveled | Expected Distance | Total Distance | Amount Delivered | Route |
| 1 | 65.20 | 7.14 | 72.34 | 117.97 | $12-4-26-40-58$ |
| 2 | 97.01 | 17.08 | 114.09 | 143.97 | $14-44-38-86-16$ |
| 3 | 77.78 | 12.48 | 90.26 | 96.97 | $15-43-37-98-59$ |
| 4 | 61.26 | 7.93 | 69.19 | 108.97 | $21-2-95-97-13$ |
| 5 | 75.95 | 12.28 | 88.23 | 96.97 | $22-41-57-42-87$ |
| 6 | 101.45 | 17.79 | 119.24 | 128.97 | $23-67-39-25-55$ |
| 7 | 98.54 | 16.80 | 115.34 | 120.97 | $45-46-47-36-48$ |
| 8 | 78.88 | 11.28 | 90.16 | 107.97 | $50-1-30-32-70$ |
| 9 | 43.22 | 4.10 | 47.32 | 92.97 | $53-6-89-52-27$ |
| 10 | 69.09 | 7.35 | 76.44 | 77.97 | $60-18-7-69-28$ |
| 11 | 112.78 | 19.09 | 131.87 | 113.97 | $63-11-64-49-19$ |
| 12 | 99.53 | 16.79 | 116.32 | 117.97 | $71-35-34-29-68$ |
| 13 | 63.64 | 12.42 | 76.06 | 101.97 | $73-74-56-75-72$ |
| 14 | 66.77 | 12.13 | 78.90 | 109.97 | $77-3-33-81-51$ |
| 15 | 113.62 | 18.51 | 132.13 | 108.97 | $78-9-65-66-20$ |
| 16 | 80.07 | 11.51 | 91.58 | 98.97 | $80-54-24-79-76$ |
| 17 | 68.38 | 11.56 | 79.94 | 104.97 | $82-8-83-84-5$ |
| 18 | 69.56 | 11.99 | 81.55 | 109.97 | $88-62-90-10-31$ |
| 19 | 43.24 | 8.34 | 51.58 | 106.97 | $96-99-93-92-94$ |
| 20 | 74.47 | 12.86 | 87.33 | 109.97 | $100-91-85-61-17$ |

Table 6.15: **Instance 3:** $n = 100$, $Q = 2177.46$, $T.D = 1809.94$, $D.T = 1560.47$, $E.D = 249.47$, $\lambda = 10$.

| $i$ | Results: Distance Traveled | Expected Distance | Total Distance | Amount Delivered | Route |
|---|---|---|---|---|---|
| 1 | 48.13 | 7.18 | 55.31 | 98.76 | 2 – 137 – 95 – 117 – 13 |
| 2 | 56.84 | 8.39 | 65.23 | 96.76 | 7 – 106 – 18 – 60 – 147 |
| 3 | 79.49 | 13.33 | 92.82 | 110.76 | 8 – 46 – 45 – 84 – 118 |
| 4 | 109.35 | 19.09 | 128.44 | 109.76 | 11 – 64 – 143 – 36 – 48 |
| 5 | 82.94 | 14.00 | 96.94 | 100.76 | 15 – 43 – 14 – 91 – 59 |
| 6 | 76.32 | 15.05 | 91.37 | 148.41 | 17 – 113 – 16 – 141 – 85 |
| 7 | 91.38 | 17.91 | 109.29 | 118.41 | 32 – 131 – 128 – 66 – 20 |
| 8 | 96.50 | 15.62 | 112.12 | 111.76 | 35 – 34 – 121 – 29 – 68 |
| 9 | 50.51 | 7.54 | 58.05 | 96.76 | 40 – 115 – 73 – 21 – 105 |
| 10 | 65.98 | 11.64 | 77.62 | 100.76 | 41 – 145 – 57 – 144 – 87 |
| 11 | 73.45 | 12.43 | 85.88 | 110.76 | 42 – 142 – 100 – 61 – 5 |
| 12 | 72.31 | 13.23 | 85.54 | 91.76 | 51 – 78 – 129 – 79 – 3 |
| 13 | 45.91 | 6.80 | 52.71 | 87.76 | 52 – 88 – 127 – 69 – 132 |
| 14 | 36.98 | 3.47 | 40.45 | 89.76 | 53 – 6 – 89 – 146 – 27 |
| 15 | 92.79 | 17.43 | 110.22 | 141.76 | 56 – 23 – 67 – 39 – 139 |
| 16 | 40.92 | 5.13 | 46.05 | 120.76 | 58 – 26 – 149 – 138 – 28 |
| 17 | 103.00 | 20.44 | 123.44 | 122.76 | 71 – 65 – 136 – 135 – 120 |
| 18 | 59.78 | 12.91 | 72.69 | 128.76 | 74 – 22 – 133 – 75 – 72 |
| 19 | 47.84 | 8.54 | 56.38 | 108.76 | 76 – 77 – 116 – 109 – 12 |
| 20 | 50.22 | 9.31 | 59.53 | 105.76 | 93 – 92 – 37 – 98 – 94 |
| 21 | 41.80 | 7.25 | 49.05 | 106.76 | 97 – 104 – 99 – 96 – 112 |
| 22 | 72.24 | 13.78 | 86.02 | 135.76 | 103 – 9 – 81 – 33 – 102 |
| 23 | 72.35 | 15.06 | 87.41 | 106.41 | 108 – 90 – 126 – 63 – 31 |
| 24 | 63.76 | 11.12 | 74.88 | 94.76 | 110 – 4 – 54 – 80 – 150 |
| 25 | 92.56 | 18.25 | 110.81 | 121.76 | 119 – 44 – 38 – 140 – 86 |
| 26 | 71.58 | 12.06 | 83.64 | 105.76 | 122 – 30 – 70 – 10 – 62 |
| 27 | 79.51 | 12.57 | 92.08 | 128.76 | 123 – 82 – 114 – 125 – 83 |
| 28 | 95.78 | 17.53 | 113.31 | 139.76 | 124 – 47 – 49 – 107 – 19 |
| 29 | 79.91 | 14.44 | 94.35 | 92.76 | 130 – 55 – 25 – 24 – 134 |
| 30 | 63.08 | 8.09 | 71.17 | 102.76 | 148 – 101 – 1 – 50 – 111 |

Table 6.16: **Instance 4:** $n = 150$, $Q = 3336.82$, $T.D = 2482.87$, $D.T = 2113.23$, $E.D = 369.64$, $\lambda = 1$.

| *i* | Results: Distance Traveled | Expected Distance | Total Distance | Amount Delivered | Route |
|---|---|---|---|---|---|
| 1 | 85.30 | 17.29 | 102.59 | 120.5 | 9 – 135 – 164 – 34 – 78 |
| 2 | 87.20 | 16.59 | 103.79 | 154.60 | 10 – 108 – 32 – 181 – 159 |
| 3 | 83.66 | 14.42 | 98.08 | 116.50 | 14 – 142 – 43 – 15 – 178 |
| 4 | 38.91 | 5.65 | 44.56 | 127.50 | 18 – 166 – 147 – 6 – 112 |
| 5 | 89.79 | 15.67 | 105.46 | 111.50 | 19 – 124 – 46 – 45 – 125 |
| 6 | 92.06 | 17.05 | 109.11 | 131.50 | 20 – 128 – 66 – 103 – 120 |
| 7 | 54.18 | 7.50 | 61.68 | 92.50 | 21 – 2 – 137 – 117 – 13 |
| 8 | 93.59 | 18.08 | 111.67 | 137.50 | 25 – 170 – 67 – 39 – 187 |
| 9 | 103.05 | 21.07 | 124.12 | 125.50 | 35 – 136 – 65 – 71 – 161 |
| 10 | 52.89 | 11.35 | 64.24 | 111.50 | 37 – 100 – 193 – 91 – 59 |
| 11 | 54.91 | 6.22 | 61.13 | 118.50 | 50 – 184 – 26 – 40 – 105 |
| 12 | 74.86 | 14.01 | 88.87 | 128.50 | 51 – 188 – 160 – 30 – 122 |
| 13 | 41.02 | 5.81 | 46.83 | 108.50 | 52 – 106 – 153 – 89 – 156 |
| 14 | 86.72 | 15.83 | 102.55 | 102.50 | 56 – 186 – 23 – 139 – 55 |
| 15 | 59.99 | 10.44 | 70.43 | 107.50 | 60 – 84 – 5 – 93 – 98 |
| 16 | 60.19 | 8.61 | 68.80 | 107.50 | 73 – 41 – 115 – 58 – 152 |
| 17 | 59.29 | 13.04 | 72.33 | 134.50 | 74 – 75 – 133 – 22 – 171 |
| 18 | 82.99 | 12.20 | 95.19 | 111.50 | 77 – 158 – 3 – 81 – 24 |
| 19 | 56.03 | 12.78 | 68.81 | 127.60 | 79 – 185 – 33 – 157 – 102 |
| 20 | 44.36 | 7.61 | 51.97 | 126.50 | 80 – 177 – 12 – 138 – 154 |
| 21 | 60.85 | 12.37 | 73.22 | 112.50 | 83 – 199 – 8 – 174 – 114 |
| 22 | 37.77 | 7.59 | 45.36 | 115.50 | 95 – 104 – 99 – 96 – 183 |
| 23 | 47.49 | 8.94 | 56.43 | 115.50 | 109 – 150 – 116 – 196 – 76 |
| 24 | 58.09 | 11.87 | 69.96 | 128.50 | 110 – 155 – 4 – 197 – 72 |
| 25 | 92.56 | 18.25 | 110.81 | 124.50 | 119 – 44 – 38 – 140 – 86 |
| 26 | 68.34 | 13.55 | 81.89 | 113.50 | 129 – 169 – 121 – 29 – 68 |
| 27 | 79.91 | 15.10 | 95.01 | 103.50 | 131 – 90 – 63 – 126 – 190 |
| 28 | 34.61 | 6.05 | 40.66 | 104.50 | 132 – 69 – 1 – 176 – 111 |
| 29 | 50.03 | 8.66 | 58.69 | 114.50 | 144 – 97 – 151 – 92 – 94 |
| 30 | 66.60 | 11.62 | 78.22 | 99.50 | 145 – 57 – 42 – 172 – 87 |
| 31 | 26.55 | 2.08 | 28.63 | 82.60 | 146 – 27 – 28 – 53 |
| 32 | 47.58 | 8.09 | 55.67 | 133.50 | 149 – 195 – 179 – 198 – 180 |
| 33 | 47.09 | 7.73 | 54.82 | 102.50 | 162 – 101 – 70 – 127 – 167 |
| 34 | 68.33 | 12.93 | 81.26 | 122.50 | 163 – 134 – 165 – 130 – 54 |
| 35 | 95.81 | 17.84 | 113.65 | 140.50 | 168 – 47 – 49 – 107 – 11 |
| 36 | 71.67 | 13.06 | 84.73 | 124.50 | 173 – 61 – 113 – 17 – 118 |
| 37 | 110.09 | 19.15 | 129.24 | 114.50 | 175 – 64 – 143 – 36 – 48 |
| 38 | 73.06 | 12.47 | 28.63 | 135.50 | 189 – 62 – 123 – 182 – 148 |
| 39 | 67.59 | 14.11 | 81.70 | 152.50 | 192 – 191 – 141 – 16 – 85 |
| 40 | 59.81 | 10.03 | 69.84 | 113.50 | 194 – 82 – 7 – 88 – 31 |

Table 6.17: **Instance 5:** $n = 199$, $Q = 4756.35$, $T.D = 3147.7$, $D.T = 2664.9$, $E.D = 482.8$, $\lambda = 1$.

| $i$ | Results Distance Traveled | Expected Distance | Total Distance | Amount Delivered | Route |
|---|---|---|---|---|---|
| 1 | 113.01 | 24.15 | 137.16 | 103.27 | $2-1-15-14-13$ |
| 2 | 94.42 | 22.01 | 116.43 | 78.27 | $6-7-9-10-5$ |
| 3 | 101.93 | 15.51 | 117.44 | 89.27 | $11-4-3-81-119$ |
| 4 | 184.14 | 40.30 | 224.44 | 73.27 | $29-28-32-26-23$ |
| 5 | 179.50 | 35.23 | 214.73 | 91.27 | $33-24-22-12-8$ |
| 6 | 175.75 | 33.77 | 209.52 | 85.27 | $34-30-27-19-87$ |
| 7 | 174.24 | 40.64 | 214.88 | 80.27 | $35-36-31-25-17$ |
| 8 | 168.47 | 38.31 | 206.78 | 105.27 | $40-43-39-38-37$ |
| 9 | 180.96 | 40.64 | 221.60 | 88.27 | $45-42-44-47-41$ |
| 10 | 184.49 | 43.23 | 227.72 | 82.27 | $46-49-50-51-48$ |
| 11 | 240.95 | 31.56 | 272.51 | 81.27 | $52-16-21-20-90$ |
| 12 | 173.83 | 28.51 | 202.34 | 87.27 | $53-54-57-115-97$ |
| 13 | 188.71 | 44.56 | 233.27 | 87.27 | $55-58-60-62-61$ |
| 14 | 174.24 | 32.37 | 206.61 | 99.27 | $56-79-80-72-75$ |
| 15 | 204.47 | 46.47 | 250.95 | 107.27 | $64-63-66-65-59$ |
| 16 | 120.79 | 24.81 | 145.60 | 97.27 | $69-70-67-73-68$ |
| 17 | 119.19 | 27.76 | 146.95 | 86.27 | $71-74-78-77-76$ |
| 18 | 30.15 | 5.01 | 35.16 | 73.27 | $85-112-84-117-82$ |
| 19 | 25.85 | 2.96 | 28.81 | 64.27 | $88-111-86-106-105$ |
| 20 | 26.60 | 4.48 | 31.08 | 76.27 | $102-93-94-96-95$ |
| 21 | 34.96 | 6.14 | 41.10 | 72.27 | $104-116-100-99-101$ |
| 22 | 53.12 | 8.28 | 61.40 | 86.27 | $109-108-114-91-92$ |
| 23 | 58.51 | 7.50 | 66.01 | 77.27 | $110-98-103-107-120$ |
| 24 | 43.78 | 7.72 | 51.50 | 81.27 | $113-83-118-18-89$ |

Table 6.18: **Instance 11:** $n = 120$, $Q = 2053.50$, $T.D = 3676.86$, $D.T = 3064.89$, $E.D = 611.97$, $\lambda = 0$.

| $i$ | Results | | | | |
|---|---|---|---|---|---|
| | Distance Traveled | Expected Distance | Total Distance | Amount Delivered | Route |
| 1 | 46.87 | 9.45 | 56.32 | 134.65 | $1-2-4-6-8$ |
| 2 | 90.51 | 19.21 | 109.72 | 144.65 | $12-14-16-19-18$ |
| 3 | 81.65 | 13.99 | 95.64 | 154.65 | $17-15-13-11-9$ |
| 4 | 22.19 | 2.02 | 24.21 | 47.86 | $20-21$ |
| 5 | 60.07 | 7.71 | 67.78 | 104.6 | $23-26-22-24-52$ |
| 6 | 46.45 | 9.06 | 55.51 | 134.65 | $28-30-29-27-25$ |
| 7 | 86.88 | 18.97 | 105.85 | 164.65 | $36-39-38-37-33$ |
| 8 | 48.96 | 9.58 | 58.54 | 104.65 | $41-40-42-47-49$ |
| 9 | 73.07 | 6.21 | 79.28 | 75.72 | $43-67-65-101-90$ |
| 10 | 55.17 | 11.06 | 66.23 | 114.65 | $44-45-48-46-50$ |
| 11 | 90.65 | 16.06 | 106.71 | 134.65 | $51-31-35-32-34$ |
| 12 | 99.72 | 21.05 | 120.77 | 164.65 | $60-56-53-54-55$ |
| 13 | 51.39 | 8.76 | 60.15 | 86.26 | $64-61-72-74-102$ |
| 14 | 48.33 | 8.51 | 56.85 | 115.19 | $69-68-66-62-63$ |
| 15 | 162.39 | 22.23 | 184.62 | 144.65 | $73-80-58-59-57$ |
| 16 | 43.57 | 7.98 | 51.55 | 114.65 | $75-5-3-7-10$ |
| 17 | 121.28 | 27.08 | 148.36 | 124.65 | $76-71-70-77-79$ |
| 18 | 111.94 | 19.65 | 131.59 | 144.65 | $84-83-82-78-81$ |
| 19 | 62.06 | 13.29 | 75.35 | 144.65 | $89-88-85-86-87$ |
| 20 | 89.89 | 16.42 | 106.31 | 144.65 | $91-98-95-97-99$ |
| 21 | 97.25 | 20.19 | 117.44 | 144.65 | $100-94-92-93-96$ |

Table 6.19: **Instance 12:** $n = 100$, $Q = 2703.15$, $T.D = 1878.87$, $D.T = 1590.32$, $E.D = 288.55$, $\lambda = 1$.

| $i$ | Results: Distance Traveled | Expected Distance | Total Distance | Amount Delivered | Route |
|---|---|---|---|---|---|
| 1 | 75.66 | 12.37 | 88.03 | 100.35 | $1-28-31-26-8$ |
| 2 | 90.70 | 16.90 | 107.60 | 113.35 | $20-35-36-3-22$ |
| 3 | 89.65 | 15.53 | 105.18 | 112.35 | $10-39-30-34-50$ |
| 4 | 69.74 | 11.41 | 81.15 | 109.35 | $16-21-29-2-32$ |
| 5 | 79.09 | 13.30 | 92.39 | 106.35 | $15-33-45-44-37$ |
| 6 | 50.64 | 6.66 | 57.30 | 145.35 | $12-17-4-18-47$ |
| 7 | 54.82 | 8.66 | 63.48 | 122.35 | $5-49-9-38-11$ |
| 8 | 51.86 | 5.94 | 57.80 | 106.35 | $6-23-48-27-46$ |
| 9 | 101.17 | 16.49 | 117.66 | 117.35 | $13-41-40-19-42$ |
| 10 | 89.38 | 12.72 | 102.10 | 127.35 | $14-25-24-43-7$ |

Table 6.20: **Instance 1:** $n = 50$, $Q = 1160.5$, $T.D = 872.73$, $D.T = 752.70$, $E.D = 120.03$, possible exchanges = 1, random selected routes = 2, maximum number of iterations = 5000.

| $i$ | Results | | | | |
|---|---|---|---|---|---|
| | Distance Traveled | Expected Distance | Total Distance | Amount Delivered | Route |
| 1 | 92.61 | 15.91 | 108.52 | 126.85 | $21-61-69-36-47$ |
| 2 | 60.70 | 10.55 | 71.25 | 134.85 | $33-73-62-28-74$ |
| 3 | 48.03 | 7.36 | 55.39 | 137.85 | $17-40-44-3-51$ |
| 4 | 81.44 | 11.51 | 92.95 | 134.85 | $6-63-49-24-32$ |
| 5 | 93.49 | 19.34 | 112.83 | 107.85 | $37-20-70-60-71$ |
| 6 | 78.78 | 12.29 | 91.07 | 140.85 | $12-39-9-50-18$ |
| 7 | 32.35 | 3.42 | 35.77 | 123.88 | $34-52-4-75$ |
| 8 | 17.12 | 1.15 | 18.27 | 65.94 | $26-67$ |
| 9 | 69.08 | 11.92 | 81.00 | 116.85 | $5-15-57-13-27$ |
| 10 | 84.41 | 15.64 | 100.05 | 128.85 | $16-23-56-41-43$ |
| 11 | 103.75 | 15.90 | 119.65 | 137.85 | $25-55-31-10-58$ |
| 12 | 47.97 | 7.49 | 55.46 | 130.85 | $45-29-48-30-68$ |
| 13 | 90.06 | 12.06 | 102.12 | 137.88 | $35-14-59-66$ |
| 14 | 76.77 | 13.12 | 89.89 | 137.85 | $53-11-65-38-72$ |
| 15 | 89.64 | 14.98 | 104.62 | 139.85 | $1-42-64-22-2$ |
| 16 | 63.74 | 9.17 | 72.91 | 133.8 | $7-19-54-8-46$ |

Table 6.21: **Instance 2:** $n = 75$, $Q = 2036.75$, $T.D = 1311.82$, $D.T = 1129.98$, $E.D = 181.84$, possible exchanges = 1, random selected routes = 2, maximum number of iterations = 5000.

| $i$ | Results Distance Traveled | Expected Distance | Total Distance | Amount Delivered | Route |
|---|---|---|---|---|---|
| 1 | 77.21 | 14.29 | 91.50 | 115.95 | $4-56-23-75-74$ |
| 2 | 93.89 | 16.49 | 110.38 | 117.95 | $39-67-25-55-54$ |
| 3 | 38.08 | 6.67 | 44.75 | 120.95 | $6-94-95-97-13$ |
| 4 | 46.35 | 3.39 | 49.74 | 114.95 | $28-27-89-58-53$ |
| 5 | 104.19 | 18.91 | 123.10 | 114.95 | $20-66-65-71-51$ |
| 6 | 101.75 | 16.93 | 118.68 | 85.95 | $9-35-34-29-24$ |
| 7 | 44.88 | 9.04 | 53.92 | 95.95 | $92-98-59-99-96$ |
| 8 | 61.30 | 12.38 | 73.68 | 131.95 | $61-16-91-85-93$ |
| 9 | 58.54 | 8.31 | 66.85 | 96.95 | $50-1-70-31-69$ |
| 10 | 79.72 | 15.24 | 94.96 | 108.95 | $10-63-90-32-30$ |
| 11 | 48.10 | 9.30 | 57.40 | 123.95 | $12-80-68-77-76$ |
| 12 | 77.67 | 13.16 | 90.83 | 101.95 | $37-14-43-42-87$ |
| 13 | 48.94 | 8.31 | 57.25 | 106.95 | $26-21-72-73-40$ |
| 14 | 56.93 | 9.11 | 66.04 | 86.95 | $18-82-7-88-52$ |
| 15 | 89.09 | 16.55 | 105.64 | 113.95 | $8-46-36-47-48$ |
| 16 | 65.53 | 13.16 | 78.69 | 111.95 | $3-79-78-81-33$ |
| 17 | 63.68 | 11.34 | 75.02 | 98.95 | $5-84-45-83-60$ |
| 18 | 74.45 | 12.76 | 87.21 | 80.95 | $2-57-15-41-22$ |
| 19 | 98.41 | 16.45 | 114.86 | 123.95 | $17-86-38-44-100$ |
| 20 | 103.35 | 18.18 | 121.53 | 122.95 | $19-49-64-11-62$ |

Table 6.22: **Instance 3:** $n = 100$, $Q = 2177$, $T.D = 1682.14$, $D.T = 1432.13$, $E.D = 250.01$, possible exchanges = 1, random selected routes = 2, maximum number of iterations = 5000.

| $i$ | Results | | | | |
|---|---|---|---|---|---|
| | Distance Traveled | Expected Distance | Total Distance | Amount Delivered | Route |
| 1 | 81.03 | 15.65 | 96.68 | 151.75 | 48 − 47 − 19 − 107 − 123 |
| 2 | 109.79 | 20.73 | 130.52 | 103.75 | 11 − 64 − 49 − 143 − 36 |
| 3 | 73.70 | 15.41 | 89.11 | 98.75 | 62 − 126 − 63 − 90 − 108 |
| 4 | 66.16 | 11.58 | 77.74 | 107.75 | 88 − 148 − 10 − 30 − 122 |
| 5 | 38.98 | 6.00 | 44.98 | 112.75 | 27 − 132 − 69 − 31 − 127 |
| 6 | 63.24 | 12.12 | 75.36 | 99.75 | 54 − 134 − 24 − 80 − 150 |
| 7 | 69.74 | 13.48 | 83.22 | 87.75 | 110 − 4 − 25 − 55 − 130 |
| 8 | 47.84 | 8.55 | 56.39 | 108.75 | 12 − 109 − 116 − 77 − 76 |
| 9 | 87.54 | 17.28 | 104.82 | 101.75 | 79 − 78 − 34 − 35 − 135 |
| 10 | 102.73 | 19.93 | 122.66 | 121.75 | 103 − 71 − 65 − 136 − 120 |
| 11 | 35.11 | 5.25 | 40.36 | 103.75 | 13 − 117 − 137 − 58 − 53 |
| 12 | 34.09 | 4.93 | 39.02 | 123.75 | 28 − 138 − 149 − 26 − 105 |
| 13 | 28.69 | 4.07 | 32.76 | 113.75 | 112 − 6 − 147 − 89 − 146 |
| 14 | 76.29 | 14.65 | 90.94 | 114.75 | 84 − 17 − 113 − 16 − 141 |
| 15 | 82.82 | 14.54 | 97.36 | 117.75 | 5 − 125 − 45 − 46 − 124 |
| 16 | 66.99 | 12.69 | 79.68 | 113.75 | 42 − 142 − 100 − 91 − 85 |
| 17 | 91.25 | 17.70 | 108.95 | 125.75 | 44 − 38 − 140 − 86 − 61 |
| 18 | 51.93 | 9.02 | 60.95 | 93.75 | 18 − 82 − 7 − 106 − 52 |
| 19 | 41.14 | 8.03 | 49.17 | 125.75 | 94 − 59 − 92 − 97 − 95 |
| 20 | 65.17 | 12.80 | 77.97 | 108.75 | 3 − 129 − 29 − 121 − 68 |
| 21 | 66.88 | 13.12 | 80.00 | 129.75 | 51 − 9 − 81 − 33 − 102 |
| 22 | 48.89 | 8.64 | 57.53 | 82.75 | 2 − 115 − 73 − 21 − 40 |
| 23 | 56.85 | 10.76 | 67.61 | 121.75 | 114 − 8 − 83 − 60 − 118 |
| 24 | 44.75 | 9.18 | 53.93 | 97.75 | 96 − 99 − 104 − 93 − 98 |
| 25 | 83.04 | 14.91 | 97.95 | 88.75 | 15 − 43 − 14 − 119 − 37 |
| 26 | 52.53 | 7.97 | 60.50 | 90.75 | 101 − 70 − 1 − 50 − 111 |
| 27 | 91.37 | 17.34 | 108.71 | 118.75 | 32 − 131 − 128 − 66 − 20 |
| 28 | 65.98 | 11.66 | 77.63 | 100.75 | 41 − 145 − 57 − 144 − 87 |
| 29 | 92.79 | 17.45 | 110.24 | 141.7 | 56 − 23 − 67 − 39 − 139 |
| 30 | 59.78 | 12.92 | 72.70 | 128.75 | 74 − 22 − 133 − 75 − 72 |

Table 6.23: **Instance 4:** $n = 150$, $Q = 3337.50$, $T.D = 2345.57$, $D.T = 1977.12$, $E.D = 368.45$, possible exchanges = 1, random selected routes = 2, maximum number of iterations = 5000.

| | Results | | | | |
|---|---|---|---|---|---|
| $i$ | Distance Traveled | Expected Distance | Total Distance | Amount Delivered | Route |
| 1 | 47.16 | 8.32 | 55.48 | 123.50 | 149 – 179 – 198 – 21 – 180 |
| 2 | 51.73 | 8.31 | 60.04 | 121.50 | 40 – 73 – 171 – 115 – 152 |
| 3 | 48.33 | 8.99 | 57.32 | 116.50 | 58 – 2 – 178 – 57 – 144 |
| 4 | 52.44 | 9.54 | 61.98 | 94.50 | 117 – 42 – 172 – 87 – 137 |
| 5 | 44.17 | 6.48 | 50.65 | 125.50 | 26 – 195 – 184 – 76 – 28 |
| 6 | 48.69 | 9.73 | 58.42 | 111.50 | 80 – 150 – 116 – 77 – 196 |
| 7 | 40.67 | 7.21 | 47.88 | 130.50 | 101 – 162 – 190 – 127 – 167 |
| 8 | 70.43 | 13.91 | 84.34 | 122.50 | 31 – 10 – 126 – 63 – 108 |
| 9 | 26.97 | 2.16 | 29.13 | 87.60 | 53 – 105 – 27 – 146 |
| 10 | 85.40 | 16.26 | 101.66 | 106.50 | 70 – 131 – 32 – 181 – 90 |
| 11 | 59.63 | 10.56 | 70.19 | 113.50 | 60 – 5 – 61 – 93 – 98 |
| 12 | 67.23 | 12.94 | 80.17 | 118.50 | 118 – 84 – 17 – 113 – 173 |
| 13 | 25.55 | 3.78 | 29.33 | 130.50 | 89 – 147 – 6 – 112 – 156 |
| 14 | 50.72 | 8.17 | 58.89 | 106.50 | 52 – 153 – 82 – 18 – 166 |
| 15 | 87.36 | 16.87 | 104.23 | 136.50 | 20 – 66 – 161 – 103 – 51 |
| 16 | 72.16 | 14.62 | 86.78 | 132.50 | 30 – 160 – 128 – 188 – 122 |
| 17 | 93.08 | 17.93 | 111.01 | 125.50 | 39 – 67 – 170 – 25 – 55 |
| 18 | 79.96 | 15.99 | 95.95 | 114.50 | 56 – 186 – 23 – 187 – 139 |
| 19 | 65.47 | 13.54 | 79.01 | 111.50 | 41 – 22 – 133 – 75 – 74 |
| 20 | 80.78 | 16.29 | 97.07 | 148.50 | 48 – 168 – 47 – 175 – 107 |
| 21 | 109.79 | 20.72 | 130.51 | 106.50 | 11 – 64 – 49 – 143 – 36 |
| 22 | 55.46 | 11.65 | 67.11 | 127.50 | 3 – 158 – 129 – 79 – 102 |
| 23 | 56.64 | 11.56 | 68.20 | 114.50 | 50 – 157 – 33 – 81 – 185 |
| 24 | 39.19 | 7.72 | 46.91 | 121.50 | 13 – 97 – 92 – 151 – 94 |
| 25 | 71.49 | 12.81 | 84.30 | 136.50 | 148 – 123 – 62 – 159 – 189 |
| 26 | 90.94 | 17.98 | 108.92 | 125.50 | 16 – 86 – 140 – 38 – 119 |
| 27 | 67.59 | 14.39 | 81.98 | 151.50 | 85 – 191 – 141 – 44 – 192 |
| 28 | 103.06 | 20.65 | 123.71 | 116.50 | 71 – 65 – 136 – 35 – 120 |
| 29 | 88.17 | 15.06 | 103.23 | 97.50 | 142 – 14 – 43 – 15 – 145 |
| 30 | 76.26 | 13.91 | 90.17 | 110.50 | 68 – 24 – 29 – 121 – 169 |
| 31 | 40.38 | 7.21 | 47.59 | 131.50 | 109 – 177 – 12 – 138 – 154 |
| 32 | 48.63 | 10.02 | 58.65 | 113.50 | 88 – 182 – 7 – 194 – 106 |
| 33 | 85.30 | 17.29 | 102.59 | 120.50 | 9 – 135 – 164 – 34 – 78 |
| 34 | 89.78 | 15.67 | 105.45 | 111.50 | 9 – 124 – 46 – 45 – 125 |
| 35 | 52.89 | 11.35 | 64.24 | 111.50 | 37 – 100 – 193 – 91 – 59 |
| 36 | 60.85 | 12.37 | 73.22 | 112.50 | 83 – 199 – 8 – 174 – 114 |
| 37 | 37.78 | 7.59 | 45.37 | 115.50 | 95 – 104 – 99 – 96 – 183 |
| 38 | 58.09 | 11.87 | 69.96 | 128.50 | 110 – 155 – 4 – 197 – 72 |
| 39 | 34.61 | 6.05 | 40.66 | 104.50 | 132 – 69 – 1 – 176 – 111 |
| 40 | 68.33 | 12.94 | 81.27 | 122.50 | 163 – 134 – 165 – 130 – 54 |

Table 6.24: **Instance 5:** $n = 199$, $Q = 4758.10$, $T.D = 3013.73$, $D.T = 2533.26$, $E.D = 480.47$, possible exchanges = 1, random selected routes = 2, maximum number of iterations = 5000.

| $i$ | Results<br>Distance Traveled | Expected Distance | Total Distance | Amount Delivered | Route |
|---|---|---|---|---|---|
| 1 | 115.22 | 16.57 | 131.79 | 106.25 | 97 − 109 − 8 − 12 − 108 |
| 2 | 158.09 | 37.59 | 195.68 | 85.25 | 17 − 20 − 23 − 26 − 21 |
| 3 | 119.62 | 27.73 | 147.35 | 88.25 | 73 − 76 − 77 − 78 − 74 |
| 4 | 129.27 | 29.07 | 158.34 | 102.25 | 68 − 79 − 80 − 72 − 75 |
| 5 | 183.29 | 42.51 | 225.80 | 50.25 | 29 − 36 − 34 − 35 − 32 |
| 6 | 175.44 | 42.19 | 217.63 | 67.25 | 27 − 33 − 30 − 31 − 28 |
| 7 | 200.52 | 46.56 | 247.08 | 86.25 | 56 − 60 − 63 − 66 − 64 |
| 8 | 190.31 | 44.78 | 235.09 | 99.25 | 55 − 58 − 62 − 61 − 65 |
| 9 | 33.60 | 5.41 | 39.01 | 83.25 | 81 − 117 − 84 − 89 − 92 |
| 10 | 91.84 | 16.77 | 108.61 | 83.25 | 83 − 3 − 4 − 5 − 113 |
| 11 | 179.42 | 41.83 | 221.25 | 91.25 | 52 − 53 − 54 − 57 − 59 |
| 12 | 162.74 | 39.22 | 201.96 | 94.25 | 16 − 22 − 24 − 25 − 19 |
| 13 | 112.74 | 21.05 | 133.79 | 94.25 | 69 − 70 − 71 − 67 − 103 |
| 14 | 23.15 | 3.76 | 26.91 | 83.25 | 87 − 85 − 112 − 86 − 111 |
| 15 | 37.12 | 7.56 | 44.68 | 77.25 | 18 − 118 − 114 − 90 − 91 |
| 16 | 51.43 | 9.32 | 60.75 | 75.25 | 100 − 116 − 98 − 110 − 115 |
| 17 | 35.80 | 2.87 | 38.67 | 85.25 | 88 − 82 − 119 − 120 − 105 |
| 18 | 173.01 | 41.62 | 214.63 | 81.25 | 41 − 47 − 49 − 46 − 44 |
| 19 | 183.36 | 42.45 | 225.81 | 89.25 | 42 − 48 − 50 − 51 − 45 |
| 20 | 27.99 | 5.02 | 33.01 | 66.25 | 101 − 99 − 104 − 107 − 106 |
| 21 | 94.71 | 22.44 | 117.15 | 79.25 | 6 − 10 − 11 − 9 − 7 |
| 22 | 113.01 | 24.21 | 137.22 | 103.25 | 2 − 1 − 15 − 14 − 13 |
| 23 | 168.47 | 38.41 | 206.88 | 105.25 | 40 − 43 − 39 − 38 − 37 |
| 24 | 26.60 | 4.49 | 31.09 | 76.25 | 102 − 93 − 94 − 96 − 95 |

Table 6.25: **Instance 11:** $n = 120$, $Q = 2053$, $T.D = 3400.29$, $D.T = 2786.79$, $E.D = 613.50$, possible exchanges = 1, random selected routes = 2, maximum number of iterations = 5000.

| $i$ | Results<br>Distance Traveled | Expected Distance | Total Distance | Amount Delivered | Route |
|---|---|---|---|---|---|
| 1 | 53.02 | 8.35 | 61.37 | 94.65 | 20 − 24 − 52 − 49 − 47 |
| 2 | 45.47 | 9.67 | 55.14 | 104.65 | 41 − 40 − 44 − 42 − 43 |
| 3 | 93.65 | 18.63 | 112.28 | 134.65 | 91 − 94 − 92 − 93 − 96 |
| 4 | 62.06 | 13.30 | 75.36 | 144.65 | 87 − 86 − 85 − 88 − 89 |
| 5 | 52.63 | 11.41 | 64.04 | 114.65 | 46 − 45 − 48 − 51 − 50 |
| 6 | 86.13 | 18.00 | 104.13 | 154.65 | 98 − 95 − 97 − 100 − 99 |
| 7 | 45.46 | 7.45 | 52.91 | 75.72 | 66 − 64 − 68 − 69 |
| 8 | 53.84 | 5.15 | 58.99 | 115.73 | 67 − 65 − 63 − 101 − 90 |
| 9 | 50.88 | 9.86 | 60.74 | 125.73 | 62 − 61 − 72 − 74 − 102 |
| 10 | 42.67 | 8.83 | 51.50 | 124.65 | 5 − 6 − 9 − 8 − 7 |
| 11 | 42.54 | 8.94 | 51.48 | 124.65 | 3 − 4 − 2 − 1 − 75 |
| 12 | 28.19 | 3.54 | 31.73 | 76.79 | 21 − 22 − 23 |
| 13 | 42.19 | 8.64 | 50.83 | 134.65 | 25 − 27 − 30 − 28 − 26 |
| 14 | 139.66 | 20.78 | 160.44 | 124.65 | 59 − 57 − 55 − 80 − 79 |
| 15 | 99.71 | 22.06 | 121.77 | 184.65 | 54 − 53 − 56 − 58 − 60 |
| 16 | 82.37 | 15.76 | 98.13 | 144.65 | 29 − 34 − 33 − 35 − 31 |
| 17 | 121.20 | 27.56 | 148.76 | 124.65 | 76 − 71 − 70 − 73 − 77 |
| 18 | 78.15 | 13.66 | 91.81 | 154.65 | 10 − 11 − 13 − 15 − 17 |
| 19 | 86.98 | 18.79 | 105.77 | 154.65 | 32 − 37 − 38 − 39 − 36 |
| 20 | 90.51 | 19.22 | 109.73 | 144.65 | 12 − 14 − 16 − 19 − 18 |
| 21 | 111.94 | 19.66 | 131.60 | 144.65 | 84 − 83 − 82 − 78 − 81 |

Table 6.26: **Instance 12:** $n = 100$, $Q = 2703.02$, $T.D = 1798.63$, $D.T = 1509.30$, $E.D = 289.33$, possible exchanges = 1, random selected routes = 2, maximum number of iterations = 5000.

# RESUMEN EN CASTELLANO

## Antecedentes

Cuando se detecta un problema, la toma de decisiones es primordial para abordarlo correctamente. Aquí surge la investigación de operaciones (IO), que permite maximizar la efectividad de las decisiones tomadas. Es por ello que la investigación de operaciones es tan antigua como la humanidad, pero no es hasta la segunda guerra mundial cuando adquiere autonomía científica. Dada la escasez de recursos, existía la necesidad de asignarlos de la mejor manera posible a las diferentes actividades militares de las que se componía cada operación. Por tanto, en el seno de los ejércitos aliados se decide reunir a grupos de científicos para asesorar sobre la optimización de las operaciones militares. Es aquí donde, con estos científicos, surge la IO. Dado el gran éxito de la IO en las diferentes operaciones, después de la guerra es introducida en problemas industriales, de negocio o de gobierno como forma de obtener una mejor relación entre costo/beneficio.

Investigar sobre las operaciones, como el nombre indica, aborda un amplio y diverso conjunto de temas. Sin embargo, podemos destacar una serie de problemas, y modelos, clásicos de especial relevancia y a la vez generales. Entre ellos, la programación lineal (ver, por ejemplo, Hillier y Lieberman, 2002) es, quizás, la herramienta más utilizada en IO dada su gran versatilidad. Esta herramienta es utilizada en problemas que buscan asignar de la manera más eficiente posible un conjunto de recursos limitados a diversas tareas que compiten entre sí, teniendo en cuenta un conjunto de restricciones a la hora de hacer este reparto, y buscando maximizar el beneficio. Asociado a la programación lineal, el método símplex propuesto por Dantzig en 1947 (Dantzig, 1963, es el primer libro escrito por el propio autor donde aparece esta metodología) es un procedimiento de resolución para este tipo de problemas eficaz y rápido, aún para problemas de gran tamaño. Dos problemas particulares de programación lineal son el problema del transporte y el de asignación. El problema del transporte, formalizado por primera vez en Monge (1781) y siendo resuelto matemáticamente en Tolstoi (1930), consiste en optimizar la manera de transportar bienes desde orígenes a destinos diferentes a mínimo costo, y el problema de asignación (Kuhn, 1955), consiste en asignar distintas tareas a personas (cabe destacar que el problema de asignación es considerado un caso especial del problema de transporte, limitando el número de restricciones y variables). Aunque ambos son modelables como problemas de programación lineal, el número elevado de restricciones y variables hace que la resolución con el método simplex requiera un elevado esfuerzo computacional; por ello, se han desarrollado diversos algoritmos específicos para este tipo de problemas. Estos dos problemas particulares entran en la subcategoría de la IO llamada modelos de optimización de redes, debido a su carácter y representación en forma de red. Otros problemas de redes son el de la ruta más corta (Dijkstra, 1959), el del árbol de mínima expansión (Kruskal, 1956) o el de flujo máximo (Ford y Fulkerson, 1956). Destacamos, a mayores, un problema de modelado de redes de gran relevancia a lo largo de este trabajo, el problema PERT (Program Evaluation and Review Techniques), o también conocido como CPM (Critical Path Method); véase Malcolm et al. (1959) y, para una revisión más completa, Punmia y Khandelwal (2002). En este tipo de problemas se formula y estudia un proyecto como una red dirigida con un único nodo inicio y final, formada por un conjunto de actividades con relaciones de precedencia entre ellas.

Todos los problemas presentados hasta ahora son deterministas, pero la IO también trata problemas estocásticos. Por ejemplo, en los problemas de teoría de colas presentados en Erlang (1909), y con una revisión más completa en Gross (2008), se estudian las líneas de espera y se trata de obtener el tiempo que pasa cada cliente en la cola, el tiempo que cada servidor está ocupado, la longitud de la cola, etc. También cómo debe ser un sistema óptimo diseñado para reducir el tiempo que los usuarios pasan en él, sin incrementar el coste de mantenimiento del sistema.

Anteriormente hemos mencionado los modelos de programación lineal. Como el nombre indica este tipo de modelos se caracteriza por ser descrito mediante funciones lineales. Esto restringe las características del problema a analizar, y no permite abordar con exactitud algunos problemas que surgen en la realidad. Es por ello que la programación matemática no lineal (para una referencia completa ver Bazaraa et al., 2013) considera modelos más generales. Sin embargo, aunque algunos de esos modelos pueden resolverse de manera eficiente, tales como los modelos convexos, en la mayoría de los casos no es posible resolverlos de manera exacta. Es por ello que algoritmos de aproximación y algoritmos heurísticos para obtener buenas soluciones suelen ser utilizados.

La IO no solo contempla problemas en los cuales un agente toma decisiones sobre un problema para optimizar los beneficios correspondientes. Existen situaciones en las cuales nos encontramos con un conflicto entre varios agentes, o jugadores, y para cada conjunto de decisiones, o estrategias, tomadas se obtiene un escenario y resultado diferente. La teoría de juegos (Von Neumann y Morgenstern, 1947) estudia este tipo de situaciones desde un punto de vista matemático y comprende dos subcategorías: la cooperativa y la no cooperativa. En la teoría de juegos no cooperativa, los jugadores toman las decisiones basadas en los conjuntos de estrategias disponibles, y dada la competición natural entre los jugadores, el resultado final de cada jugador depende de la decisión tomada por todos ellos. De manera diferente, en la teoría de juegos cooperativos los jugadores buscan formar coaliciones para maximizar las ganancias obtenidas, y un reparto "justo" entre los diferentes jugadores es uno de los principales objetivos del problema; el valor de Shapley (Shapley 1953) es una de las reglas de reparto más conocidas y utilizadas en la teoría de juegos cooperativos.

A mayores, se debe destacar que gran parte del avance que ha experimentado la investigación de operaciones a lo largo de su (corta) historia ha sido gracias a la programación informática. El avance informático ha sido esencial a la hora de resolver problemas de grandes dimensiones. Además, la creación de software específico y libre hace que cualquier usuario pueda desarrollar su problema y obtener una solución de manera relativamente sencilla y cómoda, lo que provoca un avance aún más rápido en el desarrollo de la IO. De mención especial, por los beneficios que aporta y por el uso del mismo en esta tesis, es el software libre R (R Core Team, 2020). R es un software estadístico formado por un conjunto de herramientas muy flexibles que permiten al usuario adecuarlas en todo momento para su uso personal. Además, está compuesto por paquetes, o librerías: un conjunto de funciones diseñadas por usuarios y accesibles para toda la comunidad mediante una sencilla descarga. Esto hace que su funcionalidad aumente día a día.

Los objetivos generales de esta tesis son desarrollar diversas herramientas para abordar nuevas variantes, o desde nuevas perspectivas, de algunos problemas de la investigación de operaciones tanto deterministas como estocásticos, así como estudiar el buen comportamiento de dichas herramientas de manera teórica y práctica, y crear librerías en el software estadístico R con estas herramientas y otras ya existentes en la literatura para ponerlas a disposición de la comunidad científica.

## Resumen general de la tesis

El desarrollo de la tesis se presenta en seis capítulos. El primer capítulo está dedicado al reparto de costos de demora en proyectos estocásticos ya finalizados (Gonçalves-Dosantos et al., 2020c). La gestión de proyectos es un campo dedicado a desarrollar técnicas para seleccionar, planificar, ejecutar y supervisar proyectos. Uno de los principales temas de interés en relación a la planificación de proyectos es la gestión del tiempo, con el objetivo de finalizar el proyecto y las diversas actividades que lo forman en su fecha de entrega. Sin embargo, a menudo los proyectos se ven retrasados, por lo que el costo asociado a ello debe ser sufragado entre los diferentes agentes responsables de la ejecución del proyecto. En la literatura existen diversas maneras de abordar el problema, desde reglas proporcionales basadas en problemas de bancarrota (Brânzei et al., 2002) hasta reglas basadas en la teoría de juegos (por ejemplo, Bergantiños et al., 2018, y Castro et al., 2007). Todas ellas suponen que la duración inicialmente esperada de cada actividad es determinista. Sin embargo es natural considerar estas duraciones como variables estocásticas, siguiendo algún modelo de distribución, lo que enriquece el modelo y lo acerca más a la realidad. Además, dadas las ventajas que supone utilizar un juego cooperativo frente a las reglas basadas en problemas de bancarrota, hemos optado por extender el juego propuesto por Bergantiños et al. (2018) al contexto estocástico y utilizar el valor de Shapley como regla de reparto. Por último, caracterizamos la regla propuesta en el contexto de estos juegos de proyectos y proponemos un algoritmo de aproximación, basado en Castro et al. (2009), para obtenerla por simulación en proyectos con un número elevado de actividades.

El Capítulo 2 desarrolla el paquete ProjectManagement, un paquete de gestión de proyectos para el software estadístico R (Gonçalves-Dosantos et al., 2020a). ProjectManagement es una herramienta que permite la gestión de proyectos computacionalmente y de forma libre y gratuita. Dada la escasez de software relativo a gestión de proyectos, o en su defecto con licencias de alto coste, creemos que un paquete de R será de gran ayuda para cualquier usuario con necesidades específicas en gestión de proyectos. ProjectManagement permite al usuario gestionar un proyecto para obtener un calendario del mismo, es decir, la fecha de finalización del proyecto, los tiempos de inicio y fin de cada actividad, así como la holgura que dispone cada actividad sin que se retrase el proyecto. En relación a costes en proyectos finalizados, el paquete dispone de repartos con diferentes reglas de asignación propuestas, tales como reglas proporcionales y/o el valor de Shapley. Todo esto también se aborda desde un punto de vista estocástico, utilizando la metodología propuesta y estudiada en el Capítulo 1. Por último, ProjectManagement incluye la gestión de recursos, lo que permite reducir el tiempo de finalización del proyecto aumentando los recursos y, por consiguiente, el coste, una redistribución del horario para un nivel más uniforme del consumo de recursos y una nueva gestión del proyecto considerando un límite máximo de recursos por unidad de tiempo.

Hemos visto como la teoría de juegos y el valor de Shapley se pueden utilizar para resolver problemas que surgen en la gestión de proyectos a la hora de repartir costes de demora. Ahora, desde un punto de vista más teórico, en el Capítulo 3 veremos cómo surgen nuevos valores para juegos cooperativos y cómo son caracterizados. Para ello, extendemos los valores igualitarios a juegos cooperativos con uniones a priori (Alonso-Meijide et al., 2020). Los valores igualitarios se basan en repartir equitativamente los beneficios obtenidos entre los diferentes jugadores cooperantes. Mucha es la literatura teórica que propone diferentes variantes de soluciones igualitarias, por ejemplo, van den Brink (2007) o Casajus y Hüttner (2014), donde estos valores son comparados con el valor de Shapley. Además, en van den Brink y Funaki (2009) y en van den Brink et al. (2016) pueden verse varias caracterizaciones del valor igualitario y el equal surplus division value. Sin embargo, estos valores nunca habían sido introducidos y estudiados en los juegos cooperativos con uniones a priori. Un juego con uniones a priori (Owen, 1977) se diferencia en contar con una partición del conjunto de jugadores, es decir, una estructura de coalición a priori entre los diferentes jugadores que condiciona la negociación entre ellos y, en consecuencia, hace variar el resultado de la negociación. Es en Owen (1977) donde, también, el valor de Shapley es extendido a este tipo de juegos, dando lugar al valor de Owen. Una vez propuestos el valor igualitario y tres posibles variantes del equal surplus division value para juegos cooperativos con uniones a priori, caracterizamos estos nuevos valores con propiedades similares a las utilizadas en los valores originales. Por último, para comprobar el buen comportamiento de estos valores, estos son aplicados a un ejemplo que surge en el reparto de costes correspondientes a la instalación de un ascensor en un edificio de viviendas.

En el Capítulo 4 se estudian nuevas caracterizaciones para los valores propuestos en el capítulo anterior (Gonçalves-Dosantos y Alonso-Meijide, 2020), de forma similar a como el valor de Owen es caracterizado en Vázquez-Brage et al. (1997). Además, dos nuevos valores, y sus respectivas caracterizaciones, para extender el equal surplus division value son introducidos. Uno de ellos es el valor obtenido aplicando el procedimiento propuesto por Owen (1977) para llegar al valor de Owen a partir del valor de Shapley, pero en esta nueva situación, utilizando el equal surplus division value. La segunda extensión surge buscando un valor coalicional para el equal surplus division value que verifique la propiedad de contribuciones equilibradas. Para comparar estos nuevos valores, entre sí y con los ya presentados en el Capítulo 3, un ejemplo similar al del capítulo anterior es usado en este capítulo.

En los dos capítulos anteriores, hemos obtenido extensiones de diversos valores ya existentes en un marco concreto de los juegos cooperativos. En el Capítulo 5 considerando un tipo especial de jugador, jugador necesario, y su pago asociado proponemos un nuevo valor para juegos cooperativos (Gonçalves-Dosantos et al., 2020d). Un jugador necesario es aquel sin el cual el valor obtenido por cualquier coalición del juego es igual a cero. En Alonso-Meijide et al. (2019a) y Béal y Navarro (2020) se utilizan este tipo de jugadores para caracterizar el valor de Shapley, el valor de Banzhaf (Banzhaf III, 1964) y el equal surplus division value. Para poder comparar teóricamente este nuevo valor con otros ya existentes en la literatura, se propone una caracterización del mismo. Además se comprueba su comportamiento, frente a estos otros valores, en un ejemplo práctico. Por último, este valor es extendido al contexto de los juegos cooperativos con uniones a priori, para caracterizarlo de igual manera utilizando propiedades de jugadores necesarios. Finalizamos con dos caracterizaciones para el valor de Owen y Banzhaf-Owen (Owen, 1982) utilizando, de nuevo, los jugadores necesarios de forma similar a como Alonso-Meijide et al. (2019a) caracterizan el valor de Shapley y Banzhaf.

En el Capítulo 6, desde un punto de vista más practico que en el Capítulo 1, resolvemos un problema real de redes con elementos estocásticos (Gonçalves-Dosantos y Casas-Mendez, 2020). Los problemas de rutas de vehículos, con origen en Dantzig y Ramser (1959), contemplan el diseño de un conjunto de rutas de costo mínimo para una flota de vehículos que deben atender la demanda de un grupo de clientes dispersos en diferentes localizaciones. De este modelo básico, surgen muchos otros al contemplar distintas restricciones, tales como flotas de vehículos con capacidades heterogéneas, ventanas de tiempo en las cuales los clientes deben ser atendidos y varios puntos de entrega o recogida, entre otros. En este capítulo se estudian los problemas de rutas de vehículos con multi-compartimentos. Estos problemas se caracterizan por la existencia de varios productos incompatibles que deben ser entregados, por lo que la flota de vehículos consta de varios multi-compartimentos independientes para impedir la mezcla de unos y otros. En la literatura existente para estos modelos no se tiene en cuenta aspectos importantes de la vida real que a menudo son aleatorios. En este sentido nosotros vamos a considerar que las demandas de los clientes son variables estocásticas. La motivación de este modelo viene dado por un problema real que surge en una cooperativa agrícola gallega. En esta cooperativa se fabrican cuatro tipos distintos de alimentos para animales de granja, consta de una flota de vehículos de varios compartimentos con diferentes capacidades y en cada compartimento solo un tipo de alimento puede ser transportado. Dado la alta complejidad computacional de este tipo de modelos, se propone un algoritmo en dos pasos. Primero, se utiliza una heurística constructiva basada en el algoritmo de Clarke y Wright (1964), para después, mediante una búsqueda tabú (Glover, 1989 y Glover, 1990) mejorar esta solución inicial. Se muestran resultados a través de los datos reales para, finalmente, realizar un estudio de simulación para comprobar el comportamiento de este algoritmo.

## Conclusiones

En el primer capítulo se estudian los problemas de reparto de costes en proyectos cuando las duraciones de las actividades son variables estocásticas. Se ha planteado un juego cooperativo para determinar la influencia entre las distintas actividades a la hora de retrasar el proyecto, y así, mediante el valor de Shapley hacer un reparto ecuánime de los costes por demora asignados a cada una de ellas. La comparación con el modelo determinista es satisfactoria, al obtener costes diferentes para actividades con duraciones iguales en media pero diferentes distribuciones. Un algoritmo de estimación de la regla de reparto ha sido propuesto, obteniendo en tiempos manejables buenos resultados y con errores aceptables. El segundo capítulo propone un software de gestión de proyectos de uso libre y gratuito, con un manual online para el soporte del mismo.

En los Capítulos 3 y 4 se extienden los valores igualitarios al contexto de juegos cooperativos con uniones a priori. Con un número aceptable de propiedades coherentes, los distintos valores han sido caracterizados, probándose la independencia de las distintas propiedades que caracterizan cada valor.

En el Capítulo 5 ha sido propuesto un nuevo valor considerando un reparto para los jugadores necesarios. En cierta manera, la propiedad propuesta para jugadores necesarios corrige las propiedades para tales jugadores que se encuentran en los valores de Shapley y Banzhaf. Además, se han propuesto caracterizaciones para el valor de Owen y Banzhaf-Owen utilizando únicamente tres propiedades.

Por último, en el Capítulo 6 se resuelve un problema real de rutas de vehículos con multi-compartimentos y demandas estocásticas. Dada la alta complejidad computacional de estos problemas un algoritmo en dos pasos ha sido propuesto. El algoritmo constructivo tiene en cuenta el hecho de que las demandas son variables estocásticas a la hora de seleccionar los clientes y, en caso de ser necesario, la vuelta al depósito para repostar bienes necesarios. En el segundo paso una búsqueda tabú mejora la solución inicial, seleccionando rutas al azar e intercambiando desde uno hasta dos clientes entre las rutas. En el ejemplo con datos reales utilizados se han comparado las soluciones obtenidas con el caso determinista. La distancia recorrida por los vehículos coincide con la recorrida en el caso determinista cuando todas las demandas (que en nuestro caso son estocásticas) pueden ser satisfechas sin que los vehículos tengan que retornar al depósito. En todo caso, la distancia recorrida en el caso estocástico aumenta en media aproximadamente un 6% por la presencia de tal aleatoriedad. Por último, en la simulación realizada se comparan las soluciones obtenidas en el primer y el segundo paso del algoritmo. Las soluciones iniciales son conseguidas en tiempos inferiores a un segundo y con buenos resultados. En el segundo paso, en tiempos razonables, se mejora la solución inicial hasta en un 10%.

## Líneas Futuras

Finalmente, existen varias tareas abiertas que tenemos intención de abordar en el futuro. En relación a los repartos de costes de demora en proyectos estocásticos nuevas reglas de reparto pueden ser propuestas para ser comparadas, tanto en la práctica como teóricamente, con la regla utilizada en Gonçalves-Dosantos et al. (2020d). Además, se pueden definir proyectos con uniones a priori, considerando que hay un conjunto de empresas, cada una de las cuales gestiona una o más tareas, y estudiar, por ejemplo, el valor de Owen como regla de reparto. En relación al paquete ProjectManagement nuevas funciones pueden ser implementadas, desde la metodología que acabamos de comentar anteriormente hasta la creación de una interfaz gráfica para un uso más intuitivo del paquete.

Un tema de interés en gestión de proyectos en los que intervienen varios jugadores es analizar diversas cuestiones en relación al esfuerzo que realizan. Se pueden considerar situaciones en las que cada uno de los jugadores tiene el control sobre el tiempo que necesita para completar su actividad, por ejemplo, destinando más o menos recursos a llevarla a cabo. En este tipo de situaciones los jugadores pueden generar ingresos destinando parte de los recursos iniciales a otros proyectos pero ello conlleva posiblemente costes debidos a sanciones por retraso. En este contexto surge un juego no cooperativo entre los jugadores que eligen las demoras para sus actividades individuales. Una solución de equilibrio en la que los jugadores eligen el retraso de sus actividades de forma simultánea o secuencial, dependiendo de la ubicación de la actividad dentro del proyecto, puede ser considerada. En este tema hemos empezado a trabajar durante una visita a la Duke University en el otoño de 2019 y tenemos un artículo en preparación con Fernando Bernstein (Duke University) y Greg DeCroix (University of Wisconsin-Madison) que pensamos terminar en un futuro próximo.

En los valores igualitarios para juegos cooperativos con uniones a priori, nuevas caracterizaciones pueden ser obtenidas, por ejemplo, basándose en las que propone Ferrières (2017) para caracterizar los valores igualitarios en juegos cooperativos. Además, los valores igualitarios pueden ser extendidos a juegos cooperativos en los que existe un grafo que limita las posibilidades de comunicación entre los jugadores (ver Borm et al., 1992), y a su vez ser caracterizados.

Con la propiedad propuesta para jugadores necesarios, un valor no eficiente surge al considerar que este cumpla las propiedades de aditividad, simetría y jugador nulo. Este valor puede ser comparado con los valores de Shapley y Banzhaf en la clase de microarray games (Lucchetti et al., 2010), lo que permite, entre otras cosas, a partir de una matriz de datos de expresión genética identificar los genes que son responsables de una enfermedad determinada.

Por último, en el problema de rutas de vehículos con multi-compartimentos y demandas estocásticas se pueden desarrollar nuevas metaheurísticas para mejorar las soluciones obtenidas y/o los tiempos de cálculo, basadas en colonia de hormigas (Rajappa et al., 2016), templado simulado (Xiao et al., 2014) o algoritmos genéticos (Vidal et al., 2013), entre otros.