Eindhoven University of Technology

MASTER

# Joint replenishment of inventory in a multi-company setting
# cost allocation and stability analysis

Pratikakis, C.

*Award date:*
2021

Eindhoven, September 14, 2021

# Joint replenishment of inventory in a multi-company setting: cost allocation and stability analysis

Charidimos Pratikakis
Student id number: 1362682

in partial fulfillment of the requirements for the degree of
**Master of Science in Operations Management and Logistics**

First Supervisor: dr. Marco Slikker
Second Supervisor: dr. ir. Loe Schlicher

TU/e Department of Industrial Engineering & Innovation Sciences
Series Master Theses Operations Management and Logistics

Keywords: joint replenishment problem, cost allocation, core, cooperative game theory, allocation rules

# Acknowledgements

The end of this project signifies the end of my Master studies at the Operations Management and Logistics program in Eindhoven University of Technology. It was a challenging program, I learned a lot and developed myself significantly during the past two years.

First of all, I would like to thank my main supervisor dr. Marco Slikker, for his guidance, support, and patience during my Master program and my Honors program at TU/e. During the past two years we had many meetings and discussed a lot about my progress, about my mini research project, and about my master thesis. I would like to thank you for all the time you put into those two projects and especially the master thesis project, for the constructive feedback and for your attention to detail which helped me refine and improve my work and learn a lot. I would further like to thank my second supervisor dr. ir. Loe Schlicher for his constructive feedback and suggestions on my master thesis, as well as for his working paper which was the starting point for my mini research project. I would like to thank Esther van der Ham for her workshops which helped me learn about personal leadership.

I would like to thank my girlfriend Ipek, for her support and for always being there for me in the past years. I would like to thank my friends for all the good times that we had and for making my study years enjoyable and fun. I would further like to thank my friends from the Operations Management and Logistics program, who studied with me and worked on assignments with me. I learned a lot from all of you.

Finally, I would like to thank my parents and my brother who have supported me in every step I have taken in my life. You may have been back in Greece during the past two years but you still supported me, encouraged me, and helped me whenever I needed it. I appreciate it.

# Abstract

The objective of the Joint Replenishment Problem is to decide on the optimal order quantities of products ordered from the same supplier in order to minimize costs. This problem may also arise in situations where multiple firms order their products from the same supplier. In such cases, the need to find fair ways to allocate costs may be more obvious because each firm may be an individual entity. This thesis deals with ways to distribute the joint replenishment costs between companies which are jointly replenishing their inventory using concepts from cooperative game theory.

# List of Abbreviations

| | |
|---|---|
| **DGS** | Direct Grouping Strategy |
| **EOQ** | Economic Order Quantity |
| **IGS** | Indirect Grouping Strategy |
| **JRP** | (Deterministic) Joint Replenishment Problem |
| **NTU-games** | Non-Transferable Utility Games |
| **POT Policy** | Power of Two Policy |
| **SJRP** | Stochastic Joint Replenishment Problem |
| **TU-games** | Transferable Utility Games |

# List of Tables

# List of Figures

# Summary


The Joint Replenishment Problem (JRP) is a well-known problem in the inventory management literature. It consists of deciding the optimal order quantity for products ordered from the same supplier. The goal is to minimize the time-average costs, which consist of holding costs and ordering/setup costs. What adds to the complexity of the problem is that the ordering costs consists of two parts: the major ordering cost and the minor ordering cost. The major ordering cost is incurred every time an order is placed, while the minor cost depends on each product and so it is incurred depending on the products that are placed in each order. There are deterministic as well as stochastic models for the Joint Replenishment Problem. The optimal strategy to solve the JRP is not known, yet the problem is solved in the literature using one of two strategies: Direct Grouping Strategy and Indirect Grouping Strategy. In the Direct Grouping Strategy products are partitioned into a number of groups and products in each group are always replenished together. In the Indirect Grouping Strategy each product is replenished according to an integer multiple of a base replenishment cycle. The choice of a base replenishment cycle and of integer multiples for the products depend on each other, which makes the JRP a complex problem. A widely used policy is the Power of Two policy which restricts the search space of the Indirect Grouping Strategy to integer multipliers which are a power of the number two. The main goal of the optimization literature when studying the JRP is to achieve the lowest possible time-average cost for the jointly replenished items when compared to the sum of the individual costs when each item is replenished on its own.


Figure 1: Replenishment of Inventory, Image source: (Western Truck Insurance Services, n.d.)

The JRP has also been studied in the game theoretic literature, with the objective of determining whether the total cost of jointly replenishing all items can be divided in a way that is fair for all the participating parties. Once a reduction of costs is achieved, it is important to further distribute/ allocate the realized cost-savings/costs. Thus the problem is not only to reduce total costs, but to come up with a redistribution of costs which will be seen as fair from each individual player's point of view (a player in this context is either an item or a retailer who replenishes his own item from his supplier). Fairness implies that an item / set of items is never allocated more cost than what they could achieve when acting on their own or when cooperating with only a subset of the participating parties. The game theoretic literature contains many rules, with desirable properties, which lead to the allocation of costs/ cost-savings achieved through the cooperation of players. There is literature for stochastic and deterministic JRP models, and there is literature that considers the minor ordering costs or assumes it is zero. Anily and Haviv (2007) study the (deterministic) JRP when a POT policy is used. Dror and Hartman (2007) and Dror, Hartman, and Chang (2012) consider the JRP when a set of items is always replenished together. There are very different dynamics in the games that follow from those two settings, which highlight the need for further research. Moreover, it can be argued that there is a need to study different policies because replenishing all items together is too simple, while the POT policy can be considered too complex.

The problem statement which was the starting point for this research is the following: *" The scientific literature does not provide sufficient insights on the properties of the joint replenishment game under the possibility of deterministic joint replenishment policies other than the optimal power-of-two policy and the single group policy. "*

We defined the following research questions:

- **Main Research Question:** Which game theoretic properties are present in the various types of joint replenishment games with major and minor ordering costs under policies different from the optimal POT policy or the single group policy?

- **Research Sub-question 1:** Does a joint replenishment game with major and minor ordering costs that is based on policies different from the optimal POT policy or the single group policy always have a non-empty core?

- **Research Sub-question 2:** What are proper cost allocation rules for this class of games and what are their properties?

- **Research Sub-question 3:** How sensitive are the results obtained in the previous sub-questions to problem parameters?

We defined Policy 2, which is a policy more flexible than Policy 1 of Dror and Hartman (2007), and Dror, Hartman, and Chang (2012) and less complex than the POT policy. Policy 2 allows the simultaneous optimization of both the base replenishment cycle and the integer multipliers of each item, but for reasons of practicality and simplicity limits the choice of integer multipliers to either 1 or 2. Thus when Policy 2 is used it is possible to order an item in every replenishment cycle or in every other replenishment cycle. In Chapter 3, we tried to answer the first sub-question. We saw that there can be cases where the imputation set and the core of the game that follows from a setting where policy 2 is used can be empty. We further saw that there can be cases where the game that follows from a setting where policy 2 is used may have an empty core, while in the same problem instance the game that follows from a setting where policy 1 is used has a non-empty core. Lastly, we also saw problem instances where the reverse was true. Those observations lead us to compare the core of the two games more extensively by means of a numerical study.

In Chapter 4, we focused on the first and third sub-questions. We examined the non-emptiness of the core for the games $(N, c^2)$ and $(N, c^1)$ which follow from the joint replenishment situations where Policy 2 and Policy 1 respectively are used. In our simulation, we examined two questions: what is the probability that the core of the games $(N, c^2)$ and $(N, c^1)$ which follow from a setting with parameters in defined ranges are non-empty, and in case that the core is empty we examined how much lower the cost of the grand coalition should be so that the core becomes non-empty. We defined metric e in order to answer the second question: $e = min_{\kappa \in M_N^B}\left(\frac{\sum_{S \in 2^N \setminus \{\emptyset\}} \kappa(S)c(S) - c(N)}{c(N)}\right)$, where $M_N^B$ is the set which contains all the balanced maps for a given number of players which follow from the minimal balancedness collections. The probability that the game $(N, c^2)$ has a non-empty core for 4-item games is 0.9823 for a high value of the major cost, 0.9577 for a medium value of the major cost and 0.8185 for a low value of the major cost. The respective values for 4-item games $(N, c^1)$ are: 0.9371, 0.8459, and 0.4549. By studying metric e we find that the maximum decrease in costs of the grand coalition needed so that a core of a 4-item game $(N, c^2)$ becomes non-empty is 5% for a high value of the major cost, 7.5% for a medium value of the major cost and 12.5% for a low value of the major cost. The respective values for 4-item games $(N, c^1)$ are: 10%, 15%, and 28%. We further used metric e to find lower bounds on the achieved cost-savings for both games when they are balanced. We found that the lower bounds on cost savings are comparable for both policies. Lastly, intuitively speaking we stated that the size of the core for the two games is comparable.

| Major Cost | Policy 2 | Policy 1 |
|---|---|---|
| High (A=15) | 0.9823 | 0.9371 |
| Medium (A=8) | 0.9577 | 0.8459 |
| Low (A=2) | 0.8185 | 0.4549 |

Figure 2: Probability of non-empty core: 4-item games

| Major Cost | Policy 2 | Policy 1 |
|---|---|---|
| High(A=15) | 5% | 10% |
| Medium (A=8) | 7.5% | 15% |
| Low (A=2) | 12.5% | 28% |

Figure 3: Maximum decrease in costs of grand coalition needed: 4-item games

In Chapter 5, we focused on the second and third sub-questions. We wanted to find allocation rules which are readily explainable, justifiable (give allocations which are consistent from a cost/ cost savings perspective), and easy to compute. We tested the performance of the Shapley value, the LOUDERBACK rule, INCREMENTAL rule and INDIVIDUAL rule on two metrics: the probability that a rule gives an allocation that belongs to the imputation set and the probability that a rule gives an allocation which belong to the core set. We compared the performance of those 4 rules for the games that follow from both policies. We find that the performance of the rules is consistent for both games. For the 8-item game $(N, c^2)$ and a high value of the major cost we found that the core metric for the Shapley value is equal to 0.8835, for the LOUDRERBACK rule 0.9726, for the INCREMENTAL rule 0.9476, for the INDIVIDUAL rule 0.8066. The respective values for the game $(N, c^1)$ are: 0.6241, 0.8889, 0.8134, 0.5307. We further performed sensitivity analyses on the performance of the 4 rules on the 2 metrics for two values of the major cost and 3-item games up to 8-item games (those analyses were performed for the game $(N, c^2)$ that follows from the setting where Policy 2 is used). The ranking of the four rules on the two metrics was consistent for both values of the major cost and for the various number of items. For the imputation set metric, we found the ranking: INDIVIDUAL > Shapley value > LOUDERBACK > INCREMENTAL. For the core set metric we found the ranking: LOUDERBACK > INCREMENTAL > Shapley value > INDIVIDUAL. We further found a lower bound on the number of games which have a non-empty core when Policy 2 is used, by finding the probability that one of the four chosen allocation rules will result in a core allocation. We compared the probability that an 8-item game $(N, c^1)$ has a non-empty core with the lower bound on the probability for the game $(N, c^2)$. We found that for a high value of the major cost the game $(N, c^2)$ has at least 8.02 % higher probability of having a non-empty core than the game $(N, c^1)$, in the observed range of parameters. Considering the trade-off between performance of the rules, properties of the rules, and ease of computations we recommend the LOUDERBACK rule and the Shapley value as high performing rules in the core metric.

| Allocation Rule | Policy 2 | Policy 1 |
|---|---|---|
| Shapley Value | 0.8835 | 0.6241 |
| INDIVIDUAL | 0.8066 | 0.5307 |
| INCREMENTAL | 0.9476 | 0.8134 |
| LOUDERBACK | 0.9726 | 0.8889 |

Figure 4: Policy 2 vs. Policy 1, Core Metric: 8-item games

A possible future extension for this research might be to check the performance of the LOUDERBACK rule and the Shapley value in a setting where the POT policy is used and the optimization of the base replenishment cycle in sub-coalitions is also allowed. A further extension might be to incorporate more operational constraints to the model and examine the game that follows. Some operational constraints which may be considered are capacity constraints or possible penalties when replenishing certain products together.

# Contents

# Chapter 1

# Introduction

## 1.1 Joint Replenishment Problem

The Joint Replenishment Problem (JRP) is a problem which has been studied extensively in the inventory management literature (Khouja and Goyal, 2008). A description of the problem is the following: a very common multi-product problem is deciding the optimal order quantities when ordering items from the same supplier (Goyal, 1974 ; Khouja and Goyal, 2008). The objective of this problem is to minimize the total cost per time unit (time-average cost) which consists of two parts: the ordering or setup cost and the holding cost. Additionally, the ordering cost consists of two components: the major ordering cost and the minor ordering cost. The major ordering cost is incurred every time an order is placed and the minor ordering cost is incurred depending on which products are contained in the order (Khouja and Goyal, 2008). It becomes obvious that due to the major ordering cost, using joint replenishment may lead to reduction of costs. The JRP literature is rich and deals with deterministic models and stochastic models. We will denote the joint replenishment problem under deterministic demand as JRP and the joint replenishment problem under stochastic demand as SJRP.

"The classic assumptions of the JRP are similar to the economic order quantity (EOQ)" (Khouja and Goyal, 2008). Those are deterministic demand, no backordering allowed, no quantity discounts allowed and linear holding costs. The optimal strategy to solve the JRP is not known, yet the problem (deterministic JRP) is solved in the literature using one of two strategies: Direct Grouping Strategy (DGS) and Indirect Grouping Strategy (IGS). A description for those strategies is given in (Khouja and Goyal, 2008):

1. "Under IGS a replenishment is made at regular time-intervals and each product has a replenishment quantity sufficient to last for exactly an integer multiple of the regular time interval. Groups in IGS are indirectly formed by products having the same integer multipliers." The problem consists of deciding the basic replenishment or ordering cycle which is common for all items and the positive integer multiplier for each of the items. The integer multiplier for each item defines whether an item is replenished in every cycle or in every second, third, etc... cycle. The time between successive replenishments for a product is its ordering cycle or cycle time. If an item is replenished in every basic ordering cycle, then the integer multiple is equal to one. If an item is replenished in every other basic cycle then the multiplier is equal to 2 and so on. This is a complex problem because those decision variables depend on each other.

2. "Under DGS, products are partitioned into a predetermined number of sets and the products within each set are jointly replenished with the same cycle time." The problem consists of deciding the number of groups to form as well as the items to include in each group.

From Van Eijs, Heuts, and Kleijnen (1992) we learn the following two differences between the IGS and the DGS: 1) the ordering cycles of the groups formed using IGS are based on the same base ordering cycle (due to the integer multipliers), while this is not the case for groups formed by the DGS. 2) The number of groups in the DGS is an input variable (number of groups is predetermined ) while the number of groups according to the IGS is an output variable. We can also notice that in the case where we form a single group according to the DGS and if the IGS policy will lead to a policy where all the integer multipliers are equal to one, the two strategies will result in the same solution for the JRP. However, Van Eijs, Heuts, and Kleijnen (1992) have identified that there are two factors which affect the relative performance of the strategies: the ratio of the major ordering cost to the average minor ordering cost and the number of products. The differences between the strategies were small and depended on the number of products but indicated that in the observed range of experiments IGS outperforms DGS. IGS and DGS resulted in the creation of a single group if the previously mentioned ratio is high and that DGS is better than IGS if the ratio has very small values (in that case one might argue that joint replenishment does not make much sense). In a paper by Olsen (2008) we can find an extension of the JRP, the JRP under interdependent minor costs. In their paper a penalty is incurred in the case that certain products are put in the same order and also certain products are prohibited from being ordered together. It is argued that handling the constraints prohibiting certain items from being placed in the same group could be handled by direct grouping methods.

Additionally, policies which are formed based on the IGS can be further categorized into cyclic policies and strict cyclic policies (Khouja and Goyal, 2008). Cyclic policies are called strict cyclic policies if at least one product has an integer multiplier equal to one, meaning that at least one product is included in every order. This means that strict cyclic policies are a subset of cyclic policies. In the literature, there are methods which are focused on optimizing either the JRP with strict cyclic policies or the JRP with cyclic policies (Khouja and Goyal, 2008). The information on the different strategies for solving the JRP are summarized in Figure 1.1.

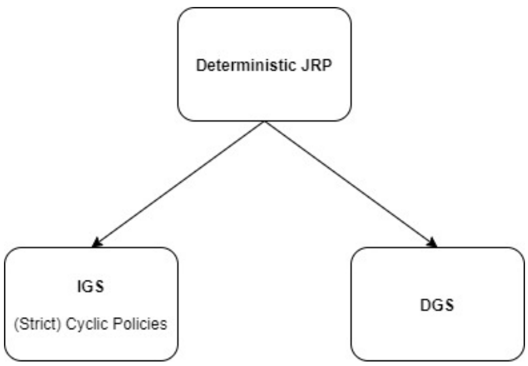



Figure 1.1: Strategies for solving the deterministic JRP

In terms of IGS, Goyal (1974) has developed an enumeration algorithm to find the optimal solution to the JRP by using a strict cyclic policy. Goyal's algorithm results in an optimal solution for the examined problem but may require a large number of computations for large problem instances due to its enumerative nature, which makes its use prohibitive for large problem instances (Van Eijs, Heuts, and Kleijnen, 1992 ; Khouja and Goyal, 2008)). However, there are many efficient heuristic algorithms that can be used as well. Kaspi and Rosenblatt (1991) developed the RAND heuristic algorithm to solve the JRP. Because the RAND is a heuristic it can solve larger problems in less computational time. Khouja and Goyal (2008) provide a nice description of the steps of the RAND algorithm and state about its performance: the RAND algorithm performed better than all the previous heuristics which were not enumerative, RAND identified the optimal solution for a large percentage of randomly generated instances of the JRP and in cases that the RAND algorithm did not identify the optimal solution its solution was very close to the optimal solution.

In terms of DGS, Bastian (1986) developed a heuristic to solve the JRP assuming that it is desirable to form a certain number of groups of items (the number of groups is an input to the algorithm, rather than an output as in the IGS). It is shown that there is always an optimal grouping in which ordered items are arranged in increasing order of their ratio of holding costs over the minor costs. A heuristic for finding the groups is also described in the paper, which turns out to be an optimal algorithm to form the groups in the case where there are no major costs considered.

Several special approaches have also been proposed in the inventory-management literature to solve the JRP (Khouja and Goyal, 2008). The Power-of-two (POT) policy (Lee and Yao, 2003) reduces the search space of the IGS policies by imposing that the integer multipliers of the base replenishment / ordering cycle are integers which are a power of number two. Khouja, Michalewicz, and Satoskar (2000) developed a genetic algorithm to solve the JRP using IGS and compared its performance with the performance of the RAND heuristic algorithm. Olsen (2005) developed a genetic algorithm to solve the JRP using the direct grouping strategy (DGS). In the more recent literature, there has been a trend in solving practical problems by considering additional situations such as capacity and resource constraints, replenishment of a main individual item and its complementary products whose demand is correlated to the former, other situations considered deterioration factors of products, and the incorporation of errors in demand prediction (Bastos, Mendes, Nunes, Melo, and Carneiro, 2017).

In the SJRP the main assumptions as stated for the JRP also hold but demand for products is stochastic and stationary and the objective is to minimize the expected total costs per time unit. Two main types of policies have been developed in the inventory-management literature to solve the SJRP: periodic replenishment policies and the can-order policy (Khouja and Goyal, 2008). In the can-order policy, each product $i$ has a reorder level $s_i$, a can-order level $c_i$, and a base stock level $S_i$, the policy is then denoted by $(s_i, c_i, S_i)$. The inventory position is reviewed continuously and each time the inventory position of a product $i$ drops to or below its reorder level $s_i$ an order is placed to bring the inventory position of the product back to the base stock level $S_i$, additionally inventory positions of products other than $i$ that are to or below their can-order levels are also brought up back to their base stock levels. Although the mechanism of the can-order policy can seem simple, determining the optimal parameters can be difficult due to the interactions between the products (Tinoco, Creemers, and Boute, 2017). Pantumsinchai (1992) has compared the performance of the can-order policy with the periodic replenishment policies proposed by Atkins and Iyogun (1988) as well as with the $(A, M)$ policy proposed by Renberg and Planche (1967). The $(A, M)$ policy is also mentioned in the literature as the $QS$ policy or the $(Q, S_1, ..., S_n)$ policy (for example in (Pantumsinchai, 1992)). Under the $(A, M)$ policy the aggregate inventory level of products is continuously reviewed and when it reaches the reorder level A, all products are replenished up to their base stock levels (which are not necessarily equal). The base stock level for a product $i$ is denoted by $S_i$ or $M_i$ depending on the notation used.

For a more detailed overview of the JRP literature under deterministic or stochastic demand we refer to Khouja and Goyal (2008) and to Bastos, Mendes, Nunes, Melo, and Carneiro (2017).

## 1.2 Cooperative Game Theory and The Core

Game theory is the science whose aim is to understand the behavior of interacting decision-makers. The basic assumptions that underlie Game theory are that decision-makers are rational and take into account their knowledge or expectations of other decision-makers' behavior, i.e. they act strategically (Osborne and Rubinstein, 1994). Game theory uses mathematical models to express its ideas formally. Game theory can be divided into cooperative game theory and non-cooperative game theory. Cooperative game theory is concerned primarily with coalitions – groups of players - who coordinate their actions and pool their winnings or share their costs. As a consequence, one of the problems that arises is how to divide the extra earnings, cost savings or the joint costs among the members of the formed coalition (Branzei, Dimitrov and Tijs, 2008). Cooperative games or coalitional games in characteristic function form, known also as transferable utility games (TU-games) are the most developed games in cooperative game theory (Branzei, Dimitrov and Tijs, 2008). A cooperative game might also be a non-transferable utility game (NTU-game). Several solution concepts for cooperative TU-games have been proposed and several interesting subclasses of TU-games have been introduced. In order to proceed further, some definitions need to be introduced. The following definitions are adapted from Slikker (2020) and Hezarkhani, Slikker, and Van Woensel (2018):

**Set Functions:**

Given a finite set $\Omega$, and $\wp(\Omega)$, $g : \wp(\Omega) \to \mathbb{R}$ is a set function that gives real values to subsets of $\Omega$. The following properties of set functions are of interest:

$g$ is non-decreasing if for every $A, B \subseteq \Omega$ with $A \subset B$ we have $g(A) \leq g(B)$.

$g$ is subadditive if for every $A, B \subset \Omega$ with $A \cap B = \emptyset$ we have $g(A \cup B) \leq g(A) + g(B)$.

$g$ is submodular if for every $A, B \subseteq \Omega$ with $A \subseteq B$ and every element $\alpha \in \Omega \setminus B$ it holds that $g(B \cup \{\alpha\}) - g(B) \leq g(A \cup \{\alpha\}) - g(A)$.

**Cooperative games:**

A transferable utility cooperative cost game (TU-game) is a pair $(N, c)$ where $N$ is a finite set of players and $c$ is a set function (also known as characteristic function) that assigns to every coalition $S \subseteq N$ the cost to be paid $c(S)$ with $c(\emptyset) = 0$.

The game $(N, c)$ is subadditive if $c$ is subadditive on the set of players and it is concave if $c$ is submodular on the set of players.

An allocation for players in $N$ is $x = (x_i)_{i \in N}$ such that $x_i \in \mathbb{R}$ for every $i \in N$ .

An allocation $x$ is efficient for $(N, c)$ if $\sum_{i \in N} x_i = c(N)$.

An allocation $x$ is individually rational for $(N, c)$ if $x_i \leq c(\{i\})$ for all $i \in N$.

An allocation $x$ is stable for $(N, c)$ if for any $S \subseteq N$ it holds that $\sum_{i \in S} x_i \leq c(S)$.

The imputation set $I(N, c)$ of a cooperative game $(N, c)$ contains all the allocations which are both efficient and individually rational.

The core $C(N, c)$ of a cooperative game $(N, c)$ contains all the allocations which are both efficient and stable.

The core is a subset of the imputation set, i.e. $C(N, c) \subseteq I(N, c)$.

There are at least two ways in which it can be shown that the core of a specific game is a non-empty set. Firstly by showing that an efficient and stable allocation exists and secondly by using the concept of balancedness. In order to proceed with the concept of balancedness, we cite some definitions which are based on Slikker (2020):

We can define for all $S \subseteq N$ the vector $e^S$ with $e_i^S = 1$ for all $i \in S$ and $e_i^S = 0$ for all $i \in N \setminus S$.

A map $\kappa : 2^N \backslash \{\emptyset\} \to [0, 1]$ is called a balanced map if $\sum_{S \in 2^N \backslash \{\emptyset\}} \kappa(S) e^S = e^N$, with $2^N$ denoting all the subsets of $N$.

A collection $\mathcal{B} \subseteq 2^N \backslash \{\emptyset\}$ is called a balanced collection on $N$ if there is a balanced map $\kappa$ on $N$ such that $\mathcal{B} = \left\{ S \in 2^N \backslash \{\emptyset\} \mid \kappa(S) > 0 \right\}$

A game $(N, c)$ is called balanced if for every balanced map $\kappa$ it holds that $\sum_{S \in 2^N \backslash \{\emptyset\}} \kappa(S) c(S) \geq c(N)$.

The balanced collections and the associated balanced maps result in certain inequalities that needs to hold, so that a game is balanced We call those inequalities balancedness conditions.

A theorem which is going to prove useful when checking whether the core of game is a non-empty set is the following theorem which is due to Bondareva (1963) and Shapley (1967).

**Theorem 1:** A game has a non-empty core if and only if it is a balanced game.

From Lohmann (2012) we learn that in order to check whether the core of a specific game is non-empty not all balanced collections are needed. Only the minimal balanced collections need to be checked. "A balanced collection of coalitions is minimal, if there does not exist a proper subset that is also balanced" (Lohmann, 2012). If we know the minimal balanced collections and the respective balanced maps for a game, then we know all the inequalities (balancedness conditions) that need to hold so that the game is balanced and so that it has a non-empty core. However finding all of the minimal balanced collections for games with many players is not an easy task. The minimal balanced collections for games with 3 and 4 players are available in the literature (e.g. Lohmann, 2012). More information on minimal balanced collections and the resulting balancedness conditions is provided in Appendix A.

In order to discuss the game theoretic literature on the JRP, some additional definitions are needed:

**Definition 1:** Let $(N, c)$ be a cost game, we can define the related cost-savings game $(N, v)$ as follows: $v(S) = \sum_{i \in S} c(\{i\}) - c(S)$ for all $S \subseteq N$.

It is important to note that a cost-savings allocation $y = (y_i)_{i \in N}$ is individually rational for $(N, v)$ if $y_i \geq v(\{i\})$ for all $i \in N$, and stable if for any $S \subseteq N$ it holds that $\sum_{i \in S} y_i \geq v(S)$. A cost savings allocation $y = (y_i)_{i \in N}$ is efficient if $\sum_{i \in N} y_i = v(N)$.

## 1.3 Inventory Games and Allocation of Costs / Benefits of Collaboration

The models in the inventory-management literature for the JRP are taken typically from the perspective of one company in a multi-product setting. However it can be the case that the joint replenishment problem arises in settings involving multiple firms or companies. For instance we can cite the following setting that has been examined by Anily and Haviv (2007):"a number of retailers, each facing a constant demand rate, lease the reordering or transportation of their supplies, as well as their storage activities, to a 3PL provider. We assume that each time a delivery is requested by any subset of retailers, a fixed reorder/transportation (leasing or renting) cost, called major setup cost, is charged. Moreover, each retailer is associated with his own retailer-dependent fixed reorder/transportation cost, called minor setup cost, which is possibly a function of the distance or the travel time between the warehouse and the retailer. This cost is being incurred whenever the retailer replenishes its stock. The warehouse's costs are assumed to be exogenous to the model." or the following setting as described by Meca, Timmer, Garcia-Jurado and Borm (2004): "One can think for example of some franchise operators restricted to a single good. Each of these firms has its own private demand and its own private storage possibilities for the good. There is a single supplier where all firms place their orders, concerning the good, at the same ordering cost. By means of placing their orders simultaneously, these firms can reduce their total cost compared to the total cost in the initial situation in which they all order separately, because of the lower total number of orders. An interesting question is what the optimal ordering policy for a group of agents will be."

It can be argued that in cases where a single company or retailer is replenishing multiple products, it may be sufficient to analyze the total cost reductions rather than the cost reductions per individual item (Tinoco, Creemers, and Boute, 2017). On the other hand, in settings where multiple firms are replenishing one product each from their supplier, as in the settings described previously, it is important to examine the conditions under which cooperation is beneficial for all of the participating parties as well as the dynamics of this cooperation. In the scientific literature the dynamics of cooperation are mainly studied by Cooperative Game Theory.

During the literature study (Pratikakis, 2021) that preceded this thesis a number of inventory game models have been identified. A paper that is heavily referenced in the literature is Meca, Timmer, Garcia-Jurado and Borm (2004). The authors have introduced inventory games in order to study how to share the joint costs by using cooperative game theory. We remark that the model examined in the paper can be seen as a special case of the joint replenishment problem in the case that minor ordering costs for all items are equal to zero (not considered in the model). This observation has also been made by Dror and Hartman (2011). The papers that were identified in the literature study which preceded the thesis (Pratikakis, 2021) and are of relevance to the JRP can be broken down into two categories: models with deterministic demand and models with stochastic demand:

### 1.3.1 Models with Deterministic Demand

The inventory games which assumed deterministic demand can be further categorized into models which included the minor ordering cost and models which assumed the minor ordering costs for all items are equal to zero (not considered in the respective models).

#### Models with Positive Minor Ordering Costs

In settings that included the minor ordering costs next to the major ordering cost and assumed deterministic demand, Dror and Hartman (2007) defined the joint replenishment game and examined the situation where all products in a coalition are ordered from the supplier according to the same ordering frequency. The authors showed that joint replenishment for the products may not necessarily be beneficial if orders for products are always placed with the same frequency (i.e. there can be cases where the core is empty). They further identified the property of inseparability which is the property that is present in situations where ordering all products always according to the same frequency is beneficial. Dror, Hartman, and Chang (2012) examined the performance of cost allocation rules in this setting and performed sensitivity analyses on the problem parameters in order to find out, among others, when joint replenishment for all the products stops being beneficial.

Anily and Haviv (2007) proved that if the optimal power-of-two (POT) policy is used, so that the retailers in a coalition are not synchronizing all of their orders every time, the resulting game is concave and so the core is always non-empty and they proposed a cost allocation rule which lies in the core. However, one might argue that although the proposed allocation rule lies in the core, it lacks fairness because certain retailers do not contribute at all to the major, while the retailers who pay towards the major cost do not have a reason to stay in the grand coalition because leaving will not increase their costs. Those two effects result in a situation where technically a core allocation is found but in practice the retailers who are paying the major cost will not accept a joint replenishment agreement according to which they need to keep "freeloaders" in the grand coalition. Zhang (2009) extended the results of Anily and Haviv (2007) to a setting where the warehouse is also included in the coalition and the costs of the warehouse are considered in the cost allocation, and obtained similar results.

It is important to note that in the inventory game model of Anily and Haviv (2007) and the one of Zhang (2009), the retailers in every coalition can only optimize over the integer multipliers of the base cycle time (while keeping the power of two restriction), but not over the base cycle time (the same, fixed base per coalition is used and is assumed to be equal to one). On the other hand, according to the model of Dror and Hartman (2007), all the integer multipliers are assumed to be equal to one for every product and only the base cycle time in each coalition can be optimized. Zhang (2009) discusses the intuition on the game that can be defined if retailers in a coalition can optimize both the cycle time and the integer multipliers which follow a POT policy and states that it can have an empty core if the major cost is close to zero. Yet one might argue that joint replenishment with an IGS policy makes a lot of sense if the major cost is high and so a lot of cost savings can be realized through the joint ordering of multiple items. In such case we don't have a lot of information, examples, and insights for the game where both the multipliers and the base cycle time are optimized over. Lastly, considering the "freeloaders" effects we discussed, we believe that the proposed allocation rule in Anily and Haviv (2007) may not lead to an effective agreement between the retailers and so more allocation rules need to be considered and effective insights need to be gained.

Fiestras-Janeiro, Garcia-Jurado, Meca, and Mosquera (2012) examine a variant of the JRP with minor ordering costs and assume that companies are located on a line from their supplier and so the minor cost is "linked" with the distance of a company from its supplier. The authors assume that when a subset of companies places an order, the ordering cost is the addition of the fixed part of the ordering cost (major ordering cost) and the maximum of the individual minor costs, as it corresponds to the longest distance travelled. This is the main difference with the game by Dror and Hartman (2007) and also makes this variation of the problem slightly less complex. The authors then identify the conditions under which cooperation leads to reduced costs and proposed rules to share the costs among the players.

### Models without Minor Ordering Costs

Several extensions of the basic inventory game by Meca, Timmer, Garcia-Jurado and Borm (2004) have been identified during the literature study that preceded the present thesis (Pratikakis, 2021). In all of those extensions we can see that different dimensions of the joint replenishment model are considered. In the models only the major ordering cost is considered but additional operational constraints are added. Fiestras-Janeiro, Garcia-Jurado, Meca, and Mosquera (2014) consider agents that face a continuous review inventory problem with deterministic demand, with no holding costs, with limited warehouse capacity and without backordering allowed. This model is related to Fiestras-Janeiro, Garcia-Jurado, Meca, and Mosquera (2012), but holding costs are now zero and the capacities of the warehouses are limited. The authors prove that for the cost game that arises from the described situation there can be conditions according to which cooperation of all agents is beneficial. Lastly they propose a proportional cost allocation rule: the two-lines rule. Saavedra-Nieves, Garcia-Jurado, and Fiestras-Janeiro (2018) consider a variant of the inventory game in which each agent faces a continuous review inventory problem, with deterministic demand, no holding costs and a limited capacity warehouse. The authors consider two acquisition costs and thus the cost function may be non-convex. Only a part of the agents cooperates by placing joint orders in this formulation.

Krichen, Laabidi, and Abdelaziz (2011) consider an extension of the inventory game where the supplier offers quantity discounts and allows retailers to delay payments. The retailers are tempted to form coalitions to increase their cost savings. In the examined model, the grand coalition is not always the preferred coalition (it does not always lead to the largest cost savings). The authors examine which coalition will be most beneficial for each retailer by assuming a cost savings game and using cooperative game theory and more specifically the concept of the generalized core. (the generalized core contains all the allocations which are stable but not necessarily efficient.) The authors define a procedure which based on the concept of the generalized core tries to find which coalition will lead to higher cost savings for each retailer. Li, Feng and Zeng (2014) studied an extension of the inventory game model that allows delay in payments and proposed a cost allocation rule that leads to a cost allocation which lies in the core.

Shi, Wang, and Xue (2019) consider a setting where retailers are ordering short-lifecycle consumer electronic products. They consider retailers who are jointly replenishing their inventory and subsequently match supply and demand with the use of transshipments. The demand in the model is dynamic but the authors make use of the EOQ model to determine the order quantity of the products and for this reason we include their inventory game in this section. They proposed a cost allocation rule that lies in the core of the game. In all of the mentioned papers the JRP model that is considered excludes the minor ordering costs and focuses on including operational constraints that are not present in the classic JRP formulation.

## 1.3.2 Models with Stochastic Demand

The joint replenishment problem under stochastic demand (SJRP) has also been studied in the game theoretic literature. In the literature study (Pratikakis, 2021) two articles with stochastic demand have been identified. Timmer, Chessa, and Boucherie (2013) extend the the model in Meca, Timmer, Garcia-Jurado and Borm (2004) to Poisson demand and study two policies to implement joint replenishment between the participating parties: a policy similar to the (A,M) policy as described in the previous section as well as a variation of the can-order policy where all players order each time one of them reaches their individual reorder level. Tinoco, Creemers, and Boute (2017) extend the model of Timmer, Chessa, and Boucherie (2013) to include minor costs and consider two companies that make use of the can-order policy to coordinate the shipment of their products. They examine cost allocations and the redistribution of costs under different cost sharing agreements and show that cooperation after redistribution of the costs is not always beneficial for both of the companies.

For more details on the literature on inventory games we point the reader to Dror and Hartman (2011) and to Fiestras-Janeiro, Garcia-Jurado, Meca, and Mosquera (2011).

An interesting article with elements both from cooperative game theory and inventory-management was written by Otero-Palencia, Amaya–Mier, and Yie-Pinedo (2019). The authors considered a stochastic demand setting yet after sufficient analysis the cost function that they use in their model, makes use of policies from the deterministic JRP literature and more specifically an IGS. An interesting comment made by the authors when referring to inventory games (such the ones we are examining in this thesis) was the following: " in our knowledge, all reported models simultaneously replenish all items together, while we consider that not all items are always jointly replenished; we use the IGS strategy, since such strategy reports lower cost solutions to the JRP". Such a comment is of interest as it refers to the two streams of literature which are of interest for this thesis. In their model the authors examine players who agree to cooperate in order to reduce their costs, the players choose to allocate the costs of the grand coalition by means of the Shapley value (more information on the Shapley value can be found in Appendix A). The authors are not interested in finding core allocations rather they are interested in finding a cost allocation based on the Shapley value. They argue that the fairness properties of the Shapley value will seem attractive to the collaborating agents and so the resulting allocation will facilitate the collaboration even if it does not belong to the core. In their model, the value of a coalition, which represents the operational cost of the coalition is obtained by solving a JRP instance with fixed parameters with the use of a genetic algorithm. Nevertheless if someone was interested in computing core allocations in the examined setting, the chosen definition of the value of a coalition might have lead to problems: the stochastic nature of genetic algorithms may lead to different results for the value of a coalition every time it is run. One might argue that the authors have not made a very good definition of the underlying game, yet for the research purposes of their paper, their choice was sufficient.

### 1.3.3 Allocation of Costs / Benefits of Collaboration

Once it has been identified that joint replenishment of inventory can lead to cost savings the problem of how to share the benefits of the collaboration arises. This is not an easy question since it is not always obvious what the contribution of each company to the total cost savings is (Tinoco, Creemers, and Boute, 2017). A simple and intuitive approach for cost allocation might be to use a proportional ratio which is based on individual indicators of each participating party (e.g. demand rate, individual costs). Literature proposes various different proportional cost allocation rules ( e.g. Dror, Hartman, and Chang, 2012; Guajardo and Rönnqvist, 2016). A more advanced approach is to use principles based on cooperative game theory. Different game theoretic allocation rules have different game-theoretic properties. A frequently used allocation rule from cooperative game theory is the Shapley value (Shapley, 1953) and another rule is the nucleolus (Schmeidler, 1969). More information on the Shapley value, the nucleolus and on the properties of allocation rules based on concepts of cooperative game theory are provided in Appendix A.

## 1.4 Outcomes

In the previous subsections it was shown that the joint replenishment problem has been studied extensively in the inventory-management literature with the objective of determining the optimal inventory policy under different operational constraints. The joint replenishment problem has also been studied from a cooperative game theoretic perspective and attempts have been made to identify the conditions under which cooperation is beneficial for all the parties. While the inventory-management literature has been studying the problem in the context of multiple items being ordered jointly, game theory literature has been studying the problem from a multi-firm perspective. The joint replenishment problem with minor ordering costs has seen slightly less attention from the game theoretic literature when compared to the less complex variant where there are no minor ordering costs considered. This means that the mechanisms of cost allocation among the participating companies are not as widely studied as the cost performance in joint replenishment. The structure of optimal policies for the deterministic joint replenishment problem is not yet known (Anily and Haviv, 2007), but many solution procedures with different underlying dynamics have been identified. Considering those observations it is important that the properties of the games that are defined from the various joint replenishment policies are studied, the properties of the allocation rules for those games are considered and insights on effective cost allocation are gained.

# Chapter 2

# Research Proposal

## 2.1 Problem Statement

In the previous chapter it was shown that the joint replenishment problem (JRP) with positive minor ordering costs has been studied in the game theoretic literature under two inventory polices: the optimal power-of-two policy (Anily and Haviv, 2007) and a direct grouping policy, under the assumption that all products are ordered according to the same ordering frequency (only one group was formed) (Dror and Hartman, 2007; Dror, Hartman, and Chang, 2012). "The power-of-two policy, being a special case of the general integer policy, restricts the search space for the optimal joint replenishment policy" Khouja and Goyal, (2008). Nevertheless, finding the optimal power-of-two policy is still complex and someone might argue that convincing retailers to cooperate based on such a complex policy will be a challenging task. Achieving the lowest possible costs through a joint replenishment agreement is important but using an intuitive, simple to understand and practical joint replenishment policy is also important as "understandability" may facilitate collaboration. On the other hand, replenishing all products with the same frequency is simple, yet it is not a very realistic policy due to operational constraints but also due to products having different characteristics.

Based on those observations we can understand that the choice of a joint replenishment policy is an important decision when trying to set up a collaborative joint replenishment agreement between companies. A proposed joint replenishment policy will need to be simple to understand, practical, allow for enough flexibility so that not only retailers with very similar cost parameters can participate but also achieve sufficient cost savings. At the same time it is vital that the dynamics of the collaboration are well understood and that the game theoretic properties of the joint replenishment game are studied so that fairness can be guaranteed for a number of policies. This will provide companies that wish to form joint replenishment agreements more options for their collaboration. At this point, we can make the following problem statement:

*" The scientific literature does not provide sufficient insights on the properties of the joint replenishment game under the possibility of deterministic joint replenishment policies other than the optimal power-of-two policy and the single group policy. "*

As a consequence, this problem may impede the collaboration agreements between companies for three reasons:

1. Companies may not be willing to form an agreement based on the optimal power-of-two policy or a policy where all products are put in a single group and replenished under the same frequency and may opt for other polices.

2. There can be cases where due to operational constraints neither the optimal power-of-two or the single group policy can fit the setting that the retailers may be facing (we have cited such a case in Chapter 1).

3. The properties of the game for other policies have not been studied and as a result the literature may lack a cost allocation rule which can be used for a wide number of policies.

## 2.2 Research Questions

After having described the problem, we formulate research questions which aim to provide a solution to the problem and will guide the research. During the research we aim to study interesting policies from a game-theoretic perspective. Studying various, interesting joint replenishment policies other than the optimal power-of-two policy and the single group policy will allow us to gain more insights on the dynamics of collaboration between companies that are jointly replenishing their inventory and are subject to major and minor ordering/transportation costs.

With this research we aim to answer the following research questions:

- **Main Research Question:** Which game theoretic properties are present in the various types of joint replenishment games with major and minor ordering costs under policies different from the optimal POT policy or the single group policy?

- **Research Sub-question 1:** Does a joint replenishment game with major and minor ordering costs that is based on policies different from the optimal POT policy or the single group policy always have a non-empty core?

- **Research Sub-question 2:** What are proper cost allocation rules for this class of games and what are their properties?

- **Research Sub-question 3:** How sensitive are the results obtained in the previous sub-questions to problem parameters?

We aim to add to the literature in the following ways: a) We aim to study the game theoretic properties for different joint replenishment policies and this will add to the literature which to the best of our knowledge has studied cost allocations and stability only for a limited number of policies. b) We aim to compare proportional as well as game theoretic cost allocation rules in the described setting and examine their performance. Game theoretic rules possess desirable fairness properties, while simpler, proportional cost allocation rules ( i.e by using a proportional ratio which is based on individual indicators of each participating party such as demand rate or individual costs) possess other desirable properties such as consistency of benefits and costs (termed justifiability) and polynomial time computability (Dror, Hartman, and Chang, 2012).

## 2.3 Research Design

After having described the problem statement and the research questions for the research, it is important to discuss the methods and solution procedures that are going to be used to answer those questions. With this research we aim to provide analytical as well as numerical results: considering the complexity of the problem it is expected that it will be challenging to find analytical results. It seems a feasible option to try and derive analytical expressions for problem instances with initially a limited number of players. In addition, through the literature study that has been conducted we have identified that determining the optimal cost of an inventory policy according to IGS or DGS is a complex problem. In terms of numerical results: we plan on defining the value of the coalition for each of the games we are going to study in one of the following ways: a) as the optimal solution of the chosen policy, or b) as the result of a heuristic algorithm. This will depend on the chosen policy and its complexity, as well as the available heuristics in the inventory-management literature and whether they are suitable to be used in this research. Another option of this research is to try and approximate the value of the coalition numerically after having defined it. Subsequently we will study the game theoretic properties for problem instances of various sizes based on those numerical results. This is in fact one of the ways that this research will add to the game theoretic literature on JRP as a great number of the papers that were identified focused on analytical results. Note that in order to study the game theoretic properties of a 3-player game we will need to solve $2^3 - 1 = 7$ optimization problems with the use of the chosen computational method.

**Numerical experiments:** A big part of the research will be focused on conducting numerical experiments for a variety of problem instances so as to get understanding on the game theoretic properties of the game. There is a lot of research already made on finding good solutions for the JRP in reasonable amount of time and a lot of experiments comparing the cost performance of different solution methodologies to the standalone setting. However game theoretic results for large problem instances are not present in the literature and so they will provide a lot of value. We aim to make use of either enumeration, or heuristics (the chosen algorithm is not decided at this point) from the JRP literature to define the value of the coalition in order to subsequently study game theoretic properties and gain insights on how various policies affect collaboration between the players. Additionally we aim to study different cost allocation rules, their properties, as well as to perform sensitivity analyses.

**Analytical results:** We aim to study non-emptiness of the core analytically for simplified policies by first assuming small problem instances consisting of initially a limited number of players or by assuming that a subset of problem parameters are identical for the players. Depending on the progress, a decision will be made as to whether other game theoretic properties can be studied.

# Chapter 3

# Model and Motivation

In this chapter we will introduce a simple, and intuitive joint replenishment policy which allows for exactly two ordering cycles within a coalition of retailers each of which orders his own unique product from the same supplier. Under this strategy each product has a cycle time which is an (positive) integer multiple of a base replenishment cycle time. However for reasons of simplicity and practicality we are going to restrict the choice of integer multipliers.

## 3.1 Assumptions and Characteristic Function

Consider a set of retailers, each of them ordering a single, unique product from the same supplier (a setting similar to the one described by Anily and Haviv (2007), and also discussed in the Introduction). The set of retailers is denoted by $N$. The major ordering cost is denoted by $A$, the minor ordering costs are denoted by $(s_i)_{i \in N}$, the holding costs are denoted by $(h_i)_{i \in N}$ (holding costs h per item on stock per time unit), and the deterministic demand rates are denoted by $(D_i)_{i \in N}$. The major cost is fixed, meaning that it is incurred whenever an order is placed irregardless of the quantity order. The minor cost $(s_i)_{i \in N}$ is also fixed and is incurred whenever an order for item $i$ (or equivalently by retailer $i$ ordering his specific item from the supplier) is included in an order irregardless of the quantity ordered. Another interpretation is that the fixed cost for an item consists of a part common to all items and a part specific to each item. Another subtle point about the demand is that in order to make sense to jointly replenish an item it means that the demand needs to be positive. We note that $A > 0, h_i > 0, D_i > 0, s_i \geq 0$ for all $i \in N$. In the special case where all $s_i = 0$, then we cannot use the present model, but in the case that only some of the minor costs are zero we can use it. After discussing the assumptions in detail we move on to the mathematical formulation of the model.

When each retailer is ordering alone, the optimal policy is the EOQ policy. The formulas for the individual costs $c(\{i\}) = c_i^*$, the individual optimal order quantities $Q_i^*$, and optimal order frequency $m_i^*$ (Dror, Hartman, and Chang, 2012) are presented next:

$$c_i = h_i Q_i / 2 + (A + s_i) D_i / Q_i \tag{3.1}$$

$$Q_i^* = \sqrt{2(A + s_i) D_i / h_i} \tag{3.2}$$

$$c_i^* = \sqrt{2(A + s_i)D_i h_i} \tag{3.3}$$

$$m_i^* = D_i/Q_i^* \tag{3.4}$$

The retailers agree on forming a coalition $S \subseteq N$ to reduce their costs. For reasons of practicality and flexibility, they agree on allowing exactly two ordering cycles in the coalition. They argue that a single ordering cycle that includes all the products will lack flexibility in case some of the retailers have vastly different cost parameters or demand rates and so if only a single ordering cycle is allowed then cooperation opportunities will be limited. They further argue that having many ordering cycles in the coalition is complex and not very intuitive so they allow exactly two possibilities for placing orders: ordering products every cycle and ordering products every other cycle. Such a policy may seem appealing to retailers replenishing fast-moving products and retailers replenishing slow-moving products. Those groups will need to replenish with different frequencies due to their different demand rates, a single ordering cycle is unlikely a good choice but a strategy that allows the option to skip a cycle and replenish every two cycles may facilitate the collaboration between those two different groups of retailers. The proposed strategy is an indirect grouping strategy and more specifically a strict cyclic policy, because the replenishment cycle for each product is an integer multiple (either 1 or 2) of a base replenishment cycle.

Aside from this initial decision of the ordering strategy, the retailers in the coalition will need to decide which products will be ordered every cycle, which products will be ordered every other cycle and subsequently use that information to find the optimal base replenishment cycle. We define $k^S = (k_i^S)_{i \in S}$ which will contain the decisions regarding the cycle that each product will be ordered. For example $k^S = (1, 2, 1)$ means that the retailers in coalition $S = \{1, 2, 3\}$ after discussions and analysis have agreed on replenishing the products of retailer 1 and 3 every cycle and the product of retailer 2 every other cycle (a different explanation for this setting may be that the retailers provide all of their information about demand, costs, etc... to a third party which is responsible for carrying out the inventory policy). The decision that the retailers in the coalition will need to take is not an easy decision because the optimal base replenishment cycle depends on the decisions of the retailers to order their products every cycle or every other cycle. Formally, if we denote the chosen base ordering cycle in coalition $S$ with $T_S$, then the optimal $T_S^*$ depends on the choice of $k^S$ and in reverse: for a fixed base replenishment cycle there is an optimal choice of $k^S$ that will lead to minimal costs.

The objective is to find which retailers will order every cycle, which retailers will order every other cycle and subsequently what is the optimal cycle denoted by $T_S^*$. The total costs per time unit (or time-average costs) $TC(T_S, k^S)$ for coalition $S$ when we restrict to the described policy are equal to the sum of holding costs per time unit plus ordering costs per time unit is given by the following formula (Khouja and Goyal, 2008):

$$TC(T_S, k^S) = \sum_{i \in S} Q_i h_i/2 + (A + \sum_{i \in S} s_i/k_i^S)/T_S \Leftrightarrow$$

$$TC(T_S, k^S) = \frac{T_S}{2} \sum_{i \in S} k_i^S D_i h_i + (A + \sum_{i \in S} s_i / k_i^S)/T_S$$

where: $Q_i = T_i D_i = k_i^S T_S D_i$ is the order quantity for product $i$.

The first term in the above equation corresponds to the total holding cost per unit of time when jointly replenishing items in coalition $S \subseteq N$ and the second term corresponds to the total ordering cost per unit of time. We are computing the time-average cost and we know that replenishments for item $i$ happen at fixed time intervals $0, k_i^S T_S, 2k_i^S T_S, ....$ This means that the holding cost for item $i$ is the area of a triangle with sides $Q_i$ and $T_i = k_i^S T_S$ multiplied by $h_i$. The holding cost then is divided by $T_i$ to compute the time-average holding cost. The ordering cost for item $i$ (major plus minor) is incurred once per ordering cycle. So we average by dividing by $T_i = k_i^S T_S$. Note that the major ordering cost is incurred in all cycles.

We can define the value of the coalition for this joint replenishment policy (in other words $c(S)$) as the optimal objective value of the following integer program:

Minimize:

$$\frac{T_s}{2} \sum_{i \in S} k_i^S D_i h_i + (A + \sum_{i \in S} s_i / k_i^S)/T_s$$

subject to:

$$k_i^S \in \{1, 2\}, \forall i \in S$$

$$T_S > 0$$

It is very important to note that the value for each coalition $S \subseteq N$, is the solution to the above optimization problem. In the case where, $|S| = 1$, the problem boils down to the EOQ model, which we have presented earlier.

For a fixed $k^S$, we can take the first derivative of $TC(T_S, k^S)$ with respect to $T_S$:

$$\frac{\partial TC(T_S, k^S)}{\partial T_S} = \frac{\sum_{i \in S} k_i^S D_i h_i}{2} - \frac{(A + \sum_{i \in S} s_i / k_i^S)}{T_S^2}$$

Then by setting the first derivative equal to zero we can compute the optimal base replenishment cycle as a function of the integer multipliers (note that the second derivative is positive for non-negative values of $T_s$):

$$T_S^* = \left[ 2 \left( A + \sum_{i \in S} s_i / k_i^S \right) / \sum_{i \in S} k_i^S D_i h_i \right]^{1/2} \tag{3.5}$$

Subsequently for any fixed $k^S$ (not necessarily the optimal, we can compute the cost as follows:

$$TC(T_S^*, k^S) = \frac{T_S^*}{2} \sum_{i \in S} k_i^S D_i h_i + (A + \sum_{i \in S} s_i / k_i^S) / T_s^* =$$

$$= [\frac{(\sum_{i\in S} k_i^S D_i h_i)^2}{4} \frac{2(A + \sum_{i \in S} s_i / k_i^S)}{\sum_{i \in S} k_i^S D_i h_i}]^{1/2} + [(A + \sum_{i \in S} s_i / k_i^S)^2 \frac{\sum_{i \in S} k_i^S D_i h_i}{2(A + \sum_{i\in S} s_i / k_i^S)}]^{1/2} =$$

$$= [\frac{(\sum_{i\in S} k_i^S D_i h_i)(A + \sum_{i \in S} s_i / k_i^S)}{2}]^{1/2} + [\frac{(A + \sum_{i \in S} s_i / k_i^S)\sum_{i\in S} k_i^S D_i h_i}{2}]^{1/2} =$$

$$= 2[\frac{(\sum_{i\in S} k_i^S D_i h_i)(A + \sum_{i \in S} s_i / k_i^S)}{2}]^{1/2}$$

We obtain the following expression for the value of coalition $S$, with $k^{S*} = (k_i^{S*})_{i \in S}$ the optimal integer multipliers for coalition $S$:

$$c^2(S) = \left[ 2 \left( A + \sum_{i \in S} s_i / k_i^{S*} \right) \sum_{i \in S} k_i^{S*} D_i h_i \right]^{1/2} \tag{3.6}$$

We can see from formula 3.6 that the optimal integer multiplier for retailer / product $i$ may be different for two coalitions $K, L \subset N$, i.e $k_i^{K*} \neq k_i^{L*}$. We will make use of this formula in the remaining of the thesis to compute the value of a coalition $S \subseteq N, |S| > 1$.

**NOTE:** The formulas derived above have been derived and cited in many papers on inventory-management for the case where the integer multipliers can have any value and not only 1 or 2. For reference: (Khouja and Goyal, 2008) and (Goyal, 1974). However to our knowledge they have not been used in a game theoretic paper that is why we derive them again and use our notation.

At this point we can define the joint replenishment situation under Policy 2 as a tuple $(N, A, (s_i)_{i \in N}, (D_i)_{i \in N}, (h_i)_{i \in N})$ with $N$ the set of retailers which are ordering from the same supplier, $A$ the major ordering cost, $(s_i)_{i \in N}$ the minor ordering costs, $(D_i)_{i\in N}$ the deterministic demand rates for the retailers, and $(h_i)_{i \in N}$ the holding costs for the retailers.

For the remaining of the thesis we are going to denote our proposed policy as Policy 2 (because we allow the possibility of choosing between two integer multipliers) and the resulting game as $(N, c^2)$. We are also going to denote the policy where all the integer multipliers are equal to 1 (or not taken in the scope of the model) , together with Policy 1. In Policy 1 all products are always replenished together. Policy 1 is the policy used in the papers by Dror, Hartman and Chang (2012) and Dror and Hartman (2007). The resulting game from Policy 1 is denoted as $(N, c^1)$. For clarity, we can define two different games from the same tuple, depending on the Policy which is used: if Policy 1 is used we have the game $(N, c^1)$, if Policy 2 is used we have the game $(N, c^2)$. The game $(N, c^1)$ is defined from the tuple $(N, A, (s_i)_{i \in N}, (D_i)_{i \in N}, (h_i)_{i \in N})$, with $c^1(S) = \left[2\left(A + \sum_{i\in S} s_i\right) \sum_{i \in S} D_i h_i\right]^{1/2}$ , $S \subseteq N$.

## 3.2 The Imputation Set and the Core of the Cost Game

After having defined our intuitive and simple to understand joint replenishment policy as well as the joint replenishment situation we are going to examine the properties of the respective games that arise from the described situation and we are going to start with examining non-emptiness of the core.

**Important Note on Rounding**

In the examples that will follow in the thesis, when we present results we will round the final result to two decimals. However, whenever the rounding has an impact on an inequality or an equality we will pay special attention to it and examine more decimals.

Consider the following example:

**Example 1:** Consider the joint replenishment game under a simple indirect grouping policy (Policy 2) given by the tuple $(N, A, (s_i)_{i\in N}, (D_i)_{i\in N}, (h_i)_{i\in N})$ with $N = \{1, 2\}$, $A = 3$,$s_1 = s_2 = 1$, $h_1 = 1, h_2 = 0.05$, and $D_1 = 2, D_2 = 1$.

In the stand-alone model each retailer orders according to the EOQ policy when his inventory level reaches zero. The optimal order quantities, optimal costs, optimal ordering frequencies and optimal time between replenishments for each retailer are summarized in Table 3.1.

| | Retailer 1 | Retailer 2 |
|---|---|---|
| optimal order quantity $Q_i^*$ | 4 | 12.65 |
| optimal cost $c_i^*$ | 4 | 0.63 |
| optimal order frequency $m_i^*$ | 0.5 | 0.08 |
| optimal time between replenishments $T_i^*$ | 2 | 12.65 |

Table 3.1: Example 1 - Stand-alone model

Subsequently we can compute the costs for a given $k^S$ according to the chosen policy and choose the combination of integer multipliers for coalition $S$ which result in the lowest costs. For a coalition $S$ there are $2^{|S|}$ possible combinations for $k^S$. As $c^2(\emptyset) = 0$ and $c^2(S)$ is a real number, we can define a cooperative cost game with transferable utility denoted by $(N, c^2)$. Since this is a small example, it is possible to enumerate over all possible $k^S$ for each coalition (Table 3.2) and pick the optimal set of integer multipliers. To conclude, $c^2(\{1, 2\}) = 4.35$. The resulting cost game can be seen on Table 3.3.

| $k_1^S$ | $k_2^S$ | $c^2(\{1,2\})$ | $T_S^*(k^S)$ |
|---|---|---|---|
| 1 | 1 | 4.53 | 2.21 |
| 1 | 2 | 4.35 | 2.07 |
| 2 | 1 | 6.04 | 1.49 |
| 2 | 2 | 5.73 | 1.40 |

Table 3.2: Chosen Policy for coalition S={1,2}

| Coalition | $c^2(S)$ |
|---|---|
| {1} | 4 |
| {2} | 0.63 |
| {1,2} | 4.35 |

Table 3.3: Example 1 - Cost Game

Note that for the specific example the imputation set and the core of the game (as they coincide for 2-player games) is non-empty since there exists at least one allocation that lies in the core: e.g. $x_1 = 3.9, x_2 = 0.45$.$\Diamond$

**NOTE:** In this example, we can achieve an even lower cost value $c^2(\{1,2\})$ for coalition $S = \{1,2\}$ by allowing more values for the integer multipliers. For instance: by setting $k^S = (1,3)$ we can find a cost equal to 4.317 and by setting $k^S = (1,4)$ we can find a cost equal to 4.324. This shows that the proposed policy is not an optimal POT policy, because by setting the integer multipliers to $k^S = (1,4)$ it is possible to achieve a lower cost. We remind that a POT policy allows for integer multipliers which are powers of the number two, so 1,2,4, etc. Since $k^S = (1,4)$ leads to a cost lower than the one obtained by our proposed policy it means that our policy in this example does not coincide with optimal power of two policy.

Moreover, our proposed policy is an improvement in this example of the policy proposed by Dror and Hartman (2007) since it achieves a cost equal to 4.35 rather than 4.53 if the policy of Dror and Hartman (2007) had been used (by setting all multipliers equal to 1 we see that products 1 and 2 are replenished according to the same cycle every time, which is the policy of Dror and Hartman (2007). The cost function also becomes identical with the one by Dror and Hartman (2007) if the multipliers are equal to 1.

For the proposed policy we can prove that the case according to which all integer multipliers are equal to 2 is always sub-optimal to the case where are all integer multipliers are equal to 1. The intuition behind this is that cycles without replenishments still incur the major setup costs of A. This means that in the case were all multipliers are equal to 2, we can imagine that we are sending an empty truck every other cycle. Formally we can write :

**Lemma 1:** In a joint replenishment situation under Policy 2 it is never optimal to set all integer multipliers equal to 2.

**Proof:**

The total costs when all multipliers are equal to 2 (this means that $k^S = (2, ..., 2)$), denoted by $TC_2$ are equal to:

$$TC_2 = \frac{T_S}{2} \sum_{i \in S} 2D_i h_i + (A + \sum_{i \in S} s_i/2)/T_S$$

$$= T_S \sum_{i \in S} D_i h_i + (A + \sum_{i \in S} s_i/2)/T_S$$

Now suppose that we choose a base replenishment cycle that is equal to $2T_S$ and integer multipliers that are equal to 1 (this means that we have $\hat{k}^S = (1, ..., 1)$), the total costs are denoted by $TC_1$ and are equal to:

$$TC_1 = \frac{2T_S}{2} \sum_{i \in S} \hat{k}_i D_i h_i + (A + \sum_{i \in S} s_i/\hat{k}_i)/(2T_S) =$$

$$= T_S \sum_{i \in S} D_i h_i + (A + \sum_{i \in S} s_i)/(2T_S) =$$

$$= T_S \sum_{i \in S} D_i h_i + \frac{1}{2} A/T_S + (\sum_{i \in S} s_i/2)/T_S$$

We can see that the holding costs and minor ordering costs are equal for the two policies but the major setup costs are double in the case that the integer multipliers are all equal to 2. This is the case due to the structure of the cost function which is assumes that cycles with no replenishments still incur a major setup cost but also due to our choice of integer multipliers (i.e we have restricted our choices of multipliers to either 1 or 2, but not greater). □

**Example 2:** Consider the game $(N, c^2)$ given by the tuple $(N, A, (s_i)_{i \in N}, (D_i)_{i \in N}, (h_i)_{i \in N})$ with $N = \{1, 2\}$, $A = 3$,$s_1 = s_2 = 1$, $h_1 = 1, h_2 = 0.05$, and $D_1 = 100, D_2 = 1$.

In the stand-alone model each retailer orders according to the EOQ policy when his inventory level reaches zero. The optimal order quantities, optimal costs, optimal ordering frequencies and optimal time between replenishments for each retailer are summarized in Table 3.4.

The resulting options for coalition $\{1, 2\}$ as well as the cost game can be seen on Table 3.5 and Table 3.6.

We can see from Table 3.6, that $c^2(\{1, 2\}) > c^2(\{1\}) + c^2(\{2\})$. This means that there can be cases where a joint replenishment game under Policy 2 is not subadditive, i.e. it is better to order the items individually rather than order them together. Consequently, we

| | Retailer 1 | Retailer 2 |
|---|---|---|
| optimal order quantity $Q_i^*$ | 28.28 | 12.65 |
| optimal cost $c_i^*$ | 28.28 | 0.63 |
| optimal order frequency $m_i^*$ | 3.54 | 0.08 |
| optimal time between replenishments $T_i^*$ | 0.28 | 12.65 |

Table 3.4: Example 2 - Stand-alone model

| $k_1^S$ | $k_2^S$ | $c^2(\{1,2\})$ |
|---|---|---|
| 1 | 1 | 31.63 |
| 1 | 2 | 30.01 |
| 2 | 1 | 42.43 |

Table 3.5: Chosen Policy for coalition {1,2}

| Coalition | $c^2(S)$ |
|---|---|
| {1} | 28.28 |
| {2} | 0.63 |
| {1,2} | 30.01 |

Table 3.6: Example 2 - Cost Game

can also see that the imputation set and the core is an empty set for the specific problem parameters.$\diamondsuit$

We also examine a 3-player game in the following example:

**Example 3:** Consider the joint replenishment game $(N, c^2)$ given by the tuple $(N, A, (s_i)_{i \in N}, (D_i)_{i \in N}, (h_i)_{i \in N})$ with $N = \{1, 2, 3\}$, $A = 3$,$s_1 = s_2 = s_3 = 1$, $h_1 = 1, h_2 = h_3 = 0.05$, and $D_1 = 100, D_2 = 1, D_3 = 3$.

| Coalition | $c^2(S)$ |
|---|---|
| {1} | 28.28 |
| {2} | 0.63 |
| {3} | 1.1 |
| {1,2} | 30.01 |
| {1,3} | 30.04 |
| {2,3} | 1.41 |
| {1,2,3} | 31.69 |

Table 3.7: Example 3 - Cost Game

Considering that this is a larger example we are trying to check the balancedness conditions in order to check if the core is an empty set. The concept of balancedness was already introduced in chapter 1 and the balancedness conditions are given in Appendix A. We can quickly see that ordering the items individually according to the EOQ formula will lead to lower total costs. This means that both the imputation set and the core of the examined 3-player game are empty sets. However we can also see that ordering items 2 and 3 together will still lead to reduced costs. This is an important finding as it shows that this simple policy can be beneficial for some of the participating parties even if it is not beneficial for the grand coalition. We speculate that the reason that this policy is not beneficial for all the retailers is due to the large differences in their cost parameters. We can see that although this simple policy added more flexibility than Policy 1, this flexibility is still not sufficient to guarantee that cooperation is always beneficial for all the retailers.$\diamondsuit$

**Findings from Dror and Hartman (2007)**

Our proposed policy is an extension of the policy used in Dror and Hartman (2007). In their paper all products are replenished under the same cycle, while in our policy it is possible to replenish products also every other cycle, if that is advantageous of course. For this reason, we are going to cite definitions and theorems from those authors, in order to use them and try to explain the behaviour of our proposed model.

We can derive the cost function used in the papers by Dror and Hartman (2007) and Dror, Hartman, and Chang (2012) by setting all $k_i^{S*}$ in equation 3.6 equal to 1, and so we derive their game which we denote by $(N, c^1)$ with $c^1(S) = \left[2\left(A + \sum_{i\in S} s_i\right)\sum_{i\in S} D_i h_i\right]^{1/2}$ if $|S| > 1$ and $c^1(\{i\}) = \sqrt{2(A+s_i)D_i h_i}$ if $S = \{i\}$. (For single player coalitions it is possible to use formula $c^1(S) = \left[2\left(A + \sum_{i\in S} s_i\right)\sum_{i\in S} D_i h_i\right]^{1/2}$ as well). This means that if we eliminate all integer multipliers from formula 3.6 then we get the same game as Dror, Hartman, and Chang (2012). Another explanation is that the optimal integer multipliers according to Policy 2 for each coalition are all equal to 1.

After having described how our model relates to the model of Dror, Hartman, and Chang (2012), we can cite the following two definitions and one theorem from Dror and Hartman (2007) and Dror, Hartman, and Chang (2012) which of course refer to their own model:

**Definition 2:** A set of items $B \subseteq N$ is called separable in the game $(N, c^1)$ if there exists a family of $k > 1$ pairwise disjoint subsets $\{B_1, B_2, ..., B_k\}$ with $B_1 \cup B_2 \cup ... \cup B_k = B$ such that the subadditivity inequality does not hold, i.e $c^1(B_1) + c^1(B_2) + ... + c^1(B_k) < c^1(B)$.

**Definition 3:** A set of items is called inseparable if it is not separable.

**Theorem 2:** The core of the joint replenishment game $(N, c^1)$ is nonempty if and only if the set of products is inseparable.

Continuing with the analysis of Policy 2: an observation for Policy 2 is that it can be the case that for a game with 3 players, it makes sense for the grand coalition to have multipliers that are set as $k^{N*} = (1, 2, 2)$ but for coalition $\{2, 3\}$ to have them as $k^{\{2,3\}*} = (1, 2)$ which simply means that the multiplier for item 2 is not the same in every coalition but can change depending on the other items in the coalition. In some examples the above happens, in others it does not happen. One example where this is the case is the following:

**Example 4:** Consider the joint replenishment game $(N, c^2)$ given by the tuple $(N, A, (s_i)_{i\in N}, (D_i)_{i\in N}, (h_i)_{i\in N})$ with $N = \{1, 2, 3\}$, $A = 1$,$s_1 = s_2 = s_3 = 1$, $h_1 = h_2 = h_3 = 1$, and $D_1 = 130, D_2 = 20, D_3 = 1$.

| Coalition | $c^2(S)$ | Optimal integer multipliers for each coalition |
|---|---|---|
| {1} | 22.8 | - |
| {2} | 8.94 | - |
| {3} | 2.0 | - |
| {1,2} | 29.15 | (1, 2) |
| {1,3} | 25.69 | (1, 2) |
| {2,3} | 10.49 | (1, 2) |
| {1,2,3} | 32.12 | (1, 2, 2) |

Table 3.8: Cost Game for Example 4

We notice that for the grand coalition it makes sense to have the multipliers set as $k^{N*} = (1, 2, 2)$ with $c^2(N) = 32.12$ and for coalition $S = \{2, 3\}$ it makes sense to set them as $k^{\{2,3\}*} = (1, 2)$ with $c^2(S) = 10.49$. The full cost game can be seen on the respective table.

If we check for non-emptiness of the imputation set we can see that it is non-empty since $c^2(N) < c^2(\{1\}) + c^2(\{2\}) + c^2(\{3\})$. Subsequently if we check for balancedness we see that the core is empty because it is cheaper to order $\{1, 2\}$ and $\{3\}$ alone. $\diamondsuit$

If we had the same parameters with the only change that $D_3 = 15$, we find a game $(N, c^2)$ which is inseparable and balanced, yet the integer multipliers for items 2,3 change depending on the coalition that is considered (in the grand coalition the multipliers for items 2 and 3 are both equal to 2, in coalition $\{2, 3\}$, the multipliers for items 2 and 3 are both equal to one). This means that there isn't a connection with when the multipliers change and inseparability or balancedness.

After looking at the definitions cited previously and after comparing the cost expressions of our proposed policy and the policy by Dror and Hartman, someone might think that inseparability may also be equivalent to balancedness and thus a non-empty core for the proposed policy as well. In the examples we have seen so far we had that inseparability and balancedness were present at the same time in the proposed policy. Inseparability is required for balancedness because the inseparability conditions are basically a subset of the balancedness conditions. But for policy 2 inseparability is not the same as balancedness as we will show in the following example:

**Example 5:** Consider the joint replenishment game $(N, c^2)$ given by the tuple $(N, A, (s_i)_{i\in N}, (D_i)_{i\in N}, (h_i)_{i\in N})$ with $N = \{1, 2, 3\}$, $A = 0.6$, $s_1 = 7, s_2 = 6, s_3 = 6$, $h_1 = 1, h_2 = 1.5, h_3 = 2$, and $D_1 = 100, D_2 = 100, D_3 = 30$.

We notice that the game is inseparable because every set partition of the items will lead to higher costs than we ordering all the items together, but the game is not balanced since: $c^2(\{1, 2, 3\}) > 0.5(c^2(\{1, 2\}) + c^2(\{1, 3\}) + c^2(\{2, 3\})) \Leftrightarrow 110.24 > 110.22$.

| Coalition | $c^2(S)$ | Optimal integer multipliers for each coalition |
|---|---|---|
| {1} | 38.99 | - |
| {2} | 44.50 | - |
| {3} | 28.14 | - |
| {1,2} | 82.46 | (1,1) |
| {1,3} | 65.97 | (1,1) |
| {2,3} | 72 | (1,2) |
| {1,2,3} | 110.24 | (1,1,1) |

Table 3.9: Cost Game for Example 5

We can see that if we examine the example from an optimization perspective, there are cost reductions by jointly replenishing the three items and so one might consider that this will benefit the retailers. Yet from a game theoretic perspective this is not the case, the core is empty. The game is inseparable (this also means that the imputation set is non-empty), yet it is not balanced and so the core is an empty set.

It is further important to note that if we have not allowed the possibility of a multiplier being equal to 2, then the game is balanced. This is an important finding as it states that extending the possibilities for optimization can come in the way of stability.$\Diamond$

In line with the comments by Zhang (2009) but by giving an example we find that for the cases that the major cost is low when compared to the minor cost, the core of the game can be empty. We add more knowledge and intuition though by showing that the property of inseparability while sufficient for the game by Dror and Hartman, (2007), it is not sufficient to guarantee balancedness for the game we are examining. Also, considering that our game considers the effect of both the integer multipliers and the optimization of the base cycle time inside a coalition, more attention needs to be paid on the game's properties.

Lastly, we can see an example where the game $(N, c^1)$ (Policy 1) has an empty core, yet the game $(N, c^2)$ (Policy 2) has a non-empty core.

**Example 6:** Consider the joint replenishment game $(N, c^2)$ given by the tuple $(N, A, (s_i)_{i\in N}, (D_i)_{i\in N}, (h_i)_{i\in N})$ with $N = \{1, 2\}$, $A = 2$, $s_1 = 1, s_2 = 1$ , $h_1 = 1, h_2 = 1$, and $D_1 = 100, D_2 = 1$ and the respective game $(N, c^1)$.

| Coalition | $c^2(S)$ | Optimal integer multipliers for each coalition | $c^1(S)$ |
|---|---|---|---|
| {1} | 24.49 | - | 24.49 |
| {2} | 2.45 | - | 2.45 |
| {1,2} | 26.72 | (1,2) | 28.43 |

Table 3.10: Cost Games for Example 6

$\Diamond$

## 3.3 Outcomes

In this chapter we have introduced our model, discussed the motivation behind it and stated our assumptions. We went on to perform an initial investigation on the behaviour of the model and have identified a number of important findings for the model which will prove useful in the numerical experiments that follow:

1. **Non-emptiness of the core and the imputation set:** The game $(N, c^2)$ is not subadditive in general. This means that there are instances where the core and the imputation set of the game $(N, c^2)$ are empty. Special note needs to be paid on two factors: 1) If the major cost is low (approaching zero), then as Zhang (2009) has pointed out the core of the game may be empty (Zhang was referring to a similar game that follows the full POT policy and a different base replenishment period per coalition can be used). 2) Even in the case that the major cost is high, then still the core may be empty because we limit the choice of multipliers to either 1 or 2. There is a trade-off to be made between simplicity of the model and possibilities for cooperation as well as for optimization.

2. **Realistic Assumptions:** By allowing the possible optimization of both the integer multipliers and the base cycle time, we might create problems in stability of the game. As we have seen in Example 5 by allowing the possibility that an integer multiplier can be 2 and the optimization of the base cycle time in each coalition we identified a game with an empty core. If Policy 1 had been applied with the same cost parameters the corresponding game would have a non-empty core. The same holds for the POT of Anily and Haviv (2007) (because it results in a concave game). Yet we believe that allowing the optimization of the base cycle time in a coalition is a very reasonable assumption, unlike the one made by Anily and Haviv (2007) which basically has the retailers cooperating upfront. We can also see from Example 6 that the core of a specific game when Policy 1 is used is empty, while when Policy 2 is used it is non-empty.

3. **Inseparability:** Inseparability in the game $(N, c^2)$ is not equivalent with balancedness and thus a non-empty core, as we have shown by means of an example. While cost optimization dictates that items in an inseparable set should be ordered together, stability is not guaranteed by cost optimization. This has been observed in cases where the major cost is low, but not when the major cost is high. It is interesting to note that while our starting point was Policy 1, by allowing more options and going to Policy 2, the same condition (inseparability) which was sufficient for Policy 1 is not sufficient for Policy 2. Intuitively one could expect that if there is no set partition of items which can be ordered at a lower cost then we might also have a fair allocation of costs but this is not the case here.

4. **Applicability of Policy 2:** We expect that Policy 2 will lead to increased possibilities for cooperation between the retailers (higher chance that a game has a non-empty core), because a separable set (empty core) of items in game $(N, c^1)$ may have a non-empty core in $(N, c^2)$ due to the added flexibility. We have seen in Example 6 that this was the case. However, the contradictory effect we have seen in Example 5 might also affect the results we get so we intend to further examine the core of the game $(N, c^2)$ with numerical experiments. We note, the mentioned effect was seen only when the major cost was low relative to the minor costs.

# Chapter 4

# Numerical Study - The Core

In the following chapter we are going to perform a numerical study in order to examine the core of the game that follows from Policy 2 and gain more understanding and insights. Comparisons with Policy 1 will be made and managerial insights will be gained. We are not going to study the non-emptiness of the Imputation set in this chapter because the inventory-optimization literature has dealt with a similar objective extensively by comparing the optimal costs achieved through joint replenishment with the sum of the individual EOQ costs. However we are going to come back and examine the Imputation set when examining the performance of allocation rule because the fact the imputation set is non-empty does not mean that all rules will lead to an allocation which belongs to it. We thus believe that for the examined model it's more interesting to know whether a certain rule provides an allocation which belongs to the imputation set rather than examine if the Imputation set is non-empty. A starting point for the numerical experiments is the paper by Dror, Hartman, and Chang (2012) and their approach, however adjustments and additions to their numerical experiments will be made if that is required.

## 4.1 Setup of the Study and Objectives

A number of important questions arise when considering how Policy 2 relates to Policy 1. Dror, Hartman, and Chang (2012) have simulated random games with cost parameters from a uniform distribution and tried to examine how many games have a nonempty core. They wanted to know this metric because for their game the core may be empty and so it is important to have a feeling of how often this happens and which cost parameters have an effect on the non-emptiness of the core and how. One important finding was that as the number of products in the grand coalition increases the frequency (or probability) of nonempty core decreases. The intuition is that as the number of simulated items increases the more probable it is that items will be included in the grand coalition which have cost parameters which don't allow them to be ordered together. Considering that our policy adds flexibility and more "options" for joint replenishment to the retailers at the cost of slightly increased computational complexity, then quantifying that flexibility and examining the trade-off between flexibility and simplicity is important. It would be interesting to see how this metric corresponds for our policy, since we have seen from examples that we can possibly combine items with different cost parameters and have a non-empty core with our policy, but the core would have been empty for the policy by Dror and Hartman. We have also seen examples with low major cost where the reverse is true! For those reasons it would be interesting to examine the core of the two games and compare the results.

We will try to answer the following two sub-question numerically:

**1) What is the probability that a random game according to the proposed policy is balanced? How does this compare to the policy by Dror and Hartman?**

Considering that the properties of the games $(N, c^1)$ and $(N, c^2)$ change depending on the problem parameters and so the games may have empty cores, one might be interested in knowing which policy leads to a nonempty core with a higher probability. From a cost perspective we know that Policy 2 leads to joint costs that are at least as good as Policy 1, yet this does not mean that cooperation will benefit all of the participating parties, i.e. the core may be empty. In order to get better understanding of this we are going to perform a numerical experiment comparing the two policies, for games with 3 and 4 players and various values of the cost parameters. Considering that we need to examine all balancedness conditions for Policy 2, this will limit the number of players in a game to 4. The reason is that finding all the balancedness conditions for more than 4 players is hard. Another way to check that the core of a specific game is non-empty is to compute an allocation and check whether it is efficient and satisifes stability (i.e. use the definition of the core). For Policy 1, it is possible to examine only the inseparability property, which as we have seen it is equivalent to balancedness. We are going to study all the balancedness conditions for both policies for consistency.

**2) In case the game is not balanced, how much lower should the cost in the grand coalition be so that the game becomes balanced?**

It is important to have an indication of the amount of cost savings that are obtained through the collaboration of the retailers. Even in the case that the core is empty one might still be interested in knowing how much lower the cost of the grand coalition would need to be so that the resulting game would have a nonempty core. By examining how the costs of the grand coalition relate to the costs of other sub-coalitions we can further understand the incentive that a retailer has to stay with the grand coalition and in consequence how important it is that a fair allocation of costs is made. We will thus further examine and gain understanding on the trade-off between optimization and stability. In order to study this question we can define the following metric (for $|N| = 2, 3, 4$) by examining all the balancedness conditions that follow from the minimal balanced collections (more information can be found in Appendix A):

$$e = min_{\kappa \in M_N^B} \left( \frac{\sum_{S \in 2^N \setminus \{\emptyset\}} \kappa(S)c(S) - c(N)}{c(N)} \right)$$

where $M_N^B$ is the set which contains all the balanced maps for a given number of players which follow from the minimal balancedness collections.

The term $(\sum_{S \in 2^N \setminus \{\emptyset\}} \kappa(S)c(S) - c(N))$ represents the difference between the two sides of a given balancedness inequality. Considering that we are going to examine games with various parameters, we divide the term $(\sum_{S \in 2^N \setminus \{\emptyset\}} \kappa(S)c(S) - c(N))$ by $c(N)$ in order to express it in relation to the costs incurred by the grand coalition. This facilitates the comparison of games with different parameters. By taking the minimum over all the inequalities for a game we get a single number (in percentage form) which gives an indication of how close a game is to being balanced or how far it is from being balanced.

If metric e is positive the balancedness inequality with the smallest difference between its two sides is satisfied (and also by definition all of the other inequalities are satisfied). On the other hand, if metric e is negative the game is not balanced. The grand coalition would need to achieve a lower cost so that the examined inequality would have been satisfied and by definition all of the other inequalities would have been satisfied also. The cost of grand coalition would need to be lower by an amount equal to $|e| \cdot c(N)$.

Formally, if we have a game $(N, c)$ and define a new game $(N, c^{'})$ with $c^{'}(S) = c(S), S \subset N$. Then it holds that if $c^{'}(N) > c(N) + ec(N)$, the core is empty and if $c^{'}(N) \leq c(N) + ec(N)$, the core is non-empty. Note that metric e is negative if the core of $(N, c)$ is empty and positive if it is non-empty.

Intuitively, if metric e is positive there are cost savings and we have some intuition about the amount of savings: The balancedness inequalities follow from a weighted map of subcoalitions. The weight of each player in each balancedness inequality is one. By comparing the two sides of an inequality we compare the cost of the grand coalition with the costs obtained if players are partitioned into subsets and multiplied by a weight. This means that we also have an indication of how the grand coalition compares to the cost obtained by the best possible partitioning of players into subcoalitions and we thus also have a conservative estimate of the actual cost savings. This happens because the cost savings correspond to either the balancedness inequality that is found by applying the min operator or is an inequality with a larger difference between its two sides. In both cases metric e provides an indication of the amount of cost savings obtained. The point is more easily understood with an example:

**Example 7:** Suppose we have the 3-player game according to Policy 2 given by the tuple $(N, A, (s_i)_{i \in N}, (D_i)_{i \in N}, (h_i)_{i \in N})$ with $N = \{1, 2, 3\}$, $A = 4$,$s_1 = s_2 = s_3 = 1$, $h_1 = h_2 = h_3 = 1$, and $D_1 = 20, D_2 = 5, D_3 = 1$.

**NOTE:** All the values that we present are rounded to two decimals, but in the actual calculations we do not round in the intermediate steps.

| Coalition | $c^2(S)$ |
|---|---|
| {1} | 14.14 |
| {2} | 7.07 |
| {3} | 3.16 |
| {1,2} | 17.32 |
| {1,3} | 15.56 |
| {2,3} | 8.49 |
| {1,2,3} | 18.73 |

Table 4.1: Example 7 - Cost Game

$c(\{1,2,3\}) \leq c(\{1\}) + c(\{2\}) + c(\{3\}) \Leftrightarrow 18.73 \leq 24.38(1)$

$c(\{1,2,3\}) \leq c(\{1,2\}) + c(\{3\}) \Leftrightarrow 18.73 \leq 20.48(2)$

By applying metric e to all of the balancedness inequalities that follow from minimal balanced collections and not only the two inequalities explicitly described above we find that inequality (2) is the inequality resulting from the min operator: $e = 0.09$.

The cost savings relative to the costs of the grand coalition are equal to: $\frac{c(\{1\})+c(\{2\})+c(\{3\})-c(\{1,2,3\})}{c(\{1,2,3\})} = 0.30$. $\Diamond$

From this example we saw how metric e gives an indication of the amount of cost savings obtained when forming the grand coalition. More specifically metric e states that for a balanced game the grand coalition achieves cost savings at least equal to $e \cdot c(N)$ (lower bound on the cost savings).

We will now try to relate metric e to a well-known solution concept in cooperative game theory, the weak-epsilon core. The weak-epsilon core was first introduced by Shapley and Shubik (1963). Shapley and Shubik (1963) introduced the weak-epsilon core and the strong-epsilon core as means to enlarge the core and explain the dynamics of games which have empty cores. The authors state that games with empty cores should be "more competitive and harder to stabilize" than games with non-empty cores. We have already seen some of the adverse effects of optimization in the game we are examining in example 5 in the previous chapter.

Shapley and Shubik (1963) have defined the strong and weak epsilon cores for a value game. The strong epsilon core imposes that there is an extra cost denoted by $\epsilon > 0$ when forming a coalition $S \subseteq N$. This extra cost lowers the value obtained by each coalition and thus makes it easier to find a stable allocation. The weak epsilon core also imposes a small positive cost $\epsilon > 0$ to each player in a coalition. This means that the total cost which is subtracted from the value obtained by each coalition also depends on the number of players in the coalition. It is possible that both the strong-epsilon core and the weak-epsilon core are empty. But for a large enough value of $\epsilon$ they will be non-empty sets. The core is a subset of the strong-epsilon core and the strong-epsilon core is a subset of the weak-epsilon core.

The concepts of strong and weak epsilon cores are also discussed in the literature on cost allocation cooperative games. Frisk, Göthe-Lundgren, Jörnsten, and Rönnqvist (2010) discuss the epsilon cores and state: "Using an epsilon-core means that we add a minimum penalized slack in the constraints defining the core." This "slack" intuitively means that the actual cost of each coalition $S \subset N$ is enlarged by a small positive amount which can be seen as the extra cost required to form a coalition.

In mathematical notation we can define the strong-epsilon core as follows:

$$C_s(N,c) = \{(x_i)_{i\in N} \in \mathbb{R}^{|N|} : \sum_{i\in N} x_i = c(N), \sum_{i\in S} x_i \leq c(S) + \epsilon \forall S \subset N\}$$

We can define the weak-epsilon core as follows:

$$C_w(N,c) = \{(x_i)_{i\in N} \in \mathbb{R}^{|N|} : \sum_{i\in N} x_i = c(N), \sum_{i\in S} x_i \leq c(S) + |S|\epsilon \forall S \subset N\}$$

We can see that when metric e is negative (so when the core is empty), it can be related to a very specific $\epsilon$ of the weak-epsilon core concept. If we denote that specific $\epsilon$ with $\epsilon_m$ we see that $\epsilon_m = \frac{c(N)|e|}{|N|}$. We know that if $\epsilon \geq \epsilon_m$ then the weak-epsilon core will be non-empty.

At this point we move on to discuss the implementation of the simulation. We remark at this point that since we are using a different random number generator to perform our numerical experiments than Dror, Hartman, and Chang (2012) and since we will examine different sets of parameters at various studies, we possibly expect slightly different numerical results. Nevertheless, we still expect that our results are comparable to the mentioned paper and will thus serve as a form of double-check for the correctness of our experiments. Moreover we intend to report results on the numerical accuracy of our experiments (confidence intervals) and this is not present in the numerical study performed by Dror, Hartman, and Chang (2012). Lastly, we present here the numerical results for one of their conducted experiments: frequency of inseparable games for varying number of items in the grand coalition (recall Theorem 2 - inseparable games with Policy 1 have non-empty core)

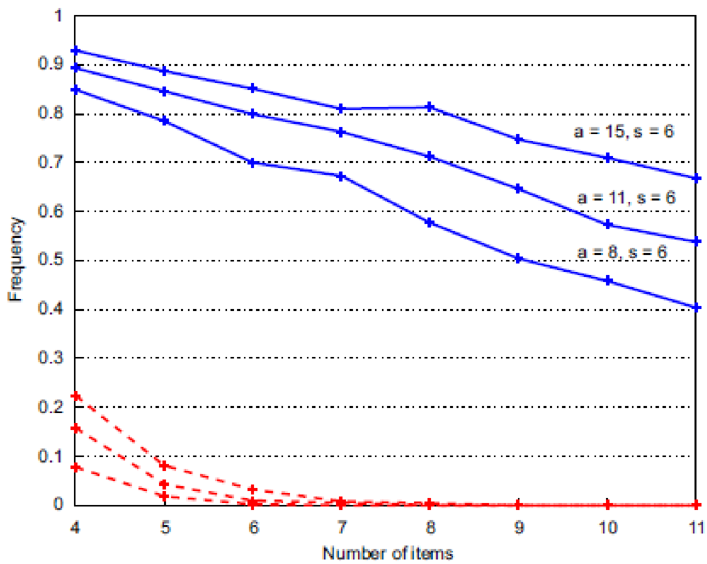



Fig. 6. Varying the number of items.

Figure 4.1: Numerical Results for non-emptiness of the core - Dror, Hartman, and Chang (2012), Blue line: games with non-empty core, Red Line: concave games

Figure 4.1 is taken from Dror, Hartman, and Chang (2012). We can see for example from Figure 4.1 (blue line) that in their numerical experiment the authors found about 90% of the randomly simulated 5-item inventory games with $A = 15$ and $\hat{s} = 6$ to be inseparable and thus have a non-empty core. We note at this point that the authors simulated 1,000 random games for each set of parameters. The remaining parameters used in the particular numerical experiment alongside a description for the study will be given in the subsequent section of the report.

After having described the objectives of our numerical study we need to discuss about the setup of the numerical experiment and the parameters that we are going to use, as well as give a brief overview of the implementation process. This will ensure that our numerical study is transparent and that it can be replicated.

**Setup of the numerical experiment:**

The range of parameters that we will check is an important decision for our numerical study and we are going to use the range of parameters used by the authors. This will allow us to validate our simulation and the results obtained by Dror, Hartman, and Chang (2012) and will serve as a starting point for our subsequent numerical experiments.

| Parameter | Nr. of Values | Sampling interval |
|---|---|---|
| $A$ | 1 | 2,8,15 |
| $s_i$ | n | [1,1+$\hat{s}$]=[1,7] |
| $h_i$ | n | [0 , $\hat{h}$]=[0,2] |
| $D_i$ | n | [$500, 500 + \hat{D}$]=[500 , 2,400] |

Table 4.2: Parameters for numerical study of non-emptiness of core

In our experiment we will perform experiments with 3 and 4 products (or equivalently retailers). The reason we are going to perform experiments with up to 4 players is that it is hard to identify all balancedness conditions for games with more than 4 players. For each number of products used in an experiment we will generate a number of random games according to the parameters given on Table 4.2 and we will examine how many of the randomly generated games (we are going to simulate tuples with random parameters and compute the games that follow from those tuples ) have a nonempty core. We note that while Dror, Hartman and Chang performed experiments with A=15, A=11, and A=8, we choose to include a low value of A=2 as well, and to exclude A=11. The reason is that we wanted to go lower on the major cost and check if similar results hold. The remaining parameters that Dror, Hartman, and Chang (2012) used in the particular numerical experiment are set as follows $\hat{h} = 2, \hat{D} = 1,900$ and can be seen in page 253 of the paper. We note that in each of the numerical studies / sensitivity analyses performed by Dror, Hartman, and Chang (2012), a different set of parameters is used and the parameters presented in their main table are not used throughout.

We will use the parameters in Table 4.2 to determine our first metric: the probability that a game according to our policy has a nonempty core. We will also compute the second metric for each game in order for us to answer our second question. The randomly generated games will be sampled according to a continuous uniform distribution and according to the parameters in Table 4.2. Once we have performed the required number of simulation runs, we will use the results to generate a report which will be discussed in the next section of the thesis.

We will try to estimate the probability that a game $(N, c^2)$ and a game $(N, c^1)$, which follow from a tuple $(N, A, (s_i)_{i\in N}, (D_i)_{i\in N}, (h_i)_{i\in N})$ with parameters drawn from uniform distributions as seen in Table 4.2, have a nonempty core. In other words assuming certain distribution of parameters (uniform) how probable is it that a game is balanced? How frequent are balanced games from those parameters? Any game that follows from a specific tuple can either have an empty/ or non-empty core, but since the cores of the games $(N, c^1)$ and $(N, c^2)$ are not always empty or non-empty, then we have a probability that the core is nonempty, and we will try to estimate it. In simple words suppose we examine a number of tuples with parameters in the described range, what proportion of the games $(N, c^1)$ and $(N, c^2)$ that follow from those tuples have a non-empty core? We can get an estimate for those 2 probabilities by simulating a number of random tuples, examining the games that follow and observing if they have a non-empty core. Subsequently we can divide the number of games which have a nonempty core by the total number of games generated and so get an estimate for the actual and unknown probabilities (or equivalently frequencies, or proportions). Additionally, we are interested in the accuracy of our estimations, so we are going to construct 95% approximate confidence intervals as well.

In Appendix B, we provide the basic textbook theory on approximate confidence intervals, which we are going to use for our numerical experiments. The main source used is (Boon, van der Boor, van Leeuwaarden, Mathijsen, van der Pol, and Resing, 2019). The main formula which we are going to use to construct 95% approximate confidence intervals is the following:

$$\widehat{p} \pm \frac{z_{\alpha/2}\sqrt{\hat{p}(1-\widehat{p})}}{\sqrt{n}}$$

where $\widehat{p}$ is the estimate for the probability or frequency or proportion and $n$ is the number of simulation runs, or simulated tuples, and $z_{\alpha/2} \approx 1.96$ for a 95% confidence interval.

We use the basic, standard confidence because we aim to simulate a large number of games (at least 1,000), and we have no reason at this point to think that the core is always empty or always non-empty for the examined parameters. In our initial investigation of the model we saw that there are games with 3 players where the core is empty and there are cases where the core is non-empty (i.e the actual probabilities are unlikely to be 1 or 0). Moreover other confidence intervals in the literature as the ones proposed by Brown, Cai, and DasGupta (2001), are still approximate confidence intervals and some of them still rely on the central limit theorem, the difference is that there is better accuracy between the achieved confidence level and the nominal confidence level.

In case that an estimated probability is really close to zero or one and so the standard approximate confidence interval will not be correct, we intend to use just for those cases the Wilson interval (Brown, Cai, and DasGupta, 2001):

$$CI_W = \frac{X + z_{\alpha/2}^2/2}{n + z_{\alpha/2}^2} \pm \frac{z_{\alpha/2} n^{1/2}}{n + z_{\alpha/2}^2} \left(\hat{p}(1-\hat{p}) + z_{\alpha/2}^2/(4n)\right)^{1/2}$$

where $X$ is the number of successes in the performed experiments.

The reason we are not using the Wilson interval for all the cases is that when we computed a number of examples in order to compare the two intervals the Wilson interval resulted in very similar approximate confidence intervals as the standard one in cases where the estimated probability was not 0 or 1 and when we are performing many simulation runs, yet the Wilson interval is more complicated and less intuitive. We are only going to use it if the main assumptions for the standard normal approximation confidence interval do not hold.

One important decision that needs to be made is how many simulation instances / simulation runs need to be performed. The number of simulation runs affect the width of the confidence interval. For example, a 95% approximate confidence interval equal to $[0.70, 0.85]$ gives less insight and is less accurate than $[0.68, 0.72]$. By applying the formula $n = \frac{1}{4}(z_{\alpha/2}/\epsilon)^2$ (Appendix B) we can see that if we simulate $n = 1,000$ random games and check the probability for nonempty core then in terms of accuracy we would get a half-width which is almost equal to 0.031. Considering that this is not extremely accurate we also apply the formula for $n = 10,000$ and we get a half-width which is almost equal to 0.01. This would mean that our approximate confidence interval for the estimated probability would be almost equal to: $(\hat{p} - 0.01, \hat{p} + 0.01)$ which we consider accurate.

The flowchart that we used in order to perform this numerical experiment and answer the sub-questions 1 and 2 can be seen in Figure 4.2. Each of the rectangles represent a function (a self-standing program) which is used to perform a task or a simpler operation. When they are combined together as in the flowchart they are used to complete the whole experiment. Each of the functions may make use of other functions in order to perform the required task. All of the functions that are used are described in detail in Appendix D. In brief the flow goes as follows: first the sampling parameters are given along with the number of products for the games (either 3 or 4 for this experiment), also the number of simulation runs is given (here 10,000). Subsequently all the random instances are generated at once in order to save computational time. Then we iterate over those 10,000 instances (tuples) and compute the values for the two cost games, then we check balancedness for each of the games as well as compute the second metric. After we have iterated over all of the randomly generated tuples and computed the games that follow, we use the results to compute the estimates for the probabilities for nonempty core and the confidence intervals and we plot the histogram for the second metric. Because we want to give an overview of the experiment rather than the full implementation details at this point, we decided to include a flowchart that explains all the main steps. Further information on the specific programs used in the flowchart, as well as details about the implementation and validation of our process are available in Appendix D.

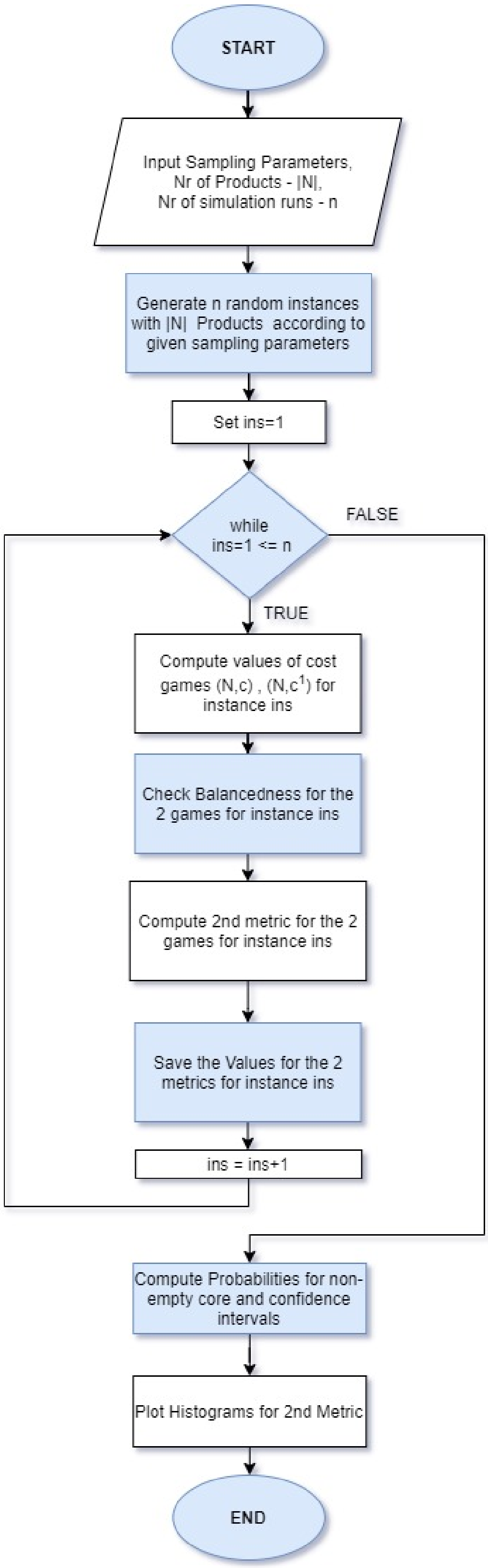



Figure 4.2: Flowchart for Numerical Study On the Core

## 4.2 Numerical Results

We are going to perform the numerical experiment for three sets of major cost ($A$) and upper limit of the minor cost ($\hat{s}$). Those three sets are: ($A = 15, \hat{s} = 6$), ($A = 8, \hat{s} = 6$), ($A = 2, \hat{s} = 6$). The remaining sampling parameters are equal to $\hat{D} = 1,900, \hat{h} = 2$. The same experiments are going to be performed again for $\hat{D} = 10,000, \hat{h} = 25$ (Appendix F). The first set of experiments will be presented in this Chapter and facilitates the comparison of our results with the ones by Dror, Hartman, and Chang (2012).

In order to perform the experiment we are going to sample 10,000 problem instances for number of items equal to 3 and 4, compute the cost games that follow, check how many are balanced according to Policy 2 and how many are balanced according to policy 1. Then we get an estimate for the probability of non-empty core by dividing the number of balanced games by the number of sampled instances (10,000). This estimate is the mean value in the tables. It is important to note that for every randomly generated instance, we create the 2 games according to the two policies and we check those 2 games from the same instance for balancedness. Subsequently for those same 2 games we are going to compute the metric e as defined in the previous section. The results from the experiment can be seen in the respective tables. The confidence intervals are rounded to the fourth decimal.

| Nr. of Products | Policy 2 | | Policy 1 | |
|---|---|---|---|---|
| | Mean Value | Confidence Interval | Mean Value | Confidence Interval |
| 3 | 0.9910 | [0.9891 , 0.9929] | 0.9603 | [0.9565 , 0.9641] |
| 4 | 0.9823 | [0.9797 , 0.9849] | 0.9371 | [0.9323 , 0.9419] |

Table 4.3: Probability of non-empty core, A=15

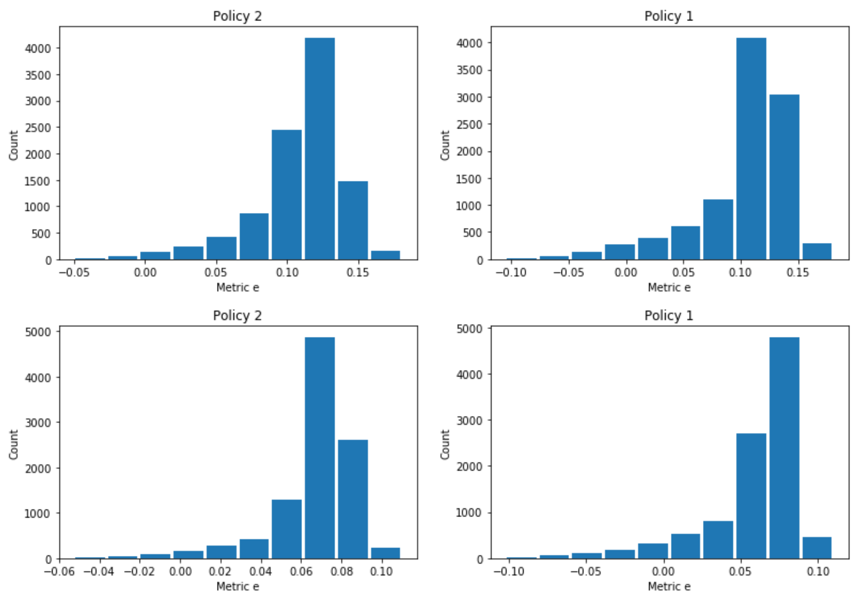



Figure 4.3: Histograms for Metric e for A=15, Upper Graphs: 3-product games , Lower Graphs: 4-product games

| Nr. of Products | Policy 2 | | Policy 1 | |
|---|---|---|---|---|
| | Mean Value | Confidence Interval | Mean Value | Confidence Interval |
| 3 | 0.9719 | [0.9687 , 0.9751] | 0.9061 | [0.9003 , 0.9118] |
| 4 | 0.9577 | [0.9538 , 0.9616] | 0.8459 | [0.8388 , 0.8530] |

Table 4.4: Probability of non-empty core, A=8

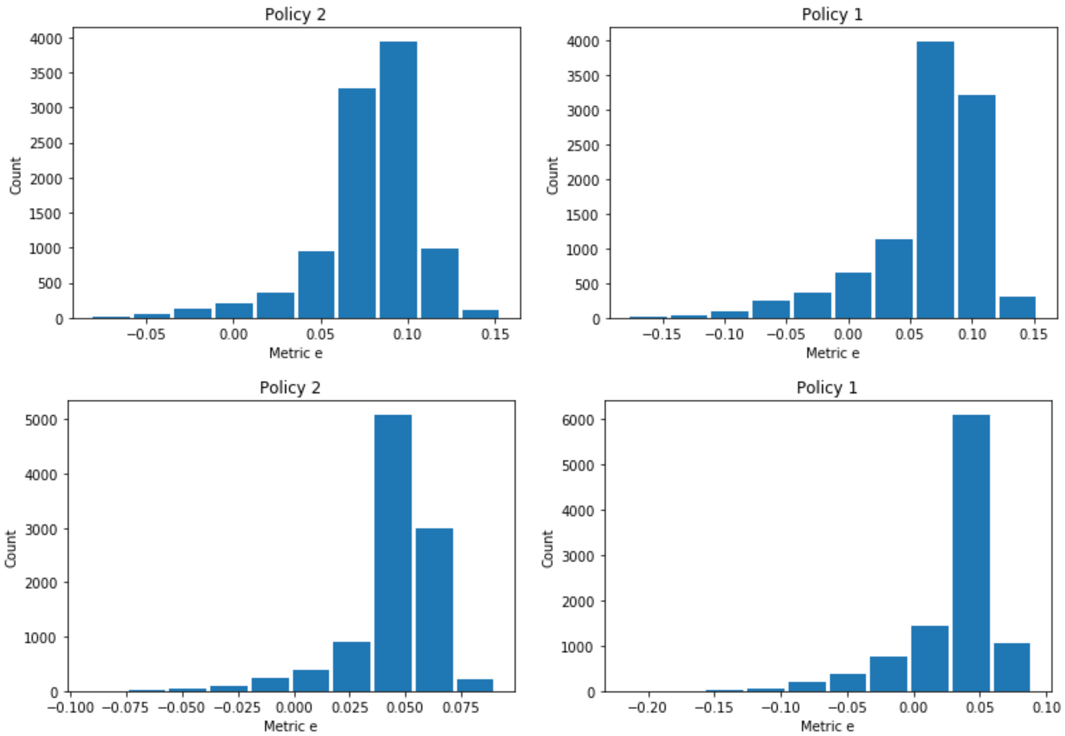



Figure 4.4: Histograms for Metric e for A=8, Upper Graphs: 3-product games , Lower Graphs: 4-product games

| Nr. of Products | Policy 2 | | Policy 1 | |
|---|---|---|---|---|
| | Mean Value | Confidence Interval | Mean Value | Confidence Interval |
| 3 | 0.8807 | [0.8743 , 0.8870] | 0.6215 | [0.6120 , 0.6310] |
| 4 | 0.8185 | [0.8109 , 0.8261] | 0.4549 | [0.4451 , 0.4647] |

Table 4.5: Probability of non-empty core, A=2

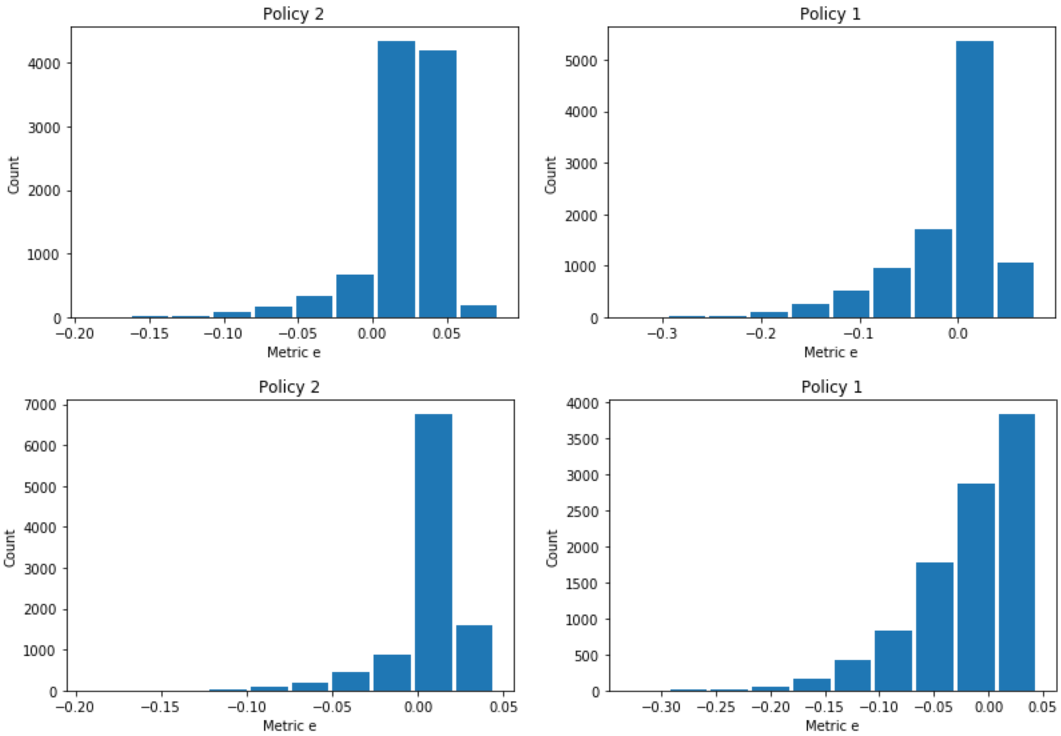



Figure 4.5: Histograms for Metric e for A=2, Upper Graphs: 3-product games , Lower Graphs: 4-product games

We first compare our results with the results by Dror, Hartman and Chang (2012): in Table 4.3 we can see that our estimate for the probability that a 4-item game which follows from a tuple with parameters in the described ranges has a nonempty core when Policy 1 is used (equivalently the proportion of sampled games that have a nonempty core when Policy 1 is used) is equal to 0.9371 and has an approximate confidence interval equal to $[0.9323, 0.9419]$. Dror, Hartman and Chang (2012) report a number of around 94% (pg. 252) and this number can also be seen in Figure 4.1. Similarly the metric for a 4-item game when $A = 8$ is equal to 0.8459 with a confidence interval $[0.8388, 0.8530]$, Dror, Hartman and Chang (2012) report 85% (pg. 252). Those results validate our simulation.

Before describing the obtained results in more detail, we would further like to see how a different range of parameters affects our results. For this reason we repeat the experiments for the following set of parameters: $A = 15, 8, 2, \hat{s} = 6, \hat{h} = 25, \hat{D} = 10,000$. Those parameters have been used by Dror, Hartman and Chang (2012) in different experiments than the one we are performing. Those parameters allow for a wider choice of demand rates and holding costs, which may be more realistic in practice. It is possible that an item has a high demand rate of $10,000$ while another item has a very low demand of $1,000$. Such an "extreme" difference in demand rate was not possible with the previous set of parameters. The results can be seen in Appendix F. The computation times per performed experiment can be seen in Appendix E.

The aim of the numerical experiments in this chapter was to compare the two simple joint replenishment policies: Policy 1 which allows for a single group and Policy 2 which allows for two groups. We wanted to examine whether an inventory game with random parameters has a higher chance of having a non-empty core according to the 2 policies. The reasoning is simple: we know that from a cost perspective Policy 2 is at least as good as Policy 1, yet as we have seen via examples the possibility for a stable cooperation is not guaranteed (the core may be empty). If we can show that our policy also leads to a higher chance of a nonempty core then we can advise the use of Policy 2 for a wider range of parameters, instead of Policy 1.

We get the following insights from our numerical experiment:

- **Number of Products - Flexibility:** The probability of having a non-empty core decreases for both policies as the number of items in the grand coalition increases from 3 to 4. This has been observed with both sets of sampling parameters. This probably has to do with the random selection of the parameters as Dror, Hartman, and Chang (2012) have already pointed out for their game (Policy 1). This means that the more items that are included in the grand coalition the more probable it is that "unmatching" items will be generated and lead to games with empty cores. We think the same holds for Policy 2 but as we can see from Table 4.3, Table 4.4, and Table 4.5 the decrease in the probability is smaller when compared to the decrease in probability for Policy 1. This means that Policy 2 is more flexible and allows the forming of the grand coalition (probability of non-empty core) for more retailers with different cost parameters. This finding states that by allowing the possibility of replenishing every other cycle then there is a higher chance that a slow-moving item can be paired with a fast-moving item than by replenishing them always together (of course this holds if we assume that the other cost parameters are equal between the two items and so only the demand rate is different between the two products). This added flexibility may also be used by sub-coalitions. We have two effects, yet the stronger one is in the grand coalition: the reductions in costs that happen in the grand coalition are greater than the reductions in costs which happen in the sub-coalitions.

- **Effect of Major Cost:** When looking at Table 4.3, Table 4.4, and Table 4.5 we notice that as the major cost decreases the estimated probability of having a non-empty core decreases. This is more profound for Policy 1 than for Policy 2. For A=15 and Number of Products=3 we can see that both policies have a non-empty core for most of the simulated games (i.e. estimated probabilities are very high for both policies). However for A=2 and Number of Products=4 the core is non-empty for 45.49 % of the simulated games for Policy 1 and 81.85 % for Policy 2. In simple words: when the major cost is high there are many alternative policies which can be used for cooperation of retailers with various cost parameters (it is possible to use either Policy 1 or Policy 2 and have a non-empty core in most of the simulated games in the observed range of parameters), but when it is low there are less options for effective policies (using Policy 1 leads to an empty-core for most of the simulated games in the observed range).

- **Games with negative metric e:** The results for the second performance metric can be seen in Figure 4.3, Figure 4.4, and Figure 4.5. On the horizontal axis we have created 10 bins to aggregate the values of metric e for the games we have simulated. On the vertical axis we can see the number of games which contain the values for metric e in a given bin. By looking at the bins containing negative values of metric e, we can see how much lower $c(N)$ would have to be so that games with an empty core have a non-empty core, expressed in percentage form. For A=15 we can see that for Policy 1 $c(N)$ would need to be at most 10% lower for a small count of 3-player games so that the related game would be balanced. For Policy 2 the metric goes as low as 5%. When we look at A=2, we see in Figure 4.5 that the lowest value for metric e is 30% for Policy 1 and around 15% for policy 2. We understand that when using policy 2, if a game is unbalanced it is "less unbalanced" than an unbalanced game with policy 1. For 4-player games, we see that the respective values are equal to 5% for policy 2 and 10% for policy 1 when A=15 and 12.5% vs 28% when A=2.

- **Games with positive metric e:** By looking at the bins containing positive values of metric e, we have an indication about what happens in games with non-empty cores and how the cost of the grand coalition relates to the costs of sub-coalitions. We first see that the patterns in the graphs are similar, yet more games are balanced with Policy 2. The metric e tells us how the cost of the grand coalition relates to the costs from the lowest balancedness inequality, and as a consequence from the best total cost that can be achieved by partitioning the items in sub-coalitions. We have also argued that metric e, when positive, gives a lower bound on the cost savings achieved. The ranges of the lower bounds on the cost-savings are comparable for the two policies (we can see the maximum value of the lower bound of the cost-savings for the two policies on the horizontal axes of the histograms ) yet Policy 2 allows more retailers (we can see the number of games which achieved a certain value for metric e by looking at the vertical axes in the histograms) to join the grand coalition especially when the major cost is low.

- **Size of the core:** By looking at the bins containing positive values of metric e, we have an indication about the size of the core. Speaking intuitively, in a game with a larger positive metric e it is easier to find a core allocation because the grand coalition achieves a cost which is much lower than the best possible cost achieved by a partitioning of players into subcoalitions (also see example 7). This further means that the core is larger. First of all we see that the pattern between both policies in the histograms is similar. When A=15 and Policy 1 is used, most 3-item games have metric e equal to 10%-15%, for Policy 2 this is also the case but the total count of games for which this is happening is higher (Figure 4.3). This means that in Policy 2 more games (with also various parameters due to the parameter distribution) have a core of approximately the same size as Policy 1. From a cost allocation perspective, a rule which performed well with Policy 1 will perform well with Policy 2, for a wider range of parameters, if we further assume that the shape of the core between the two policies is also comparable. We do not have an indication of the shape of the core at this point, but we intend to further study the concept of cost allocation / division of cost-savings in the next chapter.

- **Range of parameters:** When we look the results in Appendix F, we see that the insights and observations we made previously for the defined range of parameters also hold, for the chosen second set of parameters. An important insight is that as we increased the ranges of parameters for the demand rates and holding costs, then less games were balanced for both policies. The intuition is that the larger the range of those parameters the more likely it is that products with "extreme" characteristics will be chosen and so the core is more likely to be empty. This observation holds for both policies but the decrease in the metric is higher for policy 1.

## 4.3 Outcomes

We can make the following comment about Policy 2: if we assume that the chosen cost parameters are representative for a wide range of real-life problems then the simple policy we propose provides increased potential for cooperation between retailers (the core is non-empty in more games with Policy 2 than with Policy 1). There is a need for a policy that results in a nonempty core for retailers with different parameters (and by that reasoning also cooperation is possible for a greater number of retailers). It is only reasonable that in certain projects retailers with different cost parameters may wish to find an allocation for the costs of the grand coalition which belongs to the core, and so a policy which corresponds to their needs should be available. In order to ensure that the retailers stay in the grand coalition though the choice of an allocation rule is important so to ensure that a fair division of costs / benefits is made. This may prove to be a challenging task, because the fact that the core is non-empty does not mean that we always have a way of finding an allocation of costs/ benefits that belongs to the core and possesses desirable properties or is easy to calculate. For example we know that the nucleolus always results in a core element as long as the core is not an empty set, yet as it is computationally expensive, we may be unable to use it for games with many players (for example more than 10 players). An interesting comment made by Dror, Hartman and Chang (2012) on the nucleolus was that according to their estimations for an 11 or 12 item game the nucleolus may require hours to be computed. The performance of allocation rules is the subject of the following chapter in the thesis.

# Chapter 5

# Numerical Study - Performance of Allocation Rules

In this chapter we are going to perform a numerical study in order to examine different ways to allocate the joint costs (or the cost savings) to the retailers when Policy 1 and Policy 2 are used. We are going to further examine the Imputation set and the Core set of the game $(N, c^2)$ for more than 4 players indirectly by examining various allocation rules. As we have discussed in the Introduction, there are at least 2 ways to study if the core is empty: by examining all the balancedness conditions or by finding an allocation which belongs to the core. While the task of finding all the balancedness conditions is hard and complex, computing an allocation which may / may not belong to the core set is relatively easy even for large player sets. The purpose of this chapter is thus twofold: on one hand to examine the performance of various allocation rules in order to provide insights on the allocation of costs/ benefits of the cooperation and on the other hand to provide some more insights on the core and the imputation set of games with more than 4 players. More specifically we aim to provide a lower bound for core non-emptiness by finding allocations using our selected rules. A wide range of cost parameters will be considered in order to ensure that many possible joint replenishment situations are taken into account when recommending an allocation rule.

## 5.1 Setup of the Study and Objectives

Once we have realized that there are benefits in joint replenishment of inventory, it is important to find a way to allocate the costs. One might argue that while a complex to compute rule such as the nucleolus satisfies many interesting game-theoretic properties, an easy to compute rule such as allocating the costs based on the proportion of individual costs may lack interesting properties. This trade-off leads to the need to examine many rules with the hopes of finding one rule which will balance this trade-off. As we have stated in Chapter 1, we want to examine both game theoretic allocation rules but also simpler, proportional allocation rules. The starting point for our numerical study on the performance of allocation rules, is to make a list with allocation rules available in the literature. Subsequently we are going to define our selection criteria, we are going to apply the selection criteria, which will lead to a narrowing of the list of rules and lastly we are going to define the performance measures which we are going to use in our numerical experiment.

The literature on allocation rules is rich, yet not every rule that is in the literature is applicable to the setting we are examining in this project. We have noticed from the literature study prior to the thesis (Pratikakis, 2021) that as most inventory game models are cost games, most of the rules found in the literature relevant to the JRP are cost allocation rules. We would like to use rules to allocate costs which have preferably been used in a similar inventory game model or that can be adapted for our examined setting. However we also intend to consider cost savings allocation rules in order to have a more complete view of the dynamics of the problem. We think that there is value in examining the allocation of cost savings as well because a big part of the game theoretic literature is focused on allocating gains (or profits) of cooperation.

After searching in the literature, the initial pool of allocation rules to be considered for the research can be seen in Table 5.1. An introduction on the Shapley Value and the nucleolus has already been made in Chapter 1 and Appendix A. The nucleolus can be defined either for a cost game or for a cost savings game. In the appendix it is defined for a cost game. The Proportional Allocation rule (as termed by Liu, Wu, and Xu (2010)) is a rule to divide the benefits of the collaboration (and also cost savings) and is often used in practice (Liu, Wu, and Xu, 2010). The remaining rules are cost allocation rules which have been identified during the course of the research on inventory game models.

| **Cost Allocation Rule** | **Description & Formula** | **Reference** |
|---|---|---|
| Shapley Value | More Details in Appendix A | (Shapley, 1953) |
| Nucleolus | More Details in Appendix A | (Schmeidler, 1969) |
| Proportional Allocation | $\left(\frac{c_i^*}{\sum_{j\in N} c_j^*} \cdot v(N)\right)_{i\in N}$ | (Liu, Wu, and Xu, 2010) |
| DEMAND Rule | $\left(\frac{D_i}{\sum_{j\in N} D_j} \cdot c(N)\right)_{i\in N}$ | (Dror, Hartman, and Chang, 2012) |
| INDIVIDUAL Rule | $\left(\frac{c_i^*}{\sum_{j\in N} c_j^*} \cdot c(N)\right)_{i\in N}$ | (Dror, Hartman, and Chang, 2012) |
| INCREMENTAL Rule | $\left(\frac{c(N)-c(N/\{i\})}{\sum_{j\in N} c(N)-c(N/\{j\})} \cdot c(N)\right)_{i\in N}$ | (Dror, Hartman, and Chang, 2012) |
| LOUDERBACK Rule | $\left(c(N)-c(N/\{i\})+\frac{[c_i^*-(c(N)-c(N/\{i\}))]\cdot[c(N)-\sum_{j\in N} c(N)-c(N/\{j\})]}{\sum_{j\in N} c_j^*-[c(N)-c(N/\{j\})]}\right)_{i\in N}$ | (Dror, Hartman, and Chang, 2012) |
| Order Rule | $\left(\frac{(m_i^*)^2}{\sum_{j\in N} (m_j^*)^2} \cdot c(N)\right)_{i\in N}$ | (Tinoco, Creemers, and Boute, 2017) |
| Square Rule | $\left(\frac{(c_i^*)^2}{\sum_{j\in N} (c_j^*)^2} \cdot c(N)\right)_{i\in N}$ | (Tinoco, Creemers, and Boute, 2017) |
| Cost Allocation Rule | More Details in the Paper | (Anily and Haviv, 2007) |

Table 5.1: List of Allocation Rules In The Literature

The Shapley rule and the nucleolus are allocation rules which are defined for games, while the remaining rules are defined for situations. Rules defined for situations require additional parameters and not only the values of each coalition in order to be computed. As an example, the DEMAND and INDIVIDUAL rules, require the demand rates and the individual costs respectively. A brief description of the rules follows:

1. **Shapley Value:** the Shapley value contains many interesting game-theoretic properties and someone might argue that it can be a fair way to allocate the costs irregardless of whether the resulting cost allocation will lie in the core or not. This has been argued for instance by Otero-Palencia, Amaya–Mier, and Yie-Pinedo (2019) in their paper. We denote the Shapley value for the game (N,c) as $\Phi(N, c)$. More details on the Shapley value, its properties, and how it is computed can be found in Appendix A.

2. **Nucleolus:** the nucleolus is a well-known rule in cooperative game theory. The nucleolus always gives a core allocation as long as the core set is non-empty. The nucleolus is defined for games with a non-empty imputation set only. More information can be found in Appendix A.

3. **Proportional Allocation:** The "proportional allocation" rule as it has been described by the authors in the mentioned paper is a way to allocate the gains of cooperation based on the ratio of individual costs. This means that the cost savings obtained in the grand coalition are divided among players according to their individual costs. A player with a higher individual costs will be allocated more cost savings. The rule is defined as follows: $y^P = \left(\frac{c_i^*}{\sum_{j\in N} c_j^*} \cdot v(N)\right)_{i\in N}$. Notice that the "proportional allocation" rule as termed by the authors results in an allocation of cost-savings.

4. **DEMAND Rule:** The demand rule allocates the cost of the grand coalition based on the fraction of the total demand generated. This rule may seem intuitive, but we remark that in most inventory models the demand rate is only one of the parameters that affect cost performance. We remark that the DEMAND rule is defined for a situation, because we need demand rates in order to compute it. The rule is defined as follows: $x^D = \left(\frac{D_i}{\sum_{j\in N} D_j} \cdot c(N)\right)_{i\in N}$

5. **INDIVIDUAL Rule:** this rule allocates an amount of the total cost of a coalition based on the ratio of individual cost of a player over the sum of the individual costs of the players. We will denote the cost allocation for game $(N, c)$ that is derived using the INDIVIDUAL rule as follows: $x^I = \left(\frac{c_i^*}{\sum_{j\in N} c_j^*} \cdot c(N)\right)_{i\in N}$. This rule is again easy to understand and easy to compute but now with the added benefit that if the retailers reveal only their individual costs then it is not possible for the other retailers to identify the other individual parameters such as the demand rate. We remark that for this rule the individual costs $c_i^*$ are parameters for the joint replenishment situation and so they do not change when we use the rule in the respective cost-savings situation (i.e. the rule may be used to allocate the cost-savings based on the ratio of individual cost of a player over the sum of the individual costs of the players). We also remind that $c_i^* = c^2(\{i\}) = c^1(\{i\})$.

6. **INCREMENTAL Rule:** The INCREMENTAL rule allocates an amount of the total cost based on the ratio of the marginal cost of a player over the sum of the marginal costs of the players. We will denote the cost allocation for game $(N, c)$ that is derived using the INCREMENTAL rule with: $x^{IN} = \left( \frac{c(N)-c(N/\{i\})}{\sum_{j\in N} c(N)-c(N/\{j\})} \cdot c(N) \right)_{i\in N}$. We remark that while we saw this rule in a joint replenishment setting (Dror, Hartman, and Chang, 2012 ), the rule does not use any information extremely specific to the joint replenishment setting. It only uses the marginal costs so it may be possible to use this rule in other settings where many items are jointly replenished. We note that the rule is not always well-defined and so it is important to consider this when performing numerical experiments.

7. **LOUDERBACK Rule:** The LOUDERBACK rule allocates an amount of the total cost according to the following formula: $x^L = \left( c(N) - c(N/\{i\}) + \frac{[c_i^* - (c(N)-c(N/\{i\}))] \cdot [c(N) - \sum_{j\in N} c(N)-c(N/\{j\})]}{\sum_{j\in N} c_j^* - [c(N)-c(N/\{j\})]} \right)_{i\in N}$. This rule allocates the marginal cost to a player and the remaining cost is allocated proportionately to the difference of standalone and marginal cost. Similarly to the INCREMENTAL rule, this rule uses the marginal costs and individual costs as indicators to allocate the joint costs. We note that this rule might give unexpected results (allocate too much cost on certain items/ negative allocations on other items) in case that the core or imputation set are empty (we have noticed this finding on some initial experiments we performed).

8. **Order Rule:** this rule allocates an amount of the total cost of a coalition based on the ratio of the squared individual order frequency of a player over the sum of the squared individual order frequencies of all the players. A variant of this rule has been found in many different papers under the name proportional or distribution rule (Meca, Timmer, Garcia-Jurado and Borm, 2004 ; Fiestras-Janeiro, Garcia-Jurado, Meca, and Mosquera, 2011) or order rule (Tinoco, Creemers, and Boute, 2017) and it was used to allocate different parts of the total costs. We will use the rule as defined in the paper by Tinoco, Creemers, and Boute (2017) in order to allocate the total costs. We will denote the cost allocation for game $(N, c)$ that is derived using the Order rule as follows: $x^O = \left( \frac{(m_i^*)^2}{\sum_{j\in N} (m_j^*)^2} \cdot c(N) \right)_{i\in N}$. This rule is defined for the joint replenishment situation because it requires information such as holding costs, demand rates and major and minor costs.

9. **Square Rule:** this rule allocates the costs of the grand coalition based on the ratio of the squared individual cost over the sum of the squared individual order costs. When we compare it with the INDIVIDUAL or the "Proportional Allocation" rule, we see that the Square rule penalizes the players with high costs as it allocates more cost from the grand coalition to them (when we compare with the allocated cost from the INDIVIDUAL rule). The original paper where we found the rule dealt with a stochastic joint replenishment setting (Tinoco, Creemers, and Boute, 2017). The rule is defined as follows: $x^S = \left( \frac{(c_i^*)^2}{\sum_{j\in N} (c_j^*)^2} \cdot c(N) \right)_{i\in N}$.

10. **Cost Allocation Rule by Anily and Haviv (2007)**: Those rules have been designed for a joint replenishment situation where the optimal POT policy is used. The following is stated: "There exists a core allocation under which all retailers pay their own minor setup costs and holding costs, and each retailer in the minimal

set, pays part of the major setup cost. A retailer not in the minimal set, does not pay anything toward the major setup cost". The minimal set refers to the retailers which order in every period. This rule has also been described in the Introduction. Considering that the rule is defined for a very specific setting, we do not give its formula here, but refer the reader to Anily and Haviv (2007) for more specific information on its calculation.

Dror, Hartman and Chang (2012) state the following properties/ criteria for selecting an allocation rule:

1. The allocation rule needs to be readily *explainable*. We interpret this property to mean that the rule can be easily explained without relying on too much specialized knowledge and is intuitive.

2. The allocation rule needs to be *easy to calculate*.

3. The allocation rule needs to be *justifiable*. This means the rule needs to be consistent when allocating costs or cost savings and result in an unambiguous allocation (an example is provided later in this chapter).

4. The allocation rule needs to be *fair* (belong in the core of the examined game).

One might argue that selecting a rule might be a subjective process, since there may be difficulties in agreeing on the criteria. Based on the literature we found those 4 criteria and we will try to select our rules based on a similar yet not the same line of thought. We would like to find rules which satisfy as many of the above properties as possible, yet this is not an easy task. When examining the 4 criteria we understand that we cannot use criterion 4 as a selection criterion because except for the nucleolus and the Shapley value, we do not know a priori when the other rules will lead to a core allocation. For the Shapley value we know that for concave games it results in a core allocation and for nucleolus we know that it results in a core allocation if the core set is non-empty. Regarding the second criterion: the proportional rules are the easiest to calculate since they require little computation time and most of them are based on a simple ratio. The Shapley value and the nucleolus require more computation time than the proportional rules because full subset enumeration is needed. The nucleolus is relatively harder to calculate than the Shapley value. To conclude, we are going to use the first 3 criteria to perform the selection of allocation rules and subsequently also judge their quality based on the performance of our numerical study. We note that while our selection procedure is not 100% objective, it is based on criteria from the scientific literature. We will start examining every rule, and in the case one or more of the three selected criteria are not satsified, we are going to exclude the rule from the numerical study.

We notice that by applying the criteria, the rule of Anily and Haviv (2007) is excluded because it is not readily explainable. This rule takes advantage of the form of the cost function for a very specific game to offer a core allocation. The retailers who order in every cycle (most frequently) pay their own holding costs and minor ordering costs plus the major costs. Retailers who order less frequently pay their holding costs and minor costs and nothing towards the major cost. This rule is counter-intuitive as it is hard to justify why someone who orders rarely never contributes to the joint cost. This argument has also been made by Anily and Haviv (2007). This rule has been designed for a very specific setting and under very specific assumptions. We can argue that although it has been tested on a joint replenishment model with good results, it is still hard to justify its use in an intuitive way. This rule is also relatively hard to "translate" to a cost-savings setting which means it might be hard to find a rule which relates to this one based on a benefit game (i.e. we do not know if the rule is justifiable).

In Dror and Hartman (1996) it is stated that the Shapley value and the nucleolus are self-justifiable / self-dual and since those rules are rules for games, we can use this result for our own setting also. The nucleolus is excluded because it is not an easy to calculate rule. The Shapley value is relatively easier to calculate than the nucleolus (yet it still requires many computations ), can be explained easily, and satisfies the justifiability criterion.

The concept of justifiability has been studied in the literature by Dror and Hartman (1996). A distinction is made between "self-justifiable" or "self-dual" allocation rules/methods and "dual justifiable" rules/methods. The "self-justifiable" rules are the rules which are applied without any changes to both the cost-savings game and the respective cost game, while the "dual justifiable" rules for a cost-savings game or a cost game have a natural dual rule which if applied to the respective cost game / cost-savings game will create a justifiable pair of allocations. We learn from the authors that the rule DEMAND is not justifiable in the described setting. We further learn that the rules INDIVIDUAL, INCREMENTAL, and LOUDERBACK are justifiable in the described setting (they are also easy to compute and easy to explain). The INDIVIDUAL rule is "self-justifiable", while the INCREMENTAL and LOUDERBACK rules are "dual justifiable" rules in the described setting. Their dual rules can be found in the Appendix of Dror and Hartman (1996). From the definition of justifiability (the player's benefit is equal to the individual cost minus his allocated cost) we understand that the Proportional rule which allocates cost savings proportional to the individual costs needs to be excluded from the experiment because we already include the INDIVIDUAL rule and so the allocations resulting from those rules will be identical. We understand that the INIDVIDUAL rule is justifiable in every setting, while we saw that Dror and Hartman (1996) were able to construct justifiable pairs for the INCREMENTAL and LOUDERBACK rules. The INCREMENTAL and LOUDERBACK rules (and their duals) have been defined for a centralized inventory setting and as we can see from their definitions in Table 5.1, they only require the values of the cost game, in order to be computed. This further means we can use the justifiability results of Dror and Hartman (1996) for those two rules.

For the order and square rule we construct some examples to see if they are self dual, because we know that they fulfill the remaining criteria. Both the Square and the Order rule are not self-dual and we do not know if they are dual-justifiable. Note the example provided below:

**Example 8:** Consider the cost game discussed in Example 1 of Chapter 3.

Note that only the final cost values are rounded to two decimals and not the intermediate steps.

Note from Table 3.1 that: $m_1 = 0.5, m_2 = 0.08$

| Coalition | $c^2(S)$ |
|---|---|
| {1} | 4 |
| {2} | 0.63 |
| {1,2} | 4.35 |

Table 5.2: Cost Game of Example 1

| Coalition | $v^2(S)$ |
|---|---|
| {1} | 0 |
| {2} | 0 |
| {1,2} | 0.29 |

Table 5.3: The Equivalent Cost-savings Game

We compute the cost allocation and the cost savings allocation by using the order rule. Subsequently we use the results from the cost savings allocation to redistribute the cost savings and subsequently find the costs that the companies would have to pay if the order rule was used in order to allocate cost savings:

| **Cost Allocation** | |
|---|---|
| Cost Allocation for Player 1: $x_1$ | 4.24 |
| Cost Allocation for Player 2: $x_2$ | 0.11 |
| **Cost-savings Allocation** | |
| Cost-savings Allocation for Player 1: $y_1$ | 0.28 |
| Cost-savings Allocation for Player 2: $y_2$ | 0.01 |
| **Resulting Costs from Cost-savings Allocation** | |
| Resulting Costs for Player 1: $c_1'$ | 3.72 |
| Resulting Costs for Player 2: $c_2'$ | 0.63 |

Table 5.4: The Equivalent Cost-savings Game

From Dror, Hartman, and Chang(2012), we also find the following formulas for the EOQ: $m_i = \sqrt{D_i h_i/(2(A+s_i))}$ and so $c_i^* = 2m_i(A+s_i)$.

Similarly for the square rule, we have the following expression:

$$\frac{c_i^{*2}}{\sum_{j\in N} c_j^{*2}} c(N) = \frac{4m_i^2(A+s_i)^2}{\sum_{j\in N} 4m_i^2(A+s_i)^2} c(N)$$

Because in this example $s_1 = s_2$, the two rules will result in the same allocations , but this is not always the case. So we have shown by means of an example that neither the square nor the order rule are self-dual.

We understand the importance of justifiability from this example: if the game $(N, c^2)$ had been defined as a cost-savings game then by using the Order rule we get two different allocations from the same rule! This might create problems and affect cooperation, because Player 2 would have more benefit from the allocation of costs using the Order rule, while Player 1 would have more benefit from the allocation of cost-savings using the

Order rule. $\Diamond$

We do not know if there is another rule that can make it possible for us to create a dual-justifiable pair for the square and order rules. Since for other rules from the list (INRCREMENTAL and LOUDERBACK) we know from literature that this was possible and since those rules are also easy to calculate and readily explainable we will prefer them over the Square and Order rules.

After applying the selection criteria to the full list of allocation rules, we come up with 4 allocation rules which are going to be used in our numerical study (the Shapley value will be calculated based on the values of the cost game, yet it was not originally defined for a cost game).We intend to study the performance of the following allocation rules when allocating the costs of Policy 2: the Shapley value, the INCREMENTAL rule, the INDIVIDUAL rule, and the LOUDERBACK rule.

After describing the allocation rules which we are considering, we need to pick appropriate measures or metrics for their performance. We are considering the following performance metrics for our research:

1. The probability that an allocation rule results in an allocation $x$ that lies in the imputation set for the game $(N, c)$. We will denote this metric as $Pr_I$. This metric is very important as every retailer is interested in knowing how the costs allocated to him compare with the costs he would have had without cooperating with any other retailer. An allocation that is not individually rational is simply unacceptable for a retailer and so the probability that an allocation rule results in an allocation that belongs to the imputation set is the first important (and preliminary) metric for the quality of an allocation rule.

2. The probability that an allocation rule results in an allocation $x$ that lies in the core for the game $(N, c)$. We will denote this metric as $Pr_C$. The second metric that we choose for the research builds upon the previous metric and further requires that an allocation obtained with the use of an allocation rule cannot be further improved upon by any coalition. While the imputation set simply states that an allocation of costs is an improvement over the individual costs, it does not say which coalition is the best from a cost perspective for a retailer. On the other hand core allocations will ensure that all of the retailers in a group have no incentive of departing from the grand coalition and further no "bargaining" for the best coalition will take place as a result of the allocation of costs.

One can imagine that the metrics correspond to two different levels of "quality" for the allocation rules: a rule that leads to a cost allocation which is often in the imputation set would be interesting for companies which are only interested in minimizing their individual costs and do not care much for possible deviations from the grand coalition and formation of sub-coalitions. For instance one can imagine a franchise company where all retailers are part of the same larger company and thus possible deviation and forming of sub-coalitions is not possible, yet still it makes sense that retailers improve through their collaboration in the grand coalition, even if a possible sub-coalition will lead to improved costs. On the other hand the core represents a set of cost allocations which cannot be improved by any other coalition and so it is a metric for the quality of an allocation rule which can be used by retailers which are parts of different companies and so they are interested in their own best possible solution under any coalition.

**NOTE:** Similarly to the metrics used in Chapter 4, we are computing numerically by means of simulation an estimator for the mentioned probabilities, based on randomly generated problem instances. We are going to use two different sets of data for our simulation. The first set of data was used by Dror, Hartman, and Chnag (2012) in their study on performance of allocation rules and can be seen on Table 5.5. The second set of parameters has been used in the experiments we performed for chapter 4 in Appendix F.

It may be the case that the rules are "correlated" with each other: the allocations that are found may be close to each other every time and so when one rule results in a core element so does another. It may also be the case that a rule results in a core element for a game while another does not. It would be interesting to know whether we can always find a core element with one of our chosen four rules as long as the core is non-empty (Probability that at least one rule will result in a core element). For games with 3 and 4 players we know when the core is nonempty so we can also compute the fraction of games with non-empty cores where we did find a core allocation with one of the four rules. However, we can always compute the fraction of games where we found a core allocation and that is the metric we are going to use for consistency. We denote the metric as the estimate for the Probability that at least one rule will result in a core element. The metric also provides a lower bound on the number of games with non-empty cores.

**Setup of the numerical experiment:**

Similarly to the setup of the numerical study in chapter 4, we are going to generate a number of random tuples for a given number of items and cost parameters, compute the games that follow, and subsequently we are going to use our chosen rules to compute cost allocations and check if they belong to the imputation set and/or to the core. After we have simulated and checked the metrics for the desired number of games/ simulation runs, we are going to compute the estimates for the probabilities and the confidence intervals.

We are going to simulate games with 3 up to 8 players, as we aim to examine how the increase in number of players affects the performance of the rules. We believe that collaborations with more than 8 retailers may not realistically arise in practice and also the computation time for the experiment increases a lot as the number of players increases (this is the case because we compute the Shapley value for each game). Moreover by examining up to 8-player games we will be able to get better understanding of the dynamics of the game since our previous experiments were limited to 4-player games. Lastly, one can imagine that many products can be aggregated into product families, because they may contain similar characteristics. If that is the case then our assumptions will be sufficient to judge the quality of the allocation rules for our game. We are going to simulate 10,000 n-player games for $n = 3, 4, 5, 6, 7, 8$. The simulated games with 3 and 4 items are not the same games as the ones in chapter 4, because a different script than in chapter 4 is used. The reasoning for this number of games/ simulation runs has been described in chapter 4. The flowchart we use for this numerical experiment is similar to the one of Chapter 4 and can be seen on the respective figure.

**NOTE:** the Shapley value will be calculated by using a script available on the internet (susobhang70/shapleyvalue: Python code to find Shapley Value of a Characteristic Form Game - GitHub, n.d.). Some adjustments have been made to the type of input and output of the script so that it can work with the rest of the program. The script was also validated by using examples which were available in the literature. More details on implementation and validation are available in Appendix D.

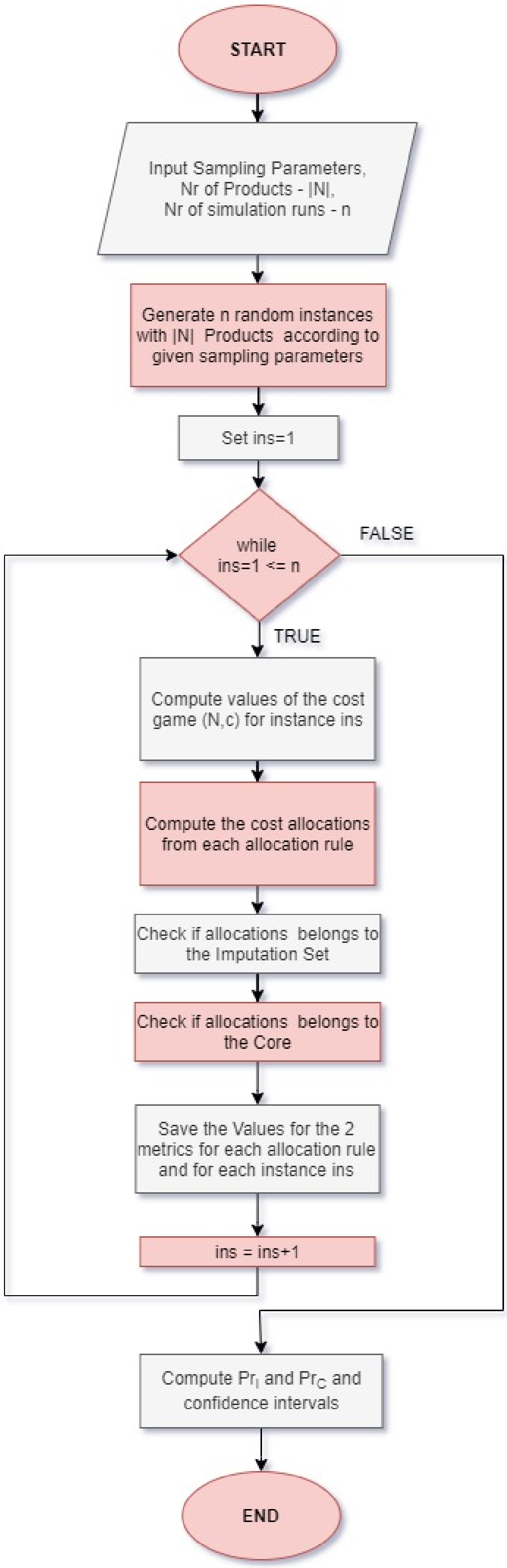



Figure 5.1: Flowchart for Numerical Study on Allocation Rules

## 5.2 Numerical Results

In the first study we are going to perform, we will use the parameter set that was used by Dror, Hartman, and Chang (2012). The parameters used for the study are presented in Table 5.5. In order to compare policy 1 and policy 2 for those parameters we are going to use the metrics defined section 5.1. We are going to simulate 10,000 8-item games for this study. The computation times per performed experiment can be seen in Appendix E.

| Parameter | Nr. of Values | Sampling interval |
|---|---|---|
| $A$ | 1 | 20 |
| $s_i$ | n | [1,1+$\hat{s}$]=[1 , 4] |
| $h_i$ | n | [0 , $\hat{h}$]=[0 , 2] |
| $D_i$ | n | $[100, 100 + \hat{D}]$=[100 , 2,100] |

Table 5.5: Parameters for first numerical study of non-emptiness of core

**NOTE:** the confidence intervals in italics have been calculated according to the Wilson interval, because the use of the standard normal approximation confidence interval resulted in either overshoot(upper bound of confidence interval which is greater than 1 ) or zero-width intervals (which falsely imply certainty).

| **Allocation Rule** | **Policy 2** | **Conf. Intervals** | **Policy 1** | **Conf. Intervals** |
|---|---|---|---|---|
| Shapley Value | 0.9830 | [0.9805 , 0.9855] | 0.9393 | [0.9346 , 0.9440] |
| INDIVIDUAL Rule | 1.0000 | *[0.9996 , 1.0000]* | 1.0000 | *[0.9996 , 1.0000]* |
| INCREMENTAL Rule | 0.9492 | [0.9449 , 0.9535] | 0.8339 | [0.8058 , 0.8210] |
| LOUDERBACK Rule | 0.9741 | [0.9710 , 0.9772] | 0.9073 | [0.8827 , 0.8951] |

Table 5.6: Imputation Set Metric - 8-Item Games , 2 Policies

| **Allocation Rule** | **Policy 2** | **Conf. Intervals** | **Policy 1** | **Conf. Intervals** |
|---|---|---|---|---|
| Shapley Value | 0.8835 | [0.8772 , 0.8898] | 0.6241 | [0.6146 , 0.6336] |
| INDIVIDUAL Rule | 0.8066 | [0.7989 , 0.8143] | 0.5307 | [0.5209 , 0.5405] |
| INCREMENTAL Rule | 0.9476 | [0.9432 , 0.9520] | 0.8134 | [0.8058 , 0.8210] |
| LOUDERBACK Rule | 0.9726 | [0.9694 , 0.9758] | 0.8889 | [0.8827 , 0.8951] |
| **$\hat{\text{Pr}}$ (of finding a core allocation - Policy 1)** | | | 0.8891 | |
| **$\hat{\text{Pr}}$ (of finding a core allocation - Policy 2)** | | | 0.9726 | |

Table 5.7: Core Metric - 8-Item Games , 2 Policies

By using the results from Table 5.7, we can compute the metric used by Dror, Hartman, and Chang (2012) and validate our results (and theirs). The metric that the authors used to judge the performance of allocation rules is the frequency of allocation rule in the core when the core is non-empty. For policy 1, it is possible to compute the metric because we know that inseparable games have a non-empty core. We remind that computing whether a game is inseparable is possible even for games with more than 4 players. In our simulation 9,004 / 10,000 games were inseparable and so balanced. We compute the metric for the Shapley value as follows: $6,241/9,004 = 0.6931$. Similarly we compute the metric for the other rules and round the results to 4 decimals: INDIVIDUAL - 0.5894, INCREMENTAL - 0.9034, LOUDERBACK - 0.9872. The authors find a frequency of 70.2% for the Shapley value, a frequency of 98.8% for the LOUDERBACK rule, a frequency of 90.0% for the INCREMENTAL rule and a value of 58.3% for the INDIVIDUAL rule (pg. 253 of the article). This numerical experiment validates our results and the results by Dror, Hartman, and Chang (2012).

Our second study will focus on the performance of allocation rules when Policy 2 is used. We have compared the performance of rules according to the two policies and we can now examine the performance of the rules when policy 2 is used, but also examine varying number of items. We are going to use the parameter set that was used in the second numerical study of chapter 4. The reason we are using this parameter set is that it contains a wider range of parameters for the demand rates, and holding costs and this will allow us to sample many different tuples and better judge the performance of the rules. We restate the sampling parameters for this study $A = 15, 8$, the holding costs are sampled from the range [0,25], the demand rates from the range [500 , 10,500] and the minor costs from the range [1,7].

The results from the study can be seen in tables in Appendix G.We summarize the results from those tables in the following graphs so comparisons can be made more easily:

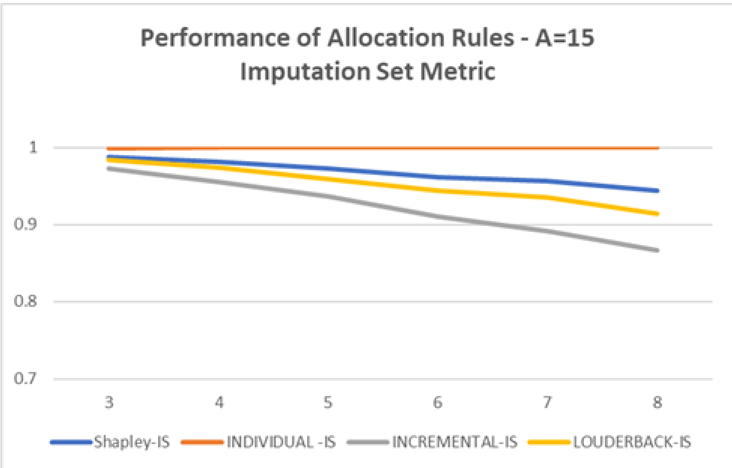



Figure 5.2: Policy 2, A=15 , Imputation Set Metric

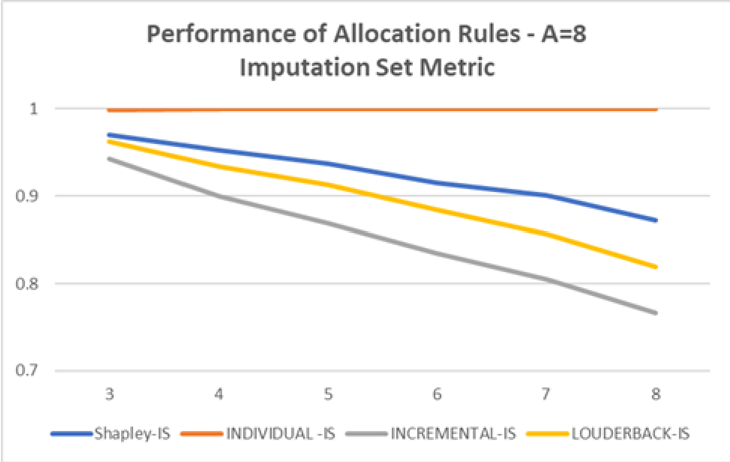



Figure 5.3: Policy 2, A=8 , Imputation Set Metric

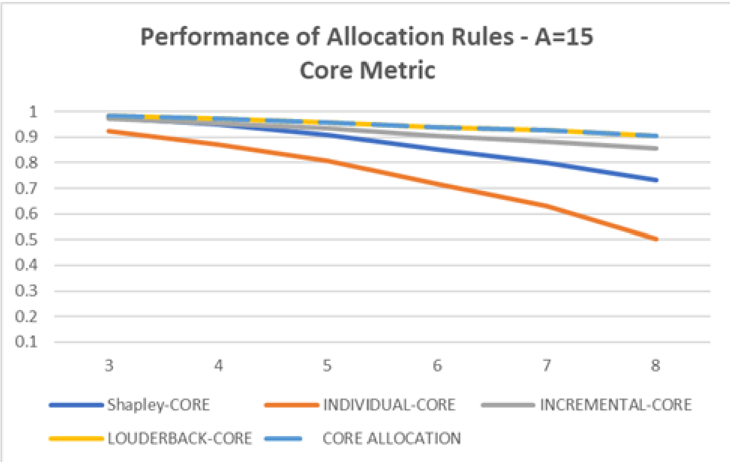



Figure 5.4: Policy 2, A=15 , Core Metric

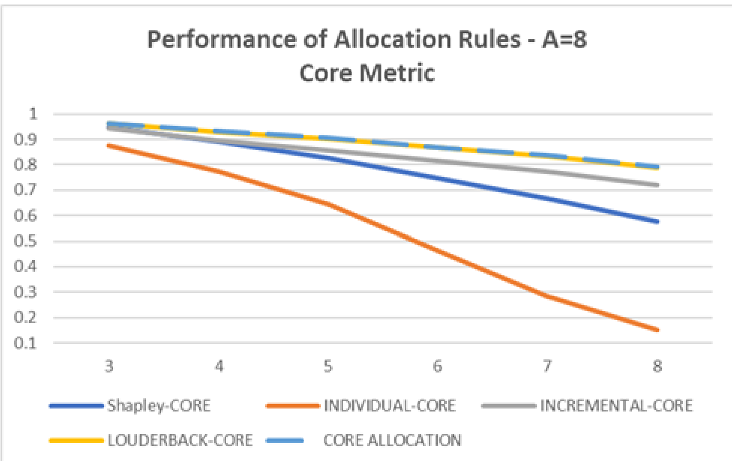



Figure 5.5: Policy 2, A=8 , Core Metric

We obtain the following insights from the numerical experiments conducted in this chapter:

- **Imputation Set Metric:** Regarding the performance of the allocation rules on the first metric, we can see that the INDIVIDUAL rule almost always results in an allocation which belongs to the imputation set for the performed experiments. More specifically by its definition we understand that the INDIVIDUAL rule always results in an allocation which belongs to the imputation set when the imputation set is non-empty: $x_i^I \leq c(\{i\}) \Leftrightarrow \frac{c_i^*}{\sum_{j \in N} c_j^*} c(N) \leq c(\{i\}) \Leftrightarrow \frac{c(\{i\})}{\sum_{j \in N} c(\{j\})} c(N) \leq c(\{i\}) \Leftrightarrow c(N) \leq \sum_{j \in N} c(\{j\})$. In the conducted experiments the imputation set was non-empty when the number of players was larger than 3.

  The performance of the allocation rules INCREMENTAL, LOUDERBACK and Shapley value declines as the number of products increases. The performance of those rules also decreases when the value of A decreases. The best performing rule is the INDIVIDUAL rule, followed by the Shapley value, then the LOUDRERBACK rule and finally the INCREMENTAL rule. The difference in the performance of the rules in the imputation set metric is moderate for both values of A.

- **Core Metric:** The performance of all the allocation rules declines as the major cost decreases. The perfomance of the rules also declines when the number of products increases and this is mediated by the major cost: the lower the major cost the more profound the decrease in performance when the number of products in the grand coalition increases. In general, we are able to identify a core element for almost nine out of ten games or higher when A=15 and for eight out of ten games or higher when A=8.

  The best performing rule is the LOUDERBACK rule followed by the INCREMENTAL rule, the Shapley value and the INDIVIDUAL rule (a small, insignificant change in ranking for the core metric between the Shapley Value and the INCREMENTAL rule is observed for 3- and 4-item games when A=15, and in 3-item games when A=8). The LOUDERBACK rule resulted in a core element in most of the simulated games, yet not in all of them. It is further interesting to note that all rules decrease in performance in the metric when the number of products increases, yet only the INDIVIDUAL rule shows a sharp decrease. The rules INCREMENTAL, LOUDERBACK, and the Shapley value do not decrease in performance sharply.

- **Game-theoretic properties vs. Computational speed:** We make a comment regarding the trade-off between game-theoretic properties and computational speed: the Shapley value requires enumeration of all the sub-coalitions in order to be computed while the rules INCREMENTAL and LOUDERBACK require a lot less computations. This makes the Shapley value a computationally expensive rule. On the other hand the Shapley value is the only rule which satisfies the properties: efficiency, zero-player property, additivity, and symmetry. The rules INCREMENTAL and LOUDERBACK do not satisfy all of those properties (more precisely: for the allocation rules which are defined for situations we can define similar properties as the ones we discussed for game theoretic rules in Appendix A). We thus understand that if we want a rule that also satisfies many game-theoretic properties and not only belonging to the core we would need to use the Shapley value. Yet at the specific setting we are examining, one might argue that the zero-player property is of little value since there is no possibility that an item does not contribute to the joint costs of a coalition (i.e. zero players do not exist in this setting). This comment makes the use of the LOUDERBACK and INCREMENTAL rule perhaps more attractive.

- **Non-emptiness of the core Policy 1 vs. Policy 2:** By looking at the metric: probability of finding a core allocation, we have a lower bound on the games that have a non-empty core when Policy 2 is used. This can be seen in our first numerical study in Table 5.7. We see that the lower bound of balanced games when Policy 2 is used is 0.9726% (9,726 games), while the number of balanced games when Policy 1 is 9,004. When we compare the lower bound of Policy 2 with the balanced games with Policy 1 we find that Policy 2 results in at least $\frac{9{,}726-9{,}004}{9{,}004} = 0.08018$ (8.02% ) more balanced games than Policy 1 in the observed range of parameters.

- **Performance of rules Policy 1 vs. Policy 2:** We have provided some intuition in chapter 4 about the size of the core and we conjectured that a rule which performs well with policy 1 may also perform well with policy 2. By looking at Table 5.6, and Table 5.7, we indeed see that the rules can be ranked in the same order for both policies according to the core metric and the imputation set metric. The performance of the rules on the core metric was: LOUDERBACK > INCREMENTAL > Shapley value > INDIVIDUAL. For the imputation set metric, the performance was INDIVIDUAL > Shapley value > LOUDERBACK > INCREMENTAL.

## 5.3 Outcomes

The goal of this chapter was to study the issue of cost allocation for Policy 1 and Policy 2 and make recommendations on appropriate allocation rules. Moreover we wanted to study the core for Policy 2 for games with more than 4 items. We started by creating a big pool of allocation rules which have been used in a joint replenishment setting or that could be adapted to fit in the setting we are examining. Subsequently, we identified selection criteria for the allocation rules based on the literature and applied them, thus limiting our list of rules to 4 rules. We defined 3 performance metrics for our numerical study: the probability that an allocation rule identifies an imputation set element, the probability that an allocation rule identifies a core element, and the probability that we can identify a core element with any of the chosen rules. Our results show that with the use of INDIVIDUAL rule we can always identify an allocation which belongs to the imputation set as long as it is non-empty. The LOUDERBACK rule can identify a core element in most of the simulated games and is easy to compute even for large problem instances. The Shapley value can also identify a core element in many (yet not all) of the simulated games and possesses favorable game-theoretic properties.

We can indirectly give some insights on the non-emptiness of the core for the setting where Policy 2 is used. As the number of items in the grand coalition increases the probability of finding a core allocation decreases. While we do not know exactly how many of the simulated games have a non-empty core, we do know that for 3 and 4 player games we have found a core allocation when the core is non-empty for most of the simulated games (see Table G.2,Table G.3 and Table G.8,Table G.9). If we assume that the same holds for games with more than 4 players, we have some insights for the core of games with more than 4 players in the setting where Policy 2 is used. More specifically, we have a lower bound. Furthermore, if we compare the games with non-empty core when Policy 1 is used and our lower bound for games where Policy 2 is used we find that: Policy 2 results in at least 8.02% more balanced games than Policy 1 in the observed range of parameters (Table 5.7).

Lastly, from Dror, Hartman, and Chang (2012), we learn that the LOUDERBACK rule and the Shapley value are very good allocation rules for Policy 1. From our numerical experiments, we can see that those two rules continue to perform very well for a more flexible policy (Policy 2). It would be perhaps interesting to examine the performance of those rules for the POT policy of Anily and Haviv (2007) or a POT policy where the subcoalitions can further optimize over their base replenishment cycle.

# Chapter 6

# Discussion and Conclusions

In this chapter, we restate the research questions, relate our findings back to them, and we make specific and applicable recommendations regarding the Joint Replenishment Problem (JRP) that will allow managers to gain more understanding on the dynamics of the collaboration and will thus complement their knowledge of inventory management and cost optimization. Moreover, insights regarding the games that follow from the settings where Policy 1/ Policy 2 are gained. A general discussion on the policies that have been studied and compared so far is also made.

## 6.1 Discussion of Main Findings

The starting point of this thesis was the statement that the JRP had been studied in the game theoretic literature under a limited number of policies. We saw that Dror and Hartman (2007) have studied the JRP using a policy where every item is always replenished with the same frequency (Policy 1). On the other hand, we saw that Anily and Haviv (2007) and Zhang (2009) examined the game when the optimal POT policy is used. It is important to restate that in the optimal POT policy considered in the previous papers, the players in a sub-coalition cannot optimize over the base replenishment cycle (each coalition has the same base period and can only optimize over the multipliers of the base period). We argued that replenishing all products under the same frequency is too simple and stated that the full POT is already a complicated policy. We can further argue that the assumption that the coalitions cannot optimize over the base replenishment cycle is a strong assumption since it means that all of the retailers are willing to agree on an important decision variable and forego possible optimization benefits in order to form the grand coalition. Additionally, in the POT game by Anily and Haviv (2007) only a limited number of allocation rules has been examined and the proposed rule results in an allocation of costs to the retailers which cannot be easily explained and is counter-intuitive ( retailers who order frequently have no reason for staying with the grand coalition, since departing will not affect their costs). We restate the research questions here:

- **Main Research Question:** Which game theoretic properties are present in the various types of joint replenishment games with major and minor ordering costs under policies different from the optimal POT policy or the single group policy?
- **Research Sub-question 1:** Does a joint replenishment game with major and minor ordering costs that is based on policies different from the optimal POT policy or the single group policy always have a non-empty core?

- **Research Sub-question 2:** What are proper cost allocation rules for this class of games and what are their properties?
- **Research Sub-question 3:** How sensitive are the results obtained in the previous sub-questions to problem parameters?

In Chapter 3, we answered the first research sub-question. We studied Policy 2 which is a joint replenishment policy more complex than Policy 1 yet less complex than the optimal POT policy. We assumed that retailers cooperate under Policy 2 as it is more intuitive and easier to understand than the POT policy, and more flexible than Policy 1. Policy 2 is a realistic policy as it allows for the optimization of both the base cycle time as well as the option to order in every other cycle. After studying the setting with Policy 2 we found that the corresponding cost game is not subadditive in general, which means that the core and the imputation set can be empty. The game is not balanced in general, yet there are cases where it can be balanced and there can be cost savings. We further saw that by allowing more flexibility and more optimization possibilities in a coalition, there can be adverse effects: a game may have an empty core when Policy 2 is used and a non-empty core when Policy 1 is used. This was noticed for an example where the major cost had a very low value, approaching zero. The reverse may also hold due to the lack of flexibility: Policy 2 may result in a game with a non-empty core, while the use of Policy 1 for the same instance would result in a game with an empty core. This means that adding more optimization opportunities and stability may be contradictory in the JRP, depending on the problem instance.

In Chapter 4, we went more in depth to research sub-question 1 and started answering research sub-question 2. We examined numerically the non-emptiness of the core in a setting where Policy 2 is used and compared with Policy 1. The main findings in chapter 4 were that Policy 2 allows the forming of the grand coalition (probability of non-empty core) for more retailers with different cost parameters. For major cost equal to 15, 98.23% of the simulated games, where policy 2 was used were balanced, while the respective percentage for policy 1 was 93.71%. For major cost equal to 2 the percentages for 4-item games were 81.85 % and 45.49 % respectively. We studied the effect of major cost on the probability that a random game has a non-empty core: when the major cost is high Policy 2 and Policy 1 resulted in a relatively high probability of non-empty core. When the major cost is low, Policy 2 results in a relatively high probability but Policy 1 in a low probability. We have also used metric e to gain some intuition on the size of the core for both policies but also to gain a lower bound on the cost savings that are achieved when forming the grand coalition. We argued that an allocation rule which performed well with policy 1, might also perform well with policy 2. We repeated the experiments for a wider range of parameters for the holding costs and demand rates and obtained similar results.

Chapter 5 was focused mainly on research sub-questions 2 and 3, and also provided further insight on research sub-question 1. We studied the performance of allocation rules for the allocation of the costs / benefits of the collaboration. We aimed to find a rule which is readily explainable, easy to calculate, justifiable and which belongs to the core (with a high probability). A rule which is readily explainable will match the intuition of managers and thus has a higher chance of being used. A rule which is easy to calculate can be used for large problem instances to allocate costs. A justifiable rule will result in the same allocation whether it is viewed from a cost or benefit perspective and thus eliminate possible reasons for non-cooperation. A rule which results in a core allocation will ensure that non sub-coalitions will be formed and so all the parties have a benefit of staying in the grand coalition. We examined 4 allocation rules in order to allocate costs and used three metrics to evaluate their performance: the probability that an allocation rule results in an imputation set element, the probability that an allocation rule results in a core element and we further gave a lower bound on the number of balanced games by computing the probability of finding a core element with any of the four rules. The rules which performed well with Policy 1 performed comparatively well with Policy 2 as well. The performance of the rules according to the two metrics was the same for both policies. For the core metric: LOUDERBACK > INCREMENTAL > Shapley value > INDIVIDUAL. For the imputation set metric, the performance was INDIVIDUAL > Shapley value > LOUDERBACK > INCREMENTAL. The best performing rule according to the imputation set metric is the INDIVIDUAL rule, which always results in an imputation set element as long as the imputation set is non-empty. The best performing rule for the core metric is the LOUDERBACK rule (for major cost equal to 20, the rule identified a core element in 97.26% of the simulated games with Policy 2, 88.89% of games with Policy 1). We further examined how the performance of the rules is affected as the number of items in the grand coalition is increased (for games which follow from Policy 2). The rules consistently followed the ranking described previously in both metrics (a small, insignificant change in ranking for the core metric between the Shapley Value and the INCREMENTAL rule is observed for 3- and 4-item games when the major cost is equal to 15, and in 3-item games when A=8). It is important to note that the LOUDERBACK rule can be computed fast for even large problem instances, but since it does not possess as many desirable properties as the Shapley value, there is a trade-off to be made between desirable properties, computational speed, and core memebership. For this reason and after considering the performance of the Shapley value in the core metric (for major cost equal to 20, the rule identified a core element in 88.35% of the simulated games with Policy 2, 62.41% of games with Policy 1), we can recommend the use of the Shapley value as well for the JRP.

## 6.2 Limitations and Future Research

The research we conducted in this project has some limitations which may provide inspiration for future research. Moreover, the findings we described in the previous chapters also point out to interesting research. One limitation in the model was the choice of integer multipliers to be equal to 1 or 2 only. While this choice provides for a simple, interesting, and intuitive policy one may wonder what happens when we restrict ourselves to two multipliers but not 1 and 2 rather 1 and the optimal second multiplier. The problem might be then to find a smart way to identify the optimal second multiplier. Another option for future research might be to use one of the heuristic algorithms described in the introduction to identify a strict cyclic policy for each coalition and examine the corresponding game. We would expect that for low values of the major cost, the core may be empty for the same reasons as in our model. When the major cost is high we would expect that the added flexibility would lead to even better results (in terms of probability of nonempty core) than Policy 2. It is only reasonable that the larger the search space the more probable it is that retailers with different cost parameters can jointly replenish their inventory.

From Dror, Hartman, and Chang (2012), we learn that the LOUDERBACK rule and the Shapley value are very good allocation rules for Policy 1. From our numerical experiments in Chapter 5, we saw that those two rules continue to perform very well for a more flexible policy (Policy 2). It would be interesting to examine the performance of those rules for either the POT policy of Anily and Haviv (2007) or the POT policy where retailers can also optimize over the base cycle time and evaluate the performance of the rules in such a setting. Lastly, a possible future extension might be to incorporate more operational constraints to the joint replenishment model and examine the game that can be derived from the model. This will make for a more realistic case that possibly arises in practice. Some operational constraints that may be considered are: capacity constraints or possible penalties when replenishing certain products at the same time.

# References


1. Anily, S., & Haviv, M. (2007). The cost allocation problem for the first order interaction joint replenishment model. Operations Research, 55(2), 292-302.

2. Atkins, D. R., & Iyogun, P. O. (1988). Periodic versus "can-order" policies for coordinated multi-item inventory systems. Management Science, 34(6), 791-796.

3. Bastian, M. (1986). Joint replenishment in multi-item inventory systems. Journal of the Operational Research Society, 37(12), 1113-1120.

4. Bastos, L. D. S. L., Mendes, M. L., Nunes, D. R. D. L., Melo, A. C. S., & Carneiro, M. P. (2017). A systematic literature review on the joint replenishment problem solutions: 2006-2015. Production, 27.

5. Bondareva, O. (1963). Certain applications of the methods of linear programming to the theory of cooperative games (In Russian). Problemy Kibernetiki, 10:119–139.

6. Boon, van der Boor, van Leeuwaarden, Mathijsen, van der Pol, and Resing (2019). Advanced Simulation - Lecture Notes, Eindhoven University of Technology

7. Branzei, R., Dimitrov, D., & Tijs, S. (2008). Models in cooperative game theory (Vol. 556). Springer Science & Business Media.

8. Brown, L. D., Cai, T. T., & DasGupta, A. (2001). Interval estimation for a binomial proportion. Statistical science, 101-117.

9. Dror, M., & Hartman, B. C. (2007). Shipment consolidation: Who pays for it and how much?. Management Science, 53(1), 78-87.

10. Dror, M., & Hartman, B. C. (2011). Survey of cooperative inventory games and extensions. Journal of the Operational Research Society, 62(4), 565-580.

11. Dror, M., Hartman, B. C., & Chang, W. (2012). The cost allocation issue in joint replenishment. International Journal of Production Economics, 135(1), 242-254.

12. Fiestras-Janeiro, M. G., García-Jurado, I., Meca, A., & Mosquera, M. A. (2011). Cooperative game theory and inventory management. European Journal of Operational Research, 210(3), 459-466.

13. Fiestras-Janeiro, M. G., García-Jurado, I., Meca, A., & Mosquera, M. A. (2012). Cost allocation in inventory transportation systems. Top, 20(2), 397-410.

14. Fiestras-Janeiro, M. G., García-Jurado, I., Meca, A., & Mosquera, M. A. (2014). Centralized inventory in a farming community. Journal of Business Economics, 84(7), 983-997.

15. Frisk, M., Göthe-Lundgren, M., Jörnsten, K., & Rönnqvist, M. (2010). Cost allocation in collaborative forest transportation. European Journal of Operational Research, 205(2), 448-458.

16. Goyal, S. K. (1974). Determination of optimum packaging frequency of items jointly replenished. Management Science, 21(4), 436-443.

17. Göthe-Lundgren, M., Jörnsten, K., & Värbrand, P. (1996). On the nucleolus of the basic vehicle routing game. Mathematical programming, 72(1), 83-100.

18. Guajardo, M., & Rönnqvist, M. (2016). A review on cost allocation methods in collaborative transportation. International transactions in operational research, 23(3), 371-392.

19. Hartman, B. C., & Dror, M. (1996). Cost allocation in continuous-review inventory models. Naval Research Logistics (NRL), 43(4), 549-561.

20. Hezarkhani, B., Slikker, M., & Van Woensel, T. (2018). Collaborative replenishment in the presence of intermediaries. European Journal of Operational Research, 266(1), 135-146.

21. How to Safely Load and Unload Truck Cargo to Prevent Injury - Western Truck Insurance Services. Retrieved September 11, 2021, from https://www.truckinsure.com/blog/how-to-safely-load-and-unload-truck-cargo-to-prevent-injury/

22. Jiang, L., & Hickernell, F. J. (2014). Guaranteed monte carlo methods for bernoulli random variables. arXiv preprint arXiv:1411.1151.

23. Kannai, Y. (1992). The core and balancedness. Handbook of game theory with economic applications, 1, 355-395.

24. Kaspi, M., & Rosenblatt, M. J. (1991). On the economic ordering quantity for jointly replenished items. The International Journal of Production Research, 29(1), 107-114.

25. Khouja, M., & Goyal, S. (2008). A review of the joint replenishment problem literature: 1989–2005. European Journal of Operational Research, 186(1), 1-16.

26. Khouja, M., Michalewicz, Z., & Satoskar, S. S. (2000). A comparison between genetic algorithms and the RAND method for solving the joint replenishment problem. Production Planning & Control, 11(6), 556-564.

27. Krichen, S., Laabidi, A., & Abdelaziz, F. B. (2011). Single supplier multiple cooperative retailers inventory model with quantity discount and permissible delay in payments. Computers & Industrial Engineering, 60(1), 164-172.

28. Lee, F. C., & Yao, M. J. (2003). A global optimum search algorithm for the joint replenishment problem under power-of-two policy. Computers & Operations Research, 30(9), 1319-1333.

29. Li, J., Feng, H., & Zeng, Y. (2014). Inventory games with permissible delay in payments. European Journal of Operational Research, 234(3), 694-700.

30. Liu, P., Wu, Y., & Xu, N. (2010). Allocating collaborative profit in less-than-truckload carrier alliance. Journal of Service science and Management, 3(01), 143.

31. Lohmann, E. R. M. A. (2012). Joint decision making and cooperative solutions.

32. Meca, A., Timmer, J., Garcıa-Jurado, I. ,& Borm, P. (2004). Inventory games. European Journal of Operational Research, 156(1), 127-139.

33. Olsen, A. L. (2005). An evolutionary algorithm to solve the joint replenishment problem using direct grouping. Computers & Industrial Engineering, 48(2), 223-235.

34. Olsen, A. L. (2008). Inventory replenishment with interdependent ordering costs: an evolutionary algorithm solution. International Journal of Production Economics, 113(1), 359-369.

35. Osborne, M. J., & Rubinstein, A. (1994). A course in game theory. MIT press.

36. Otero-Palencia, C., Amaya–Mier, R., & Yie-Pinedo, R. (2019). A stochastic joint replenishment problem considering transportation and warehouse constraints with gainsharing by Shapley Value allocation. International Journal of Production Research, 57(10), 3036-3059.

37. Pantumsinchai, P. (1992). A comparison of three joint ordering inventory policies. Decision Sciences, 23(1), 111-127.

38. Pratikakis, C. (2021). Literature Review, Eindhoven University of Technology

39. Renberg, B., & Planche, R. (1967). Un modele pour la gestion simultanee des n articles d'un stock. Revue Francaise d'Informatique et de Recherche Operationelle, 6, 47-59.

40. Saavedra-Nieves, A., García-Jurado, I., & Fiestras-Janeiro, M. G. (2018). On coalition formation in a non-convex multi-agent inventory problem. Annals of Operations Research, 261(1-2), 255-273.

41. Schmeidler, D. (1969). The nucleolus of a characteristic function game. SIAM Journal on applied mathematics, 17(6), 1163-1170.

42. Shapley, L. S. (1953). A value for n-person games. Contributions to the Theory of Games, 2(28), 307-317.

43. Shapley, L. S., & Shubik, M. (1966). Quasi-cores in a monetary economy with nonconvex preferences. Econometrica: Journal of the Econometric Society, 805-827.

44. Shapley, L. (1967). On balanced sets and cores. Naval Research Logistics Quarterly, 14:453–460.

45. Shi, X., Wang, H., & Xue, W. (2019). Allocation rules for joint replenishment and inventory sharing via transshipment. Procedia Computer Science, 159, 1872-1879.

46. Slikker, M. (2020). Game Theory with Applications to Supply Chain Management-Lecture Notes, Eindhoven University of Technology

47. susobhang70/shapley_value: Python code to find Shapley Value of a Characteristic Form Game - GitHub. Github. Retrieved June 23, 2021, from https://github.com/susobhang70/shapley_value

48. Timmer, J., Chessa, M., & Boucherie, R. J. (2013). Cooperation and game-theoretic cost allocation in stochastic inventory models with continuous review. European journal of operational research, 231(3), 567-576.

49. Tinoco, S. V. P., Creemers, S., & Boute, R. N. (2017). Collaborative shipping under different cost-sharing agreements. European Journal of Operational Research, 263(3), 827-837.

50. Van Eijs, M. J. G., Heuts, R. M. J., & Kleijnen, J. P. C. (1992). Analysis and comparison of two strategies for multi-item inventory systems with joint replenishment costs. European Journal of Operational Research, 59(3), 405-412.

51. Zhang, J. (2009). Cost allocation for joint replenishment models. Operations research, 57(1), 146-156.

# Appendix A

# Cooperative Game Theory - Additional Definitions

In this appendix, we explain in more detail game theoretic concepts that are used throughout the thesis. The definitions and explanations for the Shapley value, the nucleolus and the properties are based on (Slikker, 2020), who wrote those definitions for a value game. The balancedness conditions are written based on the minimal balanced collections as cited in (Lohmann, 2012).

**The Shapley value:**

Consider the cooperative cost game $(N, c)$. Formally, $\sigma : N \to \{1, ..., n\}$ is a bijection, where $\sigma(i) = k$ states that player $i$ is in position $k$. Alternatively, $\sigma^{-1}(k)$ denotes the index of the player that is in position $k$. Hence, the set of players that precede player $i \in N$ (according to $\sigma$) can be described by

$$P(\sigma, i) = \{j \in N | \sigma(j) < \sigma(i)\}.$$

The (marginal) contribution of player $i$ can then be described as the cost of player $i$ together with his predecessors minus the cost of his predecessors (excluding player $i$),

$$m_i^\sigma(c) = c(P(\sigma, i) \cup \{i\}) - c(P(\sigma, i)).$$

The vector $m^\sigma = (m_1^\sigma, ..., m_n^\sigma)$ is called the marginal vector associated with order $\sigma$. Let $\Pi(N)$ be the set of all orders of player set $N$. The Shapley value $\Phi$ of game $(N, c)$ is defined as the average of all marginal vectors

$$\Phi(N, c) = \frac{1}{n!} \sum_{\sigma \in \Pi(N)} m^\sigma(c).$$

The Shapley value satisfies many interesting properties. The properties that will be defined now are not just defined for the Shapley value but for any arbitrary allocation rule that assigns for each coalitional game to each of the players a payoff.

**I. Efficiency:** An allocation rule $\gamma$ for coalitional games is efficient if for each coalitional game $(N, c)$ it holds: $\sum_{i \in N} \gamma_i(N, c) = c(N)$.

**II. Additivity:** An allocation rule $\gamma$ for coalitional games is additive if for two coalitional games $(N, c)$ and $(N, w)$ with the same player set it holds: $\gamma(N, c + w) = \gamma(N, c) + \gamma(N, w)$.

**III. Symmetry:** An allocation rule $\gamma$ for coalitional games is symmetric if for each coalitional game $(N, c)$ it holds that: $\gamma_i(N, c) = \gamma_j(N, c)$ for any two players $i, j \in N$ that are symmetric in $(N, c)$.

Two players $i, j$ are symmetric in $(N, c)$ if $c(S \cup i) = c(S \cup j)$ for all coalitions $S \subseteq N \setminus \{i, j\}$.

**IV. Zero-Player Property:** An allocation rule $\gamma$ for coalitional games satisfies the zero-player property if for each coalitional game $(N, c)$ it holds that: $\gamma_i(N, c) = 0$ for any zero player $i \in N$.

A player $i \in N$ is a zero player in coalitional game $(N, c)$ if $c(S \cup i) = c(S)$ for all $S \subseteq N$.

There is exactly one allocation rule for coalitional games that satisfies efficiency,additivity, zero-player property, and symmetry: the Shapley value.

**The nucleolus:**

The nucleolus was introduced in Schmeidler (1969) and is a single-valued allocation rule that always picks an allocation in the core whenever it is nonempty.The nucleolus is defined for games with a nonempty imputation set only. In order to define the nucleolus we first need the concepts of 'ordering function' and 'lexicographic order'. If $K$ is a finite set then the ordering function on $\mathbb{R}^K$ is the function $n^K : \mathbb{R}^K \to \mathbb{R}^{|K|}$, defined by the following subsequent steps: $n_1^K(x) = min\{x_j | j \in K\}$. Choosing $j_1 \in K$ such that $n_1^K(x) = x_{j1}$ we have $n_2^K(x) = min\{x_j | j \in K \setminus \{j_1\}\}$, etc. For $x, y \in \mathbb{R}^K$ we say that $x$ is lexicographically larger than $y$ so $x \geq_L y$ if there exists an $s$ such that $x_i = y_i$ for all $i < s$ and $x_s > y_s$.

The nucleolus is defined as follows: Let $(N, c)$ be a cooperative game, for a payoff vector $x \in \mathbb{R}^N$ define satisfaction of coalition $S \subseteq N$ as $s(S, x) = c(S) - \sum_{i \in S} x_i$ (this definition for the satisfaction is based on Göthe-Lundgren, Jörnsten, and Värbrand (1996), who defined the nucleolus for a cost game and in that paper satisfaction was termed as excess) . We can see that an allocation $x$ leads to higher satisfaction if the sum of allocated costs to players in $S$ is low. Let $\theta(x) \in \mathbb{R}^{2^n}$ have the satisfactions of payoff vector $x$ ordered increasingly. Then the nucleolus $v(N, c)$ is defined in case the imputation set is not empty, as the vector in the imputation set whose $\theta$ is lexicographically maximal.

**Core Non-emptiness and Minimal Balanced Collections for Games with 3 and 4 Players:**

In the introduction we have discussed the concept of balancedness and have given the necessary definitions and preliminaries. As the number of players in a game grows larger identifying the minimal balanced collections and the resulting balancedness conditions becomes a harder task. As stated in (Kannai, 1992) the determination of all the minimal balanced collections is not an easy task especially for a large number of players. For games with 3 and 4 players, we can identify whether the core is non-empty by examining the balancedness conditions that follow from the minimal balanced collections. The minimal balanced collections can be found in (Lohmann, 2012) for a value game and we present the balancedness conditions that are defined from those collections for a cost game here:

3-Player Game:

$$c(\{1,2,3\}) \leq c(\{1\}) + c(\{2\}) + c(\{3\})$$

$$c(\{1,2,3\}) \leq 0.5c(\{1,2\}) + 0.5c(\{1,3\}) + 0.5c(\{2,3\})$$

$$c(\{1,2,3\}) \leq c(\{1\}) + c(\{2,3\})$$

$$c(\{1,2,3\}) \leq c(\{2\}) + c(\{1,3\})$$

$$c(\{1,2,3\}) \leq c(\{3\}) + c(\{1,2\})$$

4-Player Game:

$$\begin{gathered}
c(\{1,2,3,4\}) \leq c(\{1,2\}) + c(\{3,4\}) \\
c(\{1,2,3,4\}) \leq c(\{1,4\}) + c(\{2,3\}) \\
c(\{1,2,3,4\}) \leq c(\{1,3\}) + c(\{2,4\})
\end{gathered}$$

$$\begin{gathered}
c(\{1,2,3,4\}) \leq c(\{1,2,3\}) + c(\{4\}) \\
c(\{1,2,3,4\}) \leq c(\{1,3,4\}) + c(\{2\}) \\
c(\{1,2,3,4\}) \leq c(\{1,2,4\}) + c(\{3\}) \\
c(\{1,2,3,4\}) \leq c(\{2,3,4\}) + c(\{1\})
\end{gathered}$$

$$\begin{gathered}
c(\{1,2,3,4\}) \leq c(\{1,2\}) + c(\{3\}) + c(\{4\}) \\
c(\{1,2,3,4\}) \leq c(\{1,4\}) + c(\{2\}) + c(\{3\}) \\
c(\{1,2,3,4\}) \leq c(\{2,4\}) + c(\{1\}) + c(\{3\}) \\
c(\{1,2,3,4\}) \leq c(\{1,3\}) + c(\{2\}) + c(\{4\}) \\
c(\{1,2,3,4\}) \leq c(\{2,3\}) + c(\{1\}) + c(\{4\}) \\
c(\{1,2,3,4\}) \leq c(\{3,4\}) + c(\{1\}) + c(\{2\})
\end{gathered}$$

$$\begin{gathered}
c(\{1,2,3,4\}) \leq 0.5c(\{1,2,3\}) + 0.5c(\{1,2,4\}) + 0.5c(\{3,4\}) \\
c(\{1,2,3,4\}) \leq 0.5c(\{1,3,4\}) + 0.5c(\{2,3,4\}) + 0.5c(\{1,2\}) \\
c(\{1,2,3,4\}) \leq 0.5c(\{2,4,1\}) + 0.5c(\{2,4,3\}) + 0.5c(\{1,3\}) \\
c(\{1,2,3,4\}) \leq 0.5c(\{2,3,1\}) + 0.5c(\{2,3,4\}) + 0.5c(\{1,4\}) \\
c(\{1,2,3,4\}) \leq 0.5c(\{1,4,2\}) + 0.5c(\{1,4,3\}) + 0.5c(\{2,3\}) \\
c(\{1,2,3,4\}) \leq 0.5c(\{1,3,2\}) + 0.5c(\{1,3,4\}) + 0.5c(\{2,4\})
\end{gathered}$$

$$c(\{1,2,3,4\}) \leq c(\{1\}) + c(\{2\}) + c(\{3\}) + c(\{4\})$$

$$c(\{1,2,3,4\}) \leq 1/3c(\{1,2,3\}) + 1/3c(\{1,2,4\}) + 1/3c(\{1,3,4\}) + 1/3c(\{2,3,4\})$$

$$c(\{1,2,3,4\}) \leq 0.5c(\{1,2\}) + 0.5c(\{1,3\}) + 0.5c(\{2,3\}) + c(\{4\})$$
$$c(\{1,2,3,4\}) \leq 0.5c(\{1,2\}) + 0.5c(\{1,4\}) + 0.5c(\{2,4\}) + c(\{3\})$$
$$c(\{1,2,3,4\}) \leq 0.5c(\{1,4\}) + 0.5c(\{1,3\}) + 0.5c(\{4,3\}) + c(\{2\})$$
$$c(\{1,2,3,4\}) \leq 0.5c(\{4,2\}) + 0.5c(\{4,3\}) + 0.5c(\{2,3\}) + c(\{1\})$$

$$c(\{1,2,3,4\}) \leq 0.5c(\{1,2,3\}) + 0.5c(\{1,4\}) + 0.5c(\{2,4\}) + 0.5c(\{3\})$$
$$c(\{1,2,3,4\}) \leq 0.5c(\{1,4,3\}) + 0.5c(\{1,2\}) + 0.5c(\{2,4\}) + 0.5c(\{3\})$$
$$c(\{1,2,3,4\}) \leq 0.5c(\{2,4,3\}) + 0.5c(\{1,2\}) + 0.5c(\{1,4\}) + 0.5c(\{3\})$$
$$c(\{1,2,3,4\}) \leq 0.5c(\{1,2,4\}) + 0.5c(\{3,1\}) + 0.5c(\{3,2\}) + 0.5c(\{4\})$$
$$c(\{1,2,3,4\}) \leq 0.5c(\{1,3,4\}) + 0.5c(\{2,1\}) + 0.5c(\{2,3\}) + 0.5c(\{4\})$$
$$c(\{1,2,3,4\}) \leq 0.5c(\{2,3,4\}) + 0.5c(\{1,2\}) + 0.5c(\{1,3\}) + 0.5c(\{4\})$$
$$c(\{1,2,3,4\}) \leq 0.5c(\{1,2,3\}) + 0.5c(\{2,4\}) + 0.5c(\{3,4\}) + 0.5c(\{1\})$$
$$c(\{1,2,3,4\}) \leq 0.5c(\{1,2,4\}) + 0.5c(\{2,3\}) + 0.5c(\{3,4\}) + 0.5c(\{1\})$$
$$c(\{1,2,3,4\}) \leq 0.5c(\{1,3,4\}) + 0.5c(\{2,3\}) + 0.5c(\{2,4\}) + 0.5c(\{1\})$$
$$c(\{1,2,3,4\}) \leq 0.5c(\{2,1,3\}) + 0.5c(\{1,4\}) + 0.5c(\{3,4\}) + 0.5c(\{2\})$$
$$c(\{1,2,3,4\}) \leq 0.5c(\{2,1,4\}) + 0.5c(\{3,1\}) + 0.5c(\{3,4\}) + 0.5c(\{2\})$$
$$c(\{1,2,3,4\}) \leq 0.5c(\{2,3,4\}) + 0.5c(\{1,3\}) + 0.5c(\{1,4\}) + 0.5c(\{2\})$$

$$c(\{1,2,3,4\}) \leq 2/3c(\{1,2,3\}) + 1/3c(\{1,4\}) + 1/3c(\{2,4\}) + 1/3c(\{3,4\})$$
$$c(\{1,2,3,4\}) \leq 2/3c(\{1,2,4\}) + 1/3c(\{1,3\}) + 1/3c(\{2,3\}) + 1/3c(\{3,4\})$$
$$c(\{1,2,3,4\}) \leq 2/3c(\{1,3,4\}) + 1/3c(\{1,2\}) + 1/3c(\{2,3\}) + 1/3c(\{2,4\})$$
$$c(\{1,2,3,4\}) \leq 2/3c(\{2,3,4\}) + 1/3c(\{1,2\}) + 1/3c(\{1,3\}) + 1/3c(\{1,4\})$$

# Appendix B

# Theory on Confidence Intervals

The approximate confidence intervals for our estimated metrics are based on the theory of the central limit theorem. These approximate confidence intervals can be used to assess the simulation accuracy, or to determine the minimum number of runs required for our simulation. Before moving on to describe how to construct the approximate confidence intervals we present some theorems and definitions. The main source used to write down the theorems and definitions is (Boon, van der Boor, van Leeuwaarden, Mathijsen, van der Pol, and Resing, 2019).

**Theorem 3 (The law of large numbers):** The law of large numbers states that if $Z_1, Z_2, ...Z_n$ are independent and identically distributed random variables with mean $\zeta := \mathbb{E}[Z1]$ and finite variance, the probability that the sample mean is close to $\zeta$ is large. In fact, for every $\epsilon > 0$, it holds that:

$$\lim_{n\to\infty} \mathbb{P}\left(\left|\frac{Z_1 + Z_2 + \cdots + Z_n}{n} - \zeta\right| < \varepsilon\right) = 1$$

Now suppose that we have a method to obtain independent identically distributed outcomes $Z_1, ..., Z_n$ while we do not know the value of $\zeta$. The discussion above suggests that we can use the sample mean:

$$\bar{Z} \equiv \bar{Z}_n := \frac{Z_1 + Z_2 + \cdots + Z_n}{n}$$

Consider a sequence of independent, identically, distributed random variables $Z_1, Z_2, ..., Z_n$, with common mean $\zeta$ and variance $\sigma^2$. We have already introduced the sample mean $\bar{Z}$. We will now define the sample variance $S^2$:

**Definition 4 (Sample variance):** The quantity $S^2$, defined by

$$S^2 := \frac{1}{n-1}\sum_{i=1}^{n}\left(Z_i - \bar{Z}\right)^2$$

is called the sample variance of the random sample $Z_1, Z_2, ..., Z_n$.

**Theorem 4 (Central Limit Theorem):** Let the sequence of random variables $Y_n$ be defined by

$$Y_n := \sqrt{n}\frac{\bar{Z} - \zeta}{\sqrt{\sigma^2}}$$

Then $Y_n$ converges in distribution to a standard normal random variable, or

$$\lim_{n\to\infty} \mathbb{P}\left(Y_n \leq y\right) = \Phi(y)$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function, given by

$$\Phi(y) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y} e^{-\frac{1}{2}t^2} dt$$

Theorem 4 is often interpreted as

$$\mathbb{P}\left(Y_n \leq y\right) \approx \Phi(y)$$

and we will use theorem 4 to construct approximate confidence intervals.

Note that $Y_n$ depends on an unknown parameter, $\sigma^2$. But, Boon, van der Boor, van Leeuwaarden, Mathijsen, van der Pol, and Resing (2019) reference a theorem known as Slutsky's theorem, according to which the central limit theorem still holds if we replace the variance by the sample variance.

We can see that the Central Limit Theorem relates the sample mean of a sufficiently large number of random variables to the normal distribution.

We define $z_\alpha$ to be the $1-\alpha$ quantile of the standard normal distribution. This means that $\Phi(z_\alpha) = 1 - \alpha$ and it further follows that: $\mathbb{P}(z_{1-\alpha} < Y_n < z_\alpha) \approx 1 - 2\alpha$.

An (approximate) confidence interval is an interval in which we can assert that the quantity we wish to estimate $(z)$ lies with a certain degree of confidence. A $100(1-2\alpha)\%$ confidence interval is a random interval (usually depending on the outcomes $Z_i$ ), such that the probability that the interval covers the true value of $\zeta$ equals $1 - 2\alpha$. We will focus on approximate confidence intervals, inspired by the central limit theorem.

Boon, van der Boor, van Leeuwaarden, Mathijsen, van der Pol, and Resing (2019) use the previous theorems to show that the approximate $100(1-2\alpha)\%$ confidence interval for the random variable $\zeta$ which we are trying to estimate can be obtained by the following formula:

$$\left(\bar{Z} - z_\alpha\sqrt{\frac{S^2}{n}}, \bar{Z} - z_{1-\alpha}\sqrt{\frac{S^2}{n}}\right)$$

Since we are often interested in 95% confidence intervals as is the case in this report, we have that $\alpha = 0.025$ and so $z_\alpha = -z_{1-\alpha} \approx 1.96$ and so the formula becomes:

$$\left(\bar{Z} - 1.96\sqrt{\frac{S^2}{n}}, \bar{Z} + 1.96\sqrt{\frac{S^2}{n}}\right)$$

We know that for the standard normal distribution, since it is symmetric around zero then it holds that: $z_{1-\alpha} = -z_\alpha$.

We note at this point that for the approximate confidence intervals which we are trying to construct here, it is important to have an idea for the number of experiments or runs for our simulation. This will allow us to reach the desired level of accuracy. In order to define the number of runs we also need to have an idea of the actual variance $\sigma^2$.

A special case which is of relevance for our study is when we are estimating probabilities. In that case we do not need to have an initial guess or to know the variance and so we can find the required number of runs for our simulation differently. When we are simulating probabilities, as is the case here, the outcome per simulation instance is either 1 or 0. This means that the estimator for the unknown probability $p$ is given by the following formula: $\widehat{p} := \frac{1}{n}\sum_{i=1}^{n} Z_i$, where $Z_i$ has a Bernoulli distribution with this (unknown) parameter $p$. The mean of the Bernoulli distribution is equal to the parameter $p$ and the variance is equal to $p(1-p)$. This means that a 95% confidence interval for the parameter $p$ is given by the following formula:

$$\widehat{p} \pm \frac{z_{\alpha/2}\sqrt{\hat{p}(1-\widehat{p})}}{\sqrt{n}}$$

where $z_{\alpha/2} \approx 1.96$ for a 95% confidence interval and $n$ is the sample size.

The interval presented above can also be seen in other articles in the literature (e.g. in Brown, Cai, and DasGupta, 2001) although with some criticism about its assumptions and its use with estimating Bernoulli random variables. Nevertheless it is a basic, textbook confidence interval frequently used in the literature (also when trying to estimate probabilities, frequencies, or proportions) and so we are going to use it for the present thesis. We note that in the literature it is argued to not use this confidence interval if the sample size ($n$) is small or if the true value for the probability is close to 0 or 1 (Jiang and Hickernell, 2014). Further directions for the correct use of the interval when trying to estimate probabilities are given in ( Brown, Cai, and DasGupta (2001): it is mainly stated that a large number of runs is required, that the confidence interval should not be used if the value 1 or 0 is contained in the interval $\hat{p} \pm 3\sqrt{\hat{p}(1-\hat{p})/n}$, and some other rules of thumb are also given.

Regarding the minimum number of runs needed in order to have a feeling about the accuracy and the width of the interval: (Boon, van der Boor, van Leeuwaarden, Mathijsen, van der Pol, and Resing, 2019) after sufficient analysis give the following formula for obtaining the number of runs for the simulation: $n = \frac{1}{4}(z_{\alpha/2}/\epsilon)^2$, where $\epsilon$ is the desired width or accuracy of the half-width (half-width is half of the spread of the confidence interval around the mean of the random variable which we are trying to estimate).

The formula is based on the fact that: $0 \leq \hat{p} \leq 1$ and that the variance $\sigma^2 = \hat{p}(1 - \hat{p})$ is maximal when $\hat{p} = 0.5$ and so by using this estimate they derive the mentioned formula.

# Appendix C

# Project Plan

In this appendix, the project plan for the Master thesis is presented. By taking into consideration the maximum duration of 25 weeks after the approval of the research proposal (4 holiday weeks are included in the duration) for the thesis the following project plan is constructed.

A few general remarks about the project plan (Figure C.1) are the following:

- Weekly progress meetings will be held with the first supervisor of the project dr. Marco Slikker. If possible meetings will be held on campus. The aim of those meetings will be to discuss about the progress of the research and to set short-term deadlines for the deliverables/ drafts of the project. Feedback will also be requested by the student.

- The student will provide to the supervisor prior to each meeting an indicative agenda for each meeting containing the main points the student wishes to discuss at the meeting. This agenda may also be used by the student and/or the supervisor to take personal notes during the meetings so each of them has sufficient information about the content and discussions of the meetings. In any case the student is responsible for taking notes from the meetings and can provide his notes to the supervisor if required.

- The project plan will be monitored and reevaluated weekly by the student and any need for deviations/ alterations will be discussed with the supervisor. The nature of the project plan is to provide a general road-map rather than a step-by-step list of tasks and deadlines. As such the project is split into 4 main phases: the conceptual, the analysis, the dissertation, and the final submission phase. Each phases builds upon the previous phase and the research moves gradually from abstract to concrete.

- In terms of deliverables, the student will provide to the supervisor every 2 weeks (unless agreed otherwise during the process) a document containing parts of the work to be included in the thesis and will request feedback. Those documents will later be compiled and edited to form the thesis.

- There is a "buffer" time of 1 week planned between the conceptual, analysis, and dissertation phase of the project in order to allow for possible extensions of the respective phases of the thesis. There is also a "buffer" time of 4 weeks at the end of the project that can be used depending on the progress of the various phases of the thesis.

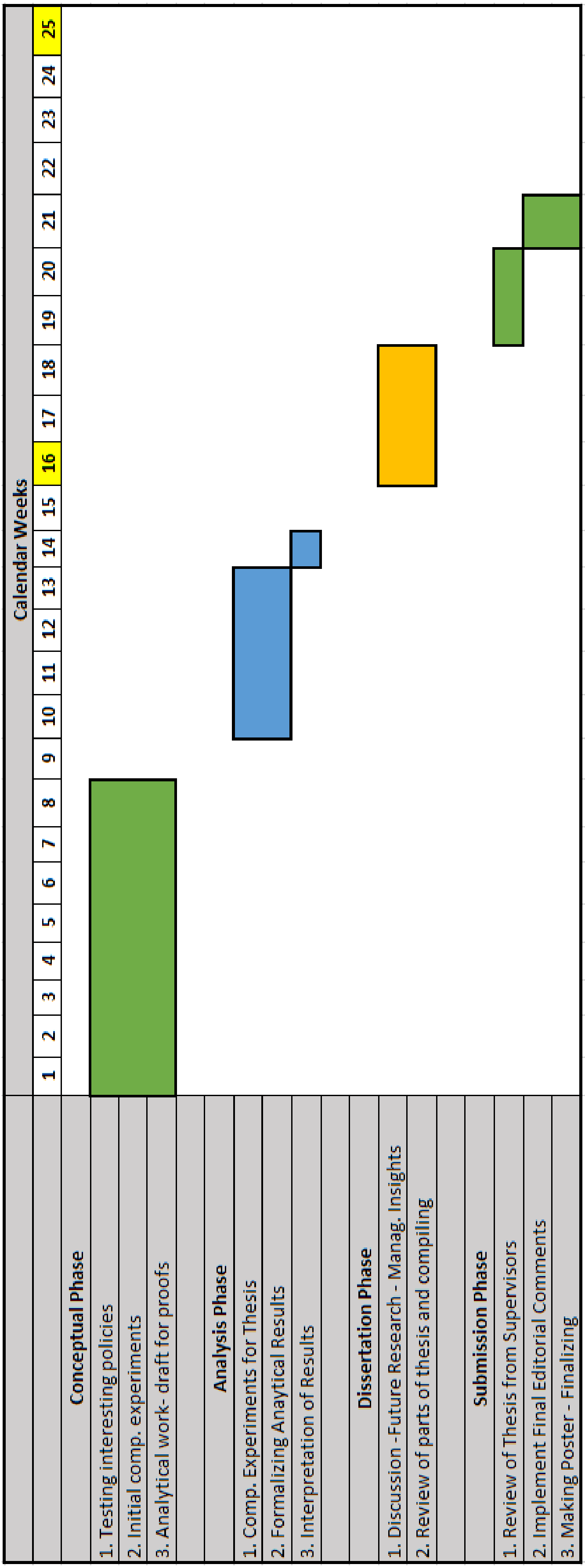



Figure C.1: Project Plan

# Appendix D

# Programs for Numerical Studies

In this appendix we describe the programs that were used to perform the numerical studies. The programs were written in Python by making use of common and well-known libraries such as NumPy, Math, Scipy, and Itertools. The library Scipy was used in generating the random samples and the library Itertools was especially useful in generating all the subsets for a given set of players. The process that we used to validate the computer programs is also described in this appendix.

## Description of the Programs

### Functions for Finding Cost for a Coalition

**EOQ:**

- **Input:** This function takes as input the major cost, the minor cost for one item, the demand rate and the holding cost for the item.
- **Output:** The function makes use of the eoq formula to compute the optimal eoq cost for the given item i

**simple_IG:**

- **Input:** This function takes as input one coalition S, the major cost, the minor costs, the demand rates and the holding costs for a given instance of the problem.
- **Output:** The function returns the minimum cost for the given coalition if we assume that the simple indirect grouping strategy where $k_i$ is either 1 or 2 is used (Policy 2). This policy is described in chapter 3 of this master thesis.
- **Description:** The steps followed are the following: 1)we create all possible combinations of $ki = 1$ and $ki = 2$ for all players in coalition S, 2) we compute the cost for each of the combinations of $k_i$'s and return the minimum cost. We exclude the combination where all of the $k_i$'s are equal to 2.

**one_frequency:**

- **Input:** This function takes as input one coalition S, the major cost, the minor costs, the demand rates and the holding costs for a given instance of the problem.

- **Output:** The function returns the minimum cost for the given coalition if Policy 1 is used.

- **Description:** The cost is computed according to the formulas by Dror and Hartman (2007), as they have been described in Chapter 3 of the thesis.

## Functions for Finding the Values for the Cost Games

**cost_game3:**

- **Input:**The input for this function is the tuple as it has been defined in chapter 3 of the thesis. This means that **EOQ** is used for coalitions with one player and the **simple_IG** is used for coalitions with multiple players.

- **Output:** The output of this function is the values for the cost game $(N, c^2)$

- **Description:** The procedure that is used is the following: we first perform check to see that the input data are correct, then we create all the possible subsets of N, excluding the empty set, finally we compute the value of each coalition according to the chosen policy. In order to compute the value for each coalition we use a simple procedure: 1) if the coalition contains a single player we use the **EOQ function**, 2) if the coalition contains multiple players then we use the **simple_IG function** as they have been described previously.

**cost_gameDH:**

- **Input:**The input for this function is the tuple as it has been defined in chapter 3 of the thesis, but different formulas are used since a different policy is used. This means that **EOQ** function is used for coalitions with one player and the **one_frequency** function is used for coalitions with multiple players.

- **Output:** The output of this function is the values for the cost game $(N, c^1)$

- **Description:** The procedure that is used is similar to the **cost_game3** function.

## Functions for Checking the Core of a Cost Game

**balancedness:**

- **Input:** This function takes as an input any cost game with either 2,3, or 4 players, which is defined as a dictionary containing for every coalition S the respective cost.

- **Output:** The function checks if the given cost game is balanced or not, if balanced it returns 1, else 0.

- **Description:** The function checks all of the inequalities that are derived from the balanced collections for the cost game of 2,3, or 4 players (check appendix A) and if they all hold it returns 1.

**metric_e:**

- **Input:** This function takes as an input any cost game with either 3, or 4 players, which is defined as a dictionary containing for every coalition S the respective cost.
- **Output:** The function computes metric e as it has been defined in chapter 4 of the thesis and returns it to the main program.
- **Description:** The description for the metric cn be seen in chapter 4 of the thesis.

## Functions for Performing the Numerical Experiments

**create_jrp_instances:**

- **Input:** The input is the upper limits for the cost parameters as well as the number of products.
- **Output:** The outputs are lists containing the parameters for each of the random games.
- **Description:** This function is used to create n random inventory games (or we can call them JRP instances as well). We are sampling the cost parameters for the games according to a uniform distribution and according to the limits and intervals as defined in chapter 4 and of course according to the experiment we are going to perform.

**numerical_question_1:**

- **Input:** The input are the sampling parameters for the sampling of random games, the number of players for each game, and the number of random games (equivalently simulation runs) to be generated.
- **Output:** The output that is printed on the screen is the probability that a game according to Policy 1 is balanced and the respective confidence intervals also the same for Policy 2. Moreover the histograms for metric e are also printed on the screen.
- **Description:**The function first calls the function **create_jrp_instances** to create the random games with the required number of players, subsequently we need to a loop to check all of the created games and see if they satisfy balancedness and to compute metric e. Once all the simulated instances have been checked for balancedness according to the two games $(N, c^2)$ and $(N, c^1)$ and the metric e has been calculated, we compute the estimates for the probabilities and the confidence intervals and we plot the histograms.

The flowchart for the described process is once again provided here for clarity:

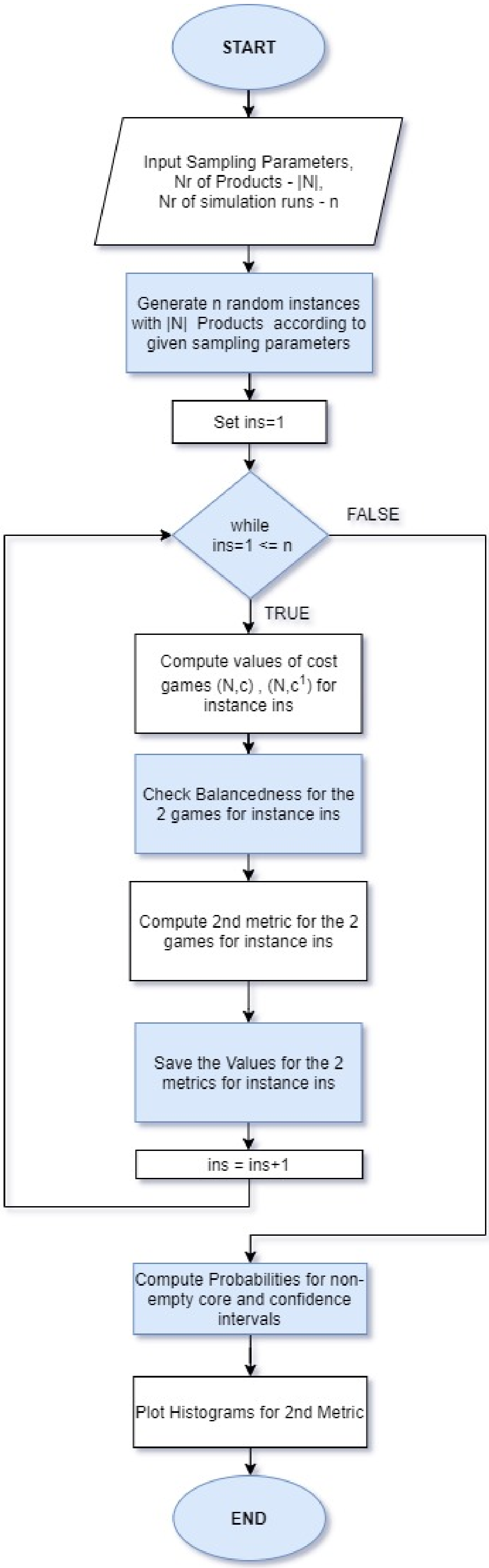



Figure D.1: Flowchart for Numerical Study On The Core

**Shapley:**

- **Input:** This function takes as an input the values for a cost game.
- **Output:** The function returns the allocation according to the Shapley value.
- **Description:** The code for this function is based on (susobhang70/shapleyvalue: Python code to find Shapley Value of a Characteristic Form Game - GitHub, n.d.). The main part of the function is based on the above script, but the input and output type of the function has been adjusted so as to work with the remaining functions in the thesis.

**check_core:**

- **Input:** The input for the function is a cost game, and a cost allocation.
- **Output:** The function returns either 1 if the allocation belongs to the core or 0 if the allocation does not belong to the core.
- **Description:** The function checks for all sub-coalitions the stability constraints and the efficiency constraint as a double check. The moment we find a coalition where stability does not hold we stop checking the remaining. Note that we do not explicitly check the stability constraint $\sum_{i \in N} x_i \leq c(N)$, because due to the efficiency constraint it is always satisfied. We have also added a rounding error check after the efficiency check and so if the two compared values are very very close, their difference is less than or equal to 1E-10, they are considered equal. The same rounding error check happens when checking inequalities in stability constraints.

**check_imputation_set:**

- **Input:** The input for the function is a cost game, and a cost allocation.
- **Output:** The function returns either 1 if the allocation belongs to the imputation set or 0 if the allocation does not belong to the imputation set.
- **Description:**The function checks the individual rationality constraints and the efficiency constraint as a double check. We have also added a rounding error check after the efficiency check and so if the two compared values are very very close, their difference is less than or equal to 1E-10, they are considered equal. The same rounding error check happens when checking inequalities.

**numerical_question_2:**

- **Input:** The input are the sampling parameters for the sampling of random games, the number of players for each game, and the number of random games (equivalently simulation runs) to be generated.
- **Output:** The output that is printed on the screen is the estimates for the probability that each allocation rule results in a core allocation / allocation in imputation set along with the confidence intervals.

- **Description:**The function first calls the function **create_jrp_instances** to create the random games with the required number of players, subsequently we need to a loop to check for all of the created games how many of the chosen allocation rules belong to the imputation set and how many belong to the core set. Once we have iterated over all of the instances, we compute the estimates for the probabilities and the confidence intervals for each rule and each metric.

The flowchart for the described process is once again provided here for clarity:

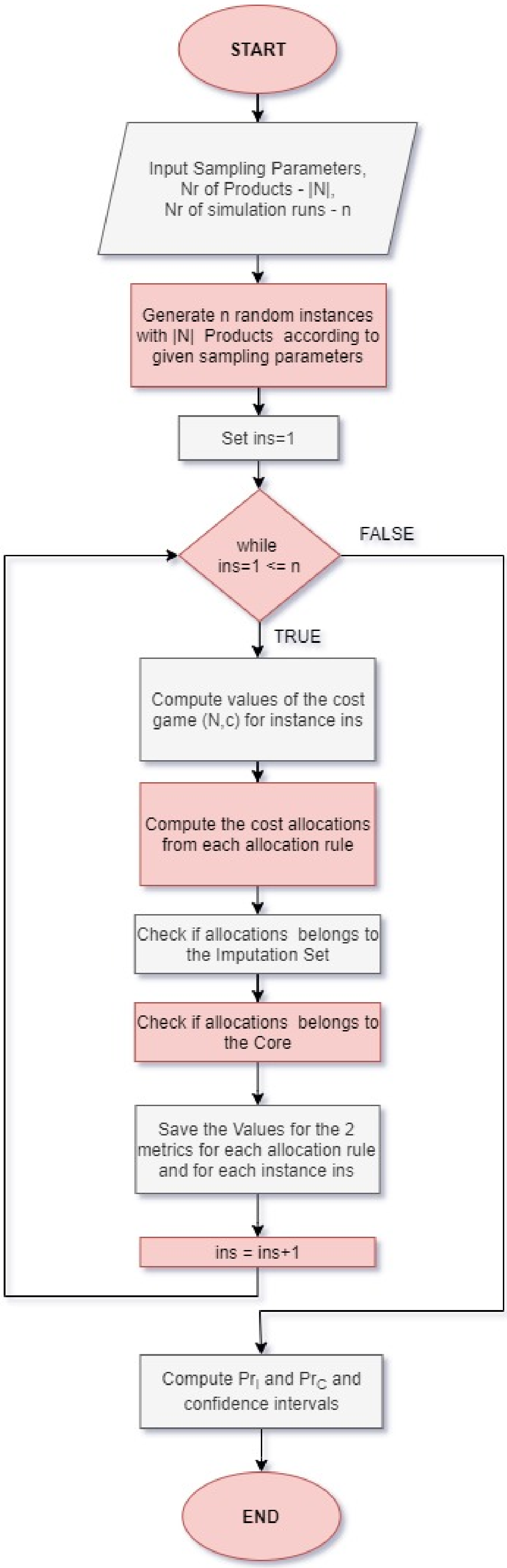


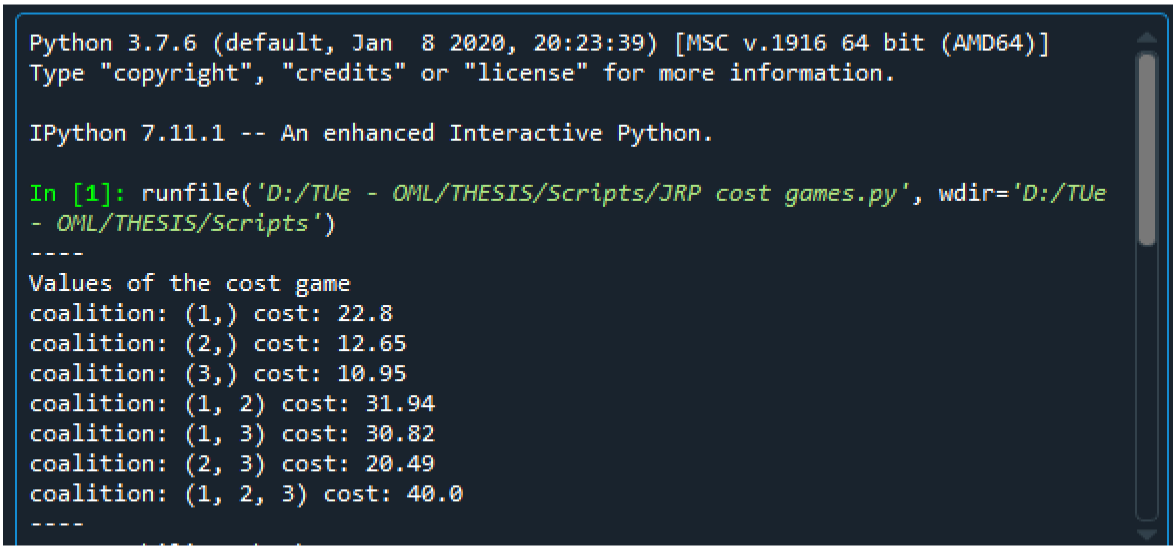

Figure D.2: Flowchart for Numerical Study On Allocation Rules

## Validation of the Programs

The validation of the computer programs was made in each individual function but also when all of them were combined to form the "larger" programs which were used to perform the numerical experiments. We used two methods to validate the programs: an Excel file where we calculated a number of intermediate steps in the computations but we also compared the results we got from our programs with certain examples that were provided in scientific papers and were of relevance to our research.

**cost_game3:** This function is very important because it is the basis of our experiments and is the function that computes the cost values for the coalitions. In order to validate we computed an example both in our Excel sheet but also using the program. The output of the Excel file and the Python script can be seen in Figure D.3, Figure D.4, and Figure D.5.

First of all we notice that the EOQ quantities and costs are computed correctly according to both Excel and the Python program. Next we notice in Figure D.4 that if all integer multipliers are set to 1, then we find the optimal costs for coalitions $\{1,2\}$, $\{2,3\}$, $\{1,2,3\}$. But for coalition $\{1,3\}$ we do not have the optimal cost because if the multiplier for player 3 is equal to 2, we find a cost which is lower. This can be clearly in Figure D.5. From the two Excel figures, we also have an indirect way of seeing if our enumeration procedure works correctly, since we are able to check the cost values for all coalitions when the integer multipliers change from 1 to 2. We conclude that our function provides the correct values for a given cost game.


```
Python 3.7.6 (default, Jan  8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.11.1 -- An enhanced Interactive Python.

In [1]: runfile('D:/TUe - OML/THESIS/Scripts/JRP cost games.py', wdir='D:/TUe
- OML/THESIS/Scripts')
----
Values of the cost game
coalition: (1,) cost: 22.8
coalition: (2,) cost: 12.65
coalition: (3,) cost: 10.95
coalition: (1, 2) cost: 31.94
coalition: (1, 3) cost: 30.82
coalition: (2, 3) cost: 20.49
coalition: (1, 2, 3) cost: 40.0
----
```

Figure D.3: Function cost_game3 - Python Console Output

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | Player 1 | Player 2 | Player 3 | | | | | | coalition formulas | | | |
| 2 | major ordering cost [euros] | a | 1 | | | | | k1 | 1 | | {1,2} grand | {1,3} | {2,3} | {1,2,3} |
| 3 | minor ordering cost [euros] | si | 1 | 1 | 1 | | | k2 | 1 | | | | | |
| 4 | holding cost [euros/(item*time)] | h | 1 | 1 | 1 | | | k3 | 1 | | | | | |
| 5 | demand [items/time] | d | 130 | 40 | 30 | | | | | | | | | |
| 6 | | | | | | | | | | | | | | |
| 7 | | h*d*k | 130 | 40 | 30 | | | | | | | | | |
| 8 | | | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | | | |
| 10 | | EOQ Formulas | | | | | | | | | | | | |
| 11 | optimal quantity [items] | q* | 22.80 | 12.65 | 10.95 | | | | | | | | | |
| 12 | | | | | | | | | | | | | | |
| 13 | optimal cost [euros/time] | c* | 22.80350850 | 12.65 | 10.95 | | | | | | 31.94 | 30.98 | 20.49 | 40.00 |
| 14 | | | 22.804 | | | | | | 35.45 | | | | | |
| 15 | optimal ordering frequency [1/time] | m* | 5.70 | 3.16 | 2.74 | | | | | | 5.32 | | | |
| 16 | optimal time between replenishments [time] | T* | 0.18 | 0.32 | 0.37 | | | | | | 0.19 | | | 0.20 |

Figure D.4: Function cost_game3 - Excel Output (1/2)

I4 | $f_x$ | 2

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | Player 1 | Player 2 | Player 3 | | | | | | coalition formulas | | | |
| 2 | major ordering cost [euros] | a | 1 | | | | | k1 | 1 | | {1,2} grand | {1,3} | {2,3} | {1,2,3} |
| 3 | minor ordering cost [euros] | si | 1 | 1 | 1 | | | k2 | 1 | | | | | |
| 4 | holding cost [euros/(item*time)] | h | 1 | 1 | 1 | | | k3 | 2 | | | | | |
| 5 | demand [items/time] | d | 130 | 40 | 30 | | | | | | | | | |
| 6 | | | | | | | | | | | | | | |
| 7 | | h*d*k | 130 | 40 | 60 | | | | | | | | | |
| 8 | | | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | | | |
| 10 | | EOQ Formulas | | | | | | | | | | | | |
| 11 | optimal quantity [items] | q* | 22.80 | 12.65 | 10.95 | | | | | | | | | |
| 12 | | | | | | | | | | | | | | |
| 13 | optimal cost [euros/time] | c* | 22.80350850 | 12.65 | 10.95 | | | | | | 31.94 | 30.82 | 22.36 | 40.12 |
| 14 | | | 22.804 | | | | | | 35.45 | | | | | |
| 15 | optimal ordering frequency [1/time] | m* | 5.70 | 3.16 | 2.74 | | | | | | 5.32 | | | |
| 16 | optimal time between replenishments [time] | T* | 0.18 | 0.32 | 0.37 | | | | | | 0.19 | | | 0.17 |

Figure D.5: Function cost_game3 - Excel Output (2/2)

**numerical_question_1:** For the validation of this function we need to check a number of issues. We first need to check if the sampling of the cost parameters is indeed as expected (according to uniform distribution), this is meant as a check for the random number generator we are using. So we created a sample of the holding costs, the demand rates, and the minor costs and we checked the mean and variance of the sample (we checked 1,000 randomly generated 2-player instances). The parameters we used can be seen in Table 4.2. We note that for a 2-player game we sample 2,000 minor costs, 2,000 holding costs and 2,000 demand rates according to the desired sampling ranges because we need 1,000 parameters for each player. The results for one of the checks we performed (a check for 1,000 random games with 2-players) can be seen in the respective table.

| Parameter | Mean of Sample | Variance of Sample |
|---|---|---|
| Holding Cost | 12.49 | 51.63 |
| Minor Cost | 3.95 | 2.97 |
| Demand Rate | 5,434.89 | 8,041,869.49 |

Table D.1: Validation of the random number generator(1/2)

The analytical results for a discrete uniform distribution according to the parameters used are the following:

| Parameter | Mean | Variance |
|---|---|---|
| Holding Cost | 12.5 | 52.08 |
| Minor Cost | 4 | 3 |
| Demand Rate | 5,500 | 8,333,333.33 |

Table D.2: Validation of the random number generator(2/2)

We can also plot the histograms for the generated samples of 2,000 parameters to see how well the samples corresponds to a uniform distribution:

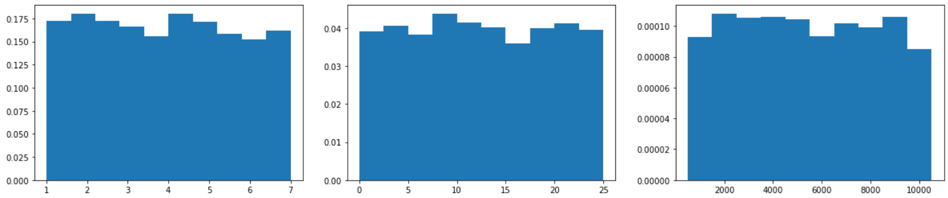


Figure D.6: Probability Mass Function of Generated Parameter Samples

We can see that the random number generator provides reasonably accurate results and this will allow us to have a valid numerical experiment.

When we check the balancedness conditions, we need to have an idea of how one side of an inequality compares with the other. This is important because if the two sides of an inequality are very very close then the two parts may be equal and due to rounding errors they seem not equal. We perform some checks on this problem by explicitly checking the computed values for metric e. Metric e compares the 2 sides of all the balancedness inequalities and takes the minimum. This means that if a game is not balanced (i.e. negative metric) the remaining inequalities have a larger difference between their two sides and so it is sufficient to check only the value of metric e. Similarly if we have a balanced game, then again checking metric e is sufficient. But since metric e is expressed in percentage form we are going to check its absolute value to have a better understanding of how close the two sides of the balancedness inequalities actually are for all the simulated games.

We check the raw data after sorting them for the case where A=2 and number of products=3 and we find that the difference between the two sides of inequalities are at least different in the second digit. This means that the two sides are different by a sufficiently big number and so no rounding errors take place. Similarly we check for A=15 and nr of products=4 and we obtain similar results.

We further print some intermediate results in the program to ensure that the functions described previously work well together. We conclude that our program works as expected and is valid.

**Shapley:** Considering that the code for this function was available on the internet, validation is extremely important to ensure that the function provides the results that it should. We can see a 3-person game along with the computed Shapley value in (Slikker, 2020). By running the code we can see that the results obtained are correct. The results can be seen on the respective figures.

| order | player 1 | player 2 | player 3 |
|---|---|---|---|
| 1-2-3 | 0 | 1 | 5 |
| 1-3-2 | 0 | 4 | 2 |
| 2-1-3 | 1 | 0 | 5 |
| 2-3-1 | 1 | 0 | 5 |
| 3-1-2 | 2 | 4 | 0 |
| 3-2-1 | 1 | 5 | 0 |
| sum | 5 | 14 | 17 |
| average | $\frac{5}{6}$ | $2\frac{2}{6}$ | $2\frac{5}{6}$ |

Figure D.7: The Shapley Value of a 3-Player Game

```
Console 2/A
Python 3.7.6 (default, Jan  8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.11.1 -- An enhanced Interactive Python.

In [1]: runfile('D:/TUe - OML/THESIS/Scripts/JRP allocation rules.py', wdir='D:/
TUe - OML/THESIS/Scripts')
{(1,): 0.8333333333333333, (2,): 2.333333333333333, (3,): 2.8333333333333335}
```

Figure D.8: Shapley Value based on the script

**NOTE:** that the value for coalition 3 is not entirely correct due to rounding errors. This happens because certain numbers cannot be represented 100% correctly numerically. The computer performs rounding in order to represent certain numbers and so we need to be careful and take those rounding errors into account when performing our numerical experiments.

**numerical_question_2:** We need to check a number of issues before conducting the experiment on the performance of allocation rules. The most important check that we need to do is whether the functions **check_core** and **check_imputation_set** work as expected. To do so we check the results we get for a given game and a given allocation: we first check the **check_imputation_set** function and we check an allocation that is not efficient and the results are as expected. Subsequently we check an allocation which is efficient but not individually rational and lastly an allocation which satisfies both. We do similar checks for the **check_core** function.

Subsequently, we check again for possible rounding errors. Considering that when we are computing the cost allocations we are doing divisions and multiplications, there is a high chance that rounding errors in the last few digits occur. Those errors are not important in a practical setting but in the simulation they may give the wrong impression on the performance of certain allocation rules. In order to check how the two sides of an inequality in a stability or individual rationality constraint perform, we simulate a small number of games at first and explicitly check the two sides of the inequalities. We notice that there are many inequalities with a difference in the 12th or 13th digit (i.e. $x_1 - c(\{1\}) = 4.55e - 13$). This check tells us that allocation $x$ allocates slightly more cost to player 1 than his standalone cost. However the difference between the allocated cost and the standalone cost is not significant and happens due to rounding errors. For this reason we add some extra checks in the **check_core** and **check_imputation_set** functions and so if the difference between allocated costs and incurred costs is higher than $1e - 10$ then we take it into account. Otherwise we assume that the two terms we are comparing are equal.

Lastly, we move on to check some intermediate results on the program and we conclude that the program works as expected.

# Appendix E

# Computation Times for Experiments

In this appendix we present the computation times for the various numerical experiments which we have conducted in the thesis:

| Number of Players | Number of Simulation Runs | Computation Time |
|---|---|---|
| 3 | 10,000 | 4.526 sec |
| 4 | 10,000 | 14.154 sec |

Table E.1: Numerical Experiments Chapter 4

| Number of Players | Number of Simulation Runs | Computation Time |
|---|---|---|
| 3 | 10,000 | 5.668 sec |
| 4 | 10,000 | 13.410 sec |
| 5 | 10,000 | 48.839 sec |
| 6 | 10,000 | 142.651 sec |
| 7 | 10,000 | 393.371 sec |
| 8 | 10,000 | 1,244.783 sec |

Table E.2: Numerical Experiments Chapter 5

We note that those computation times refer to a given set of sampling parameters (specific number of players, major cost, etc).

# Appendix F

# Additional Experiments - Chapter 4

In this appendix we present the results obtained after repeating the experiments of Chapter 4 with different sampling parameters for $\hat{h}$ and $\hat{D}$. A table containing the sampling parameters for the experiments along with the results of the experiments can be seen.

| Parameter | Nr. of Values | Sampling interval |
|---|---|---|
| $A$ | 1 | 2,8,15 |
| $s_i$ | n | [1,1+$\hat{s}$]=[1,7] |
| $h_i$ | n | [0 , $\hat{h}$]=[0,25] |
| $D_i$ | n | $[500, 500 + \hat{D}]$=[500,10,500] |

Table F.1: Parameters for numerical study of non-emptiness of core - second set

| Nr. of Products | Policy 2 | | Policy 1 | |
|---|---|---|---|---|
| | Mean Value | Confidence Interval | Mean Value | Confidence Interval |
| 3 | 0.9832 | [0.9807 , 0.9857] | 0.9427 | [0.9381 , 0.9473] |
| 4 | 0.9753 | [0.9723 , 0.9783] | 0.9047 | [0.8989 , 0.9105] |

Table F.2: Probability of non-empty core, A=15, second set of parameters

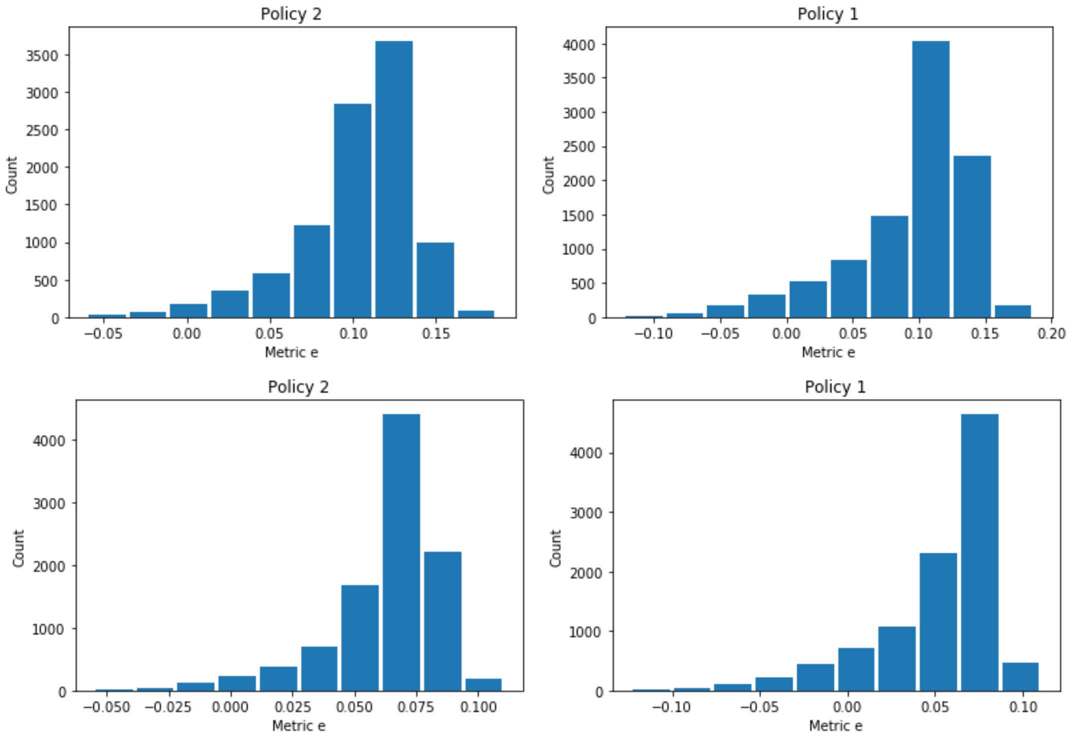



Figure F.1: Histograms for Metric e for A=15, Upper Graphs: 3-product games , Lower Graphs: 4-product games, second set of parameters

| Nr. of Products | Policy 2 | | Policy 1 | |
|---|---|---|---|---|
| | Mean Value | Confidence Interval | Mean Value | Confidence Interval |
| 3 | 0.9638 | [0.9601 , 0.9675] | 0.8716 | [0.8650 , 0.8782] |
| 4 | 0.9338 | [0.9290 , 0.9387] | 0.7785 | [0.7704 , 0.7866] |

Table F.3: Probability of non-empty core, A=8, second set of parameters

| Nr. of Products | Policy 2 | | Policy 1 | |
|---|---|---|---|---|
| | Mean Value | Confidence Interval | Mean Value | Confidence Interval |
| 3 | 0.8327 | [0.8254 , 0.8400] | 0.5218 | [0.5120 , 0.5316] |
| 4 | 0.7461 | [0.7376 , 0.7546] | 0.3576 | [0.3482 , 0.3670] |

Table F.4: Probability of non-empty core, A=2, second set of parameters

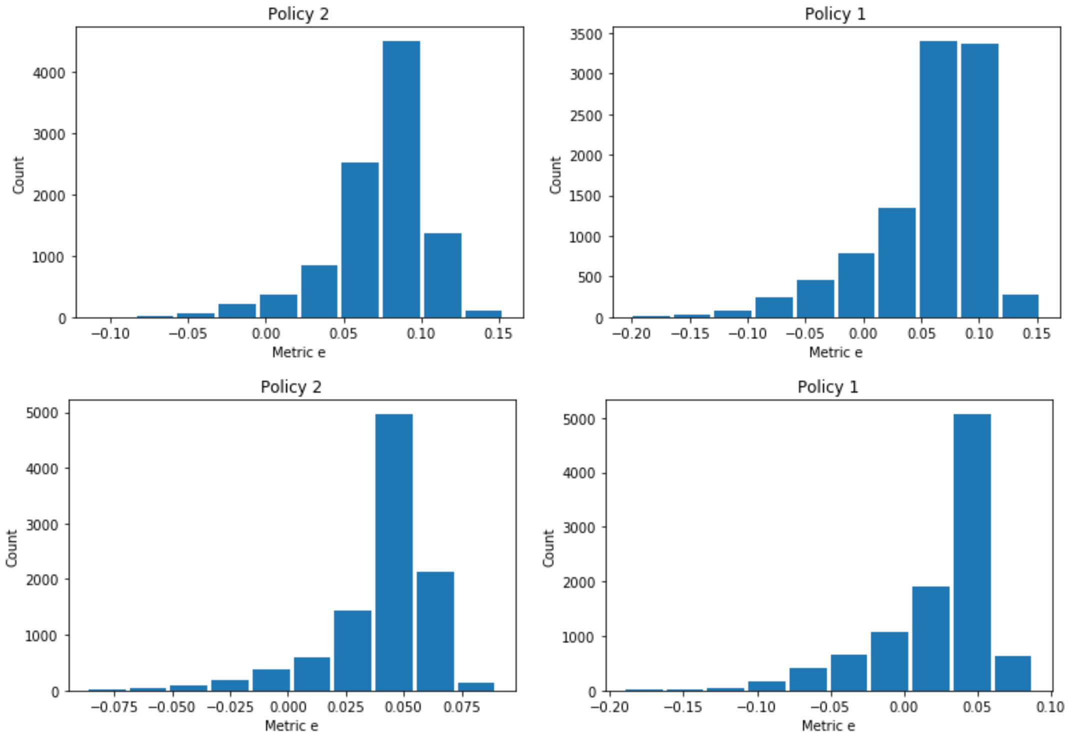



Figure F.2: Histograms for Metric e for A=8, Upper Graphs: 3-product games , Lower Graphs: 4-product games, second set of parameters

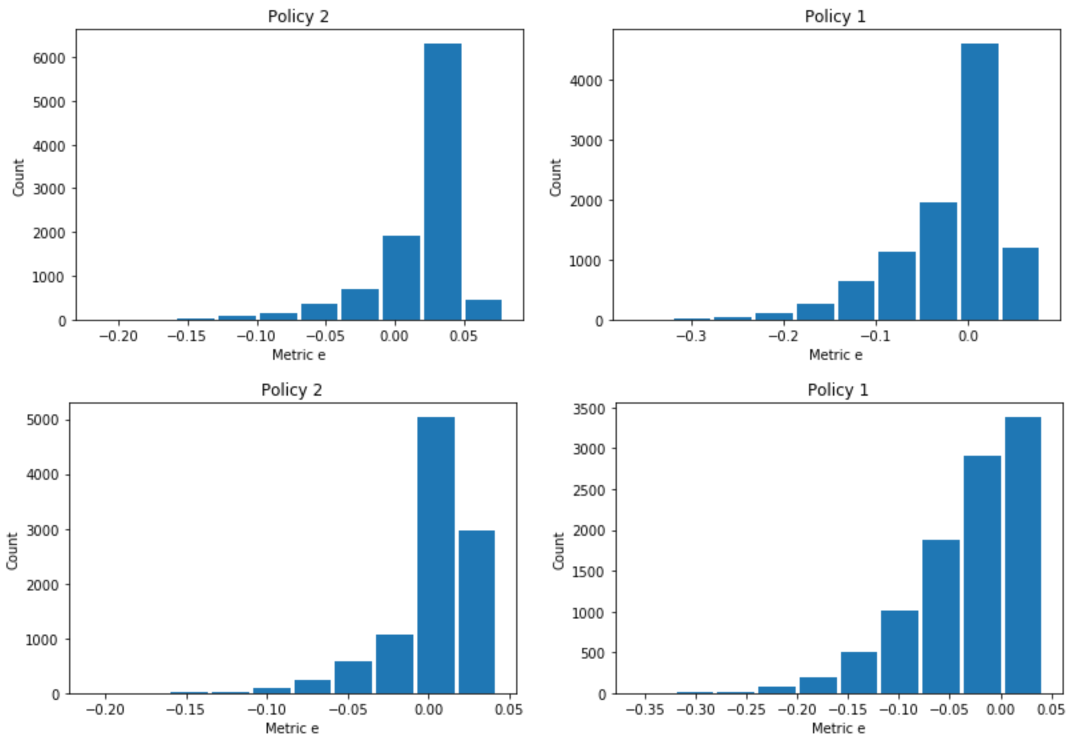



Figure F.3: Histograms for Metric e for A=2, Upper Graphs: 3-product games , Lower Graphs: 4-product games, second set of parameters

# Appendix G

# Additional Experiments - Chapter 5

In this appendix we present the results obtained after performing the experiments of Chapter 5 with a different set of sampling parameters. A table containing the sampling parameters for the experiments along with the results of the experiments can be seen.

**Second Set of Sampling Parameters**

| Parameter | Nr. of Values | Sampling interval |
|---|---|---|
| $A$ | 1 | 8,15 |
| $s_i$ | n | [1,1+$\hat{s}$]=[1,7] |
| $h_i$ | n | [0 , $\hat{h}$]=[0,25] |
| $D_i$ | n | $[500, 500+\hat{D}]$=[500 , 10,500] |

Table G.1: Parameters for numerical study on performance of allocation rules - second set

**NOTE:** the confidence intervals in italics have been calculated according to the Wilson interval, because the use of the standard normal approximation confidence interval resulted in either overshoot(upper bound of confidence interval which is greater than 1 ) or zero-width intervals (which falsely imply certainty).

Major Cost A=15

| **Allocation Rule** | $\hat{\mathbf{Pr}}_{\mathbf{I}}$ | **Confidence Intervals** | $\hat{\mathbf{Pr}}_{\mathbf{C}}$ | **Confidence Intervals** |
|---|---|---|---|---|
| Shapley Value | 0.9881 | [0.9860 , 0.9902] | 0.9776 | [0.9747 , 0.9805] |
| INDIVIDUAL Rule | 0.9999 | *[0.9994 , 1.0000]* | 0.9234 | [0.9182 , 0.9286] |
| INCREMENTAL Rule | 0.9734 | [0.9702 , 0.9766] | 0.9734 | [0.9702 , 0.9766] |
| LOUDERBACK Rule | 0.9842 | [0.9818 , 0.9866] | 0.9842 | [0.9818 , 0.9866] |
| $\hat{\mathbf{Pr}}$ **(of finding a core allocation)** | | | 0.9842 | |

Table G.2: Policy 2, 3-Item Games , A=15, **Balanced games: 9,842 /10,000**

| **Allocation Rule** | $\hat{\mathbf{Pr}}_{\mathbf{I}}$ | **Confidence Intervals** | $\hat{\mathbf{Pr}}_{\mathbf{C}}$ | **Confidence Intervals** |
|---|---|---|---|---|
| Shapley Value | 0.9822 | [0.9796 , 0.9848] | 0.9514 | [0.9472 , 0.9556] |
| INDIVIDUAL Rule | 1.0000 | *[0.9996 , 1.0000]* | 0.8724 | [0.8659 , 0.8789] |
| INCREMENTAL Rule | 0.9558 | [0.9518 , 0.9598] | 0.9547 | [0.9506 , 0.9588] |
| LOUDERBACK Rule | 0.9742 | [0.9711 , 0.9773] | 0.9727 | [0.9695 , 0.9759] |
| $\hat{\mathbf{Pr}}$ **(of finding a core allocation)** | | | 0.9732 | |

Table G.3: Policy 2, 4-Item Games , A=15, **Balanced games: 9,739 /10,000**

| **Allocation Rule** | $\hat{\mathbf{Pr}}_{\mathbf{I}}$ | **Confidence Intervals** | $\hat{\mathbf{Pr}}_{\mathbf{C}}$ | **Confidence Intervals** |
|---|---|---|---|---|
| Shapley Value | 0.9725 | [0.9693 , 0.9757] | 0.9088 | [0.9032 , 0.9144] |
| INDIVIDUAL Rule | 1.0000 | *[0.9996 , 1.0000]* | 0.8087 | [0.8010 , 0.8164] |
| INCREMENTAL Rule | 0.9372 | [0.9324 , 0.9420 ] | 0.9342 | [0.9293 , 0.9391] |
| LOUDERBACK Rule | 0.9591 | [0.9552 , 0.9630 ] | 0.9568 | [0.9528 , 0.9608] |
| $\hat{\mathbf{Pr}}$ **(of finding a core allocation)** | | | 0.9570 | |

Table G.4: Policy 2, 5-Item Games , A=15

| **Allocation Rule** | $\hat{\mathbf{Pr}}_{\mathbf{I}}$ | **Confidence Intervals** | $\hat{\mathbf{Pr}}_{\mathbf{C}}$ | **Confidence Intervals** |
|---|---|---|---|---|
| Shapley Value | 0.9621 | [0.9584 , 0.9658] | 0.8520 | [0.8450 , 0.8590] |
| INDIVIDUAL Rule | 1.0000 | *[0.9996 , 1.0000]* | 0.7168 | [0.7080 , 0.7256] |
| INCREMENTAL Rule | 0.9100 | [0.9044 , 0.9156] | 0.9049 | [0.8992 , 0.9106] |
| LOUDERBACK Rule | 0.9437 | [0.9392 , 0.9482] | 0.9398 | [0.9351 , 0.9445] |
| $\hat{\mathbf{Pr}}$ **(of finding a core allocation)** | | | 0.9405 | |

Table G.5: Policy 2, 6-Item Games , A=15

| **Allocation Rule** | $\hat{\mathbf{Pr}}_{\mathbf{I}}$ | **Confidence Intervals** | $\hat{\mathbf{Pr}}_{\mathbf{C}}$ | **Confidence Intervals** |
|---|---|---|---|---|
| Shapley Value | 0.9572 | [0.9532 , 0.9612] | 0.8016 | [0.7938 , 0.8094] |
| INDIVIDUAL Rule | 1.0000 | *[0.9996 , 1.0000]* | 0.6308 | [0.6213 , 0.6403] |
| INCREMENTAL Rule | 0.8920 | [0.8859 , 0.8981] | 0.8839 | [0.8776 , 0.8902] |
| LOUDERBACK Rule | 0.9348 | [0.9300 , 0.9396] | 0.9275 | [0.9224 , 0.9326] |
| $\hat{\mathbf{Pr}}$ **(of finding a core allocation)** | | | 0.9281 | |

Table G.6: Policy 2, 7-Item Games , A=15

| **Allocation Rule** | $\hat{\mathbf{Pr}}_{\mathbf{I}}$ | **Confidence Intervals** | $\hat{\mathbf{Pr}}_{\mathbf{C}}$ | **Confidence Intervals** |
|---|---|---|---|---|
| Shapley Value | 0.9447 | [0.9402 , 0.9492] | 0.7322 | [0.7235 , 0.7409] |
| INDIVIDUAL Rule | 1.0000 | *[0.9996 , 1.0000]* | 0.5040 | [0.4942 , 0.5138] |
| INCREMENTAL Rule | 0.8666 | [0.8599 , 0.8733] | 0.8548 | [0.8479 , 0.8617] |
| LOUDERBACK Rule | 0.9140 | [0.9085 , 0.9195] | 0.9051 | [0.8994 , 0.9108] |
| $\hat{\mathbf{Pr}}$ **(of finding a core allocation)** | | | 0.9060 | |

Table G.7: Policy 2, 8-Item Games , A=15

Major Cost A=8

| **Allocation Rule** | $\hat{\mathbf{Pr}}_{\mathbf{I}}$ | **Confidence Intervals** | $\hat{\mathbf{Pr}}_{\mathbf{C}}$ | **Confidence Intervals** |
|---|---|---|---|---|
| Shapley Value | 0.9704 | [0.9671 , 0.9737] | 0.9484 | [0.9441 , 0.9527] |
| INDIVIDUAL Rule | 0.9991 | [0.9985 , 0.9997] | 0.8749 | [0.8684 , 0.8814] |
| INCREMENTAL Rule | 0.9427 | [0.9381 , 0.9473] | 0.9427 | [0.9381 , 0.9473] |
| LOUDERBACK Rule | 0.9627 | [0.9590 , 0.9664] | 0.9627 | [0.9590 , 0.9664] |
| $\hat{\mathbf{Pr}}$ **(of finding a core allocation)** | | | 0.9627 | |

Table G.8: Policy 2, 3-Item Games , A=8, **Balanced games: 9,627/10,000**

| **Allocation Rule** | $\hat{\mathbf{Pr}}_{\mathbf{I}}$ | **Confidence Intervals** | $\hat{\mathbf{Pr}}_{\mathbf{C}}$ | **Confidence Intervals** |
|---|---|---|---|---|
| Shapley Value | 0.9522 | [0.9480 , 0.9564] | 0.8899 | [0.8838 , 0.8960 ] |
| INDIVIDUAL Rule | 1.0000 | *[0.9996 , 1.0000]* | 0.7744 | [0.7662 , 0.7826] |
| INCREMENTAL Rule | 0.9001 | [0.8942 , 0.9060] | 0.8959 | [0.8899 , 0.9019] |
| LOUDERBACK Rule | 0.9341 | [0.9292 , 0.939 ] | 0.9307 | [0.9257 , 0.9357] |
| $\hat{\mathbf{Pr}}$ **(of finding a core allocation)** | | | 0.9316 | |

Table G.9: Policy 2, 4-Item Games , A=8, **Balanced games: 9,329 /10,000**

| **Allocation Rule** | $\hat{\mathbf{Pr}}_{\mathbf{I}}$ | **Confidence Intervals** | $\hat{\mathbf{Pr}}_{\mathbf{C}}$ | **Confidence Intervals** |
|---|---|---|---|---|
| Shapley Value | 0.9376 | [0.9329 , 0.9423] | 0.8259 | [0.8185 , 0.8333] |
| INDIVIDUAL Rule | 1.0000 | *[0.9996 , 1.0000]* | 0.6453 | [0.6359 , 0.6547] |
| INCREMENTAL Rule | 0.8684 | [0.8618 , 0.8750] | 0.8578 | [0.8510 , 0.8646] |
| LOUDERBACK Rule | 0.9126 | [0.9071 , 0.9181] | 0.9037 | [0.8979 , 0.9095] |
| $\hat{\mathbf{Pr}}$ **(of finding a core allocation)** | | | 0.9054 | |

Table G.10: Policy 2, 5-Item Games , A=8

| **Allocation Rule** | $\hat{\mathbf{Pr}}_{\mathbf{I}}$ | **Confidence Intervals** | $\hat{\mathbf{Pr}}_{\mathbf{C}}$ | **Confidence Intervals** |
|---|---|---|---|---|
| Shapley Value | 0.9156 | [0.9102 , 0.9210] | 0.7452 | [0.7367 , 0.7537] |
| INDIVIDUAL Rule | 1.0000 | *[0.9996 , 1.0000]* | 0.4611 | [0.4513 , 0.4709] |
| INCREMENTAL Rule | 0.8350 | [0.8277 , 0.8423] | 0.8147 | [0.8071 , 0.8223] |
| LOUDERBACK Rule | 0.8843 | [0.8780 , 0.8906] | 0.8693 | [0.8627 , 0.8759] |
| $\hat{\mathbf{Pr}}$ **(of finding a core allocation)** | | | 0.8698 | |

Table G.11: Policy 2, 6-Item Games , A=8

| **Allocation Rule** | $\hat{\mathbf{Pr}}_{\mathbf{I}}$ | **Confidence Intervals** | $\hat{\mathbf{Pr}}_{\mathbf{C}}$ | **Confidence Intervals** |
|---|---|---|---|---|
| Shapley Value | 0.9013 | [0.8955 , 0.9071] | 0.6655 | [0.6563 , 0.6747] |
| INDIVIDUAL Rule | 1.0000 | *[0.9996 , 1.0000]* | 0.2856 | [0.2767 , 0.2945] |
| INCREMENTAL Rule | 0.8054 | [0.7976 , 0.8132] | 0.7740 | [0.7658 , 0.7822] |
| LOUDERBACK Rule | 0.8573 | [0.8504 , 0.8642] | 0.8333 | [0.8260 , 0.8406] |
| $\hat{\mathbf{Pr}}$ **(of finding a core allocation)** | | | 0.8363 | |

Table G.12: Policy 2, 7-Item Games , A=8

| **Allocation Rule** | $\hat{\mathbf{Pr}}_{\mathbf{I}}$ | **Confidence Intervals** | $\hat{\mathbf{Pr}}_{\mathbf{C}}$ | **Confidence Intervals** |
|---|---|---|---|---|
| Shapley Value | 0.8722 | [0.8657 , 0.8787] | 0.5751 | [0.5654 , 0.5848] |
| INDIVIDUAL Rule | 1.0000 | *[0.9996 , 1.0000]* | 0.1506 | [0.1436 , 0.1576] |
| INCREMENTAL Rule | 0.7661 | [0.7578 , 0.7744] | 0.7219 | [0.7131 , 0.7307] |
| LOUDERBACK Rule | 0.8192 | [0.8117, 0.8267] | 0.7886 | [0.7806 , 0.7966] |
| $\hat{\mathbf{Pr}}$ **(of finding a core allocation)** | | | 0.7912 | |

Table G.13: Policy 2, 8-Item Games , A=8

# Appendix H

# Scientific Poster

**Industrial Engineering and Innovation Sciences, OPAC Group**

TU/e Technische Universiteit Eindhoven University of Technology

## Joint replenishment of inventory in a multi-company setting: cost allocation and stability analysis


**Master Thesis Project by Charidimos Pratikakis– Student ID: 1362682**
**1st supervisor: dr. Marco Slikker**
**2nd supervisor: dr. ir. Loe Schlicher**


### Introduction

**The Joint Replenishment Problem:**

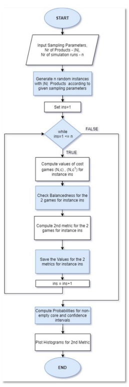
- Well-known problem in inventory management literature
- Decide on optimal order quantities for products ordered from the same supplier
- Minimize time-average cost: holding plus ordering/setup cost
- Major ordering cost when an order is placed, minor ordering cost depending on which products are placed in each order

### Research questions

"Which game theoretic properties are present in the various types of joint replenishment games with major and minor ordering costs under policies different from the optimal POT policy or the single group policy?"

1. Does a joint replenishment game with major and minor ordering costs that is based on policies different from the optimal POT policy or the single group policy always have a **non-empty core**?
2. What are proper **cost allocation rules** for this class of games and what are their properties?
3. How **sensitive** are the results obtained in the previous sub-questions to problem parameters?

### Methodology



- **Comparison of 2 joint replenishment policies:**
  - ✓ Policy 1: all products are always replenished together
  - ✓ Policy 2: products may be replenished every other cycle
- **Numerical Studies – Simulation**
- Simulation of a large number of problem instances and studying of the games that follow, their properties, and the performance of allocation rules
- Provide insight on allocation of cost/benefits of cooperation

- **Non-emptiness of the Core**
  - ✓ Probability that the core is non-empty
  - ✓ Empty core – how much lower should the costs be
- **Performance of Allocation Rules**
  - ✓ Probability that an allocation belongs to the imputation set
  - ✓ Probability that an allocation belongs to the core

### Main Results

**Numerical Study on The Core – Policy 1 vs. Policy 2**

| Major Cost | Policy 2 | Policy 1 |
|---|---|---|
| High (A=15) | 0.9823 | 0.9371 |
| Medium (A=8) | 0.9577 | 0.8459 |
| Low (A=2) | 0.8185 | 0.4549 |

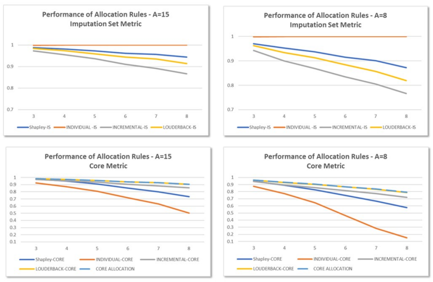
*Probability of Non-empty core: 4-item games*

| Major Cost | Policy 2 | Policy 1 |
|---|---|---|
| High(A=15) | 5% | 10% |
| Medium (A=8) | 7.5% | 15% |
| Low (A=2) | 12.5% | 28% |

*Empty core – max. decrease in costs of grand coalition needed: 4-item games*

- Flexibility – more optimization opportunities in all coalitions, stronger effect in the grand coalition
- Ranges of lower bound on cost savings – comparable for 2 policies
- The size of the core is comparable for the two policies
- Using different sampling ranges for holding costs and demand rates gives comparable results

**Numerical Study on The Performance of Allocation Rules**




*Policy 2 – Performance of allocation rules for 3-8-item games*

| Allocation Rule | Policy 2 | Policy 1 |
|---|---|---|
| Shapley Value | 0.8835 | 0.6241 |
| INDIVIDUAL | 0.8066 | 0.5307 |
| INCREMENTAL | 0.9476 | 0.8134 |
| LOUDERBACK | 0.9726 | 0.8889 |

*Policy 2 vs. Policy 1, Core Metric: 8-item games*

### Conclusions

- ➢ The game with Policy 1 / Policy 2 may have an empty core. In the examined range of parameters settings, Policy 2 lead to a higher probability of non-empty core.
- ➢ The LOUDERBACK rule and the Shapley value are well-performing rules in the core metric for settings where Policy 1 or Policy 2 are used and satisfy desirable properties.

Industrial Engineering and Innovation Sciences