UNIVERSIDAD DE INVESTIGACIÓN DE TECNOLOGÍA EXPERIMENTAL YACHAY

Escuela de Ciencias Matemáticas y Computacionales

TÍTULO: RESOURCE ALLOCATION AND SOCIAL WELFARE

Trabajo de integración curricular presentado como requisito para la obtención
del título de Ingeniero en Tecnologías de la Información

Autor:

Zhunio Ochoa Cristopher Julián

Tutor:

Ph.D. Camacho Franklin José

Urcuquí, diciembre 2020

Urcuquí, 23 de octubre de 2020

SECRETARÍA GENERAL
(Vicerrectorado Académico/Cancillería)
ESCUELA DE CIENCIAS MATEMÁTICAS Y COMPUTACIONALES
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
ACTA DE DEFENSA No. UITEY-ITE-2020-00036-AD

A los 23 días del mes de octubre de 2020, a las 15:00 horas, de manera virtual mediante videoconferencia, y ante el Tribunal Calificador, integrado por los docentes:

Presidente Tribunal de Defensa	Dr. MENDEZ PEREZ, MIGUEL ANGEL , Ph.D.
Miembro No Tutor	Dr. CUENCA LUCERO, FREDY ENRIQUE , Ph.D.
Tutor	Dr. CAMACHO , FRANKLIN , Ph.D.

El(la) señor(ita) estudiante **ZHUNIO OCHOA, CRISTOPHER JULIAN**, con cédula de identidad No. **0105016414**, de la **ESCUELA DE CIENCIAS MATEMÁTICAS Y COMPUTACIONALES**, de la Carrera de **TECNOLOGÍAS DE LA INFORMACIÓN**, aprobada por el Consejo de Educación Superior (CES), mediante Resolución **RPC-SO-43-No.496-2014**, realiza a través de videoconferencia, la sustentación de su trabajo de titulación denominado: **RESOURCE ALLOCATION AND SOCIAL WELFARE**, previa a la obtención del título de **INGENIERO/A EN TECNOLOGÍAS DE LA INFORMACIÓN.**

El citado trabajo de titulación, fue debidamente aprobado por el(los) docente(s):

Tutor	Dr. CAMACHO , FRANKLIN , Ph.D.

Y recibió las observaciones de los otros miembros del Tribunal Calificador, las mismas que han sido incorporadas por el(la) estudiante.

Previamente cumplidos los requisitos legales y reglamentarios, el trabajo de titulación fue sustentado por el(la) estudiante y examinado por los miembros del Tribunal Calificador. Escuchada la sustentación del trabajo de titulación a través de videoconferencia, que integró la exposición de el(la) estudiante sobre el contenido de la misma y las preguntas formuladas por los miembros del Tribunal, se califica la sustentación del trabajo de titulación con las siguientes calificaciones:

Tipo	Docente	Calificación
Presidente Tribunal De Defensa	Dr. MENDEZ PEREZ, MIGUEL ANGEL , Ph.D.	10,0
Tutor	Dr. CAMACHO , FRANKLIN , Ph.D.	10,0
Miembro Tribunal De Defensa	Dr. CUENCA LUCERO, FREDY ENRIQUE , Ph.D.	9,5

Lo que da un promedio de: **9.8 (Nueve punto Ocho)**, sobre 10 (diez), equivalente a: **APROBADO**

Para constancia de lo actuado, firman los miembros del Tribunal Calificador, el/la estudiante y el/la secretario ad-hoc.

ZHUNIO OCHOA, CRISTOPHER JULIAN
Estudiante

Dr. MENDEZ PEREZ, MIGUEL ANGEL , Ph.D.
Presidente Tribunal de Defensa

Dr. CAMACHO , FRANKLIN , Ph.D.
Tutor

Dr. CUENCA LUCERO, FREDY ENRIQUE , Ph.D.
Miembro No Tutor

MEDINA BRITO, DAYSY MARGARITA
Secretario Ad-hoc

AUTORÍA

Yo, **CRISTOPHER JULIÁN ZHUNIO OCHOA**, con cédula de identidad 0105016414, declaro que las ideas, juicios, valoraciones, interpretaciones, consultas bibliográficas, definiciones y conceptualizaciones expuestas en el presente trabajo; así cómo, los procedimientos y herramientas utilizadas en la investigación, son de absoluta responsabilidad de el/la autora (a) del trabajo de integración curricular. Así mismo, me acojo a los reglamentos internos de la Universidad de Investigación de Tecnología Experimental Yachay.

Urcuquí, diciembre y 2020.

Z Cristopher

Cristopher Julián Zhunio Ochoa
CI: 0105016414

AUTORIZACIÓN DE PUBLICACIÓN

Yo, **CRISTOPHER JULIÁN ZHUNIO OCHOA,** con cédula de identidad 0105016414, cedo a la Universidad de Tecnología Experimental Yachay, los derechos de publicación de la presente obra, sin que deba haber un reconocimiento económico por este concepto. Declaro además que el texto del presente trabajo de titulación no podrá ser cedido a ninguna empresa editorial para su publicación u otros fines, sin contar previamente con la autorización escrita de la Universidad.

Asimismo, autorizo a la Universidad que realice la digitalización y publicación de este trabajo de integración curricular en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior

Urcuquí, diciembre 2020.

Z Cristopher

Cristopher Julián Zhunio Ochoa
CI: 0105016414

Dedicatoria

Dedico este proyecto a todas las personas que me han ayudado a convertirme en la persona que soy hoy; particularmente a mis profesores y amigos, quienes son la familia que uno escoge.

Cristopher Julián Zhunio Ochoa

Agradecimiento

Expreso mi profundo sentido de gratitud a mis distinguidos y respetados mentores, Prof. Franklin Camacho y Prof. Rigoberto Fonseca por su gran ayuda y guía. Estoy muy agradecido con ellos por haberme alentado a completar el proyecto. Ellos me han ayudado desde el comienzo hasta el final del proyecto, el cual no lo hubiese podido terminar sin su ayuda.

Cristopher Julián Zhunio Ochoa

Resumen

En este trabajo se establecen condiciones que garantizan la existencia de asignaciones eficientes y justas a través de un enfoque matricial.
Un problema de asignación de recursos suele tener tres componentes: un conjunto finito de agentes, un conjunto finito de recursos y preferencias individuales de agentes sobre recursos.
A partir de las preferencias individuales, se establecen criterios que miden tanto la satisfacción social (eficiencia) como la individual (justicia). La eficiencia de una asignación se establece a través de la eficiencia de Pareto.
Cuando se reparten recursos indivisibles, el criterio más común, para medir la justicia, es la libre envidia de hasta un recurso.

Las preferencias individuales también permiten generar relaciones de bienestar social; esta son relaciones binarias sobre el conjunto de todas las asignaciones.
Estudiaremos el bienestar social utilitario y el bienestar social de Nash, veremos que: las asignaciones que maximizan ambas relaciones son eficientes, las que maximizan Nash son justas; pero, las que maximizan el bienestar social utilitario no siempre son justas.

Presentaremos condiciones que garantizan cuando existe justicia en las asignaciones que maximizan el bienestar social utilitario. La estrategia que se propone es darle un enfoque matricial al problema. A partir de este enfoque, se definen las asignaciones transitorias y se muestra que maximizan el bienestar social utilitario. Se define la propiedad de parcialmente justa y se demuestra que las asignaciones transitorias son parcialmente justas. Por otra parte, se muestra que las propiedades de justicia y parcialmente justas, en algunos casos, son equivalentes.

Como subproducto de este trabajo se propone un lenguaje de programación de dominio específico, denominado Resource Allocation Programming Language (RAPL).

Palabras Clave:

Libre de envidia de hasta un recurso, eficiencia de Pareto, bienestar social utilitario, bienestar social de Nash, utilidades aditivas.

Abstract

In this work, conditions are established that guarantee the existence of efficient and fair allocations of indivisible resources through a matrix approach. A resource allocation problem usually has three components: a finite set of agents, a finite set of resources, and individual preferences from agents over resources. Based on individual preferences, criteria are established to measure both social satisfaction (efficiency) and individual satisfaction (justice). The efficiency of an allocation is established through Pareto efficiency. When indivisible resources are distributed, the most common criterion, to measure justice, is the envy-free up to one good criterion.

Individual preferences also make it possible to generate social welfare relations; these are binary relations over the set of all allocations.
We will study the utilitarian social welfare and Nash social welfare. We will see that: the allocations which maximize both relations are efficient, those which maximize Nash are fair; but, those which maximize the utilitarian social welfare are not always fair.

We will present conditions that guarantee when there is justice in the allocations which maximize the utilitarian social welfare. The strategy proposed is to give a matrix approach to the problem. From this approach, the transitory allocations are defined and it is shown that they maximize the utilitarian social welfare. The property of partial fairness is defined and it is shown that the transitory allocations are partially fair. On the other hand, it is shown that the properties of fair and partially fair are equivalent for some cases.

As a by-product of this work, a domain-specific programming language is proposed, called Resource Allocation Programming Language (RAPL).

Key Words:

Envy-free up to one good, Pareto efficiency, utilitarian social welfare, Nash social welfare, additive utilities.

Contents

List of Figures

Chapter 1

Introduction

Assigning a group of available resources to a group of agents is a very common situation in our daily life: how to assign household tasks among the members of a family, how a government institution can distribute its budget. Without a doubt, this is a common problem, easy to pose, but perhaps, complicated to solve.

This problem usually has three ingredients: a finite set of agents, a finite set of resources, and individual preferences. The set of agents, depending on the application of the problem, represents: individuals, objects, government institutions, among others. It is common not to make any additional assumptions about this set. With respect to the resources to be distributed, they can be divisible or indivisible. A divisible resource is divided and assigned to different agents, while an indivisible resource is not. An example with divisible resources is to divide a piece of land among a certain number of individuals. On the other hand, an example with indivisible resources are the assets to be distributed in an inheritance: a house, a vehicle, among others. Individual preferences define the needs, tastes, interests, or requirements that agents have over the subsets of resources. These preferences are established from a qualitative or quantitative point of view. In the qualitative, preferences are structural or ordinal, that is, each agent defines a binary relation over the subsets of the resources. While in the quantitative approach, individual preferences are given through numerical functions. These functions are known as utility functions.

Group and individual satisfaction is sought based on individual preferences. Group satisfaction, also known as "efficiency" or Pareto optimal, seeks to establish when an allocation benefits the group, ensuring that no other allocation improves one agent without harming another. Individual satisfaction, known as *fairness*, seeks allocations that decrease envy among the individuals involved.

A classic example, that use divisible resources, to understand efficiency and fairness criterion is the"fair cake-cutting" example where two people try to divide a cake fairly. In order to guarantee the individual preference of the piece that each one receives, the first agent is in charge of dividing the cake in half and the second agent chooses first. Then, the first agent divides the cake ensuring equal preference for any piece. The second agent has two options: to choose the piece he prefers the most, or to choose the piece with least preference. If the second agent chooses the piece he prefers most, agent 1 will not envy what agent 2 chooses and therefore, agent 2 will not envy what agent 1 receives.

This allocation is fair because there is no envy between the two agents. Moreover, any other allocation that raises one agent's utility, decreases the other agent's utility; therefore, this allocation is also Pareto optimal. Other examples, using indivisible resources, will be reviewed through chapter 2 and chapter 3.

Another aspect that is considered, based on individual preferences, is how to define binary relations over the whole of all allocations. These relations are known as welfare relations. In the quantitative framework, utility functions define, on each allocation, social welfare functions (SW). An example of such functions is the utilitarian social welfare function (SW_U), which is the sum of the profits of the resources assigned to each agent. From a SW the set of all the allocations are ordered. This is, given a SW the social welfare relation $\succeq$ is defined as follows: given the allocations A and B,

$$A \succeq B \iff SW(A) \geq SW(B)$$

where $A \succeq B$ is interpreted as "the A allocation has better or equal social welfare than the B allocation".

The set of all possible allocations could be very large. This amount depends on the number of agents and the number of resources. For example, when distributing 20 resources among 3 agents, there are 1048576 possible allocations. In a problem like this, comparing the social welfare of each pair of allocations is quite complicated and perhaps unnecessary. Even more so, if one wants to analyze the criteria of justice and efficiency. For this reason, we want to reduce the size of the search space and consider the set formed by all those that maximize social welfare (MSW).

1.1 Background

How to fairly divide one piece of cake between two or more people? This problem took interest in the middle of the last century, when in 1947, before the Econometric Society of Washington D.C., Hugo Steinhaus posed this question before a community of mathematicians and economists [1]. From that moment on, the field of resource allocation became important in the scientific community. Important scientists such as John Nash [2], Von Neumman and Morgenstern [3] and Hugo Steinhaus [4, 5] made some contributions to this area. Most of the contributions that were made during the last half of the last century were to the problem with divisible resources [4, 2, 6, 7, 8]. Among the contributions to the problem with indivisible assets are [9, 10].

Fair resource allocation is present in many areas of knowledge; mathematics, computer science, political science and economics are the most common. In computer science a wide range of applications are found in Multi-Agent Systems and Artificial Intelligence. These problems are known as MultiAgent Resource Allocations (MARA) problems. MARA includes applications such as: task allocations in robotics scenarios [11], task programming in manufacturing systems[12, 13], school and course allocations for students [14], division of property in situations of divorce or inheritance [7, 14], among others. In computer science, there have been theoretical contributions in the field; among which it is worth mentioning the computational complexity of the algorithms [15, 16, 17] and the study of the properties of justice and efficiency [18, 19, 20]. A good review of the applications and theoretical aspects is presented by Chevaleyre et al. [21].

Allocations that maximize the utility SW, (MSW_U) are Pareto optimal; however, they are generally not fair, as it may be that only some agents get all the resources and the rest get no resources. Another SW is the Nash social welfare function (SW_{Nash}), this is obtained by multiplying the profits of the resources assigned to each agent. Caranaguis et al. [20], proposed the *ultimate solution* for allocating indivisible resources by combining efficiency and justice.

Finding an envy-free allocation (EF), in indivisible goods, is not always possible [22]. For this reason, Caragiannnis et al. [20] use a relaxed version of free of envy to measure justice, which is free of envy except for one good (EF1)[19, 22, 23, 20]. An allocation is EF1, if the envy between two agents disappears by eliminating exactly one resource in the individual evaluations. Now, under the consideration that efficiency is measured through Pareto optimal and justice through EF1, Caragiannis et al., Theorem 3.2 in [20], demonstrated that, under the assumption that agents define their preferences with additive utility functions, any allocation that maximizes the SW_{Nash} (MSW_{Nash}), is efficient and fair.

The result of Caragannis et al. is theoretically very important. It ensures that there is always an allocation that is Pareto optimal and EF1. In practice finding such an allocation remains a difficult problem, as finding a MSW_{Nash} is NP-Hard [16]. Therefore, getting allocations that are Pareto optimal and EF1, without having to find MSW_{Nash}, is an interesting problem.

On the other hand, Camacho et al., en [24], propose a SW using qualitative preferences and hierarchy among agents. In that work they used matrix properties to demonstrate some of their results. Now, taking advantage of the advantages that matrices have, both for visualizing information and for the algebraic properties they possess, we want to explore the matrix approach to find allocations that are Pareto optimal and, redefining the problem in some way, look for allocations that have the EF1 property.

1.2 Problem statement

In the problem of fair allocation of indivisible resources, in the quantitative framework, assuming: additive utility functions and the same budget per agent. We want to explicitly find allocations that are Pareto optimal, and within them to look for allocations that meet some criterion of justice. In addition, to propose a domain specific programming language that allows us to easily handle this type of problem.

1.3 Objectives

1.3.1 General objective

To formulate the fair allocation of indivisible resources from a matrix point of view, based on maximum preferences, to explicitly find Pareto optimal allocations and search among them for those that maximize Nash social welfare function.

1.3.2 Specific objectives

To achieve this general objective, the following specific objectives were set:

1. To raise the issue of fair allocation of indivisible resources following a matrix approach.
2. To create a method to find the allocations that maximize the utilitarian social welfare explicitly, based on the maximum preferences.
3. To propose a method to discriminate all the allocations which maximize the utilitarian social welfare using the Nash social welfare function.
4. To propose a domain specific language, based on the proposed matrix approach, as a tool to develop examples in this area.

1.4 Main contributions & results

This final degree project contributes to the research field of the fair allocation of indivisible resources with a matrix approach to the problem. This new approach proposes several matrices that allow us to represent and process the information of preferences and utilities of the agents. These matrices facilitate the development of new methods to satisfy the desired criteria of efficiency and justice.

- Valuation matrix: It is a matrix that represents the individual preferences of agents over resources.
- Allocation matrix: It is a binary matrix, with 0's and 1's inputs. The agents are represented in the rows and the resources in the columns. In each column, there is exactly a 1, which represents to which agent the resource was assigned.
- Transition matrix: It is a binary matrix, with entries of 0's and 1's, which describes which agents maximize each resource.
- Utility matrix: Given all the subsets of resources distributed in an allocation, this matrix shows the evaluation that each agent perceives on each subset.

Following the proposed matrix approach, two methods were created for the efficient and fair allocation of indivisible resources:

- The first method finds all the possible allocations which maximize the utilitarian social welfare based on the transition matrix. We showed that all these allocations are Pareto optimal.
- The second method looks for the allocations which meet an EF1 justice criterion. This maximizes Nash social welfare overall allocations found in the first method.

Another contribution is the first version of a domain-specific language (DSL), called RAPL. This DSL works with the different matrices proposed in this research and aims to facilitate the treatment of non-divisible resource allocation problems.

1.4.1 Disclosures and publications

The results of this degree work were shown and published in several national and international events:

- "Justicia y eficiencia en el bienestar social de Nash", presented at the first Ecuadorian Mathematics Conference. Protoviejo-Manabi. November, 2019.
- "Asignación de recursos con eficiencia y justicia débil en el bienestar social utilitario", approved to be presented in the III International Congress of Intelligent Systems and New Technologies: Interdisciplinary Trends in Health, October, 2020.
- "RAPL: A Domain Specific Language for Resource Allocation of Indivisible Goods", submitted to the TICEC 2020 scientific track.

1.5 Organization of final degree project

Chapter 2 exposes the theoretical foundations and works related to the allocation of indivisible resources. Chapter 3 presents the formulation of the problem following the matrix approach and the proposed methods. Chapter 4 presents the theoretical foundations of domain-specific languages and introduce the proposed DSL, which works with the results of chapter 3. In the end, chapter 5 summarizes the conclusions and future perspectives of this final project.

Chapter 2

Theoretical background review

The theoretical review explains all the concepts needed to understand and evaluate problems of allocating indivisible resources. The concepts reviewed in this chapter are defined through sets and functions, which is the most used point of view in the literature. These basic concepts are necessary to follow and understand chapter three, which exposes the same concepts from a matrix point of view.

2.1 Theoretical foundations

This section breaks down the components involved in the resource allocation problem. A clear nomenclature is now defined which will be used throughout the document to refer to these components and their characteristics.

The problem of "fair" resource allocation is to distribute fairly the available goods, at a given time, among the different actors involved. There are three components to this problem: the agents, the resources or goods, and justice.

The agents are the alternatives available at the time of the repercussion. The agents can be: people, objects, machines, among others. The set of all agents is denoted by N and it is assumed that this set is finite.

$$N = \{1, \dots, n\}.$$

Resources are the aspects or factors to be distributed and depend on the nature of the problem; they can be related to human, material, financial, technical, and other factors. The set formed by all the resources is denoted with M. In general, there is a limited amount of resources to be distributed and, therefore, it is rational to consider M as a finite set. That is to say,

$$M = \{1, \dots, m\}.$$

Justice is a moral principle that seeks to judge, respecting the 'truth', if each individual has what is his due. Now then, how do you determine if an allocation is fair? In the field of resource allocation, there are several arguments for determining fairness. Perhaps the most common, because it is the most natural, is "justice". Envy is defined as a feeling of unhappiness for not possessing what another has. Therefore, a criterion of justice is to find allocations that do not possess envy or that the envy is as small as possible.

On the other hand, resources are divided into two groups: divisible and non-divisible. As its name indicates, a divisible resource is one that can be distributed among several agents; that is, several agents obtain a fraction of this resource. In this case, envy can be avoided and it is feasible to find fair allocations. Whereas, an indivisible resource is distributed, in its entirety, to a single agent. Therefore, controlling envy among agents is more complicated or impossible. To clarify this, let us look at the following example.

Example 1. *A father wants to give his four children \$10 and five books. In this case,* $N = \{1, 2, 3, 4\}$ *and* $M = \{10\,dollars, 5\,books\}$*. When distributing the goods, the father wishes that all his children be satisfied with what they received; in the sense that, each son is satisfied with what he has received and is not envious of what his brothers received. So, the father observes that a fair way to distribute the \$10 is to divide it among the 4 sons; that is, to give each son \$2.50. Since each son receives the same value, none of them envies the other. But when it comes to dividing the books, it seems impossible to avoid envy among the brothers.*

This example shows that when working with non-divisible resources, such as books, envy among agents cannot always be eliminated, making it difficult to meet the criterion of justice.

From now on, unless the contrary is clarified, it is assumed that resources are indivisible (or indivisible). Furthermore, it is assumed that m is the number of resources and n is the number of agents; that is, $|M| = m$ and $|N| = n$.

Definition 1. *Let* $\{A_1, A_2, \cdots, A_n\}$ *be a partition of* M*; that is,* $A_1, A_2, \cdots A_n \subseteq M$ *such that* $\cup_{i=1}^{n} A_i = M$ *and for all* $i, j \in N$ *with* $i \neq j$*,* $A_i \cap A_j = \emptyset$*. A feasible allocation, or simply an allocation,* A*, is a* $n-$*tuple.*

$$A = (A_1, \dots, A_n)$$

where for every $i \in N$*,* A_i *are the resources assigned to the agent* i*.*

Since $\{A_1, A_2, \cdots A_n\}$ is a M partition, it guarantees that:

- All resources are allocated, that is to say, $\forall r \in M$, $\exists i \in N$ such that $r \in A_i$. This is because $\cup_{i=1}^{n} A_i = M$;
- no resource is allocated twice, for $A_i \cap A_j = \emptyset$.

The set formed by all the allocations will be denoted by M^N. The capital letters $A, B, C, \dots, F, G, \dots$ will denote allocations that belong to M^N. The cardinality of M^N is m^n.

Example 2. *Pablo* (1) *and Maria* (2) *won a contest and were awarded 4 ice creams, each with a different flavor. The flavors are: Chocolate* (c), *Blackberry* (m), *Vanilla* (v), *and Strawberry* (f). *In this case,* $N = \{1, 2\}$ *and* $M = \{c, m, v, f\}$, *there are* m^n *possible allocations, for this example* $|M^N| = 4^2 = 16$. *Pablo suggests spreading the ice cream as expressed in the allocation* A,

$$A = (A_1, A_2) = (\{f, m\}, \{v, c\})$$

where the first agent, Pablo, receives the strawberry and blackberry ice cream ($A_1 = \{f, m\}$*); while, the second agent, Maria, receives the vanilla and chocolate ice cream (*$A_2 = \{v, c\}$*). On the other hand, Maria suggests handing out the ice creams as expressed in the allocation* B,

$$B = (B_1, B_2) = (\{c, m, v\}, \{f\})$$

The allocation B *gives the chocolate, blackberry and vanilla ice cream to the first agent (*$B_1 = \{c, m, v\}$*), leaving the other agent with the strawberry ice cream only (*$B_2 = \{f\}$*).*

This example shows that there are many ways of allocating resources among agents. Choosing among the allocations depends on the preferences of the agents as it is reviewed below.

2.2 Individual Preferences

An interesting question is how to measure envy in an allocation. To answer this question it is necessary to know the "truth"; this is obtained from the individual preferences of the agents over the subsets of resources. It is denoted by $\mathcal{P}(M)$ to the set of all subsets of M.

Definition 2. *For* $i \in N$, *the individual preference of agent* i *over* $\mathcal{P}(M)$, *denoted by* $\geq_i$ *is a total preorder over* $\mathcal{P}(M)$. *That is,* $\geq_i$ *is a subset of* $\mathcal{P}(M) \times \mathcal{P}(M)$ *and satisfies the following conditions:*

1. *It is total;*

   $\forall A, B \in \mathcal{P}(M),\ (A \geq_i B) \vee (B \geq_i A)$

2. *It is transitive;*

   $\forall A, B, C \in \mathcal{P}(M),\ [(A \geq_i B) \wedge (B \geq_i C)] \Rightarrow (A \geq_i C)$.

It is worth noting that, for each agent $i \in N$, from $\geq_i$, the strict and indifferent part of $\geq_i$ is defined, denoted by $>_i$ and $\sim_i$, respectively. That is;

$$A >_i B \Leftrightarrow (A \geq_i B) \wedge \neg(B \geq_i A)$$

$$A \sim_i B \Leftrightarrow (A \geq_i B) \wedge (B \geq_i A).$$

The interpretation of these relations is as follows:

- $A \geq_i B$ is interpreted as "agent i, prefers A at least as much as B";
- $A >_i B$ is interpreted as "agent i, strictly prefers A to B";
- $A \sim_i B$ is interpreted as "agent i, prefers equally to both A and B"

The following example shows how individual preferences determine whether there is envy in an allocation.

Example 3. *Let us consider again the example 2, where $N = \{1, 2\}$ and $M = \{c, m, v, f\}$. The agents have to split the four ice creams based on each other's preferences. For Pablo, the agent 1, he is indifferent to any flavor:*

$$\{f\} \sim_1 \{c\} \sim_1 \{v\} \sim_1 \{m\}$$

and has a greater preference for ice cream with more flavors; that is,

$$A \geq_1 B \Leftrightarrow |A| \geq |B|.$$

While Maria, the agent 2, has a greater preference for strawberry than any other flavor. Chocolate, vanilla and blackberry are equally preferred. This is:

$$\{f\} >_2 \{c\} \sim_2 \{v\} \sim_2 \{m\}$$

and if they combine the flavors, she prefers anything that has the flavor of strawberry; that is to say,

$$A \geq_2 B \Leftrightarrow f \in A.$$

Now, let us consider the allocations B, C and D given as follows: $B = (\{c, v, m\}, \{f\})$, $C = (\{c\}, \{f, m, v\})$ and $D = (\{c, v\}, \{f, m\})$. For the allocation C, Pablo prefers what Maria was given, as he has a single flavor ice cream and Maria gets three flavors; that is, $C_2 = \{f, m, v\} >_1 \{c\} = C_1$; therefore, Pablo envies what Maria received. On the other hand, Maria does not envy Pablo because she has strawberry-flavored ice cream; that is,

$C_2 = \{f, m, v\} >_2 \{c\} = C_1$. *On the contrary, in the allocations* B *and* D *there is no envy among the agents, since Maria always received strawberry and Pablo had the same or more flavors than Maria's.*

From the previous example, we can observe that there are many ways to express individual preferences over $\mathcal{P}(M)$ and, from these, to establish if an assignation has individual satisfaction (envy) among the agents. For example, if Pablo, in his preferences, defines strawberry flavor strictly over any other flavor or flavor combination, then in the allocation D there is envy.

Generally, as shown in the example above, agents' preferences can be subjective. This approach is known as *qualitative*. Another approach is the quantitative or numerical one, in this case the preferences on the resources are objective; in the sense that, they use numerical functions to define the preferences. In other words, given a $R \subseteq M$, each agent $i \in N$ assigns a numerical value to R using a function, known in the literature as a *utility function*:

Definition 3. *Any function* u *of* $\mathcal{P}(M)$ *in the real numbers,* $u : \mathcal{P}(M) \to \mathbb{R}$, *is a utility function, where* $u(R)$ *is the number that represents the utility of the set* R.

In addition, a utility function u *is said to:*

- *not be negative if, and only if,* $u(R) \geq 0$ *for all* $R \in \mathcal{P}(M)$*;*
- *be positive if, and only if,* u *is not negative and for all* $R \in \mathcal{P}(M)$ *with* $R \neq \emptyset$ *you have to* $u(R) \neq 0$*;*
- *be additive if, and only if,* u *is positive and for all* $R \in \mathcal{P}(M)$,

$$u(R) = \sum_{r \in R} u(\{r\});$$

- *has* K *budget, with* $k \in \mathbb{R}$ *and* $K \geq 0$ *if, and only if,* $u(M) = K$

Let us note that if u is additive with a K budget, then

$$u(M) = \sum_{r \in M} u(\{r\}) = K. \tag{2.1}$$

Next we will see that if an agent has a utility function, this function defines an individual preference.

Lemma 1. *Let* i *be an agent,* $i \in N$, *and* u_i *be a utility function, then the relation* $\geq_i$ *over* $\mathcal{P}(M)$ *given by:* $\forall A, B \in \mathcal{P}(M)$,

$$A \geq_i B \Leftrightarrow u_i(A) \geq u_i(B) \tag{2.2}$$

is the individual preference of agent i.

Proof. *We want to prove that $\geq_i$ is a total preorder. Let $A, B, C \in \mathcal{P}(M)$. Let us see the totality of it. As $u_i(A), u_i(B) \in \mathbb{R}$, for the trichotomy of all real numbers, you have that $u_i(A) \geq u_i(B)$ or $u_i(B) \geq u_i(A)$. Then by the equation (2.2), $A \geq_i B$ or $B \geq_i A$. So, $\geq_i$ is total. For transitivity, let us assume that $A \geq_i B$ and $B \geq_i C$. . By the equation (2.2), $u_i(A) \geq u_i(B)$ and $u_i(B) \geq_i u_i(C)$. For the transitivity of $\geq$ in $\mathbb{R}$, $u_i(A) \geq u_i(C)$. Again by the equation (2.2), $A \geq_i C$. Then, $\geq_i$ is transitive. It has been shown that $\geq_i$ is total and transitive; then, $\geq_i$ is a total preorder.*

Let us look at an example to clarify this approach.

Example 4. *Let us consider again the examples 2 and 3. $N = \{1, 2\}$ and $M = \{c, m, v, f\}$. Suppose that Pablo, agent 1, considers the utility function, u_1, given by*

$$u_1(\{c\}) = u_1(\{m\}) = u_1(\{v\}) = u_1(\{f\}) = 2.5$$

And for all $R \in \mathcal{P}(M)$, $u_1(R) = \sum\limits_{r \in R} u_1(\{r\})$. That is, u_1 is an additive utility function. On the other hand; Maria, the agent (2), has an additive utility function, u_2, given by

$$u_2(\{c\}) = u_2(\{m\}) = u_2(\{v\}) = 2 \quad \textit{and} \quad u_2(\{f\}) = 4.$$

Note that both u_1 and u_2 have the same budget $K = 10$.
Now, for the allocation $C = (\{c\}, \{f, m, v\})$,agent 1, "evaluates" what he was assigned with 2.5; that is, $u_1(C_1) = u_1(\{c\}) = 2.5$. While he evaluates with 7.5 what was assigned to Maria, $u_1(C_2) = u_1(\{f, m, v\}) = 7.5$. Pablo gives more use to what was assigned to Maria, $u_1(C_2) \geq u_1(C_1)$. In this case, Pablo envies Maria. For agent 2, Maria, she evaluates her allocation better; that is, $u_2(C_2) > u_2(C_1)$, because $u_2(C_2) = u_2(\{f, m, v\}) = u_2(\{f\}) + u_2(\{m\}) + u_2(\{v\}) = 4 + 2 + 2 = 8$ and $u_2(C_1) = u_2(\{c\}) = 2$. Then, Maria does not envy Pablo.

In the example above, additive utility functions were considered for both agents with the same budget. Having the same budget guarantees that the agents have equal conditions to evaluate their preferences. On the other hand, it was observed that for the allocation C, there is no individual satisfaction, this was natural since Pablo received fewer flavors of ice cream than Maria. Evaluating individual satisfaction is complicated and sometimes "rational" assumptions do not work. For example, if you consider the allocation $B = (\{c, v, m\}, \{f\})$, it seems rational to think that both agents are satisfied because Maria received the strawberry ice cream, which is the flavor she prefers most, and Pablo received three flavors. However, $u_2(B_1) = u_2(\{c, v, m\}) = 6$ and $u_2(B_2) = u_2(\{f\}) = 4$. Then, Maria envies Pablo; that is, the allocation B has envy. In the next section we will study in more detail some properties of individual satisfaction (justice) and the property of social satiation (efficiency).

2.3 Allocation's properties: efficiency and justice.

At the moment of distributing m indivisible resources between n agents, m^n possible allocations can be calculated. Of course; not all allocations are good, but how can we determine this? In this section we will study properties to determine when an allocation is good. These properties are related to efficiency and justice. On the other hand, we will assume that for all $i \in N$, the individual preferences $\geq_i$ are given through utility functions, u_i. That is, $\geq_i$ are given by the equation (2.2). Then, for every $i \in N$, the utility function u_i will be referred to instead of the relation $\geq_i$.

2.3.1 Efficiency: Pareto optimal

The efficiency of an allocation will be measured according to the Pareto efficiency criterion, also known as the Pareto optimal [22]. This criterion is named after Vilfrido Pareto (1848-1923). It consists of maximizing a relation of social satisfaction on all allocations. This satisfaction is measured by comparing two allocations A and B, from the individual preferences of each agent and their corresponding evaluations, that is: "A is socially preferred to B, if there is an agent that improves its valuation utility in the allocation B, with respect to that assigned by A and no agent worsens its evaluation in B with respect to A". Let us formally look at the definition:

Definition 4. *Let $A, B \in M^N$ and $i, j \in N$. A is Pareto dominated by B if $u_i(A_i) \leq u_i(B_i)$ and there is at least one j where $u_j(A_j) < u_j(B_j)$.*

Below is an example where one allocation dominates another respective to Pareto.

Example 5. *We want to distribute 3 resources among 3 agents; that is, $M = \{a,\ b,\ c\}$ and $N = \{1,\ 2,\ 3\}$. Each agent $i \in N$ considers additive utility functions u_i given in the following way:*

$$u_1(\{a\}) = 20, \quad u_1(\{b\}) = 30, \quad u_1(\{c\}) = 50;$$
$$u_2(\{a\}) = 15, \quad u_2(\{b\}) = 50, \quad u_2(\{c\}) = 35;$$
$$u_3(\{a\}) = 80, \quad u_3(\{b\}) = 20, \quad u_3(\{c\}) = 0.$$

Note that the utilities functions have budget $K = 100$ and $u_i(\emptyset) = 0$ for all i. Let us *consider the following allocations:*

$$A = (A_1, A_2, A_3) = (\{a, b\}, \emptyset, \{c\});$$
$$B = (B_1, B_2, B_3) = (\{c\}, \{b\}, \{a\}).$$

Each agent's valuations on allocations A and B are given by

$$u_1(A_1) = u_1(\{a, b\}) = u_1(\{a\}) + u_1(\{b\}) = 50, u_1(B_1) = u_1(\{c\}) = 50;$$
$$u_2(A_2) = u_1(\emptyset) = 0,\ u_2(B_2) = u_2(\{b\}) = 50;$$
$$u_3(A_3) = u_1(\{c\}) = 0, u_3(B_3) = u_3(\{a\}) = 80.$$

Note that the first agent, in the allocations A and B, has equal utility; but, in B the second and third agent improve their utility. Thus, A is Pareto dominated by B.

In order to obtain social satisfaction, we look for allocations that are not dominated by any other allocation, in the sense of Pareto. Below is the formal definition.

Definition 5. *Let $A, B \in M^N$, we will say that B is Pareto optimal (PO) if it is not Pareto dominated by any other A.*

PO allocations are also called efficient in the Pareto sense, or simply efficient. We will denote as $\mathcal{PO}$, the set of all PO allocations. Caragiannis et al.[20], showed that under additive utility functions, there is always a PO allocation in M^N; that is, $\mathcal{PO} \neq \emptyset$. Figure 2.1, shows that the allocations in M^N can be grouped into two sets. The first, $\mathcal{PO}$, conformed by the PO allocations, and the second, $\mathcal{PO}^c$, conformed by the non-PO allocations. In other words, $M^N = \mathcal{PO} \cup \mathcal{PO}^c$.

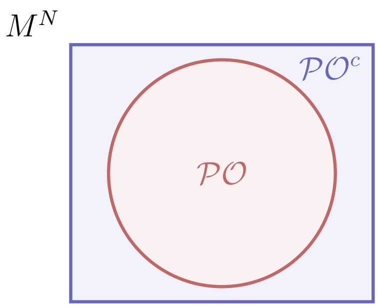


Figure 2.1: $M^N = \mathcal{PO} \cup \mathcal{PO}^c$

The easiest PO allocations to find are those that allocate all resources to a single agent. Let us look at some examples.

Example 6. *Let us consider again the example 5, $M = \{a, b, c\}$ and $N = \{1, 2, 3\}$. There are $3^3 = 27$ possible allocations. Let us consider C, the allocation that distributes all the resources to agent 1; that is to say, $C = (M, \emptyset, \emptyset)$. In this case,*

$$u_1(C_1) = u_1(M) = 100, \quad u_2(C_2) = u_2(\emptyset) = 0 \quad u_3(C_3) = u_3(\emptyset) = 0.$$

As the utility function of agent 1 is positive, then any allocation that improves agent 2 or agent 3, harms agent 1; then, C is PO. Another PO allocation is $B = (\{c\}, \{b\}, \{a\})$; one way to verify this is by checking that none of the remaining 26 allocations dominate B. Since $A = (\{a, b\}, \emptyset, \{c\})$, A is Pareto dominated by B, therefore A is not PO.

In the example above, we saw the allocations B and C that are PO and the allocation A that is not PO. From these non-OP allocations it is possible to arrive at OP allocations, through negotiations between agents. This was demonstrated by Endris et al. en [18], under the assumption that all agents consider additive utility functions.

It seems attractive to determine that an allocation is good if it satisfies PO. However, this property does not consider the individual satisfaction of the agents; that is, Pareto efficiency is indifferent to justice. As can be seen in the example 6, allocation C, which is PO, does not distribute any resources to agents 2 and 3. Although this allocation satisfies the efficiency criterion, it can be seen that not all agents are satisfied.

In general, determining when an allocation is in $\mathcal{PO}$ is a complicated problem; because it requires comparing all possible allocations. This increases the computational complexity of the problem, see [15].

Two important conclusions are drawn from this section: first, there can be more than one Pareto optimal allocation within a single problem. Second, this criterion is not sufficient to decide or label an allocation as *good*. More criteria are needed to classify allocations. In the following subsection we will study the property of free envy between agents.

2.3.2 Fairness: Envy-free

A property (PO) has been studied that allows classifying the allocations by comparing the agent utilities among the allocations. We want to find a criterion that compares the utilities of the agents within the same allocation and evaluates the "satisfaction" of each one. This criterion establishes the fairness of the allocation. Now, justice and efficiency (PO), will be the conditions to determine when an allocation is good. As mentioned earlier, one criterion related to justice is the property free of envy, [22]. An allocation is said to be envy-free if all agents value the set of resources assigned equally or better than the set of resources assigned to the other agents. This describes the agents' satisfaction with the allocation. Unfortunately, as we will see in this subsection, in the case of indivisible resources this property does not always exist. The following is the formal definition of an envy-free allocation.

Definition 6. *An allocation A is envy-free (EF) if for all $i, j \in N$, $u_i(A_i) \geq u_i(A_j)$.*

Note that if there is an agent $i \in N$ such that $u_i(A_i) < u_i(A_j)$ for some $j \in N$, then the agent i envies agent j. On the other hand, we will denote with $\mathcal{EF}$ the set formed by all the EF allocations. In the following example, we will see a problem where $\mathcal{EF} \neq \emptyset$.

Example 7. *We want to distribute 3 resources between 2 agents; in this case,* $M = \{a, b, c\}$ *and* $N = \{1, 2\}$*. Let us suppose that both agents consider additive utility functions given by*

$$u_1(\{a\}) = 20, \quad u_1(\{b\}) = 30, \quad u_1(\{c\}) = 50;$$

$$u_2(\{a\}) = 15, \quad u_2(\{b\}) = 50, \quad u_2(\{c\}) = 35.$$

As for every i*,* u_i *is additive, then* $u_i(\emptyset) = 0$*. Let us consider the allocations:*

$$A = (A_1, A_2) = (\{a, c\}, \{b\});$$

$$B = (B_1, B_2) = (\{b\}, \{a, c\})$$

each agent's valuations over A *are given by*

$$u_1(A_1) = u_1(\{a, c\}) = u_1(\{a\}) + u_1(\{c\}) = 20 + 50 = 70, \quad u_1(A_2) = u_1(\{b\}) = 30;$$

$$u_2(A_1) = u_2(\{a, c\}) = u_2(\{a\}) + u_2(\{c\}) = 15 + 35 = 50, \quad u_2(A_2) = u_2(\{b\}) = 50.$$

Then,

$$70 = u_1(A_1) > u_1(A_2) = 30 \quad \textit{and} \quad 50 = u_2(A_2) = u_2(A_1) = 50$$

no agent envies the other. That is, $A \in \mathcal{EF}$*. While valuations over* B *are given by*

$$u_1(B_1) = 30, \quad u_1(B_2) = 70 \quad \textit{and} \quad u_2(B_1) = 50, \quad u_2(B_2) = 50.$$

Then,

$$u_1(B_1) < u_1(B_2) \quad \textit{and} \quad u_2(B_2) = u_2(B_1);$$

thus, agent 1 envies agent 2. So that, $B \notin \mathcal{EF}$*.*

Another example where $\mathcal{EF} \neq \emptyset$ is the example 6, where the allocation $B \in \mathcal{EF}$ while $A, C \notin \mathcal{EF}$ (in both allocations agent 2 envies at least agent 1). In this example, the allocation C is efficient (PO) but not fair (no EF), A is neither efficient (no PO) nor fair (no EF). Finally, allocation B is efficient (PO) and fair (EF), it can be concluded that allocation B is good.

Unfortunately, when considering non divisible resources and additive profits, there are problems where it is **not** possible to find any EF allocation. For example, if $|N| > |M|$, there is always an agent that is envious of the distribution; in this case $\mathcal{EF} = \emptyset$. For this reason, the property EF does not always exist and, for this reason, it is not possible to establish the justice of an allocation. In the next section we study a relaxed version of the envy-free property known as *Envy-free up to one good* [20].

2.3.3 Envy-free up to one good

There are problems where, in all possible allocations, agents are jealous, $\mathcal{EF} = \emptyset$. In this subsection it is studied a property that establishes when the envy between the agents is minimal; in the sense that if exists envy between a pair of agents, it disappears by eliminating a resource. It is worth clarifying that, when we say to eliminate a resource, we do not mean to remove that resource from the problem; this refers to that it is guaranteed that the agents are one resource away from being satisfied with the assigned resources. Next the formal definition.

Definition 7. *An allocation A is said to be envy free up to one good (EF1), if for every pair of agents $i, j \in N$, there is $g \in A_j$ such that $u_i(A_i) \geq u_i(A_j \setminus \{g\})$.*

We will denote by $\mathcal{EFO}$ the set formed by all the EF1 allocations. Unlike envy-free property, the envy-free up to one good property always exists, see [20]; that is, $\mathcal{EFO} \neq \emptyset$. For this reason, we will assume that an allocation is fair if it is in $\mathcal{EFO}$. The figure 2.2 shows that M^N is classified into two sets: $\mathcal{EFO}$ (fair) and $\mathcal{EFO}^c$ (not fair), where $\mathcal{EFO}^c$ are all allocations that are not EF1. That is, $M^N = \mathcal{EFO} \cup \mathcal{EFO}^c$.

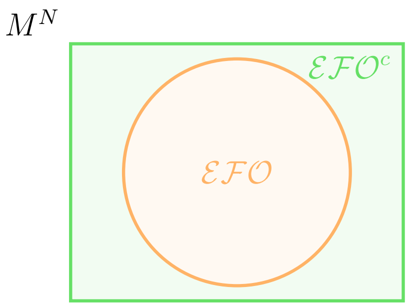


Figure 2.2: $M^N = \mathcal{EFO} \cup \mathcal{EFO}^c$

We will now demonstrate that $\mathcal{EF} \subseteq \mathcal{EFO}$.

Lemma 2. *Let us suppose that all agents consider additive utility functions, be $A \in M^N$. If A is envy free, then A envy free up to one good.*

Proof. *If $\mathcal{EF} = \emptyset$, the result is direct, $\mathcal{EF} \subseteq \mathcal{EFO}$. Suppose that $\mathcal{EF} \neq \emptyset$. Let $A \in \mathcal{EF}$, then for all $i, j \in N$ we have that $u_i(A_i) \geq u_i(A_j)$. If $g \in A_j$,then, for the additivity of u_i,*

$$u_i(A_j) = \sum_{s \in A_j} u_i(\{s\}) \geq \sum_{s \in A_j \setminus \{g\}} u_i(\{s\}) = u_i(A_j \setminus \{g\})$$

So, for any $g \in A_j$

$$u_i(A_i) \geq u_i(A_j \setminus \{g\})$$

such that, $A \in \mathcal{EFO}$. Then, $\mathcal{EF} \subseteq \mathcal{EFO}$.

Here is an example.

Example 8. *Let us consider again the example 7.* $M = \{a,\ b,\ c\}$ *and* $N = \{1,\ 2\}$. *Both agents have additive utility functions given by*

$$u_1(\{a\}) = 20, \quad u_1(\{b\}) = 30, \quad u_1(\{c\}) = 50;$$

$$u_2(\{a\}) = 15, \quad u_2(\{b\}) = 50, \quad u_2(\{c\}) = 35.$$

We know that $A = (A_1, A_2) = (\{a, c\}, \{b\}) \in \mathcal{EF}$*; by the lemma 2,* $A \in \mathcal{EFO}$*.*

On the other hand, the allocation $B = (B_1, B_2) = (\{b\}, \{a, c\}) \notin \mathcal{EF}$*. Let us see that* B *is EF1. Indeed, agent 1 envies agent 2*

$$u_1(B_1) = u_1(\{b\}) = 30 < 70 = u_1(\{a, c\}) = u_1(B_2).$$

We must determine which resource in B_2 *can be suppressed to end envy. If we remove the resource* $\{a\}$ *from* B_2*, the envy does not disappear; as,* $u_1(B_1) = 30$ *and* $u_1(B_2 \setminus \{a\}) = u_1(\{c\}) = 50$*. Now, if we take away the resource* $\{c\}$*, instead of the resource* $\{a\}$*, then the envy disappears; since,* $u_1(B_1) = 30$ *and* $u_1(B_2 \setminus \{c\}) = u_1(\{a\}) = 20$*. Then, the resource that generates envy in the* B *is* $\{c\}$*.*

In the example above, we see two allocations, A and B, which are EF1. The allocation A is EF and in the distribution there is not a resource that generates envy between agents; but, B is not EF and in the distribution there is a resource that produces envy in agent 1. The property EF1, allows us to determine these *problematic resources* n the following examples we will see allocations that involve the properties PO and EF1. Below is an example of a problem where there is an allocation that is neither efficient nor fair.

Example 9. *Consider that* $M = \{a,\ b,\ c,\ d\}$ *and* $N = \{1,\ 2\}$*. Let us suppose that both agents consider additive utility functions given by*

$$u_1(\{a\}) = 32, \quad u_1(\{b\}) = 34, \quad u_1(\{c\}) = 34, u_1(\{d\}) = 2;$$

$$u_2(\{a\}) = 33, \quad u_2(\{b\}) = 33, \quad u_2(\{c\}) = 33, u_2(\{d\}) = 3.$$

The utility functions have the same budget $K = 102$*. Consider the allocations:*

$$A = (A_1, A_2) = (\{a\}, \{b, c, d\}) \quad \textit{and} \quad B = (B_1, B_2) = (\{b\}, \{a, c, d\}).$$

The valuations of each agent on the allocations A *and* B *are given, respectively, by*

$$u_1(A_1) = 32, \quad u_1(A_2) = 70 \quad \textit{and} \quad u_2(A_1) = 33, \quad u_2(A_2) = 69;$$

$$u_1(B_1) = 34, \quad u_1(B_2) = 68 \quad \textit{and} \quad u_2(B_1) = 33, \quad u_2(B_2) = 69.$$

Such that,

$$u_1(A_1) < u_1(A_2) \quad and \quad u_2(A_2) > u_2(A_1);$$
$$u_1(B_1) < u_1(B_2) \quad and \quad u_2(B_2) > u_2(B_1).$$

In both cases agent 1 envies agent 2, $A, B \notin \mathcal{EF}$. *If in A (or in B) we eliminate some resource* α *from* $A_2 = \{b, c, d\}$ *or* α *from* $B_2 = \{a, c, d\}$, *envy is not eliminated; because,* $u_1(A_2 \setminus \{\alpha\}) \geq 36 > 32 = u_1(A_1)$ *and* $u_1(B_2 \setminus \{\alpha\}) \geq 35 > 34 = u_1(B_1)$. *Thus,* $A, B \notin \mathcal{EFO}$.

On the other hand, $A \notin \mathcal{PO}$, *as A is Pareto dominated by B;*

$$u_1(A_1) = 32 < 69 = u_1(B_1) \quad and \quad u_2(A_2) = 34 < 69 = u_2(B_2).$$

Then, $A \notin \mathcal{PO}$.

Here is another example of a problem with an allocation that is EF1 but not PO and another that is EF1 and PO.

Example 10. *Let us consider* $M = \{a,\ b,\ c\}$ *and* $N = \{1,\ 2\}$. *Let us suppose that both agents consider additive utility functions given by*

$$u_1(\{a\}) = 25, \quad u_1(\{b\}) = 35, \quad u_1(\{c\}) = 40;$$
$$u_2(\{a\}) = 30, \quad u_2(\{b\}) = 35, \quad u_2(\{c\}) = 35.$$

Let us consider the allocations:

$$A = (A_1, A_2) = (\{b\}, \{a, c\}) \quad and \quad B = (B_1, B_2) = (\{c\}, \{a, b\}).$$

The valuations of each agent on the allocations A and B are given, respectively, by

$$u_1(A_1) = 35, \quad u_1(A_2) = 65 \quad and \quad u_2(A_1) = 35, \quad u_2(A_2) = 65;$$
$$u_1(B_1) = 40, \quad u_1(B_2) = 60 \quad and \quad u_2(B_1) = 35, \quad u_2(B_2) = 65.$$

Such that,

$$u_1(A_1) < u_1(A_2) \quad and \quad u_2(A_2) > u_2(A_1);$$
$$u_1(B_1) < u_1(B_2) \quad and \quad u_2(B_2) > u_2(B_1).$$

In both cases agent 1 envies agent 2. In A, removing the resource c from A_2 *removes the envy; that is,* $u_1(A_1) = 35 > 25 = u_1(A_2 \setminus \{c\}) = u_1(\{a\})$. *Thus,* $A \in \mathcal{EFO}$. *Whereas, in B when the resource b is deleted from* B_2*, the envy is eliminated,* $u_1(B_1) = 40 > 25 = u_1(B_2 \setminus \{b\}) = u_1(\{a\})$*, with* $B \in \mathcal{EFO}$.

On the other hand, $A \notin \mathcal{PO}$*, as A is Pareto dominated by B;*

$$u_1(A_1) = 35 < 40 = u_1(B_1) \quad \textit{and} \quad u_2(A_2) = 65 = 65 = u_2(B_2).$$

Furthermore, making the corresponding comparisons shows that $B \in \mathcal{PO}$ *(in the example 13 it is shown in a different way that* $B \in \mathcal{PO}$*).*

Now we will see an example of a problem where there is a PO allocation but no EF1.

Example 11. *Let us consider again* $M = \{a,\ b,\ c\}$ *and* $N = \{1,\ 2\}$ *and the additive utility functions given as in the example 10. Let*

$$C = (C_1, C_2) = (M, \emptyset).$$

The valuations of each agent are given by

$$u_1(C_1) = 100, \quad u_1(C_2) = 0;$$
$$u_2(C_1) = 100, \quad u_2(C_2) = 0.$$

Clearly C is PO; because agent 2 gets better whenever agent 1 gets worse. Furthermore, C is an allocation where agent 2 envies agent 1 and C is not EF1, because if you take away a resource $\{\alpha\} \in C_1 = M$ *the envy persists, because* $u_2(C_1 \setminus \{\alpha\}) \geq 65 > 0 = u_2(C_2)$. *Thus, C is PO but not EF1.*

The examples above show that the EF1 and PO properties are independent. To be more precise: in the example 9, a problem was presented where there is an allocation that is neither fair nor efficient,

$$\exists A \in M^N : A \in \mathcal{EFO}^c \cap \mathcal{PO}^c;$$

while a problem where there is an allocation that is fair but not efficient,

$$\exists A \in M^N : A \in \mathcal{EFO} \cap \mathcal{PO}^c,$$

and another allocation that is fair and efficient,

$$\exists A \in M^N : A \in \mathcal{EFO} \cap \mathcal{PO}$$

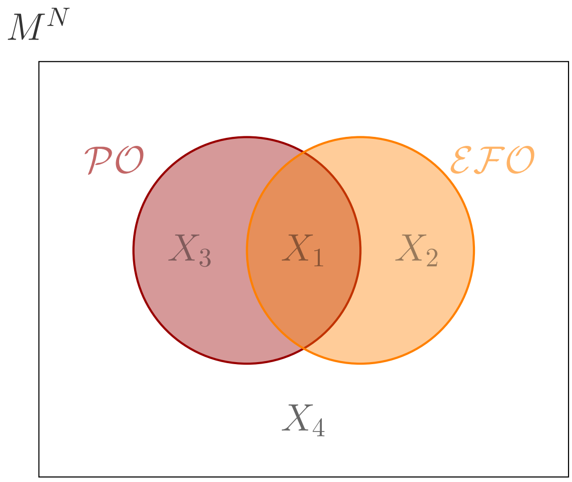


Figure 2.3: Efficient $\mathcal{PO}$ and Fair $\mathcal{EFO}$ allocations classification

. It was shown in the example 10; and using the same problem, in the example 11, an allocation was presented that is not fair but if it is efficient,

$$\exists A \in M^N : A \in \mathcal{EFO}^c \cap \mathcal{PO}.$$

Two important conclusions and one question are drawn from the subsections 2.3.2 and 2.3.3. The first conclusion is that the property of justice is established from the property EF1. This is because there are problems where EF does not exist. The property EF1 always exists, and furthermore, by the lemma 2, every EF allocation is also EF1. The second conclusion is that the properties $EF1$ and PO are independent and we established that an allocation is good if it satisfies EF1 and PO. On the other hand, the set M^N is the union of four separate sets:

$$M^N = X_1 \cup X_2 \cup X_3 \cup X_4$$

where $X_1 = \mathcal{EFO} \cap \mathcal{PO}$ is the set of the fair and efficient (good) allocations; the set $X_2 = \mathcal{EFO} \cap \mathcal{PO}^c$ is formed by the fair but not efficient allocations; while, $X_3 = \mathcal{EFO}^c \cap \mathcal{PO}$ the efficient and not fair allocations; finally, $X_4 = \mathcal{EFO}^c \cap \mathcal{PO}^c$ s 2.3.

The natural question is: given a problem of allocation of resources that are indivisible under additive profits, are there always good allocations? That is, in any problem, $\mathcal{EFO} \cap \mathcal{PO} \neq \emptyset$? The answer to this question will be seen in the next section.

2.4 Social welfare: how to measure?

These methods consist in defining binary relations $\succeq$ over M^N. When individual preferences are given by utility functions, these relations are defined through social welfare functions. Let us look at the formal definition:

Definition 8. *A social welfare function over M^N is any function $SW : M^N \to \mathbb{R}$, where the number $SW(A)$ represents the welfare of allocation A.*

Now, given a SW, the social welfare relation $\succeq$ is defined as follows: for every pair $A, B \in M^N$,

$$A \succeq B \iff SW(A) \geq SW(B) \tag{2.3}$$

where $A \succeq B$ is interpreted as "allocation A has better or equal social welfare than allocation B". The relation $\succeq$ is a total preorder over M^N; where $\succ$ is the strict part and $\sim$ is the indifferent part. In some cases we will use sub-indexes to label the social welfare function that defines it. In this section we will focus on two SW: the utilitarian social welfare and the Nash social welfare [22].

2.4.1 Utilitarian social welfare

In this subsection we will study the utilitarian social welfare, we will see that every allocation that maximizes this social welfare is PO; but, they are not always EF1. Next, the definition of utilitarian social welfare:

Definition 9. *For all $A \in M^N$, the utilitarian social welfare, SW_U, of A is defined as:*

$$SW_U(A) = \sum_{i \in N} u_i(A_i) \tag{2.4}$$

where for all $i \in N$, u_i is a utility function.

We will denote by $\succeq_U$ the social welfare relation that is defined from SW_U, according to the equation (2.3). Let us see an example:

Example 12. *Be $M = \{a,\ b,\ c\}$ and $N = \{1,\ 2\}$ and the additive utility functions given as in the example 10. Let us consider the allocations*

$$A = (\{b\}, \{a, c\}), \quad B = (\{c\}, \{a, b\}), \quad C = (M, \emptyset),\ D = (\{a\}, \{b, c\}).$$

Let's remember that

$$u_1(\{a\}) = 25; \quad u_1(\{b\}) = 35, \quad u_1(\{c\}) = 40$$
$$u_2(\{a, b\}) = 65; \quad u_2(\{a, c\}) = 65; \quad u_2(\{b, c\}) = 70$$

Then,

$$SW_U(A) = u_1(\{b\}) + u_2(\{a, c\}) = 35 + 65 = 100;$$
$$SW_U(B) = u_1(\{c\}) + u_2(\{a, b\}) = 40 + 65 = 105;$$
$$SW_U(C) = u_1(M) + u_2(\emptyset) = 100 + 0 = 100;$$
$$SW_U(D) = u_1(\{a\}) + u_2(\{b, c\}) = 25 + 70 = 95.$$

In such a way that,

$$SW_U(B) > SW_U(A) = SW_U(C) > SW_U(D)$$

then,

$$B \succ_U A \sim_U C \succ_U D$$

In the above example, we observe that B produces more utilitarian social welfare than A, C and D; while, D produces the least social welfare of these allocations. On the other hand, we will denote by $\mathcal{MSW_U}$ the set of all the allocations which maximize the utilitarian social welfare; that is,

$$\mathcal{MSW_U} = \{A \in M^N : SW_U(A) \geq SW_U(B),\ \forall B \in M^N\}$$

Let's observe that $\mathcal{MSW_U} \neq \emptyset$; as, M^N is finite and for all $B \in M^N$ exists A such that $SW_U(A) \geq SW_U(B)$. Then, is classified into those allocations that maximize SW_U and those that do not are described in the figure 2.4.

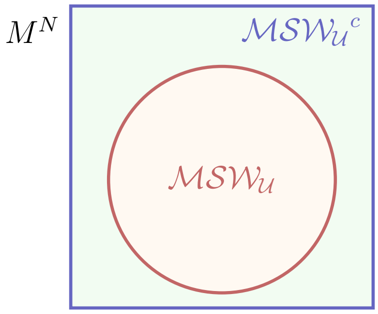


Figure 2.4: $M^N = \mathcal{MSW_U} \cup \mathcal{MSW_U}^c$

The following theorem guarantees that any allocation that maximizes SW_U is efficient.

Theorem 1. *Under additive utility functions, every allocation in MSW_U is Pareto optimal.*

Proof of this theorem is common in the literature, [22, 18]. This theorem is presented again in the subsection 3.2.3, Theorem 3, performing a demonstration using a matrix approach. To illustrate the theorem let us consider the following example.

Example 13. *We have $M = \{a,\ b,\ c\}$ and $N = \{1,\ 2\}$. Each agent have additive utility function defined as in the example 10. For the allocations*

$$A = (\{b\}, \{a, c\}), \quad B = (\{c\}, \{a, b\}), \quad C = (M, \emptyset),\ D = (\{a\}, \{b, c\}).$$

We have that

$$SW_U(B) = 105;\ SW_U(A) = 100;\ SW_U(C) = 100;\ SW_U(D) = 95.$$

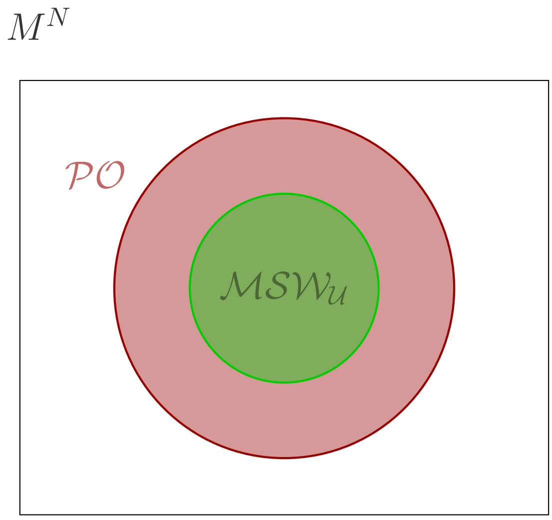


Figure 2.5: $\mathcal{MSW}_{\mathcal{U}} \subseteq \mathcal{PO}$

Such that,

$$SW_U(B) > SW_U(A) = SW_U(C) > SW_U(D).$$

It can be shown that $\mathcal{MSW}_{\mathcal{U}} = \{(\{c\},\{a,b\}),(\{b,c\},\{a\})\}$*. Clearly,* $A, C, D \notin \mathcal{MSW}_{\mathcal{U}}$ *and* $B \in \mathcal{MSW}_{\mathcal{U}}$ *. As a consequence of the theorem 1,* $B \in \mathcal{PO}$ *and, by the examples 10 and 11,* $A \notin \mathcal{PO}$ *and* $C \in \mathcal{PO}$*.*

Theorem 1, establishes that $\mathcal{MSW}_{\mathcal{U}} \subseteq \mathcal{PO}$.In the previous example, it was shown that this contention is strict; as, $C \in \mathcal{PO}$ and $C \notin \mathcal{MSW}_{\mathcal{U}}$. In the figure 2.5 it is described that any allocation that maximizes SW_U is efficient.

On the other hand, allocations that maximize SW_U do not guarantee $EF1$. The following example shows an allocation that maximizes SW_U and is not EF1.

Example 14. *Let us consider* $M = \{a, b, c\}$ *and* $N = \{1, 2, 3\}$*. Each agent has an additive utility function given by*

$$u_1(\{a\}) = 30, \quad u_1(\{b\}) = 30, \quad u_1(\{c\}) = 40;$$
$$u_2(\{a\}) = 40, \quad u_2(\{b\}) = 30, \quad u_2(\{c\}) = 30;$$
$$u_3(\{a\}) = 36, \quad u_3(\{b\}) = 22, \quad u_3(\{c\}) = 22.$$

Let us consider the allocations

$$A = (\{c\}, \{a, b\}, \emptyset) \quad and \quad B = (\{b, c\}, \{a\}, \emptyset)$$

Making the corresponding calculations, we have that $SW_U(A) = 110 = SW_U(B)$ *and* $\mathcal{MSW}_{\mathcal{U}} = \{A, B\}$. *On the other hand, in both allocations agent 3 envies agents 1 and 2. Moreover,* $A, B \notin \mathcal{EFO}$ *because in* A *if we suppress some resource in* $A_2 = \{a, b\}$ *the envy that agent 3 has over agent 2 does not disappear; in the same way, in* B *the envy of agent 3 over agent 1 does not disappear.*

In the example 14, we can see that

$$\mathcal{MSW}_{\mathcal{U}} \cap \mathcal{EFO} = \emptyset$$

that is, an allocation that maximizes utilitarian social welfare does not ensure justice. In contrast, in the examples 10 and 13 we observe that there is an allocation that maximizes the SW_U, consequently it is PO and EF1; that is

$$\mathcal{MSW}_{\mathcal{U}} \cap \mathcal{EFO} \neq \emptyset$$

Then, looking for good allocations in $\mathcal{MSW}_{\mathcal{U}}$ is not always possible.

2.4.2 Nash social welfare

In this subsection we will study Nash social welfare, we will see that any allocation that maximizes Nash social welfare is efficient and fair (good). The following is the definition of Nash social welfare [20]:

Definition 10. *For all* $A \in M^N$, *the Nash social welfare of* A *is defined as:*

$$SW_{Nash}(A) = \prod_{i \in N} u_i(A_i) \tag{2.5}$$

where for all $i \in N$, u_i *is a utility function.*

The social welfare relation that is defined, according to the equation (2.3), from SW_{Nash} will be denoted by $\succeq_{Nash}$. Let us consider again the example 12 and order the allocations using SW_{Nash}.

Example 15. *Be* $M = \{a,\ b,\ c\}$ *and* $N = \{1,\ 2\}$ *and the additive utility functions given as in the example 10:*

$$u_1(\{a\}) = 25, \quad u_1(\{b\}) = 35, \quad u_1(\{c\}) = 40;$$

$$u_2(\{a\}) = 30, \quad u_2(\{b\}) = 35, \quad u_2(\{c\}) = 35.$$

Let us consider the allocations again:

$$A = (\{b\}, \{a, c\}), \quad B = (\{c\}, \{a, b\}), \quad C = (M, \emptyset), \ D = (\{a\}, \{b, c\}).$$

as

$$u_1(\{a\}) = 25; \quad u_1(\{b\}) = 35, \quad u_1(\{c\}) = 40$$
$$u_2(\{a, b\}) = 65; \quad u_2(\{a, c\}) = 65; \quad u_2(\{b, c\}) = 70$$

then

$$SW_{Nash}(A) = u_1(\{b\}) \cdot u_2(\{a, c\}) = 35 \cdot 65 = 2275;$$
$$SW_{Nash}(B) = u_1(\{c\}) \cdot u_2(\{a, b\}) = 40 \cdot 65 = 2600;$$
$$SW_{Nash}(C) = u_1(M) \cdot u_2(\emptyset) = 100 \cdot 0 = 0;$$
$$SW_{Nash}(D) = u_1(\{a\}) \cdot u_2(\{b, c\}) = 25 \cdot 70 = 1750.$$

Such that

$$SW_{Nash}(B) > SW_{Nash}(A) > SW_{Nash}(D) > SW_{Nash}(C)$$

then,

$$B \succ_{Nash} A \succ_{Nash} D \succ_{Nash} C.$$

In the previous example it is observed that the order obtained from the allocations A, B, C and D, when using SW_{Nash}, is different from the one obtained when using SW_U. The set of all allocations that maximize the Nash social welfare is denoted by $\mathcal{MSW_N}$. That is,

$$\mathcal{MSW_N} = \{A \in M^N : SW_{Nash}(A) \geq SW_{Nash}(B),\ \forall B \in M^N\}$$

As M^N is finite, then $\mathcal{MSW_N} \neq \emptyset$. In the example 15, it can be shown that, $\mathcal{MSW_N} = \{B\}$. In the figure 2.6, it is described that M^N is classified into those allocations that maximize SW_{Nash} and those that do not.

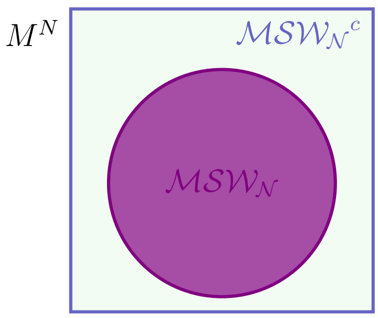


Figure 2.6: $M^N = \mathcal{MSW_N} \cup \mathcal{MSW_N}^c$

The following example shows that, unlike utilitarian social welfare, Nash social welfare is not indifferent to justice.

Example 16. *Consider the example 14. We have $M = \{a, b, c\}$ and $N = \{1, 2, 3\}$. Each agent has an additive utility function, which are given by*

$$u_1(\{a\}) = 30, \quad u_1(\{b\}) = 30, \quad u_1(\{c\}) = 40;$$
$$u_2(\{a\}) = 40, \quad u_2(\{b\}) = 30, \quad u_2(\{c\}) = 30;$$
$$u_3(\{a\}) = 36, \quad u_3(\{b\}) = 22, \quad u_3(\{c\}) = 22.$$

we show that the allocations

$$A = (\{c\}, \{a, b\}), \emptyset) \quad \text{and} \quad B = (\{b, c\}, \{a\}, \emptyset)$$

are PO; as $MSW_U = \{A, B\}$, and also $A, B \notin \mathcal{EFO}$. Now, let's consider $C = (\{c\}, \{b\}, \{a\})$. Observe that

$$SW_{Nash}(A) = 0 = SW_{Nash}(B) \quad \text{and} \quad SW_{Nash}(C) = 4320$$

Moreover, making the corresponding comparisons you have that $C \in \mathcal{MSW}_{\mathcal{N}}$ and also that $C \in \mathcal{EFO}$. Then, C maximizes Nash social welfare, and is good.

At the end of the section 2.3, we asked ourselves, "Do good allocations always exist?" The answer to this question is *yes* and was given by Caragiannis and his colleagues, [20], in the following result.

Theorem 2. *(Caragiannis et al. [20]) Under additive utility functions, an allocation $A \in \mathcal{MSW}_{\mathcal{N}}$ is envy free up to one good and Pareto Optimal.*

In the examples 8 and 13 , we affirm that the allocation B is PO and EF1; in this sense, the previous theorem confirms our affirmation since $B \in \mathcal{MSW}_{\mathcal{N}}$. On the other hand, in general $\mathcal{MSW}_{\mathcal{N}} \neq \emptyset$ and that guarantees the existence of good allocations. Moreover, finding good allocations comes down to finding an allocation that maximizes Nash social welfare. In the figure 2.7, it is shown that any allocation that maximizes the SW_{Nash} is good.

The problem of finding allocations in $\mathcal{MSW}_{\mathcal{N}}$ is a NP hard problem, [16]. The exhaustive search method is the best known and consists of exploring all possible solutions and then determining which allocation maximizes the SW_{Nash}. Another method is the MNW solution described by Caragannis et al. [20]. This solution is performed in two steps: first, the largest set S of agents that generate positive profit simultaneously is found. Then, an allocation of resources among the agents in S is found, so as to maximize the product of their individual profits. Certainly, the exhaustive search method is the most computationally expensive, for more detail related to complexity see [20, 16, 15].

The following conclusions can be drawn from the subsections 2.4.1 and 2.4.2: first, from the individual preferences, preference relations are defined over M^N that measure

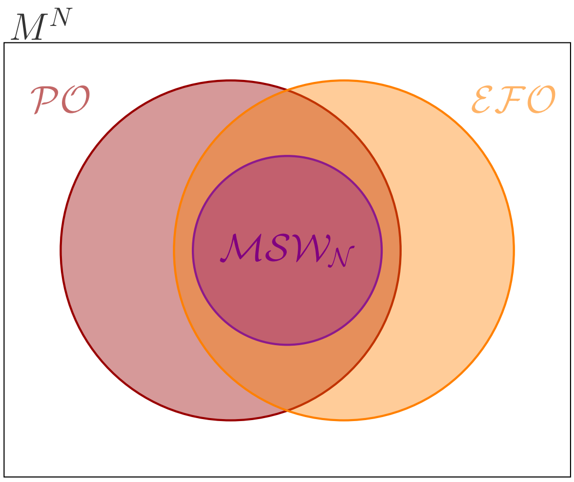


Figure 2.7: $\mathcal{MSW_N} \subseteq \mathcal{PO} \cap \mathcal{EFO}$

social welfare. Since the individual preferences used are additive utility functions, two preference relations are considered, $\succeq_U$ and $\succeq_{Nash}$, both of which are total preorders over M^N, obtained from the utility social welfare functions (SW_U) and Nash social welfare (SW_{Nash}), respectively. Second, the sets of allocations that maximize both SW, $\mathcal{MSW_U}$ and $\mathcal{MSW_N}$ are considered. It was shown that the allocations in $\mathcal{MSW_U}$ are efficient but not necessarily fair; while, thanks to the Theorem 2, the allocations in $\mathcal{MSW_N}$ are efficient and fair, that is, they are good. Third, finding allocations in $\mathcal{MSW_N}$ is still a problem with strong computational complexity.

2.5 Summary

In this chapter we study two desirable properties of an indivisible resource allocation: efficiency and justice. Efficiency is attributed to Pareto's property of optimality; while justice is measured through the property of envy free up to one good. Those allocations that are efficient and fair we call good allocations. It was established that a good allocation always exists.

It was shown that there are ways of ordering allocations through social welfare functions; as particular cases, the utilitarian social welfare function (SW_U) and the Nash social welfare function (SW_{Nash}). The corresponding sets, $\mathcal{MSW_U}$ and $\mathcal{MSW_N}$, formed by the allocations that maximize the functions SW_U and SW_{Nash}, respectively, were considered.

It was observed that efficiency and justice are properties, which in some cases, are present in $\mathcal{MSW_U}$ and $\mathcal{MSW_N}$, to be specific:

1. any allocation in $\mathcal{MSW_N}$ is good, this is because of the Theorem 2 (see figure 2.7);

2. it is guaranteed, through the theorem 1, that any allocation that maximizes the SW_U is efficient (see figure 2.5);

3. Allocations $\mathcal{MSW_U}$ are not always good, because there are allocations in $\mathcal{MSW_U}$ that are not fair. In other words, as we observe in the figures 2.8 and 2.9, depending

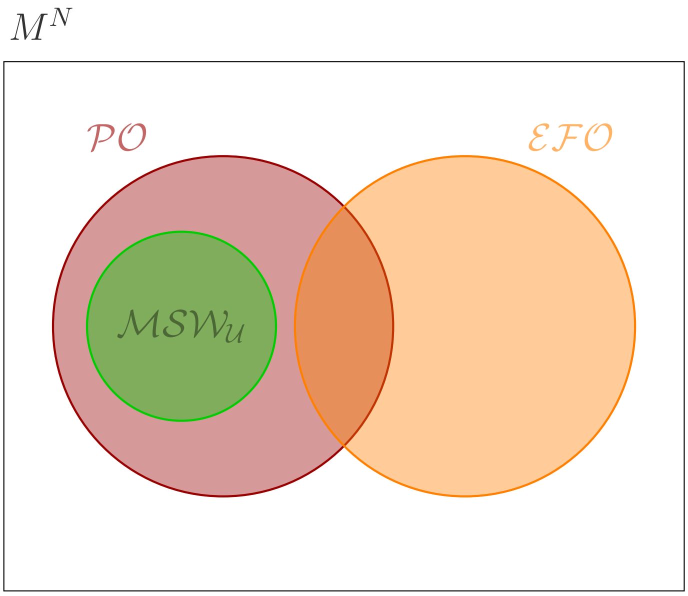


Figure 2.8: There are no allocations in $\mathcal{MSW}_{\mathcal{U}}$ that are good

on the problem, one of the following situations may occur: $\mathcal{MSW}_{\mathcal{U}} \cap \mathcal{EFO} = \emptyset$ or $\mathcal{MSW}_{\mathcal{U}} \cap \mathcal{EFO} \neq \emptyset$.

Now, in the literature, the utilitarian social welfare is the most used parameter to establish social welfare relations. As good allocations always exist, it is of interest to know if a good allocation can be found through SW_U. In the next chapter we will establish a method to find fair, or partially fair, allocations in $\mathcal{MSW}_{\mathcal{U}}$.

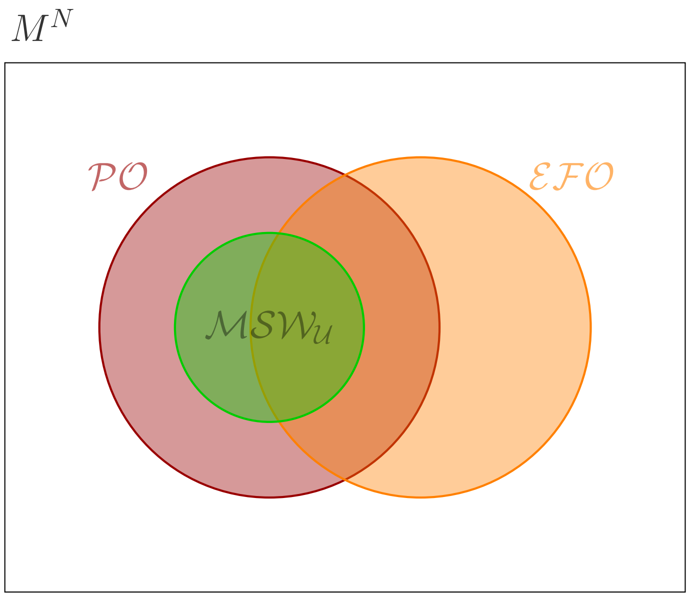


Figure 2.9: There are allocations in $\mathcal{MSW}_{\mathcal{U}}$ that are good

Chapter 3

Proposed matrix approach

This chapter presents a matrix perspective of the theory reviewed in the Chapter 2. This approach facilitates the analysis and obtaining the properties of efficiency and justice in the allocations. To ensure the criterion of efficiency, transitory allocations are presented. It is shown that these allocations are equivalent to $\mathcal{MSW}_{\mathcal{U}}$ and therefore are PO. Within the transitional allocations we want to determine whether there are fair allocations. Depending on the problem of resource allocation, certain conditions are established on a matrix that we will call transition matrix. Another important piece in this chapter will be the utility matrix associated with each allocation. From the utility matrix we will define PO, EF1, SW_U and SW_{Nash}. In the next section we establish the matrix representation of an allocation, the valuation matrix and the utility matrix of an allocation. It is worth noting that throughout this chapter it will be assumed that the utility functions associated to each agent are additive utility functions.

3.1 Matrix perspective

Having a matrix perspective of the problem of resource allocation of indivisible goods will lead to designing various methods to process the agent-resource information and thus present the results in an intuitive way.

In order to formalize the proposed approach, we will establish some notations; let us remember that $|M| = m$ and $|N| = n$. The set of all matrices of size $n \times m$ with entries in the set K, will be denoted by $\mathcal{M}_{n\times m}(K)$. The rows of these matrices represent the agents, and their columns represent the resources. Then, if $\mathcal{A} \in \mathcal{M}_{n\times m}(K)$, the position (i, r), denoted by $[\mathcal{A}]_{ir}$, refers to the agent i and the resource r. Furthermore, $[\mathcal{A}]_{i*}$ denotes the i-th row of $\mathcal{A}$ and $[\mathcal{A}]_{*r}$ denotes the r-th column of $\mathcal{A}$.

3.1.1 Allocation matrix

This subsection defines the matrices that represent all allocations. These types of matrices are binary; for this reason, we identify these matrices as a subset of $\mathcal{M}_{n\times m}(B)$ where

$B = \{0, 1\}$.

Definition 11. *Let be $A = (A_1, A_2, \cdots, A_n) \in M^N$ and $F \in \mathcal{M}_{n\times m}(B)$. We say that matrix F represents allocation A, if*

$$[F]_{ir} = \begin{cases} 1, & \textit{if} \quad r \in A_i \\ 0, & \textit{otherwise.} \end{cases} \tag{3.1}$$

Let us observe that $[F]_{ir} = 1$ is interpreted as: the resource represented in the column r corresponds to agent 1. While $[F]_{ir} = 0$, the agent i does not have the resource represented in the r column.

Example 17. *We want to distribute 4 resources $M = \{a, b, c, d\}$ among 3 agents $N = \{1, 2, 3\}$. Let us consider $F \in \mathcal{M}_{3\times 4}(B)$ given as in the figure 3.1. Note that the rows*

$$\begin{array}{c|c} & \text{Resources} \\ \text{Agents} & F = \begin{bmatrix} 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 \\ \mathbf{1} & 0 & 0 & \mathbf{1} \end{bmatrix} \end{array}$$

Figure 3.1: A matrix representing an allocation.

correspond to the agents and the columns to the resources. The resource "a" is represented in column 1, the "b" in column 2; "c" and "d" in columns 3 and 4, respectively. In row 1 and column 2 is 1, $[F]_{12} = 1$, this is interpreted as: "agent 1 has the resource b"; while $[F]_{33} = 0$ is interpreted as: "agent 3 does not have the resource c". Moreover, matrix F represents the $A = (\{b\}, \{c\}, \{a, d\})$ allocation.

Note that an allocation matrix is a binary matrix of 0's and 1's; because, when working with indivisible resources, each resource must be assigned in its entirety to a single agent and the matrix, which represents an allocation, will have a single 1 per column. To set ideas, see matrix F in the example 17. In the following lemma we will see that for each allocation there is exactly one matrix that represents it.

Remark. *For all $A \in M^N$, there is an unique $F \in \mathcal{M}_{n\times m}(B)$ that represents A. Indeed, let us suppose that there are two different matrices F and G in $\mathcal{M}_{n\times m}(B)$ such that both represent allocation A. As $F \neq G$ there are $r \in M$ and $i \in N$ such that $[F]_{ir} \neq [G]_{ir}$. Then, one of the two does not represent A.*

To end this subsection, it is worth noting that, when we say F is an **allocation matrix** we mean that F is the matrix that represents an allocation. That is to say, F is a allocation matrix of resources M on agents N, if for each column r with $1 \leq r \leq m$, exist only one i with $1 \leq i \leq n$ such that $[F]_{ir} = 1$ and $[F]_{jr} = 0$ for all $j \neq i$.

In the following section we will study the valuation matrix, which is stored by all the information of the utility functions of the agents on each resource.

3.1.2 Valuation matrix

From the utility functions u_i that each agent defines, we define the valuation matrix. This matrix contains the original data of the problem. That is, within a resource allocation problem, the valuation matrix holds the utilities that each agent gives to each resource. As considering additive utilities, we will denote with $\mathbb{R}^*$ the set of non-negative real numbers, $R^* = \{x \in \mathbb{R} : x \geq 0)$. Let us see formally the valuation matrix definition:

Definition 12. *Let u_i be the utility function of agent $i \in N$. A matrix $V \in \mathcal{M}_{n\times m}(\mathbb{R}^*)$ is a valuation matrix if $[V]_{ir} = u_i(r)$ for every agent $i \in N$ and for every resource $r \in M$.*

Example 18. *It is required to distribute 4 resources among 3 agents, $M = \{a, b, c, d\}$ and $N = \{1, 2, 3\}$. The agents define additive utility functions given by:*

$$u_1(\{a\}) = 30,\ u_1(\{b\}) = 40,\ u_1(\{c\}) = 20,\ u_1(\{d\}) = 10;$$

$$u_2(\{a\}) = 30,\ u_2(\{b\}) = 10,\ u_2(\{c\}) = 50,\ u_2(\{d\}) = 10;$$

$$u_3(\{a\}) = 30,\ u_3(\{b\}) = 24,\ u_3(\{c\}) = 16,\ u_3(\{d\}) = 30.$$

Matrix V shown in the figure 3.2. In each row the numerical preferences of each agent

$$\begin{array}{c|c} & \text{Resources} \\ \text{Agents} & V = \begin{bmatrix} 30 & 40 & 20 & 10 \\ 30 & 10 & 50 & 10 \\ 30 & 24 & 16 & 30 \end{bmatrix} \end{array}$$

Figure 3.2: Example of a valuation matrix.

are distributed. The position $[V]_{13}$ represents the utility that agent 1 gives to the resource represented in the 3rd column; that is, $[V]_{13} = 20 = u_1(\{c\})$.

Each resource receives a certain utility from each agent, the sum of all these utilities will be called the **quote of the resource**. In the following lemma we will see that from the valuation matrix is calculated the budget of each agent and the quotation of each resource.

Lemma 3. *If V is the valuation matrix of a resource allocation problem, then*

- *From the sum of all the positions in the i row, you get the budget of the agent i utility.*
- *From the sum of all the positions in the column r, you get the quote of the resource represented in the column r.*

Proof. *Let $i \in N$ and suppose that K_i is the budget of u_i, then*

$$\sum_{r=1}^{m}[V]_{ir} = \sum_{r=1}^{m} u_i(r) = \sum_{r\in M} u_i(r) = K_i.$$

On the other hand, given $r \in M$. If C_r is the quotation of the resource r, then

$$\sum_{i=1}^{n}[V]_{ir} = \sum_{i=1}^{n} u_i(r) = \sum_{r\in M} u_i(r) = C_r.$$

Both the budget of each agent and the quote for each resource can be found using the norm 1 of $\mathbb{R}^k$, $|\cdot|_1$; that is; if $\vec{x} = (x_1, \cdots, x_k) \in \mathbb{R}^k$, then

$$|\vec{x}|_1 = \sum_{i=1}^{k} |x_i|.$$

Example 19. *In the example 18, all agents have the same budget $K = 100$, therefore,*

$$|[V]_{1*}|_1 = \sum_{r=1^4} [V]_{1r} = 30 + 40 + 20 + 10 = 100;$$
$$|[V]_{2*}|_1 = \sum_{r=1^4} [V]_{2r} = 30 + 10 + 50 + 10 = 100;$$
$$|[V]_{3*}|_1 = \sum_{r=1^4} [V]_{3r} = 30 + 24 + 16 + 30 = 100.$$

While the quotations of each resource are given by

$$|[V]_{*1}|_1 = \sum_{i=1}^{3}[V]_{i1} = 30 + 30 + 30 = 90;$$
$$|[V]_{*2}|_1 = \sum_{i=1}^{3}[V]_{i2} = 40 + 10 + 24 = 74;$$
$$|[V]_{*3}|_1 = \sum_{i=1}^{3}[V]_{i3} = 50 + 20 + 16 = 86$$
$$|[V]_{*4}|_1 = \sum_{i=1}^{3}[V]_{i3} = 10 + 10 + 30 = 50$$

then, the resource "a" is the most quoted and "d" is the least quoted.

So far, we can represent two important aspects of a resource allocation problem: The allocations and all the utilities of all the agents on each resource, using matrices. In the next section we will study the utility matrix associated with each allocation.

3.1.3 Utility matrix

In this subsection we will study the utility matrix. Each allocation has an associated utility matrix and it will be used to look for properties on the allocation. Let us see the formal definition:

Definition 13. *Let matrix V be a valuation matrix, and let F be an allocation matrix that represents the allocation A. The utility matrix of F is defined by:*

$$U_F = V \cdot F^T \tag{3.2}$$

where F^T is the transposed allocation matrix of F.

The utility matrix U_F is a square matrix in $\mathcal{M}_{n\times n}(\mathbb{R}^*)$ where the position (i,j) is given by:

$$[U_F]_{ij} = \sum_{k=1}^{m}[V]_{ik} \cdot [F^T]_{kj}$$

Therefore, $[U_F]_{ii}$ is the valuation given by the agent i to the assigned through F; while $[U_F]_{ij}$ is the valuation given by the agent i to the assigned, through F, to the agent j. In other words: if F represents the allocation $A = (A_1, \cdots, A_n)$, then $\forall i.j \in N$

$$[U_F]_{ij} = u_i(A_j)$$

where u_i is the utility function of agent i.

Example 20. *Let us consider $M = \{a, b, c, d\}$, $N = \{1, 2, 3\}$ and the valuation matrix given by*

$$V = \begin{pmatrix} 30 & 40 & 20 & 10 \\ 30 & 10 & 50 & 10 \\ 30 & 24 & 16 & 30 \end{pmatrix}$$

Let F lbe the allocation matrix that represents the allocation $A = (\{b\}, \{c\}, \{a, d\})$; that is,

$$F = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}.$$

The utility matrix of F, U_F is given by

$$U_F = V \cdot F^T = \begin{pmatrix} 30 & 40 & 20 & 10 \\ 30 & 10 & 50 & 10 \\ 30 & 24 & 16 & 30 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 40 & 20 & 40 \\ 10 & 50 & 40 \\ 24 & 16 & 60 \end{pmatrix}$$

Now then, $[U_F]_{11} = 40$ *is the utility that agent 1 gives to the resources assigned through* F*;* $[U_F]_{12} = 20$ *is the utility that agent 1 gives to the resources assigned, through* F*, to agent 2; while,* $[U_F]_{33} = 60$ *is the utility that agent 3 gives to the resources assigned.*

In this section we studied how to describe in matrix form: the preferences of the agents over each resource (valuation matrix), an allocation (allocation matrix), and the valuations that the agents have over the assigned and unassigned resources (utility matrix). In the next section we will study how to find properties on an allocation from its utility matrix.

3.2 Useful properties from utility matrix

In this section we will study how the properties of justice and efficiency are determined from the utility matrix; furthermore, how to calculate the social welfare of an allocation.

3.2.1 Utility matrix and fairness

In the previous chapter we studied two properties of justice: EF and EF1. In this section we will see these properties using the utility matrix. We will begin with the property of envy-free (EF):

Lemma 4. *Let* $A \in M^N$ *and* F *be the allocation matrix that represents it. Then,* A *is envy-free if, and only if,*

$$\forall i, j \in N, [U_F]_{ii} \geq [U_F]_{ij} \tag{3.3}$$

where U_F *is the corresponding utility matrix of* F*.*

Proof. *The proof is obtained directly from the definitions 6 and 13. Be* $i, j \in N$*. Let us assume that* A *is envy-free, by the definition 6,* $u_i(A_i) \geq u_i(A_j)$*. But,* F *represents* A*, then by definition 13,* $[U_F]_{ii} \geq [U_F]_{ij}$*. Thus, the equation 3.3 is fulfilled. Reciprocally, if the equation* (3.3) *is fulfilled, as* F *represents* A*, by the definition 13,* $u_i(A_i) \geq u_i(A_j)$*. Then,* A *is envy-free.*

Let us see the example 20 again.

Example 21. *The valuation matrix is* $V = \begin{pmatrix} 30 & 40 & 20 & 10 \\ 30 & 10 & 50 & 10 \\ 30 & 24 & 16 & 30 \end{pmatrix}$*. For the allocation matrix* $F = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$ *you have that* $U_F = \begin{pmatrix} 40 & 20 & 40 \\ 10 & 50 & 40 \\ 24 & 16 & 60 \end{pmatrix}$*. Let us observe that in each row* *of the matrix U_F, the elements of the diagonal (in red color) are greater or equal than the other entries. Thus, the equation* (3.3) *is fulfilled and, as F represents the allocation $A = (\{b\}, \{c\}, \{a, d\})$, $A \in \mathcal{EF}$.*

The EF property does not always exists in problems with indivisible resources, for that reason, we established that an allocation is fair if it satisfies EF1; that is, $A \in M^N$ is fair if only and only if $A \in \mathcal{EFO}$. The following lemma shows how from the allocation matrix, the represented allocation is fair. The proof is made through a similar reasoning to the demonstration of the lemma 4, using the definitions 7 and 13.

Lemma 5. *Let $A \in M^N$ and F be the allocation matrix that represents it. Then, A is envy-free up to one good if, and only if,*

$$\forall i, j \in N, \exists r \in M \text{ such that} [F]_{jr} = 1 \,\&\, [U_F]_{ii} \geq [U_F]_{ij} - [V]_{ir} \tag{3.4}$$

where U_F is the corresponding utility matrix of F.

Here is an example:

Example 22. *Consider the example 21, whose valuation matrix is given by:*

$$V = \begin{pmatrix} 30 & 40 & 20 & 10 \\ 30 & 10 & 50 & 10 \\ 30 & 24 & 16 & 30 \end{pmatrix}$$

Let us look at the EF1 property in the allocations represented by the following allocation matrices:

$$F = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \quad G = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

the corresponding utility matrices are:

$$U_F = \begin{pmatrix} 40 & 20 & 40 \\ 10 & 50 & 40 \\ 24 & 16 & 60 \end{pmatrix} \quad U_G = \begin{pmatrix} 30 & 50 & 20 \\ 30 & 20 & 50 \\ 30 & 54 & 16 \end{pmatrix} \quad U_H = \begin{pmatrix} 30 & 60 & 10 \\ 30 & 60 & 10 \\ 30 & 40 & 30 \end{pmatrix}$$

As the allocation represented by F is in $\mathcal{EF}$, by the lemma 2, this allocation is $\mathcal{EFO}$. The allocations represented by G and H are not in $\mathcal{EF}$. Let us see that the allocation given by

H is in $\mathcal{EFO}$, while allocation given by G is not. Indeed, is U_H it can be seen that, agents 1 and 3 envy agent 2; while, agent 2 does not evy any agent.

$$30 = [U_H]_{11} < [U_H]_{12} = 60 \quad \& \quad 30 = [U_H]_{33} < [U_H]_{32} = 40$$

Agent 2 is assigned the resources $\{b, c\}$. To eliminate the envy of agent 1, let us consider the resource "b", described in the second column of V; that is, $[V]_{12} = 40$, then

$$30 = [U_H]_{11} > [U_H]_{12} - [V]_{12} = 60 - 40 = 20$$

for $i = 1$ and $j = 2$ taking the resource "b", $r = 2$, the equation (3.4) is fulfilled. In the same way, to eliminate the envy of agent $i = 3$ with agent $j = 3$, the resource "b" is considered again, and you have that

$$30 = [U_H]_{33} > [U_H]_{32} - [V]_{32} = 40 - 24 = 16$$

thus, the equation (3.4) is fulfilled. Then, H is in $\mathcal{EFO}$.

For the allocation matrix G, there is envy among all agents; that is, in each row if U_G, there is an entry that is larger than the element in the diagonal. In particular, agent 3 envies agents 1 and 2. Agent 2 received $\{b, c\}$, represented in columns 2 and 3 of V. As

$$30 = \min\{[U_G]_{32} - [V]_{32}, [U_G]_{32} - [V]_{33}\} > [U_G]_{33} = 16$$

then, for $i = 3$ and $j = 2$, the equation (3.4) is not fulfilled. Thus, the allocation represented by G is not in $\mathcal{EFO}$.

From the diagonal of the utility matrix associated with an allocation, it can be determined whether the allocation is EF. If it is not, the agents (rows) that generate envy are found, and it is verified that the equation (3.4) is fulfilled. In the next subsection, we will see the matrix version of the Pareto dominated and Pareto efficient properties.

3.2.2 Utility matrix and efficiency

Next we will see the matrix version of the properties of Pareto dominant and Pareto optimal. In the following lemma it is observed how using the utility matrices, it is possible to determine if the Pareto dominant property exists between them; this is:

Lemma 6. *Let $A, B \in M^N$ and F, G be two allocation matrices representing A and B, respectively. Then A is Pareto dominated by B if, and only if, for all i, $[U_F]_{ii} \leq [U_G]_{ii}$ and there is at least one j where $[U_F]_{jj} < [U_G]_{jj}$.*

The proof of this lemma is followed by the definitions 4 and 13. Here is an example:

Example 23. *Let us consider 3 resources and 3 agents,* $M = \{a, b, c\}$ *and* $N = \{1, 2, 3\}$*. Let us suppose that the valuation matrix is given by*

$$V = \begin{pmatrix} 20 & 30 & 50 \\ 15 & 50 & 35 \\ 80 & 20 & 50 \end{pmatrix}. \tag{3.5}$$

Let us consider the allocation matrices:

$$F = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad and \quad G = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

with utility matrices

$$U_F = \begin{pmatrix} 50 & 0 & 50 \\ 65 & 0 & 35 \\ 100 & 0 & 0 \end{pmatrix} \quad and \quad U_G = \begin{pmatrix} 50 & 30 & 20 \\ 35 & 50 & 15 \\ 0 & 20 & 80 \end{pmatrix}$$

As

$$[U_G]_{11} = 50 = [U_F]_{11}; \quad [U_G]_{22} = 50 > 0 = [U_F]_{22}; \quad [U_G]_{11} = 80 > 0 = [U_F]_{33}$$

then G *dominates* F

Next we will enunciate the Pareto Optimal criterion from a matrix point of view. The proof is obtained from the definitions 5 and 13.

Lemma 7. *If* F *is the allocation matrix representing* A*, then* A *is Pareto Optimal if, and only if,*

$$\forall F' \in \mathcal{M}_{n\times m}(\mathcal{B})\ [\exists i \in N, [U_{F'}]_{ii} > [U_F]_{ii}] \Rightarrow \left[\exists j \in N, [U_{F'}]_{jj} < [U_F]_{jj}\right]$$

To determine if an allocation is PO using the lemma 7 you need to compare it with all possible matrix allocations; depending on the problem it could be very challenging. However, as we saw in the Theorem 1, every allocation that maximizes utilitarian social welfare is PO. In the next section we will see how to find the utilitarian social welfare of an allocation using the corresponding utility matrix.

3.2.3 Utility matrix and utilitarian social welfare.

In this subsection the relation between the utility matrix and utilitarian social welfare is established. Besides, we will show that every allocation which maximizes SW_U is Pareto optimal.

Lemma 8. *Let $A \in M^N$. If F is an allocation matrix that represent the allocation A, then the utilitarian social welfare of A is given by:*

$$SW_U(A) = trace(U_F) \tag{3.6}$$

where U_F is the utility matrix of F.

Proof. *Let F be the allocation matrix that represents $A = (A_1, \cdots, A_n)$. By definitions 9 and 11, we have that*

$$SW_U(A) = \sum_{i \in N} u_(A_i) = \sum_{i \in N} [U_F]_{ii} = trace(U_F)$$

Example 24. *Let us consider 3 resources and 3 agents, $M = \{a, b, c\}$ and $N = \{1, 2, 3\}$, as in the example 23. The valuation matrix V V is as in (3.5). For allocation matrices:* $F = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ *and* $G = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$ *the utility matrices are:*

$$U_F = \begin{pmatrix} 50 & 0 & 50 \\ 65 & 0 & 35 \\ 100 & 0 & 0 \end{pmatrix} \quad and \quad U_G = \begin{pmatrix} 50 & 30 & 20 \\ 35 & 50 & 15 \\ 0 & 20 & 80 \end{pmatrix}$$

If A and B are represented by F and G, respectively, then

$$SW_U(A) = trace(U_F) = 50 + 0 + 0 = 50 \quad and \quad SW_U(B) = trace(U_G) = 50 + 50 + 80 = 180$$

then, $B \succ_U A$.

Next we will give the demonstration of the theorem1. In the demonstration we will use the utility matrix.

Theorem 3. *Let A be an allocation. If $A \in MSW_U$, then $A \in \mathcal{PO}$.*

Proof. *Let $A \in \mathcal{MSW}_{\mathcal{U}}$ and F the allocation matrix that represents A. In search of a contradiction, suppose that $A \notin \mathcal{PO}$; exists $B \in M^N$ such that A is Pareto dominated by B. Then, for all $i \in N$, $[U_F]_{ii} \leq [U_G]_{ii}$ and exists $j \in N$ such that $[U_F]_{jj} < [U_G]_{jj}$, where G represents B. Since agents' utilities are additive and no negative:*

$$SW_U(B) = trace(U_G) = \sum_{k=1}^{n}[U_G]_{kk} > \sum_{k=1}^{n}[U_F]_{kk} = trace(U_F) = SW_U(A)$$

which is a contradiction since $A \in MSW_U$. So, $A \in \mathcal{PO}$.

3.2.4 Utility matrix and Nash social welfare.

The Nash social welfare function can also be found from the diagonal of the utility matrix; that is:

Lemma 9. *If F is an allocation matrix that represents the allocation A, then the Nash social welfare of A is given by:*

$$SW_{Nash}(A) = prod(U_F) = \prod_{i \in N}[U_F]_{ii} \tag{3.7}$$

where U_F is the utility matrix of F.

Proof. *Let F be the allocation matrix that represents $A = (A_1, \cdots, A_n)$. By definition 9 and 11, we have that*

$$SW_{Nash}(A) = \prod_{i \in N} u_(A_i) = \prod_{i \in N}[U_F]_{ii} = trace(U_F)$$

Once the utility matrix is obtained, the SW_{Nash}. can be calculated directly. To do this you must multiply the elements of the main diagonal of the utility matrix. The following example explains the calculation of this function from a matrix point of view.

Example 25. *In the example 23, the matrices F and G representing the allocations $A = (\{a,b\}, \emptyset, \{c\})$ and $B = (\{c\}, \{b\}, \{a\})$; respectively, have utility matrices:*

$$U_F = \begin{pmatrix} 50 & 0 & 50 \\ 65 & 0 & 35 \\ 100 & 0 & 0 \end{pmatrix} \quad \textit{and} \quad U_G = \begin{pmatrix} 50 & 30 & 20 \\ 35 & 50 & 15 \\ 0 & 20 & 80 \end{pmatrix}.$$

Then,

$$SW_{Nash}(A) = prod(U_F) = 50 \cdot 0 \cdot 0 = 0 \quad \textit{and} \quad SW_{Nash}(B) = prod(U_G) = 50 \cdot 50 \cdot 80 = 200000$$

In this section we study how from the matrix representation, specifically from the associated utility matrix, the properties of justice and efficiency are described; also how the utilitarian and Nash social welfare are obtained. In the next section we propose an explicit way to find all the allocations which maximize the utilitarian social welfare.

3.3 Matrix representations for the maximums utilitarian social welfare

In this section we will define a transitory allocation and we will demonstrate that these allocations represent those which maximize the utilitarian social welfare.

3.3.1 Transition matrix and transitory allocations

We will start by defining the transition matrix associated to the valuation matrix and, from this, we will define the transitory allocations.

Definition 14. *Let matrix $T \in \mathcal{M}_{n\times m}(B)$. T is a transition matrix if $[T]_{ir} = 1$ whenever $[V]_{ir} \in \max\{[V]_{kr} : 1 \leq k \leq n\}$ and $[T]_{ir} = 0$ otherwise, for all agent $i \in N$ and for all resource $r \in M$ where matrix V is a valuation matrix.*

The transition matrix T is a binary matrix. This matrix is constructed based on the valuation matrix V; that is, $[T]_{ir}$ is 1, if $[V]_{ir}$ is a maximum value of the ones in the column r and $[T]_{ir}$ is 0 at any other position in the column. The information provided by the transition matrix is interpreted as *which agent offers more valuation for which resource.* Let us look at the following example:

Example 26. *If* $V = \begin{pmatrix} 30 & 40 & 20 & 10 \\ 30 & 10 & 50 & 10 \\ 30 & 24 & 16 & 30 \end{pmatrix}$ *is the valuation matrix for a resource allocation problem involving 4 resources and 3 agents. The transition matrix T is built from the valuation matrix V. That is, for the resource represented in column 1, all agents maximize it, for all $i = 1, 2, 3$, $[T]_{i1} = 1$; the resource in column 2 is maximized by agents 2 and 3, $[T]_{21} = 1 = [T]_{22}$; while, in column 3, only the second agent maximizes it, $[T]_{23} = 1$; for column 4, only the third agent maximizes it, $[T]_{34} = 1$ Thus,* $T = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$

Using the transition matrix, particular matrix allocations are defined; that is, any allocation matrix F that is constructed from a transition matrix T, will be called a transitory allocation. Let us see the definition:

Definition 15. *An allocation matrix F is transitory if,*

$$\forall r \in M, \exists i \in N, \text{ such that} ([F]_{ir} = 1 \Rightarrow [T]_{ir} = 1) \tag{3.8}$$

where T is the transition matrix.

Example 27. *In the example 26, the transition matrix is given by*

$$T = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

then, the matrices

$$F = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad G = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad H = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

are matrices of transitory allocations.

In the next section we will see that transition allocations are efficient.

3.3.2 An explicit way to identify maximums in utilitarian social welfare

Let us remember that the transition matrix identifies the position of the maximum resource values; that is, it is interested in pointing out the agents that maximize their preferences on the resources, without taking into account the utility value. As we will see below, any transitory allocation is an allocation in $\mathcal{MSW}_{\mathcal{U}}$.

Theorem 4. *Let A be an allocation and F the allocation matrix that represents A. Then, F is transitory if, and only if, $A \in \mathcal{MSW}_{\mathcal{U}}$.*

Proof. *Let $A \in M^N$ be an allocation, let us suppose that F is the allocation matrix that represents A.*

Let us suppose that F is transitory. We want to demonstrate that $trace(U_F) \geq trace(U_G)$, for any allocation matrix G. First let us observe that: for all G,

$$trace(U_G) = \sum_{i \in N} [U_G]_{ii} = \sum_{i \in N} \sum_{k \in M} [V]_{ik}[G^t]_{ki} = \sum_{k \in M} \sum_{i \in N} [V]_{ik}[G^t]_{ki}. \tag{3.9}$$

where V is the valuation matrix. Now, the proof that $trace(U_F) \geq trace(U_G)$ will be done using induction over $m = |M|$, proving that

$$trace(U_F) = \sum_{k \in M} \sum_{i \in N} [V]_{ik}[F^t]_{ki} \geq \sum_{k \in M} \sum_{i \in N} [V]_{ik}[G^t]_{ki} = trace(U_G). \tag{3.10}$$

For $|M| = 1$, there exists $i_1 \in N$ such that $[F]_{i_1 1} = 1$ and for $j \neq i_1$, $[F]_{j1} = 0$. As F is transitory, for all $j \in N$

$$[V]_{i_1 1} \geq [V]_{j1}.$$

On the other hand, there is $i^* \in N$ such that $[G]_{i^* 1} = 1$ and for all $j \neq i^*$, $[G]_{j1} = 0$. In this way,

$$[V]_{i_1 1}[F^t]_{1 i_1} \geq [V]_{i^* 1}[G^t]_{1 i^*}$$

and therefore,

$$\sum_{j \in N, j \neq i_1} [V]_{j1}[F^t]_{1j} = 0 \quad and \quad \sum_{j \in N, j \neq i^*} [V]_{j1}[G^t]_{1j} = 0.$$

Then,

$$\sum_{j \in N} [V]_{j1}[F^t]_{1j} \geq \sum_{j \in N} [V]_{j1}[G^t]$$

So, for $|M| = 1$, $trace(U_F) \geq trace(U_G)$.

Suppose that, the induction hypothesis, for $|M| = m - 1$ the equation (3.10) is fulfilled. Let us see that the equation (3.10) is fulfilled for $|M| = m$.

$$\begin{aligned} trace(U_F) = \sum_{k \in M} \sum_{i \in N} [V]_{ik}[F^t]_{ki} &= \sum_{i \in N} [V]_{i1}[F^t]_{1i} + \sum_{k=2}^{m} \sum_{i \in N} [V]_{ik}[F^t]_{ki} \\ &\geq \sum_{i \in N} [V]_{i1}[F^t]_{1i} + \sum_{k=2}^{m} \sum_{i \in N} [V]_{ik}[G^t]_{ki} \\ &= [V]_{i_1 1}[F^t]_{1 i_1} + \sum_{k=2}^{m} \sum_{i \in N} [V]_{ik}[G^t]_{ki} \\ &\geq [V]_{i_1^* 1}[G^t]_{1 i_1^*} + \sum_{k=2}^{m} \sum_{i \in N} [V]_{ik}[G^t]_{ki} \\ &= \sum_{i \in N} [V]_{i1}[G^t]_{1i} + \sum_{k=2}^{m} \sum_{i \in N} [V]_{ik}[G^t]_{ik} \\ &= \sum_{k \in M} \sum_{i \in N} [V]_{ik}[G^t]_{ki} = trace(U_G) \end{aligned}$$

So, (3.10) is fulfilled for all m. Then, $A \in \mathcal{MSW}_{\mathcal{U}}$.

Let us look at the reciprocal, let us suppose that F is not transitory and show that $A \notin \mathcal{MSW}_{\mathcal{U}}$. By definition 15, there are $r \in M$ and $i_r \in N$ such that $[F]_{i_r r} = 1$ and $[T]_{i_r r} = 0$. Consequently, there is some agent j_r, different to i_r, that maximizes r; that is, there is $j_r \in N$ with $j_r \neq i_r$ such that $[T]_{j_r r} = 1$.

Now, we must find an allocation matrix G such that trace$(U_G) >$ trace(U_F). Let us consider the matrix G as follows: $G = F$ except in the column r and, in column r; $[G]_{j_r r} = 1$ and $[G]_{jr} = 0$ for all $j \neq j_r$. Clearly, G is an allocation matrix (in each column there is exactly a 1). On the other hand, since j_r maximizes r,

$$[V]_{j_r r} > [V]_{i_r r}$$

then

$$[V]_{j_r r}[G^t]_{r j_r} > [V]_{i_r r}[F^t]_{r i_r}.$$

So that,

$$\sum_{k \in M} \sum_{i \in N} [V]_{ik}[G^t]_{ki} > \sum_{k \in M} \sum_{i \in N} [V]_{ik}[F^t]_{ki}.$$

Then, trace$(U_G) >$ trace(U_F). Therefore, $A \notin \mathcal{MSW}_{\mathcal{U}}$.

An immediate result of Theorems 3 and 4 is:

Corollary 1. *Let $A \in M^N$ and T be the allocation matrix that represents A. If T is transitory, then A is Pareto optimal.*

The above result does not state that the problem of searching for efficient allocations reduces the search set to finding all transitory allocations.

Here is an example:

Example 28. *Four resources are distributed among four agents, $|M| = 4$ and $|N| = 4$. Consider the valuation matrix V given by*

$$V = \begin{pmatrix} 20 & 20 & 20 & 20 \\ 25 & 10 & 25 & 20 \\ 25 & 35 & 0 & 20 \\ 25 & 15 & 20 & 20 \end{pmatrix}$$

Each agent has a budget of 80. It is required to distribute the resources in an efficient way. With a brute force search you would have to compare 4^4 allocations. It is now known that *all transient allocations are Pareto Optimal. First we calculate the transition matrix from* V*.*

$$T = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

There are $3 \times 1 \times 1 \times 4 = 12$ *possible transitory allocations, among them,*

$$F = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

this allocation is Pareto efficient; however, it can be seen that the allocation represented by F *is not fair, agents 1 and 4 do not receive any resources; while agent 3 receives three resources.*

In this section we define the transition matrix as the one that identifies the agents that maximize the preferences over each resource. We also established the transitory allocations; demonstrating that these represent all the allocations that maximize the utilitarian social welfare. Finally, transitory matrices allow us to determine in an explicit way an efficient, but not necessarily fair, group of allocations (this is because they represent $\mathcal{MSW}_{\mathcal{U}}$). In the next section we will discriminate $\mathcal{MSW}_{\mathcal{U}}$ and see that in a certain sense we will find a justice among this set, which we will call partial justice.

3.4 Finding a good allocation within the transitory allocations

In the chapter 2 we observe that $\mathcal{MSW}_{\mathcal{U}} \subset \mathcal{PO}$. Unfortunately, in some cases $\mathcal{MSW}_{\mathcal{U}} \cap \mathcal{EFO} = \emptyset$; that is, the allocations which maximize the utilitarian social welfare are not fair allocations. In this section we establish conditions which will allow us to determine allocations in $\mathcal{MSW}_{\mathcal{U}}$ which are fair, $\mathcal{MSW}_{\mathcal{U}} \cap \mathcal{EFO} \neq \emptyset$, or partially fair $\mathcal{MSW}_{\mathcal{U}} \cap \mathcal{EFO} = \emptyset$.

In the next subsection we will see that one way to identify justice in $\mathcal{MSW}_{\mathcal{U}}$ is through SW_{Nash}.

3.4.1 Transitory allocations and envy-free

In this subsection we will consider resource allocation problems with the same number of agents and resources; we will look for conditions that will allow us to determine when $\mathcal{MSW}_{\mathcal{U}} \cap \mathcal{MSW}_{\mathcal{N}} \neq \emptyset$. We will begin by defining the trivial allocations.

Definition 16. *Let $A \in M^N$ with $n = |N| = |M|$ and let F be the allocation matrix representing A. We say that A is a trivial allocation if F is transitory and there is, F^*, a permutation of F such that $trace(F^*) = n$.*

Let us see an example of a problem where there are trivial allocations

Example 29. *Consider again the example 28 where $|M| = |N| = 4$. Their respective valuation and transition matrices are*

$$V = \begin{pmatrix} 20 & 20 & 20 & 20 \\ 25 & 10 & 25 & 20 \\ 25 & 35 & 0 & 20 \\ 25 & 15 & 20 & 20 \end{pmatrix} \qquad T = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

Consider the allocation matrices F and G given by

$$F = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad G = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

1 Clearly, F and G are transitory. Furthermore, it is observed that there is no permutation of F, F^*, such that $trace(F^*) = 4$; therefore, by the Definition 16, F is not trivial. On the other hand, there does exist a permutation of G, G^*, given by

$$G^* = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

such that $trace(G^*) = 4$; therefore G represents a trivial allocation.

In the following example we will see a problem where there are no trivial allocations.

Example 30. *4 resources are distributed among 4 agents,* $|M| = |N| = 4$. *The valuation matrix* V *and its respective transition matrix* T *are given by*

$$V = \begin{pmatrix} 4 & 1 & 2 & 3 \\ 1 & 4 & 3 & 2 \\ 5 & 5 & 0 & 0 \\ 0 & 0 & 5 & 5 \end{pmatrix} \qquad T = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

Note that each agent has a budget of 10. Let F *be the allocation matrix that represents the allocation* $A = (\emptyset, \emptyset, \{1,2\}, \{3,4\})$*; that is,* A *distributes the resources* $\{1,2\}$ *to agent 3, and the resources* $\{3,4\}$ *to agent 4.*

$$F = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

Clearly, F *is transitory; moreover, from the transition matrix it can be seen that* F *is the only transitory matrix. On the other hand, there is no permutation of* F, F^*, *such that* $trace(F^*) = 4$*; therefore, there is no trivial solution to this problem.*

Note that if V is the valuation matrix of an allocation problem, then any permutation of V represents the same problem; but some agents or resources are represented in different positions. In the following theorem we will see some conditions that allow us to determine when there is an allocation that maximizes both SW_U and SW_{Nash}.

Theorem 5. *Let* $V \in \mathcal{M}_{n\times n}(\mathbb{R}^*)$ *be a valuation matrix and* T *be the corresponding associated transition matrix. If it exists,* T^*, *a permutation of* T *such that* $trace(T^*) = n$, *then* $\mathcal{MSW}_\mathcal{U} \cap \mathcal{MSW}_\mathcal{N} \neq \emptyset$.

Proof. *Let* T^* *be a permutation of* T *such that* $trace(T^*) = n$. *Consider* $F = I_n$, *where* I_n *is the identity matrix in* $\mathcal{M}_{n\times n}(B)$. *As* $trace(T^*) = n$, *then all agents maximize at least one resource; besides,* F *gives to each agent a resource that maximizes its preference, then* F *is a transitory allocation matrix. If* A *is the allocation represented by* F, *then, by the Theorem 4,* $A \in \mathcal{MSW}_\mathcal{U}$.

On the other hand, for all $i \in N$*,*

$$[U_F]_{ii} = \sum_{r \in M} [V]_{ir}[F^t]_{ri} = [V]_{ir_i}[F]_{ir_i} = [V]_{ir_i}$$

(each agent i *is assigned only one resource) and also,*

$$[V]_{ir_i} \in \max\{[V]_{jr_i} : j \in N\}.$$

However,

$$\begin{aligned} prod(U_F) = & \ [U_F]_{11} \cdot [U_F]_{22} \cdots [U_F]_{nn} \\ = & \ [V]_{1r_1} \cdot [V]_{2r_2} \cdots [V]_{nr_n} \end{aligned}$$

and each factor is maximum; then, $prod(U_F)$ *is maximum Nash social welfare. So,* $A \in \mathcal{MSW}_{\mathcal{N}}$*. Therefore,* $\mathcal{MSW}_{\mathcal{U}} \cap \mathcal{MSW}_{\mathcal{N}} \neq \emptyset$*.*

The immediate result of the above theorem is that every trivial allocation is efficient and fair.

Corollary 2. *If* A *is a trivial allocation, then* $A \in \mathcal{MSW}_{\mathcal{U}} \cap \mathcal{MSW}_{\mathcal{N}} \neq \emptyset$

The previous corollary assures that all trivial allocations are EF1, this because of the fact that it is a maximum Nash social welfare; however, because of the way trivial allocations are defined and the fact that $|M| = |N|$ the following corollary is obtained:

Corollary 3. *If* A *is a trivial allocation, then* A *is envy-free.*

In summary, for the case in which $|M| = |N| = n$, from the transition matrix T, it is possible to determine whether $\mathcal{MSW}_{\mathcal{U}} \cap \mathcal{EFO} \neq \emptyset$. For them it is possible to verify the existence of a permutation of T such that its trace is equal to n. If so, any trivial allocation is good; moreover, any trivial allocation is envy-free. Otherwise, or in the case of $|M| \neq |N|$, we must look for another strategy to determine whether $\mathcal{MSW}_{\mathcal{U}} \cap \mathcal{EFO} \neq \emptyset$.

3.4.2 Transitory allocations and and the envy-free up to one good

In this subsection we propose other conditions to determine good allocations which maximize utilitarian social welfare. For this, we will define the "oligarch" agents; an agent is oligarch if it maximizes at least one resource.

Definition 17. *Let* V *be the valuation matrix and* T *the associated transition matrix. An agent* $i \in N$ *is said to be oligarchic if there is* $r \in M$ *such that* $T_{ir} = 1$*. Thus, the set of all oligarchic agents, denoted by* S*, is defined as*

$$S = \{i \in N : T_{ir} = 1, \textit{ for some } r \in M\} \tag{3.11}$$

It is worth pointing out that as M and N are finite, then at least one agent must maximize one resource; that is, $S \neq \emptyset$. On the other hand, if an agent has more budget than another one, this one will have the possibility of maximizing more resources. Assuming that all the agents have an equal budget, establishes equal conditions to maximize the resources; thus, all the agents can be oligarchs. If the budget of all agents is $K = 0$, then $S = N$.

Example 31. *In the example 30 we have that the valuation matrix and the transition matrix are given, respectively, by*

$$V = \begin{pmatrix} 4 & 1 & 2 & 3 \\ 1 & 4 & 3 & 2 \\ 5 & 5 & 0 & 0 \\ 0 & 0 & 5 & 5 \end{pmatrix} \quad \textit{and} \quad T = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

Let us remember that there are no trivial allocations. Let us observe that although all agents have the same budget $K = 10$, only agents 3 and 4 are oligarchs; that is, $S = \{3, 4\}$.

Below is the definition of a partially fair allocation.

Definition 18. *Let V be the valuation matrix, $A \in M^N$ and F be the allocation matrix that represents it. We say that A is partially fair (or partially EF1) if*

$$\forall i, j \in S, \exists r \in M \ \ \textit{such that} [F]_{jr} = 1 \,\&\, [U_F]_{ii} \geq [U_F]_{ij} - [V^*]_{ir} \tag{3.12}$$

where S is the set of oligarchic agents and V^* is defined by

$$[V^*]_{ir} = \begin{cases} [V]_{ir}, & \textit{if} [V]_{ir} \in \max\{[V]_{jr} : j \in N\} \\ 0, & \textit{otherwise} \end{cases}$$

Let us note that in the definition, envy-free is established only in the oligarchic agents. In the example 31, we have that $|M| = |N|$ and F, the only transitory allocation, represents a partially fair allocation. In the following example, we will see this property considering a problem with $|M| \neq |N|$ and $\mathcal{MSW}_{\mathcal{U}} \cap \mathcal{EFO} = \emptyset$.

Example 32. *Four resources are distributed among three agents, $|M| = 4$ and $|N| = 3$. The valuation matrix V and its respective transition matrix T are considered:*

$$V = \begin{pmatrix} 350 & 100 & 500 & 50 \\ 350 & 350 & 50 & 250 \\ 100 & 300 & 400 & 200 \end{pmatrix} \quad \textit{and} \quad T = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

In this problem, there are two transitory matrices

$$F = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \textit{and} \quad G = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

which represent the allocations $A = (\{1,3\},\{2,4\},\emptyset)$ *and* $B = (\{3\},\{1,2,4\},\emptyset)$, *respectively. For each allocation matrix the utility matrix is given by:*

$$U_F = \begin{pmatrix} 850 & 150 & 0 \\ 400 & 600 & 0 \\ 500 & 500 & 0 \end{pmatrix} \quad \textit{and} \quad U_G = \begin{pmatrix} 500 & 500 & 0 \\ 50 & 950 & 0 \\ 400 & 600 & 0 \end{pmatrix}$$

then, neither F *nor* G *represent EF1 allocations. This is a problem where* $\mathcal{MSW}_{\mathcal{U}} \cap \mathcal{EFO} = \emptyset$.

Now, each agent has a budget of 1000 and only two agents are oligarchs; that is, $S = \{1,2\}$. *In search of verifying that there is a transitory allocation that is partially fair; let us consider the matrix* V^* *defined from* V *taking the value of those positions where they maximize each resource and in the other entries they take the value of zero:*

$$V^* = \begin{pmatrix} 350 & 0 & 500 & 0 \\ 350 & 350 & 0 & 250 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Note that V^* *is a valuation matrix and* T *is the transition matrix. Then, the transitory allocations in* V *and* V^* *are the same. Using* V^*, *the utility matrices are given by:*

$$U_F^* = \begin{pmatrix} 850 & 0 & 0 \\ 350 & 600 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \textit{and} \quad U_G^* = \begin{pmatrix} 500 & 350 & 0 \\ 0 & 950 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{3.13}$$

Nash social welfare is calculated, using the valuation matrices V *or* V^* *restricted to oligarchic agents,* $SW_{Nash}(\cdot)_S$, *we have that:*

$$SW_{Nash}(A)_S = \prod_{i\in S}([U_F]_{ii}) = 850 \cdot 600 = 510000$$
$$SW_{Nash}(B)_S = \prod_{i\in S}([U_G]_{ii}) = 950 \cdot 500 = 475000$$

and so, in S,

$$A \succ_{Nash} B.$$

Is S, A maximizes SW_{Nash} and is therefore EF1. So, F fulfills the equation (3.12) *and so is partially fair.*

In the previous example we found a problem where there is no transitory allocation that is EF1 ($\mathcal{MSW}_{\mathcal{U}} \cap \mathcal{EFO} = \emptyset$); but, partially fair. The following theorem ensures that there are always transitory allocations that are partially fair.

Theorem 6. *There is an allocation which maximizes the utilitarian social welfare and is partially fair.*

Proof. *Let V be the valuation matrix of the problem of distributing the M resources among the N agents. Let us consider V^* the matrix such that for all $i \in N$ and all $r \in M$,*

$$[V^*]_{ir} = \begin{cases} [V]_{ir}, & si\,[V]_{ir} \in \max\{[V]_{jr} : j \in N\} \\ 0, & otherwise \end{cases}$$

Note that V^* is another valuation matrix. Clearly, the transitory allocations in V and in V^* are the same and, by Theorem 4, they maximize the utilitarian social welfare.

Let us denote with $\mathcal{T}$ the set of all the transitory matrices and with S the set of all the oligarchic agents. Using V^* we classify $\mathcal{T}$ through $\succeq_{Nash}$, restricted to agents in S. Since $\mathcal{T}$ is finite, there is an allocation matrix of G in $\mathcal{T}$ which represents a maximum of SW_{Nash} restricted to S. By Theorem 2, G represents an EF1 allocation restricted to S; therefore, the equation (3.12) *is fulfilled. Thus, G maximizes the social welfare of the utility and is partially fair.*

The above theorem guarantees the existence and does not establish an explicit method for determining partially fair allocations. The existence of partially fair allocations is established using Nash social welfare and the result of Caragannis et al., [20]. In the following example a problem is shown where $|M| \neq |N|$ and $\mathcal{MSW}_{\mathcal{U}} \cap \mathcal{EFO} \neq \emptyset$. That is, an example where all the agents are oligarchs and there are good allocations (a transitory allocation representing an EF1 allocation).

Example 33. *Five resources are distributed among three agents, $|M| = 5$ and $|N| = 3$. Consider the valuation matrix V and its respective transition matrix T given by*

$$V = \begin{pmatrix} 500 & 200 & 50 & 50 & 200 \\ 500 & 100 & 50 & 100 & 250 \\ 500 & 200 & 25 & 100 & 175 \end{pmatrix} \quad and \quad T = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Each agent has a budget of 1000. In this problem there are $3 \times 2 \times 2 \times 2 = 24$ *transitory allocations and in addition all the agents are oligarchs;* $S = N$*.*

The matrix V^* *is considered from the oligarch positions in* T*:*

$$V^* = \begin{pmatrix} 500 & 200 & 50 & 0 & 0 \\ 500 & 0 & 50 & 100 & 250 \\ 500 & 200 & 0 & 100 & 0 \end{pmatrix}$$

We know that a transitory allocation in V*, will also be in* V^**. Let us consider the* F *allocation matrix that represents the allocation* $A = (\{1\}, \{3, 5\}, \{2, 4\})$*. The utility matrix* U_F^**, considering the valuation matrix* V^* *is given by*

$$F = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \quad \textit{and} \quad U_F^* = \begin{pmatrix} 500 & 50 & 200 \\ 500 & 300 & 100 \\ 500 & 0 & 300 \end{pmatrix}$$

From the matrix U_F^* *it is observed that both agent 2 and agent 3 envy agent 1. Let us remember that agent 1 receives only one resource in the distribution, so agent 2 and 3 are one resource away from not feeling envious of agent 1. Therefore,* F *represents an EF1 allocation using* V^* *as the valuation matrix.*

If we now consider the utility matrix U_F*, which is built from* V*:*

$$U_F = \begin{pmatrix} 500 & 250 & 250 \\ 500 & 300 & 200 \\ 500 & 200 & 300 \end{pmatrix}$$

then agents 2 and 3 envy agent 1. But, again, since agent 1 received only one resource, F *represents an allocation that is EF1 in* N*. Therefore,* F *is good.*

The following result states that if all agents are oligarchs, then there is an allocation that maximizes both SW_{Nash} and SW_U over V^*.

Corollary 4. *If* $S = N$*, there is a transitory allocation that maximizes Nash social welfare over* V^**.*

Proof. *If all the agents are oligarchs, then the equations* (3.12) *and* (3.4) *are equal. According to Theorem 6, there is a transitory allocation* F *that satisfies* (3.12)*; but, this property is fulfilled because* F *represents a maximum* SW_{Nash}*. Thus, there is an allocation that maximizes* SW_U *and* SW_{Nash}*.*

In the example 33, a problem was presented where there is a transitory allocation that maximizes social welfare and is EF1. If we calculate the maximum Nash social welfare, we can show that the allocation $A = (\{1\}, \{3,5\}, \{2,4\})$, represented by F, is a maximum SW_{Nash}. In other words, we find a transitory allocation that maximizes Nash social welfare.

In summary, in this subsection we established conditions that determine when in a resource allocation problem, it is possible to find, or not, allocations that maximize the utilitarian social welfare and that are envy-free up to one resource. In the absence of such allocations, we establish that a subset of agents, which we call oligarchs, can achieve justice among themselves. As a by-product, it is established when there is a transitory allocation that maximizes Nash social welfare.

3.5 Summary

This chapter presented the matrix approach to the problem of assigning indivisible resources. Of the matrices studied, two are worth highlighting: The utility matrix of an allocation and the corresponding transition matrix. The utility matrix describes how each agent evaluates the way in which the allocation distributes resources. This matrix provides direct information on the existence, or not, of envy among each pair of agents. On the other hand, the transition matrix identifies the agents that maximize the resources; from this, we define the transitory allocations and we demonstrate in the Theorem 4, that these allocations represent the allocations that maximize the utilitarian social welfare.

Efficiency in transitory matrices always exists, see Corollary 1. Subsequently, conditions are provided on the allocation problem to determine the existence of justice. That is:

1. If $|M| = |N|$ and there is a permutation of T whose trace is n, then there is an envy-free allocation. We call this allocation a trivial allocation.

2. If $|M| = |N|$ and there is no permutation of T whose trace is n or $|M| \neq |N|$. In this case it was established:

   - The oligarchic agents and was considered the set formed by all these agents.
   - Partial fairness, which is the property of EF1 over the oligarchic agents.
   - That there are always transitory allocations that satisfy partial justice (Theorem 6). The demonstration is based on the transition matrix to determine the oligarchic agents and the transitory allocations. Then, using Nash social welfare, restricted to oligarchic agents, the transitory matrices are ordered; those that maximize social welfare will be partially fair.
   - That if all agents are oligarchs, then there is a transitory allocation that satisfies EF1.

Another aspect to highlight is that if all the agents are oligarchs, then $\mathcal{MSW}_{\mathcal{U}} \cap \mathcal{MSW}_{\mathcal{N}} \neq \emptyset$ when considering V^*, see figure 3.3.

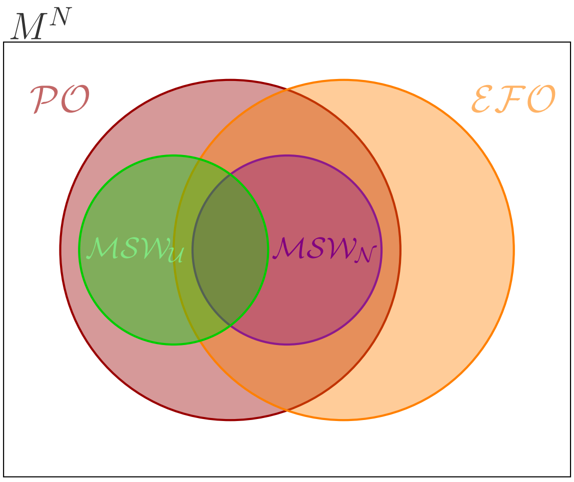


Figure 3.3: If $S = N$, then $\mathcal{MSW}_{\mathcal{U}} \cap \mathcal{MSW}_{\mathcal{N}} \neq \emptyset$ in V^*

Chapter 4

A computational tool

This section presents a new domain specific language (DSL) for solving resource allocation of indivisible goods problems, based on the matrix approach proposed in the Chapter 3. In the current literature, there is no DSL focused on solving this kind of problems. A tool of this type is important, because it would speed up the development and analysis of examples and it will motivate research on resource allocation problems. This section starts with a review of basic concepts related to programming languages, and then explains the general process followed by the proposed DSL, including usage examples.

4.1 Programming languages

A programming language allows humans to type instructions that can be executed by a computer in order to solve specific problems. This instruction set is known as *source code* or just *code*. With the evolution of technology, several programming languages have been developed; the most popular today being C, Java and Python. These languages have a *general purpose* design; this means that they are used to solve any problem that can be computed. However, the time required to properly learn and use these languages is immense, and to deal with new domains requires developing several routines.

A *Domain Specific Language* (DSL) is a specialized programming language whose commands have been designed to support the resolution of a problem or set of problems within a certain domain. A DSL has to maintain a compromise between usability and efficiency; it must be easily understood by humans and it must be executable by a computer. In this sense, the effectiveness of a DSL depends on the clarity with which it defines and tackles problems of the target domain. This specific approach makes DSLs easy to learn, use, and maintain [25].

The programming language interacts with the operative system (OS). There are two types of interactions: directly and indirectly. The direct interaction requires a compiling process. The compiler transforms the source code in a set of instructions, that the operative system can run directly. On the other hand, an interpreted programming language uses a program called interpreter that is responsible for translating each instruction at the time it is executed.

4.1.1 Compiled programs

The life cycle of a compiled program starts from the source code file. The compiler creates a target program. Then, the user runs the target program directly over the operative system. Figure 4.1 shows this life cycle.

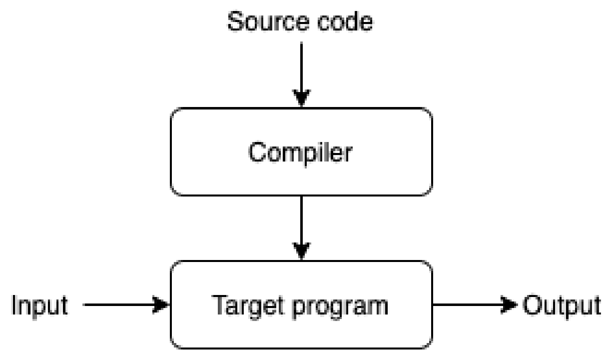


Figure 4.1: Compiled program life cycle.

The compilation stages can be divided into two groups *analysis* and *synthesis*, as can be seen in Figure 4.2. The stages of analysis start receiving a character stream of the source code to compile as input The first step is the *lexical analysis* of the code, where the *scanner* extracts tokens from the input stream. A token corresponds to a specific sequence of characters that have a special meaning within the programming language. At the end of this lexical analysis, the scanner returns a token stream.

The second step corresponds to the *syntactic analysis*, carried out by the parser. The parser follows the token stream and creates a parse tree, sorting the tokens according to constraints of the language, *grammar*. Each node of this tree corresponds to a well-defined operation to execute.

A grammar is a set of constraints that the language imposes in order to determine if the program is syntactically correct. For example, what character set makes up each valid token, how to define identifiers, what are the reserved words, operators, how to reference values, among others. Inside a grammar there is a list of production rules. Each production rule consists of a term followed by its declaration, which shows how the term can be decomposed. The next example shows a production rule for a "power" function, that receives two numbers: first the base, then the reserved word "POW", and followed by the exponent, which is a number.

$\langle power \rangle ::= \langle number \rangle \ \langle POW \rangle \ \langle number \rangle$

This grammar defines how the programming language would recognize a power operation. If the parser identifies the character sequence 2 POW 4, it knows that it corresponds to a declaration of a power operation (2^4).

The third step is the semantic analysis, which means to check the semantic constraints such as: data type, correct definition, that variables have been initialized before the decision process. This step rejects incorrect programs and shows warnings. If the semantic analysis completes without errors, it generates the Abstract Syntax Tree (AST).

The stages of synthesis start creating the intermediate code, which is a flow graph with pseudo instructions. The last compilation steps correspond to code improvements,

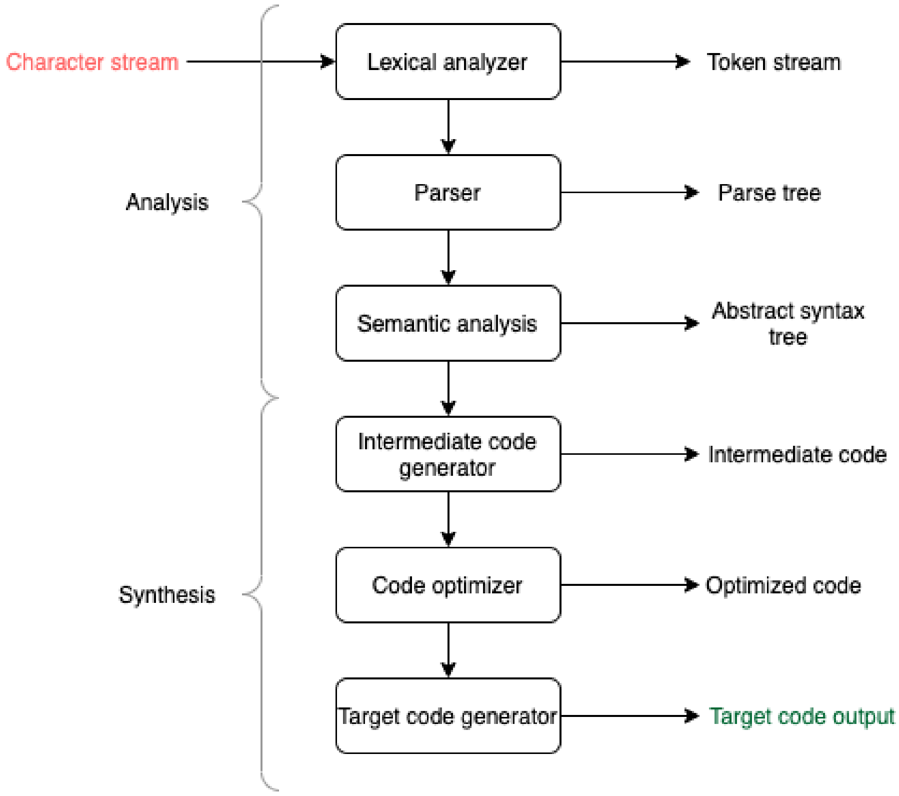


Figure 4.2: Stages of compilation.

both dependent and independent of the architecture of the used computer. Finally, the compiler returns an executable file and disappears without taking part in the execution of the program at any time.

4.1.2 Interpreted programs

As shown in Figure 4.3, an interpreted programming language uses a program called interpreter; which analyzes and executes the source code without generating an executable file. Therefore, the interpreter remains active throughout the execution of the program. The most famous examples of interpreted languages are Python and Ruby.

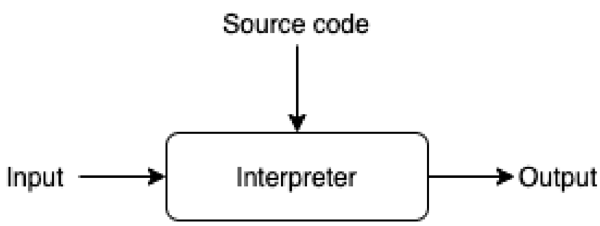


Figure 4.3: Interpreted program life cycle.

The interpretation process shares the same stages of analysis as the compilation process, see Figure 4.4. If the abstract syntax tree is returned without errors, the interpreter is

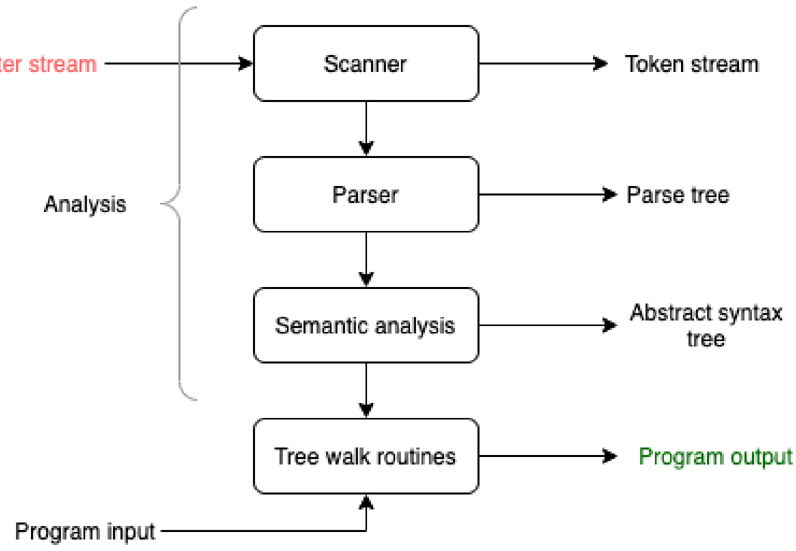


Figure 4.4: Stages of interpretation.

called to execute the instructions and display the results of the program. Then, unlike the compiler, the interpreter visits each node of the AST, performs the operations described in that node and returns the results. In other words, the interpreter is present throughout the program execution process.

In terms of advantages and disadvantages, a compiled program will have a better execution time than an interpreted one, because the interpreter performs the stages of analysis each time the program is executed. The executable file returned by the compilation process depends on the hardware and the operative systems. To run the same executable file in another OS the source code file requires to be re-compiled in the best case scenario; otherwise, it requires to use specific libraries of the target OS. An interpreted program runs directly the source code, then the only requirement is to have the same interpreter installed. Additionally the programmer can execute line by line the source code, and also he can change the values of the variables in running time thanks to the interpreter. This is a useful property in the development of new examples of a target problem, and also for searching logic errors in the program.

In order to facilitate the development of examples of resource allocation, an interpreted DSL will be a valuable computational tool in this research area.

4.2 Proposed DSL

This section proposes a new Domain Specific Language (DSL), called *Resource Allocation Programming Language*, *RAPL* to represent and analyze resource allocation problems of indivisible goods, following the matrix approach developed in Chapter 3. It includes the minimal functionality of a complete DSL.

As explained in chapter 3, resource allocation problems can be well represented using matrices. The matrix approach for this kind of problems is flexible and has interesting mathematical properties that can be exploited in order to find optimal solutions based on efficiency and fairness criteria. However, the programming of such problems using a general-purpose programming language might include an unnecessary level of complexity. For these two main reasons, the results from the proposed new approach and the lack of a DSL for these kinds of problems, a new Domain Specific Language (DSL) is proposed.

This new DSL allows the user to declare the agents' preferences over resources. The language can also manipulate those elements with the proper operations involved in the resource allocation problem. The proposed DSL can measure efficiency criteria such as Pareto optimality, measure fairness criteria such as Envy-free, and represent results using matrices. We hope that the ease of use of this DSL can motivate further research on this topic.

RAPL is an interpreted, high-level, domain-specific programming language developed in Python. Since RAPL is an interpreted language, its instructions are executed directly without a previous compilation of the program into machine-language instructions. The advantage of this feature is that the programs written in RAPL will be easier to execute. RAPL is also dynamically typed and includes type inference, which means that the programmer does not have to specify object types within the code. The programming paradigm supported by RAPL is declarative. This paradigm was chosen in the design because most of the code for this DSL is expected to be logical-mathematical operations.

RAPL follows the standard processing steps of an interpreter. Figure 4.5 shows the overall flow of the process that takes the program written in RAPL, from the character stream to its execution.

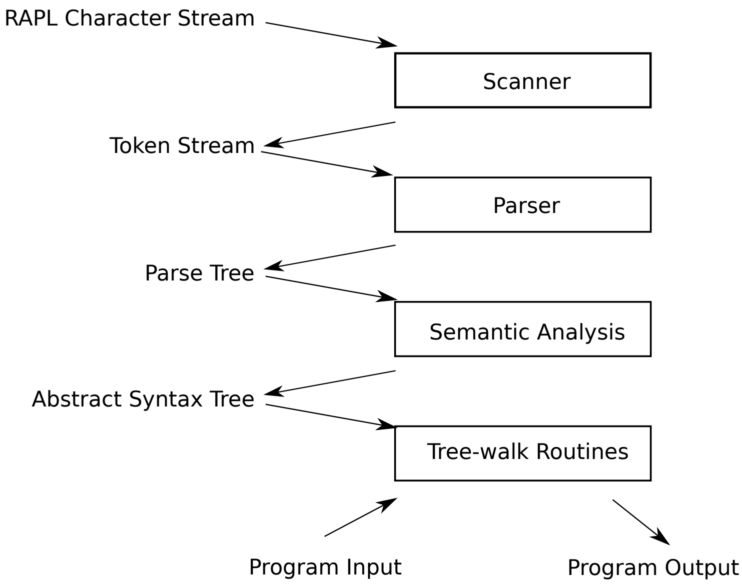


Figure 4.5: RAPL Interpretation Process

4.2.1 RAPL scanner

A RAPL program has to be typed in a Command-Line Interface (CLI). This means that the code to be interpreted is introduced in a shell line by line. The scanner takes a line of code as a sequence of characters and converts it into tokens. A token object consists of the type of the token, a value, a start position, and an end position. The scanner ignores empty spaces and tabs. Then, depending on the characters found in the introduced line of code, a type of token is created. The full list of token types can be found in the Listing 4.1. There are three groups of constants that the scanner uses to classify the tokens *DIGITS, LETTERS* and *LETTERS_DIGITS*. If the current character is a digit the scanner returns a token type of number. If the current character is a letter the scanner checks if it is a keyword or an identifier, and creates the respective token type. The complete list of supported keywords is presented in Listing 4.2. An illegal-character error is returned by the interpreter if it finds an error during this step.

Listing 4.1: Tokens of the RAPL Programming Language

```
TT_INT = ‘TT_INT’
TT_FLOAT = ‘FLOAT’
TT_IDENTIFIER = ‘IDENTIFIER’
TT_KEYWORD = ‘KEYWORD’
TT_PLUS  = ‘PLUS’
TT_MINUS = ‘MINUS’
TT_MUL = ‘MUL’
TT_DIV = ‘DIV’
TT_POW = ‘POW’
TT_EQ = ‘EQ’
TT_AT = ‘AT’
TT_PER = ‘PER’
TT_UTIL = ‘UTIL’
TT_NASH = ‘NASH’
TT_EVAL = ‘EVAL’
TT_PO = ‘PO’
TT_LPAREN = ‘LPAREN’
TT_RPAREN = ‘RPAREN’
TT_LSQUARE = ‘LSQUARE’
TT_RSQUARE = ‘RSQUARE’
TT_LCURLY = ‘LCURLY’
TT_RCURLY = ‘RCURLY’
TT_EE = ‘EE’
TT_NE = ‘NE’
TT_LT = ‘LT’
TT_GT = ‘GT’
TT_LTE = ‘LTE’
TT_GTE = ‘GTE’
TT_COMMA = ‘COMMA’
TT_ARROW = ‘ARROW’
TT_EOF = ‘EOF’
```

Listing 4.2: Keywords of the RAPL Programming Language

```
‘VAR’
‘AND’
‘OR’
‘NOT’
‘IF’
‘THEN’
‘ELIF’
‘ELSE’
‘FOR’
‘TO’
‘STEP’
```

4.2.2 Parser and semantic analysis of RAPL

Once the scanner returns the list of tokens, the parser builds the abstract syntax tree (AST) out of that list. Depending on the encountered tokens, the parser creates a type of node which determines the correct sequence of token types. If there is an error, a syntax error or an expected character error is returned. The preference of interpretation within nodes is determined based on the grammar used in the design of the language.

RAPL grammar

The grammar of the language is presented next and shows the complete grammar of RAPL using the Extended Backus–Naur Form (EBNF). RAPL has four object types so far: *Value, Number, List, Matrix.* A number can be an integer or a float. A list is a sequence of Numbers, separated by commas ‘,’ and enclosed by brackets. A matrix is a sequence of Lists, separated by commas ‘,’ and enclosed by curly brackets.

⟨*expr*⟩ ::= ⟨*KEYWORD:VAR*⟩ ⟨*IDENTIFIER*⟩ ⟨*EQ*⟩ ⟨*expr*⟩
  | ⟨*comp-expr*⟩ ((⟨*KEYWORD:AND*⟩|⟨*KEYWORD:OR*⟩)
  ⟨*comp-expr*⟩)*

⟨*comp-expr*⟩ ::= ⟨*NOT*⟩ ⟨*comp-expr*⟩
  | ⟨*arith-expr*⟩ ((⟨*EE*⟩|⟨*LT*⟩|⟨*GT*⟩|⟨*LTE*⟩|⟨*GTE*⟩)
  ⟨*arith-expr*⟩)*

⟨*arith-exprr*⟩ ::= ⟨*term*⟩ ((⟨*PLUS*⟩|⟨*MINUS*⟩) ⟨*term*⟩)*

⟨*term*⟩ ::= ⟨*factor*⟩ ((⟨*MUL*⟩|⟨*DIV*⟩) ⟨*factor*⟩)*

⟨*factor*⟩ ::= ⟨*INT*⟩|⟨*FLOAT*⟩
  | ⟨*power*⟩

⟨*power*⟩ ::= ⟨*atom*⟩(⟨*POW*⟩ ⟨*factor*⟩)

⟨*atom*⟩ ::= (⟨*PLUS*⟩|⟨*MINUS*⟩) ⟨*factor*⟩
  | ⟨*LPAREN*⟩ ⟨*expr*⟩ ⟨*RPAREN*⟩
  | ⟨*matrix-expr*⟩
  | ⟨*list-expr*⟩
  | ⟨*if-expr*⟩
  | ⟨*for-expr*⟩
  | ⟨*while-expr*⟩

⟨*list-expr*⟩ ::= ⟨*LSQUARE*⟩ (⟨*expr*⟩ (⟨*COMMA*⟩ ⟨*expr*⟩)*)?
  ⟨*RSQUARE*⟩

⟨*matrix-expr*⟩ ::= ⟨*LSQUARE*⟩ (⟨*list-expr*⟩)+ ⟨*RSQUARE*⟩

⟨*if-expr*⟩ ::= ⟨*KEYWORD:IF*⟩ ⟨*expr*⟩ ⟨*KEYWORD:THEN*⟩
  ⟨*expr*⟩ (⟨*KEYWORD:ELIF*⟩ ⟨*expr*⟩ ⟨*KEYWORD:THEN*⟩ ⟨*expr*⟩)* (⟨*KEYWORD:ELSE*⟩
  ⟨*expr*⟩)?

$\langle$*for-expr*$\rangle$ ::= $\langle$*KEYWORD:FOR*$\rangle$ $\langle$*IDENTIFIER*$\rangle$ $\langle$*EQ*$\rangle$ $\langle$*expr*$\rangle$
$\langle$*KEYWORD:TO*$\rangle$ $\langle$*expr*$\rangle$ ($\langle$*KEYWORD:STEP*$\rangle$ $\langle$*expr*$\rangle$)? $\langle$*KEYWORD:THEN*$\rangle$ $\langle$*expr*$\rangle$

$\langle$*while-expr*$\rangle$ ::= $\langle$*KEYWORD:WHILE*$\rangle$ $\langle$*expr*$\rangle$ $\langle$*KEYWORD:THEN*$\rangle$
$\langle$*expr*$\rangle$

RAPL operators

RAPL supports basic mathematical binary operations. The binary operators for two NUMBERS are: $+, -, /, *,$ ^. These operators represent summation, subtraction, division, multiplication and exponentiation respectively. Parentheses are also supported and can be used to change the priority on calculations.

Unary operators are also supported by RAPL. Thanks to this, expressions like *-5* make sense. Other unary operators are also available for matrices operations and are explained below.

RAPL supports (1) comparison operators, such as $>, <, >=, <=$ *and* $==$, (2) logical operators, such as *AND, OR and NOT* and (3) conditional and repetitive statements, such as those declared with the keywords *IF, ELIF, ELSE, FOR.*

For resource allocation problems purpose, in RAPL, the matrix operations that the DSL supports are:

Unary matrix operations

- / for matrix index value,
- @ for matrix transpose,
- % to get transition matrix from a valuation matrix,
- # to get utilitarian social welfare from utility matrix,
- $\sim$ to get Nash social welfare from utility matrix.

Binary matrix operations

- $*$ for standard matrix multiplication,
- $+$ for standard matrix addition,
- $-$ for standard matrix subtraction,
- **:** to get utility matrix from a valuation matrix and an allocation.

Obtaining the AST

To build the AST, the parser takes the list of tokens returned by the lexer and reads it to identify if the sequence corresponds to an specific node. The parser is a Python class with different methods like *matrix_expr, list_expr, if_expr, for_expr, while_expr, atom, power, factor, term, arith_expr, comp_expr* and *expr*, which the parser uses to identify specific nodes within a token stream.

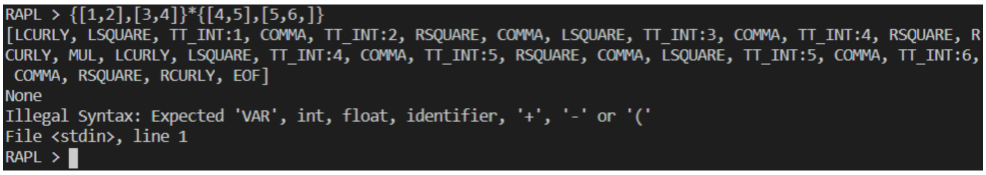
There are different nodes in RAPL: *NumbreNode, ListNode, MatrixNode, VarAccessNode, VarAssignNode, BinOpNode, UnaryOpNode, IfNode, ForNode* and *WhileNode*. Each node is a Python class that stores its position start, position end and its corresponding methods depending on the Node.

So the parser uses its implemented methods to validate that the token stream follows an specific understandable sequence, if so, the method returns a node based on that sequence. If there are no errors during this process, the pareser then returns the AST to be interpreted. In figure 4.6, image *(a)* shows a list of tokens and image *(b)* shows a token list followed by a AST, which consists on a single BinOpNode.

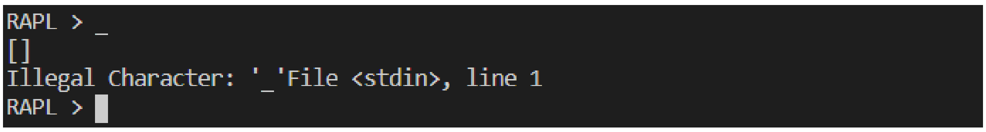
4.2.3 RAPL interpreter

If the parser returns the AST with no errors, the interpreter is called to execute operations and show results. The AST is a map of nodes that the interpreter has to visit in a specific order. The interpreter then visits a node and tries to return the desired result from every node. The interpreter returns a run-time error if there is an error during this stage. For example, if the interpreter visits a binary operation node, it is expected to have a left node, an operator node, and a right node. Once the entire AST has been visited by the interpreter with no errors, the result is shown on the screen.

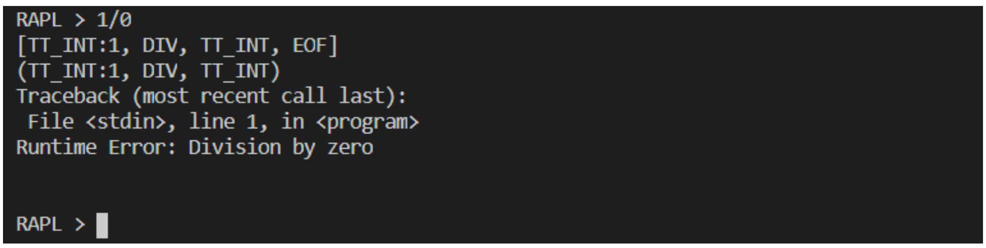
Fig. 4.6 shows examples of the possible errors that are detected by RAPL through the overall flow.



(a) Parser error



(b) Scanner error



(c) Interpreter error

Figure 4.6: Errors detected in the interpretation process by RAPL.

4.2.4 Examples using RAPL

RAPL will then be used to analyze some of the problems reviewed throughout chapter 3. A few lines of code will be used to obtain the matrices proposed in this research in order to conduct an analysis of fairness and efficiency criterion of an allocation.

Example 34. *Figures 4.7, 4.8 and 4.9, shows how to perform the analysis carried out in the example 29. With the help of RAPL, two allocations will be analyzed: matrix F which represents an allocation that maximizes the utilitarian social welfare, and the matrix G which represents an allocation that maximizes the Nash social welfare of a reduced problem.*

First, the valuation matrix V is defined, which stores the valuations that each agent gives to each resource in a resource allocation problem between 5 resources and 3 agents. Then, with the help of the operator %, the transition matrix T is obtained from V.

```
RAPL > VAR V = {[500, 200, 50, 50, 200],[500, 100, 50, 100, 250],[500, 200, 25, 100, 175]}
{[500 200 50 50 200]
 [500 100 50 100 250]
 [500 200 25 100 175]
}
RAPL > %V
{[1 1 1 0 0]
 [1 0 1 1 1]
 [1 1 0 1 0]
}
```

Figure 4.7: Declaring valuation matrix and its transition matrix using % operator.

Next, the transitory matrices F and G are defined. The variables F and G are then replaced by the respective transposed allocation matrix.

```
RAPL > VAR F = {[1, 1, 1, 0, 0],[0, 0, 0, 1, 1],[0, 0, 0, 0, 0]}
{[1 1 1 0 0]
 [0 0 0 1 1]
 [0 0 0 0 0]
}
RAPL > VAR G = {[1, 0, 0, 0, 0],[0, 0, 1, 0, 1],[0, 1, 0, 1, 0]}
{[1 0 0 0 0]
 [0 0 1 0 1]
 [0 1 0 1 0]
}
RAPL > VAR F = @F
{[1 0 0]
 [1 0 0]
 [1 0 0]
 [0 1 0]
 [0 1 0]
}
RAPL > VAR G = @G
{[1 0 0]
 [0 0 1]
 [0 1 0]
 [0 0 1]
 [0 1 0]
}
```

Figure 4.8: Declaring F and G allocation matrices and its transposed form using @ operator.

Finally, as defined in 13, the utility matrices U_F and U_G are obtained by multiplying the valuation matrix with its respective transposed allocation matrix.

```
RAPL > VAR UF = V*F
{[750 250 0]
 [650 350 0]
 [725 275 0]
}
RAPL > VAR UG = V*G
{[500 250 250]
 [500 300 200]
 [500 200 300]
}
```

Figure 4.9: Computing utility matrices by multiplying corresponding valuation and allocation matrices.

Alternatively, as shown in the figure 4.10, the utility matrices U_F and U_G can be calculated directly with the ':' operator.

Finally, and as shown in the figure 4.11, once the utility matrices U_F and U_G are obtained, we proceed to calculate the values of the utilitarian social welfare with the operator '#' and the values of Nash social welfare with the operator '$\sim$'.

```
RAPL > VAR UF = V:F
{[500 500 500 200 200]
 [500 500 500 100 100]
 [500 500 500 200 200]
}
RAPL > VAR UG = V:G
{[500 50 200 50 200]
 [500 50 100 50 100]
 [500 25 200 25 200]
}
RAPL >
```

Figure 4.10: Usage of operator ':'.

```
}
RAPL > # UF
1100
RAPL > ~ UF
0
RAPL > # UG
1100
RAPL > ~ UG
45000000
RAPL >
```

Figure 4.11: Get SW_U and SW_{Nash} using '#' and '∼' operators.

Our results show that RAPL is able to describe the allocation problem very succinctly with a clean syntax. It is important to note that the calculation of the allocations is done with few lines of code. This syntax avoids the clutter that general-purpose programming languages produce when trying to represent and solve this sort of problems. Therefore, this tool will help in the development of this research area.

4.3 Summary

This Chapter presented the interpreted DSL, called RAPL, which is focused on the problem of resource allocation of indivisible goods. It follows the matrix approach presented in Chapter 3. The description of RAPL included details of the interpreter, grammar, operators, and the different types of errors. The examples proposed showed the succinctly and clean syntax of RAPL and its specialized operators. In conclusion, it is important to highlight that RAPL allows us to represent and solve any problem of resource allocation. This computational tool will facilitate the study of new examples and properties of interest in the research of the resource allocation problem.

Chapter 5

Conclusions and perspectives

In this paper, a matrix approach to the problem of allocating indivisible resources was presented. The traditional approach was shown, Chapter 2, which is based on a set theory perspective; explaining the most important aspects of this field and the existing problem of finding efficient and fair allocations, considering indivisible resources and additive utility functions. These aspects were presented from a matrix approach in Chapter 3. This new vision facilitates analysis to identify the properties of efficiency and fairness in allocations.

From the matrices defined in section 3.1, interesting aspects are presented. That is: from the allocation matrix, the distribution of resources of an allocation is described; from the valuation matrix, a large part of the initial data of an allocation problem is described, the transition matrix is defined and, as a consequence, the transitory allocations; while, from the utility matrix of an allocation, it is obtained the value of the utilitarian social welfare, the value of the Nash social welfare, and it is established the existence, or not, of envy among a couple of agents.

The main theoretical results obtained in this work, under the assumption of additive profits, are:

1. a matrix characterization for the allocations which maximize the utilitarian social welfare;

2. a matrix representation for a class of efficient allocations;

3. if $|M| = |N| = n$ and there is a permutation of the transition matrix whose trace is n, then there is an efficient allocation that is envy-free;

4. if $|M| = |N|$ and there is no permutation of the transition matrix whose trace is n, then there is an efficient allocation that is partially fair;

5. if $|M| \neq |N|$, then there is an efficient allocation that is partially fair;

6. if all agents are oligarchs, then there is an efficient and partially fair allocation;

7. if all the agents are oligarchs, then $\mathcal{MSW}_{\mathcal{U}} \cap \mathcal{MSW}_{\mathcal{N}} \neq \emptyset$ in V^*.

On the other hand, this matrix approach motivated the development of a domain specific language called RAPL. This DSL is in its beginnings, but it can already perform the basic matrix operations necessary to obtain all the arrays defined in this work, with few lines of code .

Perspectives

As future work related to this research is wanted to:

1. find a matrix characterization to identify those allocations, or at least one, that will maximize Nash social welfare;
2. find a matrix characterization to identify the property of envy-free up to one good;
3. study other properties of justice such as proportionality and maximin share (MMS);
4. continue to develop RALP.

Bibliography

[1] J. Robertson and W. Webb, *Cake-cutting algorithms: Be fair if you can.* CRC Press, 1998.

[2] J. Nash, "The bargaining problem," *Econometrica*, vol. 18, no. 2, pp. 155–162, 1950. [Online]. Available: https://EconPapers.repec.org/RePEc:ecm:emetrp:v:18:y:1950:i:2:p:155-162

[3] J. von Neumann and O. Morgenstern, *Theory of games and economic behavior.* Princeton University Press, 1947.

[4] H. Steinhaus, "The problem of fair division," *Econometrica*, vol. 16, pp. 101–104, 1948.

[5] ——, "Sur la division pragmatique," *Econometrica*, vol. 17, pp. 315–319, 1949.

[6] S. J. Brams and A. D. Taylor, "An envy-free cake division protocol," *The American Mathematical Monthly*, vol. 102, pp. 9–18, 1995.

[7] S. Brams and A. Taylor, *Fair division - from cake-cutting to dispute resolution.* Cambridge University Press, 1996.

[8] A. D. T. Steven J. Brams and W. S. Zwicker, "A moving-knife solution to the four-person envy-free cake division problem," *Proceedings of The American Mathematical Society*, vol. 125, pp. 547–554, 1997.

[9] L. Shapley and H. Scarf, "On cores and indivisibility," *Journal of Mathematical Economics*, vol. 1, pp. 23–37, 1973.

[10] A. Roth and M. Oliveira Sotomayor, "Two-sided matching: A study in game-theoretic modeling and analysis," *Games and Economic Behavior*, vol. 1, pp. 161–165, 1992.

[11] P. Ferreira, F. dos Santos, A. Bazzan, D. Epstein, and S. Waskow, "Robocup rescue as multiagent task allocation among teams: experiments with task interdependencies," *Autonomous Agents and Multi-Agent Systems*, vol. 20, pp. 421–443, 2010.

[12] P. Sousa and C. Ramos, "A distributed architecture and negotiation protocol for scheduling in manufacturing systems." *Computers in Industry*, vol. 38, no. 2, pp. 103 – 113, 1999.

[13] S. Jackson, J. R. Wilson, and B. L. MacCarthy, "A new model of scheduling in manufacturing: tasks, roles, and monitoring." *Human Factors*, vol. 46, no. 3, p. 533–550, 2004.

[14] D. Abraham, R. Irving, and D. Manlove, "Two algorithms for the student-project allocation problem," *Journal of Discrete Algorithms*, vol. 5, no. 1, pp. 73–90, 2007.

[15] B. de Keijzer, S. Bouveret, T. Klos, and Y. Zhang, "On the complexity of efficiency and envy-freeness in fair division of indivisible goods with additive preferences," in *Algorithmic Decision Theory*, F. Rossi and A. Tsoukias, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 98–110.

[16] A. Darmann and J. Schauer, "Maximizing nash product social welfare in allocating indivisible goods," *European Journal of Operational Research*, vol. 247, no. 2, pp. 548 – 559, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S037722171500483X

[17] H. Aziz, P. Biró, J. Lang, J. Lesca, and J. Monnot, "Optimal reallocation under additive and ordinal preferences." in *In Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems, AAMAS '16*, A. S. K. J. Arrow and K. Suzumura, Eds. Elsevier, 2016, pp. 402–4010.

[18] U. Endriss, N. Maudet, F. Sadri, and F. Toni, "Negotiating socially optimal allocations of resources," *J. Artif. Intell. Res.*, vol. 25, pp. 315–348, 2006. [Online]. Available: https://doi.org/10.1613/jair.1870

[19] Y. Chevaleyre, U. Endriss, S. Estivie, and N. Maudet, "Reaching envy-free states in distributed negotiation settings," in *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, 2007, pp. 1239–1244. [Online]. Available: http://ijcai.org/Proceedings/07/Papers/200.pdf

[20] I. Caragiannis, D. Kurokawa, H. Moulin, A. D. Procaccia, N. Shah, and J. Wang, "The unreasonable fairness of maximum nash welfare," *ACM Transactions on Economics and Computation (TEAC)*, vol. 7, no. 3, pp. 1–32, 2019.

[21] Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. A. Padget, S. Phelps, J. A. Rodríguez-Aguilar, and P. Sousa, "Issues in multiagent resource allocation," *Informatica (Slovenia)*, vol. 30, no. 1, pp. 3–31, 2006.

[22] U. Endriss, "Lecture notes on fair division," *CoRR*, vol. abs/1806.04234, 2010. [Online]. Available: http://arxiv.org/abs/1806.04234

[23] Y. Chevaleyre, U. Endriss, and N. Maudet, "Distributed fair allocation of indivisible goods," *Artif. Intell.*, vol. 242, pp. 1–22, 2017. [Online]. Available: https://doi.org/10.1016/j.artint.2016.09.005

[24] F. Camacho, G. Chacón, and R. P. Peréz, "A qualitative framework for resource allocation," *Revista Ibérica de Sistemas e Tecnologias de Informação*, vol. E19, pp. 121–133, 2019.

[25] M. Fowler and R. Parsons, *Domain Specific Languages.* Addison-Wesley Professional, 2010.