An Agent-Based Approach for Distributed Resource Allocations

Antoine Nongaillard

A thesis

in the Department

of

Computer Science & Software Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Doctor of Philosophy at

Concordia University, Montreal, Quebec

And Université Lille 1, Villeneuve d'Ascq, France

In accordance with the co-tutelle agreement between Quebec and France

December 2009

Your file Votre référence
ISBN: 978-0-494-67358-4
Our file Notre référence
ISBN: 978-0-494-67358-4

ABSTRACT

An Agent-Based Approach for Distributed Resource Allocations

Antoine Nongaillard, PhD

Concordia University

Université Lille 1, 2009

Resource allocation problems have been widely studied according to various scenarios in literature. In such problems, a set of resources must be allocated to a set of agents, according to their own preferences. Self-organization issues in telecommunication, scheduling problems or supply chain management problems can be modeled using resource allocation problems.

Such problems are usually solved by means of centralized techniques, where an omniscient entity determines how to optimally allocate resources. However, these solving methods are not well-adapted for applications where privacy is required. Moreover, several assumptions made are not always plausible, which may prevent their use in practice, especially in the context of agent societies. For instance, dynamic applications require adaptive solving processes, which can handle the evolution of initial data. Such techniques never consider restricted communication possibilities whereas many applications are based on them. For instance, in peer-to-peer networks, a peer can only communicate with a small subset of the systems.

In this thesis, we focus on distributed methods to solve resource allocation problems. Initial allocation evolves step by step thanks to local agent negotiations. We seek to provide agent behaviors leading negotiation processes to socially optimal allocations. In this work, resulting resource allocations can be viewed as emergent phenomena. We also identify parameters favoring the negotiation efficiency. We provide the negotiation settings to use when four different social welfare notions are considered. The original method proposed in this thesis is adaptive, anytime and can handle any restriction on agent communication possibilities.

Keywords: Distributed problem solving, individual based reasoning, social networks, social welfare, information privacy.

RÉSUMÉ

Une Approche Centrée Individu de l'Allocation de Ressources Distribuée

Antoine Nongaillard, PhD

Concordia University

Université Lille 1, 2009

Les problèmes d'allocation de ressources suscitent un intérêt croissant aussi bien en Économie qu'en Informatique. Dans ces problèmes, un ensemble de ressources doit être alloué à un ensemble d'entités selon leurs propres préférences. De nombreux problèmes dans des domaines aussi divers que variés peuvent être modélisés grâce à un problème d'allocation de ressources. L'auto-organisation de réseaux en télécommunications, la planification en logistique, ou des problèmes basés sur des réseaux sociaux peuvent en effet être représentés par des problèmes d'allocation de ressources.

Ordinairement, ces problèmes sont résolus grâce à des méthodes centralisées, dans lesquelles une entité omnisciente détermine comment allouer les ressources de manière optimale. Cependant, ces approches font des hypothèses qui ne correspondent pas toujours à la réalité. Dans bien des contextes, il n'est pas possible d'avoir une entité omnisciente. Certaines applications sont dynamiques et nécessitent une méthode de résolution adaptative qui puisse prendre en compte de nouvelles informations au cours de la résolution. Ces approches considèrent toujours que les possibilités de communication entre les différents participants ne sont pas restreintes, ce qui n'est évidemment pas le cas dans la plupart des cas, comme dans les réseaux pair-à-pair par exemple où un pair ne peut communiquer qu'à un ensemble restreint du système.

Dans cette étude de doctorat, nous nous focalisons sur les approches de ré-allocation distribuées, basées sur des systèmes multi-agents, qui transforment une allocation initiale par des séquences de transactions locales entre agents. Nous cherchons à concevoir des comportements d'agents menant un processus de négociation à une allocation socialement optimale. Cette allocation peut alors être vue comme un phénomène émergent. Nous voulons également identifier les paramètres favorisant l'efficacité des négociations ainsi que ceux qui la restreignent. Nous considérons différentes mesures de bien-être social et nous fournissons les comportements à implémenter pour négocier efficacement dans chaque cas. Nous proposons une méthode adaptative et "anytime" où n'importe quel type de réseau d'accointances peut être considéré.

Mots-clés: Résolution distribuée de problèmes, raisonnement individuel, réseau social, bien-être social, information privée.

Contents

List of Figures

List of Tables

List of Algorithms

Introduction

This thesis is related to the distributed solving of resource allocation problems. We argue that simple agent behaviors always exist to efficiently solve allocation problems. In this chapter, we describe the main issues of this thesis. After a presentation of the context and a description of the motivations of this study, our objectives are detailed. The outline of this thesis is then presented.

Context and motivations

Auction, manufacturing scheduling, supply chain management or critical resource sharing are applications that can be modeled by resource allocation problems. A set of resources must be allocated to a set of entities who have preferences on them. The aim is to allocate all resources to entities, usually maximizing a given objective. Allocation problems are usually solved by centralized approaches. A central entity, who is omniscient, optimally allocates resources to entities. However, such solving methods do not suit many applications. Indeed, an omniscient central entity may not be available. Moreover, dynamic applications, in which data evolve constantly, cannot be solved efficiently by centralized techniques since any change in the initial data leads to a restart of whole solving processes. As well, applications in which entities keep private some information cannot be handled using centralized solving processes. For instance, Internet related applications require more and more privacy for users who do not want to reveal their preferences to everybody.

We choose in this thesis to focus on distributed approaches, solving allocation problems by agent negotiations. According to such techniques, agents are autonomous and act according to their own behavior. They locally negotiate with other agents in order to identify resource transactions satisfying their own acceptability criteria. An initial allocation evolves little by little by means of resource transactions among agents, until nobody is able to use acceptable transactions. Completed resource allocations can then be viewed as emergent phenomena. Each agent's decision-making is only based on its acceptability criterion. Indeed, agents are only aware of a restricted part of the system: No agent knows the whole resource allocation at a given time. In the literature, different notions of the social welfare theory can be used to collectively evaluate allocations. We focus on the four main social welfare notions: The utilitarian welfare that only considers the global efficiency, the egalitarian welfare that focuses only on the poorest agent, the Nash welfare that is a compromise between global efficiency and fairness, and the elitist notion that considers only the richest agent.

Objectives and contributions

In this thesis, we seek to design agents' behaviors leading negotiation processes to optimal allocations, or to socially close allocations when the need arises. We assume that the agent population is homogeneous, i.e. that all agents act similarly. Agents express their preferences thanks to additive utility functions. Different authors have studied agent negotiations in the literature. Some of them focus on a specific welfare notion, whereas others focus on mathematical properties of preference representations favoring the achievement of socially optimal allocations. However, none of them considers that agents have restricted communication abilities. Indeed, in real life, in social networks like *Facebook* or *MySpace*, users only have a restricted number of "friends". Each user has its own list of contacts, which is different from the lists of other agents. Usually in the literature, agents can negotiate with all other agents in the population, which is not a plausible assumption for most applications.

Our contribution is first to consider restricted communications between agents, which are represented by social graphs. Their price are also evaluated, i.e., their impact on the quality of achieved allocations. We provide a complete study identifying characteristics favoring the achievement of socially interesting allocations for each social welfare notion. We compare the efficiency of our negotiation processes to optimal solutions, which are provided by centralized methods. We finally provide negotiation settings to use in order to achieve optimal allocations.

Thesis outline

Chapter 1: Resource allocation problems. This chapter presents the general context of allocation problems. Their main characteristics are described and their impact on negotiation processes are discussed: the nature of the resources and representation of agent preferences are presented. Individual and collective evaluations of allocations are detailed. Centralized solving approaches and distributed ones are then presented. Contexts favoring the efficiency of each method is presented, with a description of their characteristics. Application examples that can be efficiently solved by each technique are also presented.

Chapter 2: Distributed negotiations. This chapter focuses on distributed solving processes for allocation problems. After a brief presentation of advantages of agent-based approaches and multi-agent systems, agent negotiations are defined and all features are detailed: Social graphs describing agent relationships, classes of transactions and their complexity, acceptability criteria and agent behaviors are successively described. Finally, issues related to the evaluation of negotiation processes are discussed.

Chapter 3: Experimental protocol. This chapter describes the simulation protocol, presenting the generation of different parameters, in order to precisely characterize experiments and to ensure their reproducibility. Algorithms required by the generation of graphs, preferences and initial allocations are detailed. Data instances and simulations are also characterized.

Chapter 4: Bilateral negotiations. In this chapter, results related to bilateral negotiation processes are presented. Each welfare notion is successively evaluated as follows. First, we present centralized methods, providing optimal solutions. Then, negotiation properties are specified according to the considered welfare notion. Different facets of negotiation processes are evaluated and important characteristics favoring the achievement of socially efficient allocations are identified. Utilitarian negotiations, egalitarian negotiations, Nash negotiations and elitist negotiations are successively investigated.
Chapter 5: Multilateral negotiations. This chapter is dedicated to multilateral transactions. Pros and cons are discussed in order to determine their effective interests within negotiation processes. A scalable method to determine acceptable multilateral transactions is described. Multilateral negotiation processes are then evaluated. Solution improvements due to their use is finally quantified.

We conclude this thesis by a summary of our contributions. Limits of this thesis are also described with a description of extensions that seem interesting to investigate.

Chapter 1

Resource Allocation Problems

Resource allocation is a research topic at the interface of two fields: Economics and Computer Science. Even if both communities study similar problems, fundamental differences appear when considering their respective objectives. While economists study qualities that resource allocations should satisfy, e.g., using the social choice theory and different welfare notions, computer scientists focus on mechanisms that identify resource allocations satisfying the required qualities. Studies carried out by both communities are thus complementary.

Recently, resource allocation problems arouse increasing interest due the large number of applications that can be modeled using this problem pattern. Up to now, most studies focused on combinatorial auction and their different facets, usually maximizing the global efficiency of the system. Representations of preferences and their mathematical properties are studied in order to decrease the problem complexity and to design efficient solving methods. Resource allocation problems are usually considered as optimization problems and solving processes are mainly centralized. Distributed approaches based on multi-agent systems have been investigated, achieving resource allocations thanks to local negotiations between agents. The aim is often to maximize a specific welfare notion in both cases.

In this chapter, resource allocation problems and their main characteristics are first described in Section 1.1. Features of resources and representations of agent's preferences are presented. Issues related to individual and collective evaluations of resource allocations are discussed, i.e., how allocations are evaluated either from the individual's point of view, or from the society's point of view. The two main solving approaches are then presented and compared. Principles of centralized techniques are described in Section 1.2. Some applications for which these methods are not well-adapted are identified. Classes of applications that can be efficiently solved by these approaches are also described. Moreover, Section 1.3 is dedicated to distributed solving processes. Examples of applications that can be efficiently solved only using distributed methods are characterized.

1.1 Problem description

This section is dedicated to the description of the main facets of resource allocation problems. Several essential questions arise during the problem definition: *"What are the properties of a resource? How does an agent express its preferences over the resource set? How is evaluated the individual welfare of an agent? How a resource allocation can be evaluated?"*. Each question corresponds to an important parameter characterizing the problem. Even the slightest change of problem settings drastically affects the properties of allocation problems, their complexity and then the way to solve them efficiently.

A resource allocation problem is defined considering a set of resources and a set of entities. Resources correspond to anything that can be owned by entities, i.e., concrete resources like books or any physical goods as well as abstract resources like CPU time or network bandwidth. Entities express preferences over the whole resource set. The aim of an allocation problem is to identify a distribution of all resources maximizing, minimizing or satisfying a given objective. It can be formally defined as follows:

Definition 1.1 (Allocation problem). An allocation problem is a tuple $\langle \mathcal{R}, \mathcal{P}, \mathcal{U} \rangle$, where $\mathcal{R}$ is a set of m available resources, $\mathcal{P}$ is a finite set of n entities, and $\mathcal{U}$ is a vector of entity's preferences on the resource set. The aim is the identification of a resource distribution of $\mathcal{R}$ over $\mathcal{P}$ satisfying an objective, according to the preferences of the entities $\mathcal{U}$.

A resource allocation problem is illustrated in Figure 1.1. Two parts can be distinguished: on the left hand side, the initial data and a result on the right hand side. The allocation problem is defined here by a set of 3 entities and a set of 9 resources. According to the preferences of each agent $\mathcal{U} = (u_0, u_1, u_2)$, the solving process leads to an allocation in which each entity gets a set of resources.

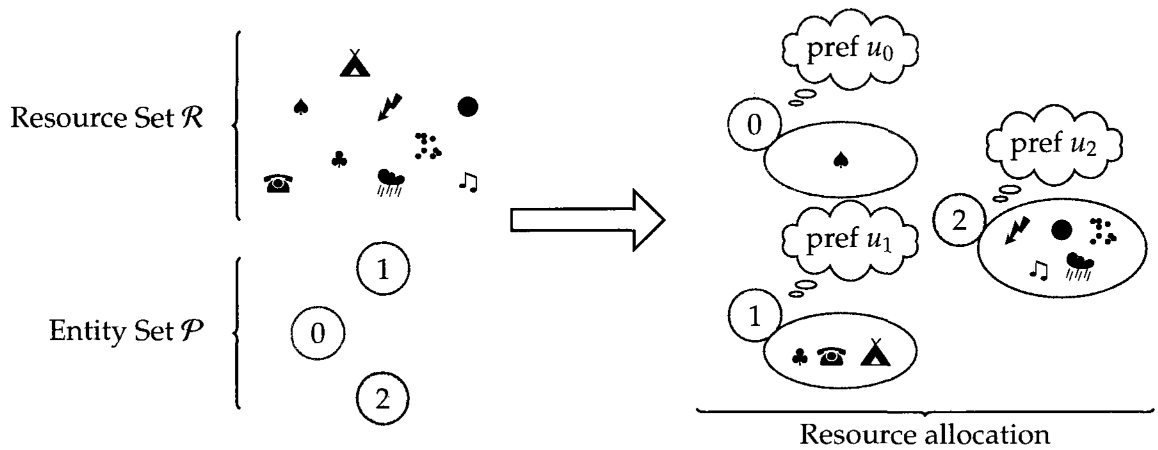


Figure 1.1: Resource allocation problems

Depending on the kind of solving method considered, a specific terminology can be used. Indeed, in a distributed context, the set of entities is usually assimilated to a population of agents. However, in a centralized context, the use of the term *"agent"* is improper, according to the standard definition of an agent (Ferber, 1999; Woolridge, 2001). Indeed, entities are neither distributed, nor autonomous and no decision is made at the entity level. Everything is decided by the central entity, which is most of the time omniscient. Entities neither have perception nor consider their neighborhood, which are important notions defining agents. Aware of the difference between entities and agents, an abuse of terminology is tolerated in this thesis, and the term *"agent"* will be used in both cases.

Each agent of the population owns a finite set of resources, called a resource bundle. A resource allocation describes how resources are distributed to agents. Then, the definition of a resource allocation can be based on the resource bundle of each agent.

Definition 1.2 (Resource allocation). Given a set $\mathcal{R}$ of m resources and a population $\mathcal{P}$ of

n agents, a resource allocation A is represented as an ordered list of n resource bundles $\mathcal{R}_i \subseteq \mathcal{R}$ describing the subset of resources owned by each agent i:

$$A = [\mathcal{R}_1, \ldots, \mathcal{R}_n], \qquad 1, \ldots, n \in \mathcal{P}, \quad A \in \mathcal{A}.$$

where $\mathcal{A}$ is the set of all possible allocations. The i-th element of an allocation A corresponds to the resource bundle of agent i. It can be written as follows:

$$A[i] = \mathcal{R}_i, \qquad i \in \mathcal{P}, \quad A \in \mathcal{A}.$$

Example 1.1. Let us consider the resource allocation described in Figure 1.1. Let $A \in \mathcal{A}$ denotes this allocation, which can be explicitly written as follows:

$$A = \left[\mathcal{R}_0, \mathcal{R}_1, \mathcal{R}_2\right] = \left[\{\text{♠}\}\{\text{♣}, \text{☎}, \text{⛺}\}\{\text{⚡}, \text{⛈}, \text{⁘}, \text{●}, \text{♫}\}\right].$$

According to allocation A, entity 0 owns only one resource $\mathcal{R}_0 = \{\text{♠}\}$. Entity 1 has three resources in its bundle $\mathcal{R}_1 = \{\text{♣}, \text{☎}, \text{⛺}\}$ whereas entity 2 owns the five remaining resources $\mathcal{R}_2 = \{\text{⚡}, \text{⛈}, \text{⁘}, \text{●}, \text{♫}\}$.

Different aspects of resource allocation problems are successively discussed in the rest of this chapter. First, various kinds of resource are presented in Section 1.1.1 and their impact on the problem model. Then, different ways to represent agent's preferences are described in Section 1.1.2. The evaluation of the individual welfare of an agent is also discussed. Finally, issues related to the collective evaluation of resource allocations are addressed in Section 1.1.3. Notions proposed by the social choice theory are presented as well as their impact on the resource distribution.

1.1.1 Resource characteristics

Resources are central elements of allocation problems. Their properties deeply affect the model, independently of the solving approach that is considered. The nature of the resources also influence the properties of allocations. The most important resource features are described in the next parts (Chevaleyre et al, 2006a).

Continuous or discrete

According to the physical properties of resources, they can be either *continuous* like water, or *discrete* like books. This influences the way that resources are exchanged.

Typically, continuous resources can be divided in as many parts as required. In such a case, resources available in the system correspond to quantities. For instance, a government aiming to fairly distribute water among cities according to their needs can be modeled thanks to continuous resources (Cormas, 2001). Indeed, the only physical resource of this problem is "water". Resources of the allocation problem are quantities of water.

Allocation mechanisms designed for discrete resources are also suitable in the case of continuous resources. However, such mechanisms are often not as efficient as mechanisms designed specifically for continuous resources. At the opposite, discrete resources are always indivisible and represent units. Continuous resources may be discretized, i.e., transformed into discrete resources. The whole quantity is divided into several parts, which are then considered as discrete and indivisible units. Allocation problems based on continuous resources are widely studied in the literature in Economics, whereas Computer Science mainly focuses on discrete resources. Only discrete resources are considered in this thesis.

Divisible or not

Resources may be either divisible or indivisible. At the opposite of the former property, this one is related to the allocation mechanism rather than to resources themselves. A resource may be divided a given number of times, beyond which the resource becomes an indivisible unit. However, only indivisible resources are considered in this thesis.

Sharable or not

Resources may also be sharable. This property affects the way that the agent welfare is determined. When resources are sharable, it is not required to own a resource in order to benefit from it. For example, if resources are assimilated to abilities, agents do not necessarily need the ability itself, they only need to know other agents which have this ability and then ask them to perform a task. Such a situation occurs in distributed service environment for instance (Chakraborty et al, 2006). Another typical example is the usage of common expensive resources like pictures got from a satellite (Lemaître et al, 1999). A single resource can be allocated to several agents. In this thesis, only not sharable resources are considered.

Static or not

Sometimes, resources may be usable. Such resources can be consumed by their owner, which perform a specific task. For instance, resources like food are edible. An agent may be able to eat some parts of its bundle in order to stay alive. Resources may then disappear from the system. Resources may also be perishable and then have a value or a quantity decreasing in time. In such cases, resources are considered as usable. At the opposite, resources are static when their properties do not change in time. As usually assumed in most of resource allocation studies, only static resources are considered in this thesis.

Single-unit or multi-unit

In multi-unit environments, resources may be identical, and then indistinguishable. Thus, they are addressed using a single name. For instance, in an egg box, all eggs are similar and cannot be distinguished. In single-unit environments, resources can always be distinguished from others. An *identification tag* is assigned to each resource. Multi-unit environments can be turn into single-unit environments by tagging all resources. For example, in the egg box, each tag allows agents to address a specific egg. The main difference between these environments comes from the resource representation. In this thesis, only single-unit environments are considered.

Resource or task

Resource allocation problems and task allocation problems can be distinguished. But, tasks are often resources associated with a negative value. While agents benefit from standard resources, tasks can be viewed as duties, and then represent a burden to their owners. An important characteristic of tasks is that they are often related to other tasks, e.g. the ones are conditions to fulfill in order to perform the others. For instance, in factories, products may require the achievement of specific tasks in order to undergo new transformations, which correspond to other tasks. This thesis focuses on resource allocation problems. Even if negative values can be assigned to resources, no dependence relationship will be considered.

Allocation properties and complexity

In this thesis, resources are assumed to be discrete, not divisible and not sharable. All resources are unique and agents cannot alter them. Since the characteristics of resources we consider is now well-defined, properties of resource allocations can be specified.

Property 1.1 (Resource allocation properties). *Since the allocation environment is single-unit and resources are assumed to be indivisible, discrete and not sharable, each resource is allocated to only one agent. Agents' resource bundle must be pairwise disjoint. We also assume that all resources must be allocated. More formally:*

$$\bigcap_{i \in \mathcal{P}} A[i] = \bigcap_{i \in \mathcal{P}} \mathcal{R}_i = \emptyset, \qquad A \in \mathcal{A};$$
$$\bigcup_{i \in \mathcal{P}} A[i] = \bigcup_{i \in \mathcal{P}} \mathcal{R}_i = \mathcal{R}, \qquad A \in \mathcal{A}.$$

Different parameters may, more or less, affect the complexity, and then the identification of optimal allocations (Chevaleyre et al, 2006a; Estivie, 2006). According to the nature of the resources we consider, the size of the solution space can be deducted.

Property 1.2 (Size of the solution space). *An allocation problem based on a population $\mathcal{P}$ of n agents where m resources of a set $\mathcal{R}$ are available has an exponential number of possible solutions:*

$$|\mathcal{A}| = n^m.$$

The explanation is quite simple since it corresponds to the number of possible combinations. Each resource of $\mathcal{R}$ can be allocated to any agent of the population $\mathcal{P}$, so n times. Similarly for all m resources, the total number of possible resource allocations can be deduced. Thus, the solution space has an exponential size:

$$|\mathcal{A}| = \prod_{r \in \mathcal{R}} n = n^m.$$

1.1.2 Representation of preferences and individual welfare

The representation of an agent's preferences is an essential issue in allocation problems. Preferences express the relative or absolute satisfaction of an agent which needs to consider several alternatives.

Five important features should be addressed when investigating preference representation languages (Chevaleyre et al, 2006a). These features allow comparisons of different representation languages, considering their different facets.

- *Elicitation*: It evaluates the difficulty of designing algorithms for an agent to get an output expressed in a given language;
- *Cognitive relevance*: This criterion describes the ease for a human to know and express its preferences in a given language;
- *Expressive power*: In a given language, it identifies the different sets of preference structures that can be expressed;

- *Computational complexity*: It evaluates the complexity of comparing two alternatives, or the complexity of determining an optimal allocation for a given language;

- *Comparative conciseness*: For two languages expressing the same content, it evaluates the size required for a given expression.

These characteristics are used to describe different classes of preference structure in this section. The most widely used preference representations are presented. The individual welfare evaluation is discussed with some issues related to so-called side payments.

Families of preference structures

The representation of agent's preferences has been studied for a long time (Doyle, 2004; Fishburn, 1970; Mas-Colell et al, 1995). Four families of preference structures can be distinguished:

- A *cardinal* preference structure is usually a utility function, denoted by u, which associates a value from the set Val to all alternatives of the set X:

  $$u : X \to Val.$$

  If Val is a set of numerical values, the preference structure is called *quantitative*, whereas if Val is an ordered set of qualitative values, like linguistic expressions, e.g., {good, excellent, ...}, the preference structure is called *qualitative*;

- An *ordinal* preference structure is a binary relation on alternatives, denoted by $\preceq$, which is reflexive and transitive;

- A *binary* preference structure is simply a partition of the set of alternatives X into two subsets representing "good" and "bad" alternatives.

- A *fuzzy* preference structure is a fuzzy relation over X allowing the expression of a degree of preference:

  $$\mu : X^2 \to [0, 1].$$

Binary and fuzzy preferences have not been used much as far as resource allocation problems are concerned. Indeed, binary preferences are very restrictive and once the "good" alternatives are regrouped, nothing indicates which one should be chosen. Fuzzy preferences allow the comparison of alternatives by pairs. However, they are not convenient when a very large number of alternatives must be compared.

Ordinal preferences can only express the satisfaction of an agent for all alternatives. Intensity cannot be expressed and, given a resource allocation, it is not possible to determine which agent is more satisfied. These preferences are not much used in a resource allocation context in Computer Science (Bouveret et al, 2009), because of the few information revealed.

Even if qualitative cardinal preferences can express intensity, they suffer of similar drawbacks than ordinal preferences. The satisfaction of two agents possessing different resources cannot be compared. More formally, there is no relationship between $u_i(r)$ and $u_j(r')$ with $i, j \in \mathcal{P}$ and $r, r' \in \mathcal{R}$, which are then not comparable.

The most widely used representation of preferences in Computer Science is the quantitative cardinal structure. In this thesis, agents express their preferences by numerical preferences. The utility function can then be defined as:

$$u : 2^{\mathcal{R}} \rightarrow \mathbb{R}.$$

According to such a definition, $2^{\mathcal{R}} - 1$ values must be specified. However, the exponential nature of an explicit specification leads to unscalability in most cases. Then, it is relevant to consider restricted preference structures.

In Economics, preferences are always represented by ordinal function nowadays (Mas-Colell et al, 1995). Since there is no comparable scale of values between two agents, it is not possible to compare the individual welfare of two agents who own two different resources. However, the context is generally different from ours. Indeed, economists always consider human actors only and purely economic applications. However, in Computer Science, in many classes of applications, agents are not necessarily humans (e.g., computers or software programs). Resources have nature pretty similar. In such a restricted context, cardinal preferences represent a plausible assumption.

Quantitative preferences

Utility functions are defined over resource bundles, and we assume that agents' preferences only depend on the resources they own. The agent's utility is independent of the utility of other agents. Such preferences said to be called *free of externalities*. More formally:

$$u_i(A) = u_i(A[i]) = u_i(\mathcal{R}_i), \qquad i \in \mathcal{P}, \quad A \in \mathcal{A}.$$

Several languages can be used to represent utility functions, and the most important ones are presented next (Chevaleyre et al, 2006a; Estivie, 2006). Let us consider $\mathcal{R} = \{r_1, r_2, \ldots, r_m\}$ the set of available resources and $\rho \subseteq \mathcal{R}$ a subset of resources in order to illustrate the different forms.

First, the **bundle enumeration**, also called **explicit form**, is the most basic form of utility function. The utility function is a set of pairs $\langle \rho, u_i(\rho) \rangle$. The bundle form is obviously *fully expressive*: Any utility function can be described. The description length is a major drawback since its exponential length increases with the number of resources m.

Example 1.2. Let us consider a resource set $\mathcal{R} = \{r_1, r_2, r_3\}$. According to the bundle enumeration form, the utility function of agent $i \in \mathcal{P}$ must explicitly describe the utility of all possible subsets of resources:

$$
\begin{aligned}
u_i(\{r_1\}) &= val_1 \\
u_i(\{r_2\}) &= val_2 \\
u_i(\{r_3\}) &= val_3 \\
u_i(\{r_1, r_2\}) &= val_4 \\
u_i(\{r_1, r_3\}) &= val_5 \\
u_i(\{r_2, r_3\}) &= val_6 \\
u_i(\{r_1, r_2, r_3\}) &= val_7
\end{aligned}
$$

Such a representation is fully expressive, but is not succinct since it requires an exponential number of expressions. As well, the computational complexity is also high: The determination of an optimal allocation requires an explicit consideration of all possible resource allocations. It represents the major drawbacks of this representation which cannot be used in practice.

The **additive form** expresses the utility associated with a given subset of resources ρ relatively to the utility associated with each resource of this subset ρ (Wellman and Doyle, 1992). The utility associated with a set of resources simply corresponds to the sum of the utilities associated with each resource of this set. More formally, a utility function is additive if and only if there exist, for all resources $r \in \rho$, coefficients α_i^r such as:

$$u_i(\rho) = \sum_{r \in \rho} \alpha_i^r, \qquad i \in \mathcal{P}, \quad \rho \subseteq \mathcal{R}.$$

Example 1.3. Let us consider a set of resources $\mathcal{R} = \{r_1, r_2, r_3\}$. According to the additive representation, the utility function of agent $i \in \mathcal{P}$ only requires the utility of each resource:

$$\begin{aligned} u_i(\{r_1\}) &= val_1 \\ u_i(\{r_2\}) &= val_2 \\ u_i(\{r_3\}) &= val_3 \end{aligned}$$

The utility value associated with any subset of $\mathcal{R}$ can be easily computed from these values. For instance:

$$u_i(\{r_1, r_2\}) = u_i(\{r_1\}) + u_i(\{r_2\}) = val_1 + val_2.$$

Such a representation is not fully expressive since no synergy among resources can be expressed. However, this representation is very succinct and has a low computational complexity. This additive form is the one used to represent agent's preferences in simulations performed in this thesis.

The k-**additive form** is inspired by the fuzzy measure theory (Grabisch, 1997; Miranda et al, 2005). It corresponds to a generalization of the additive representation. A utility function is k-additive if and only if there exists a coefficient α_i^t for each resource set t of size at most k.

$$u_i(\rho) = \sum_{t \subseteq \rho} \alpha_i^t, \qquad i \in \mathcal{P}, \quad \rho \subseteq \mathcal{R}.$$

The coefficient α_i^t represents the synergy value of owning all resources in the set t. If agent i owns all resources in a term t, its utility value increases of α_i^t.

Example 1.4. Let us consider a set of resources $\mathcal{R} = \{r_1, r_2, r_3\}$. A utility function can be written in a polynomial form, where variables r_j (the resources) are Boolean values. For instance, a 2-additive function, which allows the expression of synergy between at most 2 resources, with two non null coefficient $\alpha\{r_1\} = val_1$ (where $\forall \rho \subseteq \mathcal{R}, \alpha\{\rho\}$ denotes the coefficient associated with the subset of resources ρ) and $\alpha\{r_2, r_3\} = val_2$ can be written as:

$$u_i = val_1 r_1 + val_2 r_2 r_3$$

The representation of this utility function according to the bundle form requires the specification of 5 terms:

$$\begin{aligned} u_i(\{r_1\}) &= val_1 \\ u_i(\{r_1, r_2\}) &= val_1 \\ u_i(\{r_1, r_3\}) &= val_1 \\ u_i(\{r_2, r_3\}) &= val_2 \\ u_i(\{r_1, r_2, r_3\}) &= val_1 + val_2 \end{aligned}$$

The k-additive form is also *fully expressive*, but only if k is large enough. Such an assumption is not true in practice since k is generally restricted to a relatively small value. This form is nevertheless more succinct than the bundle form.

The **weighted propositional form** makes an explicit use of logic (Bonzon et al, 2009; Chevaleyre et al, 2006b; Coste-Marquis et al, 2004; Lang, 2004; Uckelman et al, 2009). It is possible to express all kinds of synergy using logic formulas. Each resource r is represented using a propositional variable, which is true if the agent owns r and false otherwise. Each propositional formula can be considered as a goal, and a goal base GB represents the whole set of formulas. Each agent has then a goal base expressing its preferences. Numerical weights represent the relative importance of the formulas. Intuitively, the degree of satisfaction associated with a particular propositional allocation A is the sum of the weights of the formulas satisfied by this allocation A. However, different kinds of aggregations can be used instead of a summation.

Example 1.5. Let us consider a set of 4 resources $\mathcal{R} = \{r_1, r_2, r_3, r_4\}$. The goal base can be expressed as follows:

$$GB = \{(val_1, r_1 \wedge r_2), (val_2, \neg r_1 \wedge r_3\}), (val_3, r_3 \rightarrow r_4)\}.$$

According to the resources owned by the agent, the individual welfare can be evaluated:

$$\begin{aligned} u_{GB}([r_1, r_2, r_3, r_4]) &= val_1 + val_3 \\ u_{GB}([\neg r_1, r_2, r_3, \neg r_4]) &= val_2 \end{aligned}$$

The weighted propositional representation is a fully expressive form since any synergy among resources can be expressed by means of formulas. It is less succinct than additive preferences and more complex computationally.

The **X-OR form** is the most widely used binding language. It became a standard in the expression of preferences in combinatorial auction (Nisan, 2000; Sandholm, 2002). Agent's preferences are a set of pairs (ρ, α^ρ) where α^ρ is the value associated with the resource set ρ. At the opposite of the k-additive form, instead of adding all active terms, the valuation of a bundle is simply the highest value offered for any of its terms.

$$u_i(\rho) = \max_{t \subseteq \rho} \alpha_i^t, \qquad i \in \mathcal{P}, \quad \rho \subseteq \mathcal{R}.$$

Example 1.6. Let us consider a set of 2 resources $\mathcal{R} = \{r_1, r_2\}$. A utility function can be written in a polynomial form, where variables r_j (the resources) are Boolean values. For instance, an X-OR utility function can be written as:

$$u_i = val_1 r_1 + val_2 r_2 + val_3 r_1 r_2$$

Let us assume that $val_1 < val_3 < val_2$. Then, the representation of this utility function according to the bundle form requires the specification of 3 terms:

$$
\begin{aligned}
u_i(\{r_1\}) &= val_1 \\
u_i(\{r_2\}) &= val_2 \\
u_i(\{r_1, r_2\}) &= val_2
\end{aligned}
$$

Evaluation function

Since the representation of agent's preferences is clearly defined, issues related to the determination of the agent individual welfare can now be considered.

Agents determine their individual welfare thanks to an evaluation function. Such a function may be based on several criteria, including not only the utility function. For instance, in the case of allocation problems where the use of money is allowed, the individual welfare of agents can be based on their resource bundle and their wallet.

In most studies of the literature, money is considered in allocation problems through side payments (Sandholm, 1998). When agents trade resources, they may get resources associated with a lower utility value than the ones they give. Most of the time, agents are assumed to be selfish, i.e., they can only accept resource transactions increasing their individual welfare. The agent selfishness prevents transactions to be performed if one of the participants is not satisfied. If an agent notices a decrease of its satisfaction, it then refuses the transaction. However, the loss of utility can be compensated thanks to side payments from other participants. Thus, a transaction, where an agent receives resources associated with a lower utility value than the ones it provides, can still be performed if the decrease of satisfaction is compensated by a side payment. In such systems, the use of money is nevertheless constrained. The quantity of money in the system is constant. In other words, when agents trade resources adding side payments, the amount of money given by an agent is equal to the amount of money received by the other agent. With such a constraint, the overall amount of money does not vary. However, this amount is not bounded. Indeed, agents have no budget limit. Then, they are always assumed as rich as required to perform any transaction that seems of interest.

Example 1.7. Let us consider a transaction δ involving two agents $i, j \in \mathcal{P}$, changing the initial resource allocation A in another one A' ($A, A' \in \mathcal{A}$). During this transaction, agent i gives one of its resources, $r \in \mathcal{R}_i$, to agent j.

Both agents are here assumed to be selfish. They only accept transactions increasing their individual welfare. The acceptability of a transaction δ is determined based on the utility participants get in the new allocation and on the payments they make or receive. More formally, an acceptability condition can be formulated for each agent as follows:

$$u_i(A') + p(i) > u_i(A)$$
$$u_j(A') + p(j) > u_j(A).$$

where $p(i)$ and $p(j)$ are respectively the side payments made during the transaction by agents i and j. The value associated with a payment function value is positive when agents receive money while it is negative when they have to pay. Since no money is created during the transaction, the following relationship is satisfied:

$$p(i) = -p(j).$$

The amount of money given by one agent is equivalent to the amount of money received by the other agent. According to the transaction, i.e., the gift of a resource r here, the acceptability conditions can be written as follows:

$$\begin{cases} u_i(A) - u_i(r) + p(i) > u_i(A) \\ u_j(A) + u_j(r) + p(j) > u_j(A) \end{cases}$$

$$\begin{cases} p(i) > u_i(r) \\ u_j(r) > -p(j) \end{cases}$$

Both expressions can be combined into a single one:

$$u_j(r) > p(i) > u_i(r).$$

If the amount of money owned by each agent is not bounded, then independently of the utility value associated with resource r by both agents, a compensatory payment satisfying this expression always exists.

Thus, independently of the conditions that transactions must satisfy, any transaction can be performed since they can be artificially satisfied using unbounded side payments. Since such an assumption is not plausible from a practical point of view, side payments are considered as being beyond the scope of this thesis. Thus, money is prohibited and the evaluation of the individual agent welfare is restricted to a utility function.

Definition 1.3 (Utility function). An agent evaluates its individual welfare thanks to an additive utility function $u_i : 2^{\mathcal{R}} \rightarrow \mathbb{R}$. When agent $i \in \mathcal{P}$ owns a set of resources $\rho \subseteq \mathcal{R}$, its utility is evaluated as follows:

$$u_i(\rho) = \sum_{r \in \rho} u_i(r), \qquad i \in \mathcal{P}, \quad \rho \subseteq \mathcal{R}.$$

Example 1.8. Let us illustrate the individual evaluation of agent welfare using a simple example, based on a population of 3 agents, $\mathcal{P} = \{1, 2, 3\}$, and a set of 3 available resources $\mathcal{R} = \{r_1, r_2, r_3\}$. The agent's preferences are described in Table 1.1. For instance, agent 1 associates with resource r_2 the following utility value: $u_1(r_2) = 7$.

If the initial resource allocation is $A = [\{r_4\}\{r_1, r_2, r_6\}\{r_3, r_5\}]$, then the utility of all agents

Table 1.1: Utility function - Example of agent's preferences

Population $\mathcal{P}$	Resource Set $\mathcal{R}$					
	r_1	r_2	r_3	r_4	r_5	r_6
1	10	7	10	9	2	1
2	6	10	3	4	8	6
3	1	2	1	2	1	3

can be easily computed as follows:

$$u_1(\mathcal{R}_1) = u_1(\{r_4\}) = u_1(r_4) = 9$$

$$u_2(\mathcal{R}_2) = u_2(\{r_1, r_2, r_6\}) = u_2(r_1) + u_1(r_2) + u_2(r_6) = 6 + 10 + 6 = 22$$

$$u_3(\mathcal{R}_3) = u_3(\{r_3, r_5\}) = u_3(r_3) + u_3(r_5) = 1 + 1 = 2$$

1.1.3 Social welfare theory

The collective evaluation of resource allocations constitutes an important issue. *"How can the evaluation of an allocation be based on the welfare of each agent in the population?"*. An answer can be found in the literature thanks to the social choice theory (Arrow, 1963; Moulin, 1988; Sen, 1970). The social choice theory, which comes from Economics, defines a set of tools used to measure the welfare of a society from the individual welfare of all agents. Several notions exist, and most of them can be applied to allocation problems according to different contexts (Arrow et al, 2002; Moulin, 2004; Sen, 1997). In this section, these notions are successively detailed with their impact on resource distributions.

Utilitarian welfare

The most widely used notion to evaluate resource allocations is the utilitarian welfare. The welfare of a population is evaluated through the sum of the individual welfares of all agents in the society. This notion is often used to maximize the global welfare of a population, without consideration for individual welfare.

Definition 1.4 (Utilitarian welfare). The utilitarian welfare of a resource allocation A, denoted by $sw_u(A)$, corresponds to the sum of individual utilities.

$$sw_u(A) = \sum_{i \in \mathcal{P}} u_i(\mathcal{R}_i), \qquad A \in \mathcal{A}.$$

The utilitarian welfare is not well-adapted to all cases, especially when fairness among agents is considered. In such cases, the egalitarian welfare is favored.

Egalitarian welfare

The egalitarian welfare of an allocation corresponds to the individual welfare of the poorest agent in the population. Its maximization tends to reduce inequalities over the population. Fair sharing is an important issue for many resource allocation problems (Brams and Taylor, 1996; Moulin, 2004; Rawls, 1999; Sen, 1995).

Definition 1.5 (Egalitarian welfare). The egalitarian welfare of an allocation A, denoted by $sw_e(A)$, corresponds to the individual utility of the poorest agent.

$$sw_e(A) = \min_{i \in \mathcal{P}} u_i(\mathcal{R}_i), \qquad A \in \mathcal{A}.$$

Nash product

The Nash product considers the welfare of the whole population and reduces the inequalities among agents at the same time (Ramezani and Endriss, 2009). The Nash product is a social notion that can be viewed as a compromise between the utilitarian and the egalitarian welfare. This notion is independent of utility scales, and it also normalizes agent's utilities. In spite of its qualities, a drawback remains: this notion becomes meaningless if non-positive values are used.

Definition 1.6 (Nash product). The Nash product of an allocation A, denoted by $sw_n(A)$, corresponds to the product of individual utilities.

$$sw_n(A) = \prod_{i \in \mathcal{P}} u_i(\mathcal{R}_i), \qquad A \in \mathcal{A}.$$

Elitist welfare

The elitist welfare is the exact opposite of the egalitarian notion. It only considers the welfare of the richest agent in the population. This notion can be useful in the context of artificial agent societies for instance, where agents have a common objective. This objective must be fulfilled, independently of the agent who achieves it. The elitist welfare notion is then suitable.

Definition 1.7 (Elitist welfare)**.** The elitist welfare of an allocation A, denoted by $sw_{e\ell}(A)$, corresponds to the individual utility of the richest agent in the population.

$$sw_{e\ell}(A) = \max_{i \in \mathcal{P}} u_i(\mathcal{R}_i), \qquad A \in \mathcal{A}.$$

Other notions

Various other notions and properties exist to evaluate resource allocations. According to the envy-free notion (Brams and Taylor, 1996), agents evaluate their individual welfare using a comparison with the resource bundle of others. Indeed, a resource allocation is envy-free if all agents are at least as happier with their bundle as they would be with the resource bundle of other agents. More formally, an allocation is envy-free when the following expression is satisfied:

$$u_i(\mathcal{R}_i) \geq u_i(\mathcal{R}_j) \qquad \forall i, j \in \mathcal{P}.$$

According to the notion of jealousy, agents evaluate their welfare using a comparison with the welfare of others. Agents are not jealous when their individual welfare is larger than the welfare of other agents. More formally, an allocation is jealousy-free when the following expression is satisfied:

$$u_i(\mathcal{R}_i) \geq u_j(\mathcal{R}_j) \qquad \forall i, j \in \mathcal{P}.$$

Pareto optimal allocations (Moulin, 1988) are allocations in which no agent can improve its individual welfare without penalizing the welfare of another agent. Such a notion does not require any numerical preferences representation. More formally, an allocation $A \in \mathcal{A}$ is Pareto optimal if:

$$\nexists A' \in \mathcal{A}, \quad sw(A) < sw(A') \quad \text{such as} \quad \forall i \in \mathcal{P}, \quad u_i(A) \leq u_i(A').$$

Lorenz optimality (Moulin, 1988) is a notion combining utilitarian and egalitarian aspects of social welfare. The idea is to favor allocations improving the utilitarian welfare without causing a loss in egalitarian welfare.

Theoretical studies on allocation properties based on these notions have been carried out (Chevaleyre et al, 2007, 2009; Endriss et al, 2006). In spite of their interest, these notions are not studied in this thesis. Thus, only the four main welfare notions are considered, i.e., the utilitarian welfare, the egalitarian welfare, the Nash welfare and the elitist welfare.

Impact on resource allocations

The four main welfare notions have different impacts on resource distributions. Indeed, the use of a specific welfare notion may induce undesirable properties, which should be avoided depending on the application context.

Example 1.9. Let us consider a population of 3 agents $\mathcal{P} = \{1, 2, 3\}$ and a set of 6 resources $\mathcal{R} = \{r_1, r_2, r_3, r_4, r_5, r_6\}$. Table 1.2 describes the agent's preferences.

Table 1.3 shows optimal social values and a corresponding resource allocation according to the different social welfare notions. Characteristics of optimal allocations are then discussed.

Table 1.2: Welfare impact - Example of agent's preferences

Population $\mathcal{P}$	Resource Set $\mathcal{R}$					
	r_1	r_2	r_3	r_4	r_5	r_6
1	10	7	10	9	2	1
2	6	10	3	4	8	6
3	1	2	1	2	1	3

Table 1.3: Optimal allocation examples for all welfare notions.

Social welfare	Value	Resource allocation
sw_u	53	$[\{r_1, r_3, r_4\}\{r_2, r_5, r_6\}\{\}]$
sw_e	8	$[\{r_1\}\{r_5\}\{r_2, r_3, r_4, r_6\}]$
sw_n	1800	$[\{r_1, r_3\}\{r_2, r_5\}\{r_4, r_6\}]$
$sw_{e\ell}$	39	$[\{r_1, r_2, r_3, r_4 r_5, r_6\}\{\}\{\}]$

The use of the utilitarian notion leads to a resource allocation where one agent, agent 3, does not get any resource. Indeed, some agents may be neglected, especially if, for each resource, there exists another agent who associates with it a larger utility value. Such a situation may be unreasonable.

The use of the egalitarian welfare leads to a resource allocation that provides at least one resource to each agent. Thus, if the number of available resources is greater than the number of agents ($n < m$), no agent is neglected in egalitarian allocations. However, the resource distribution can be very unbalanced. Agents with low preferences, like agent 3, drain resources. Such agents may obtain most of the resources in order to compensate for their low preferences.

The Nash product also leads to a resource allocation where all agents get at least one resource, as in the egalitarian case. However, the optimal allocation is more balanced, avoiding the draining phenomenon. Nevertheless, this notion can only be used if the agent's utility values are positive.

The elitist social welfare neglects all agents except one and leads to an allocation where all resources are in the same bundle (when utility values are positive). This last notion is mainly used when it is essential to achieve the objective, independently of the agent who achieves it.

Finally, let us note that the welfare value achieved according to these different notions are quite different. However, their comparison is meaningless since the different welfare notions are used for different purposes. The choice of a specific welfare notion only depends on the application context.

1.2 Centralized approaches

Obviously, resource allocation problems can be solved using centralized approaches. Such approaches consider resource allocation problems as optimization problems. They are appropriate to solve many application classes. However, they are not adapted to all cases since they make specific assumptions, even implicitly, which may prevent their use under different conditions. These assumptions are detailed in this section. Then, we discuss application characteristics according to which centralized solving processes are not adapted. Finally, we describe applications that can be solved efficiently using centralized models.

1.2.1 Description

All centralized techniques are based on the same principle. The solving process can be decomposed into three main steps as described in Figure 1.2: information gathering, computations, and the notification of the outcome to all agents.

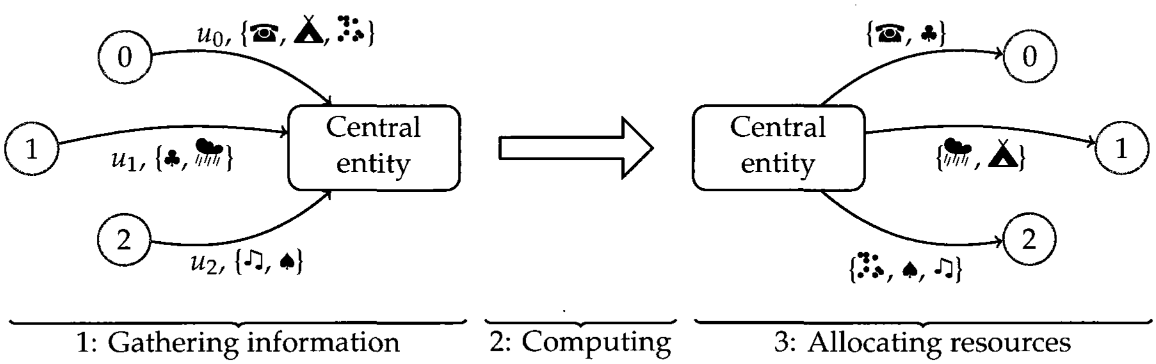


Figure 1.2: Principles of centralized techniques

First, all agents of the population must send their private information to a central entity, i.e., their preferences and a list of the resources they own. This central entity can be either an agent or an external entity. The central entity can be considered as omniscient since it gathers all information: It knows preferences of all agents and a complete list of resources available in the system. According to a predefined objective, e.g., a social welfare notion as defined in Section 1.1.3, the central entity determines a resource allocation maximizing this objective. Finally, once computations are over, it notifies all agents what they get and then it allocates resources accordingly. Let us note that these methods do not consider that resources are initially allocated anywhere. They assume that all resources are available and just determine optimal allocations.

Resource allocation problems are assimilated to optimization problems. Such centralized approaches can be used to solve some classes of applications, while other classes cannot use them in a reasonable time. Since the solution space is finite according to Proposition 1.2, an exact centralized method always exists. Indeed, the explicit enumeration of all solutions, keeping the one maximizing the objective, is always possible. However, since the solution space is exponentially large, this method is not scalable at all. Other limitations are described in the next section.

1.2.2 Limiting cases

Centralized solving processes cannot be used to efficiently solve some classes of problems. These applications have specific features, described in this section, that can be viewed as drawbacks. Some are directly related to the application context, whereas others are related to implicit assumptions made by centralized approaches.

Dynamic applications cannot be solved efficiently using centralized approaches. Indeed, in dynamic applications, data evolves constantly and these methods cannot handle such evolutions. In order to consider new data, they have to restart a complete solving process. A continuous evolution of the initial data cannot be handled properly. Adaptive processes are then essential to efficiently solve dynamic applications. For instance, in peer-to-peer applications, e.g., file sharing applications (Deshpande and Venkatasubramanian, 2004; Ge et al, 2003), agents continuously enter and leave the system, bringing new files to share with others. Thus, centralized techniques are not well-adapted to the solution of dynamic problems. Applications can be considered as dynamic when the time required for the solving process exceeds the time between two data evolutions.

Scalability issues may quickly arise according to the population size and the number of resources available in the system. Indeed, a resource allocation problem based on a population $\mathcal{P}$ of n agents and a resource set $\mathcal{R}$ of m resources leads to an exponential number of allocations, according to Proposition 1.2. Thus, large problems may be unscalable. Even if the solving process is centralized, computations may be distributed. Indeed, there exists optimization problems with distributed constraints (Petcu et al, 2006). According to the problem structure, the distribution of computations may be more or less efficient. The improvement of the scalability due to the distribution is limited anyway.

Information privacy. An important limitation is related to the issue of information privacy. Indeed, depending on the application context, agents may need to keep information private. Especially when Internet-based applications are considered, more and more people do not want to disclose private information to everybody, such as personal preferences on resources. However, no privacy notion is possible when using centralized processes. The central entity must gather the resource list and the preferences of all agents in order to determine the best resource allocation, according to the objective. Thus, when privacy is required, centralized methods are not suitable. Two notions must be distinguished: selfishness and privacy. Even if agents want to keep some information private, they are not necessarily selfish. Agents may have a common objective and hence a cooperative behavior, but also do not want to report all private information. At the opposite, selfish agents generally refuse to disclose private information (Nisan, 1999; Sen, 1996). These notions are not equivalent and must be distinguished.

Communication abilities. Centralized processes provide allocations but do not consider the way to achieve them in practice. They assume that provided solutions can always be applied. It is also possible to find transaction sequences leading to such solutions using centralized methods, but it may not be scalable even with data instances of moderate size. They implicitly assume that any agent can communicate with all the other agents in the population. Such an assumption is not plausible for many applications. Indeed, in any application based on a community, an agent only knows a small subset of the overall population, and it can only talk to this subset. For instance, in a peer-to-peer network, a peer only knows a subset of the whole population. Centralized approaches do not focus on the way to achieve the provided resource allocations in practice. Since they assume complete communication abilities, they also assume that a resource can always circulate without restriction from its initial owner to its final owner in the final allocation. Thus, when restricted communication abilities are considered, the solutions provided by centralized techniques may not be achievable in practice. Restricted communication abilities interfere in the resource circulation, which may then prevent their achievement. The identification of a transaction sequence leading to optimal allocations cannot be solved in a scalable way by centralized approaches. When restricted communication abilities are considered, no simple test can determine whether or not a path of satisfying transactions exists leading to an optimal solution. The complexity of such problems is exponentially larger than the complexity of simple allocation problems.

1.2.3 A typical application: combinatorial auction

Centralized methods are very efficient for several classes of applications. Indeed, one of the most popular applications in Economics can be solved using centralized approaches. Auction problems have been widely studied (Bellosta et al, 2006; Boutilier et al, 1999; Cramton et al, 2006; De Vries and Vohra, 2003; Nisan, 2000; Sandholm, 2002). Various kinds of auctions exist, and different models can be used to solve them.

Auction problems fit very well with centralized solving methods. Indeed, agents represent auction clients. In practice, clients report their preferences over resources to an auctioneer, who is the central entity. This auctioneer can then determine optimal outcomes and allocates resources to clients accordingly.

Many kinds of auction exists (Krishna, 2002): English auction also called "open ascending price auction", dutch auction also called "open descending price auction", sealed first-price auction or first-price sealed-bid auction, Vickrey auction or sealed-bid second-price auction, ... These four types of auctions are only the most important, but many others exist. Each kind requires a specific model in order to be solved efficiently. This application class is very rich and many issues are still open (Lehmann et al, 2006; Sandholm, 2002).

Generally, centralized approaches are efficient when the applications exists have some suitable features. Any static application where no specific relationship among agents can be solved efficiently using centralized approaches. If no privacy is required or if we are just interested in results themselves and not in the way of achieving them, centralized solving techniques are favored.

1.3 Distributed approaches

Alternative methods have been developed in order to overcome the limits of centralized methods. These methods are based on the notions of agents and multi-agent systems (Ferber, 1999; Woolridge, 2001). Moreover, standard allocation problems become multi-agent resource allocation problems that can be solved thanks to agent negotiations. In contrast to centralized techniques, agent-based approaches are scalable and adaptable, i.e., large dynamic systems can be handled as well as restricted relationships among agents. We first describe the principles of these solving processes and identify the main characteristics of suitable applications. We then discuss several issues related to the importance of considering agent relationships. We also explain why provided allocations can be viewed as emergent phenomena. Finally, some application examples, for which centralized approaches are not adapted, are finally presented.

1.3.1 Description

The principles of distributed approaches (Moulin and Chaib-Draa, 1996) are fundamentally different from the centralized ones. In agent-based methods, agents participate actively in the determination of allocations optimizing the objective. Solving processes start here from initial resource allocations, which evolve step by step using local negotiations among agents. Such solving processes correspond to negotiation processes, which are illustrated in Figure 1.3.

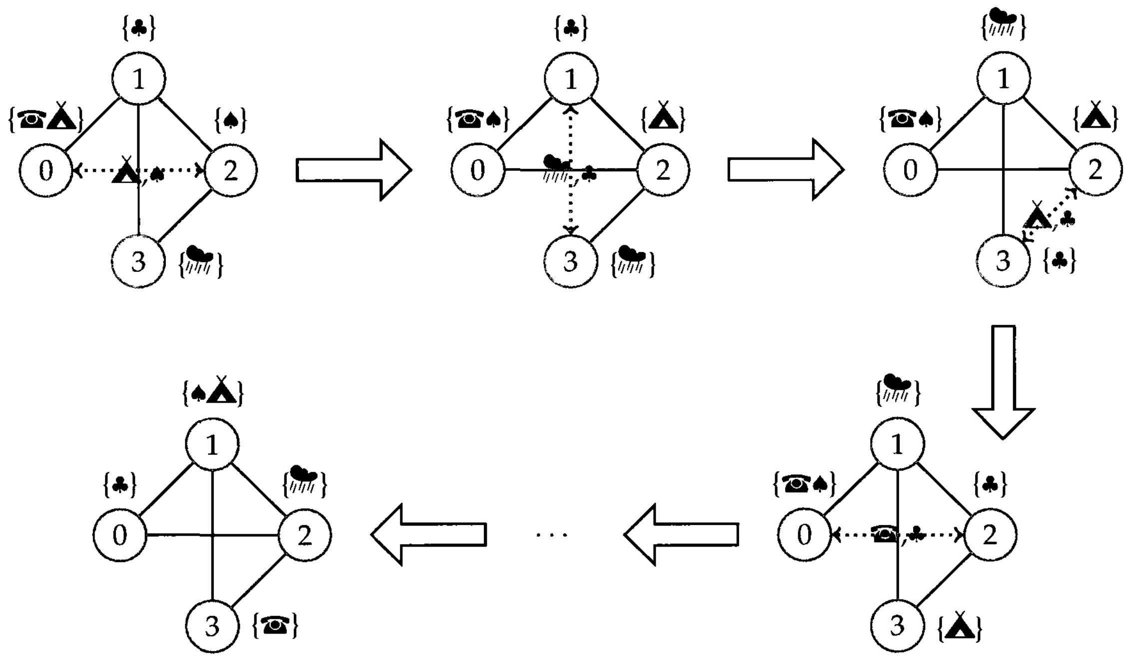


Figure 1.3: Principles of distributed methods

Figure 1.3 shows a negotiation process based on a population of 4 agents, $\mathcal{P} = \{0, 1, 2, 3\}$, and a set of 5 resources, $\mathcal{R}$ = {☎, ⛺, 🌧, ♣, ♠}. The communication possibilities are represented by a graph: Two nodes directly linked can communicate. Different steps of the negotiation process are illustrated. It starts from an initial resource allocation A_0 = [{☎,⛺}{♣}{♠}{🌧}]. Agents 0 and 2 negotiate first and finally provide two resources, respectively ♠ and ⛺. Exchanges are represented by dotted links on the figure. The initial resource allocation A_0 evolves into a new allocation A_1 = [{☎♠}{♣}{⛺}{🌧}]. Thus, a sequence of local transactions among agents leads to the final resource allocation, which constitutes the solution provided by the negotiation process.

Agent-based solving techniques handle resource allocation problems by considering different aspects. Situations for which distributed solving processes are more suitable than centralized ones are discussed next.

1.3.2 Characteristics

Adaptivity. Multi-agent systems are widely used to model dynamic phenomena. Resource allocation problems which are solved using a multi-agent system can model dynamic applications. Arrivals and departures of agents during the solving process do not lead to restart the whole process. Since multi-agent systems are naturally expandable, they can manage the continuous evolution of data.

Social graph. Relationships among agents can be considered and modeled using multi-agent systems. The communication possibilities of agents are represented thanks to social graphs. According to the topology of a social graph, two agents can communicate if they are directly related in the graph. Restricted communication abilities influence a lot the efficiency of negotiation processes since they restrict the resource circulation. Such restrictions are widely encountered in various applications. Especially in the case of large systems, like Internet, complete communications possibilities are neither possible nor plausible.

Applicability. Since the solution of resource allocation problems is the result of local negotiations among agents, a sequence of transactions is identified, from initial resource allocations to the final solutions. Thus, allocations provided by multi-agent approaches can always be achieved independently of the agents' communication abilities. However, the applicability should be moderated. A transaction sequence leading from the initial solution to the final solution is identified with respect to the agents' communication abilities. However, since agents only have a limited view of the system, they cannot be sure that the negotiation process will end. In order to be sure that negotiation processes end, a centralized coordinator is required.

Scalability. Multi-agent systems are also highly scalable compared to centralized approaches. Indeed, a multi-agent system is populated by autonomous agents. This characteristic allows an easy distribution of the computations. Very large populations can be handled in a scalable way since computation costs can be split over the population. In the case of negotiation processes, while one negotiation remains scalable, large populations can be handled. However, if the determination of a transaction requires an exponential time, it is obvious that the overall solving process cannot be scalable.

Heterogeneity. Multi-agent systems handle homogeneous populations, where all agents act according to the same behavior, as well as heterogeneous populations, where each agent acts according to its own behavior. Such notions are not taken into account in centralized techniques. Large heterogeneous populations can be managed quite easily using new design approaches, like the IODA methodology (Interaction Oriented Design Agent simulation) which focuses on the agent interaction instead of the agent behavior (Kubera et al, 2008). A simulation is characterized by a matrix, which defines the interactions that occur between agents with respect to their type. The combined use of interaction matrices with generic interactions allows convenient simulation designs. For instance, free-rider issues are widely studied in file sharing problems (Groves and Ledyard, 1977; Morge and Mathieu, 2007). In such applications, two kinds of agents can be distinguished: purely selfish agents who only get the media contents without providing anything to others, and generous agents who do both operations. They study the rate of free-riders in a population and its impact on the service efficiency. Their aim is to design agent behaviors discouraging others to act selfishly. Even if the management of heterogeneous populations is possible, this thesis only focuses on homogeneous populations. Indeed, studies on heterogeneous populations deal mainly with evolutionary issues (Hofbauer and Sigmund, 2003; Weibull, 1997), which is a topic beyond the scope of this study.

Information privacy. Depending on the required level of privacy, a negotiation can be designed in a suitable way. Depending on the quantity of information that agents accept to disclose, different negotiation protocols can be designed. Indeed, negotiation protocols influence the efficiency of solving processes. The more information agents accept to reveal, the more accurate are negotiations, and hence more efficient is the identification of acceptable transactions. Indeed, a negotiation protocol based on binary information is very limited, i.e., a yes/no answer contains only few information, whereas the expression of a degree of envy on resources may help to identify acceptable transactions.

The social graph: an essential issue?

Various studies have been carried out to solve multi-agent resource allocation problems, but only few of them consider restricted communication among agents. Distributed solving methods can model them using social graphs. Studies focusing on distributed techniques can be classified in two main classes. The first set of studies is mainly theoretical and aim to prove the existence of solutions or to identify mathematical properties favoring the achievement of optimal solutions. The second set of studies mainly concentrate on mechanisms required to achieve these solutions.

An approach to solve the task reallocation problem uses marginal cost for different classes of transactions (Sandholm, 1998). He analyzed the characteristics of local optima avoided by each transaction class. He also established theorems on the existence or not of transaction sequences leading from any initial resource allocation to optimal ones, depending on the transaction allowed during negotiation processes. These transaction classes and their efficiency have been assessed (Andersson and Sandholm, 1998) but the evaluation is restricted to a small number of resources with a small population (less than 10 resources and 10 agents). Each agent can always communicate with all other agents. Other authors consider the contract sequencing (Andersson and Sandholm, 2000). They described a protocol to solve the multi-agent Traveling Salesman Problem, and they compared strategies. However, only relative comparisons are performed and the communication possibilities are always assumed to be complete. Their work has been extended with studies mainly related to the transaction sequence length (Dunne, 2005). The author established bounds on the length of transaction sequences required to achieve optimal allocations. He also introduced a new transaction class and evaluated its efficiency. However, restricted communication possibilities were still not considered. A multi-agent system was proposed to solve distributed resource allocations relying on a market biding model(Chavez et al, 1997; Clearwater, 1996). These authors specifically addressed CPU time allocation problems. However, they did not compare the efficiency of their allocation processes with optimal solutions. Classes of utility functions and payment functions have also been studied to design convergent negotiation processes (Chevaleyre et al, 2005). Authors analyzed the mathematical properties of different functions and identified sufficient and necessary conditions to ensure the convergence of negotiation processes. These negotiation processes have been evaluated using social welfare theory (Moulin, 1986; Arrow et al, 2002). They established convergence results depending on the transaction classes allowed during the negotiations. They also studied different scenarios corresponding to different preference representations and to different acceptability criteria (Endriss et al, 2006). However, they never considered restricted agent's communication abilities. These studies designed abstract frameworks, but none of them is able to exhibit acceptable transaction sequences leading to optimal allocations. Moreover, none of these studies proposed the agent's behaviors to implement in order to negotiate efficiently in practice. Negotiation protocols were also designed where no common knowledge is available (Saha and Sen, 2007) or when agents express multi-criteria preferences (Hemaissia et al, 2007). However, the communication possibilities of agents are always assumed to be complete. Such an hypothesis restricts a lot the field of applicability once more.

Since the "agents' communication abilities" facet of resource allocation problems was not considered in former studies, it is legitimate to investigate the importance of such a parameter. Negotiation processes, which lead to optimal solutions according to complete communication possibilities (i.e., based on complete social graphs), may only lead to solutions far from the optimum, when communications are restricted.

Property 1.3 (Social graph impact). *Independently of the objective function considered, a re-*

stricted social graph may prevent the achievement of optimal resource allocations.

Proof. Let us prove this proposition using a counter-example, based on a population of 3 agents $\mathcal{P} = \{1, 2, 3\}$, and a set of 3 available resources $\mathcal{R} = \{r_1, r_2, r_3\}$. The agent's preferences are described in Table 1.4.

Table 1.4: Social graph impact - Example of agent's preferences

Population $\mathcal{P}$	Resource Set $\mathcal{R}$		
	r_1	r_2	r_3
0	3	1	9
1	1	4	1
2	10	2	3

The social graph is described in Figure 1.4. In this social graph, agents 0 and 2 cannot communicate. This figure also describes an initial resource allocation, which is $A = [\{r_1\}\{r_2\}\{r_3\}]$.

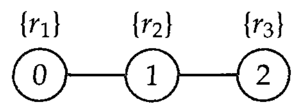


Figure 1.4: Social graph impact - Social graph example

Agents are assumed to be selfish in this example, i.e., they only accept transactions increasing their own utility. Under such conditions, no transaction can be performed as described in Table 1.5. This table lists the possible resource exchanges and shows that none increases the utility value of all participants. Only two exchanges are possible: between agents 0 and 1 who respectively exchange resources r_1 against r_2, and between agents 1 and 2 who then exchange resources r_2 against r_3. Both cases lead to a decrease of the utility of at least one participant.

However, the exchange of r_1 against r_3, which leads to an increase of the utility of both participants, is not possible since they cannot communicate according to the graph topology. If the utilitarian welfare is the social objective to maximize, then the allocation $A = [\{r_3\}\{r_2\}\{r_1\}]$ represents an optimal solution, which cannot be achieved since the agent communication possibilities are restricted. □

Table 1.5: Social graph impact - Set of possible transactions

Transaction	Agent's utility u_i		
	u_0	u_1	u_2
Initially	3	4	3
$r_1 \leftrightarrow r_2$	1	1	3
$r_2 \leftrightarrow r_3$	3	1	2

Restricted social graphs also have an indirect influence on the negotiation process. The order according to which agents negotiate is not important when complete social graphs are considered. Indeed, resources can always be traded with all other agents. However, this order becomes an important parameter when considering a restricted social graph.

Property 1.4 (Negotiation order). *Independently of the objective function which is considered, the order in which agents negotiate with each other may prevent the achievement of optimal resource allocations.*

Proof. Let us prove this proposition using a counter-example, based on a population of 3 agents $\mathcal{P} = \{1, 2, 3\}$, and a set of 3 available resources $\mathcal{R} = \{r_1, r_2, r_3\}$. The agents' preferences are described in Table 1.6.

Table 1.6: Negotiation order - Example of agent's preferences

Population $\mathcal{P}$	Resource Set $\mathcal{R}$		
	r_1	r_2	r_3
0	2	10	4
1	5	3	9
2	2	7	1

The social graph and the initial resource allocation are described in Figure 1.5. In this social graph, agent 0 cannot communicate to agent 2 and the initial allocation is: $A = [\{r_1\}\{r_2\}\{r_3\}]$.

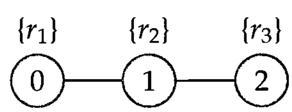


Figure 1.5: Negotiation order - Social graph example

Let us assume that agent 1 initiates a negotiation. Depending on which neighbor it selects to negotiate first, with respect to its behavior, the negotiation process may end with sub-optimal allocations instead of optimal ones. We assume here that the objective is the *maximization of the utilitarian welfare, but examples can be designed for all other notions.*

Table 1.7 lists the possible transactions depending on who the initiator selects to negotiate with. As described in this table, if agent 1 first chooses agent 0, the exchange leads to a sub-optimal allocation from which the negotiation process cannot leave. If the initiator first selects agent 2, then the negotiation process ends on a socially optimal allocation. Hence, the optimum can only be achieved using a specific order of negotiation.

Table 1.7: Utility of agents according to the initiator partner

Negotiating agents	Agent's utility u_i			Welfare
	u_0	u_1	u_2	sw_u
Initially	2	3	1	6
Agent 0 $\leftrightarrow$ Agent 1	10	5	1	16
Agent 1 $\leftrightarrow$ Agent 2	3	7	2	17

□

Thus, the social graph represents an important issue since its topology may prevent the achievement of an optimal resource allocation in practice. Its influence on the efficiency of negotiation processes must be considered and should not be omitted as it has been done in former studies.

An emergent phenomenon?

The concept of *"emergence"* is used by different communities but there still does not exist an agreement around a common definition (Corning, 2002; Goldstein, 1997; Serugendo et al, 2006). Indeed, there are as many definitions as users of this concept!

However, an operational definition of the concept of emergence is established (Capera et al, 2003). According to (Di Marzo Serugendo et al, 2006), this definition, proposed by computer scientists, is based on two main issues:

- **The subject**. The objective of a computational system is to achieve an adequate function. This function, which may evolve during time, has to emerge.

- **The condition**. This function is emergent if the coding of the system does not depend on any knowledge related to this function. This coding has to contain the mechanisms allowing the adaptation of the system, so as to tend anytime towards the adequate function.

According to this operational definition of emergence, our distributed approach based on agent negotiations provides resource allocations that can be viewed as emergent phenomena. Conditions to have an emergent phenomenon is to have a global objective function, and local mechanisms that have no knowledge on this objective function. In practice, agents have only a local view of the system. At most, they can collect information from their neighborhood with respect to the social graph topology. No agent is able to know what is exactly the current resource allocation at a given time. Agents only know their own resource bundle. It is then not possible for them to know the value associated with the objective function.

1.3.3 Application examples

Many problems studied by the Computer Science community can be modeled as resource allocation problems.

Over the past years, there has been an increasing interest for routing problems and network designs, in relation to self-organization issues (Serugendo et al, 2006). How a network is designed is an important issue that influences its efficiency. Multi-agent approaches can be used to design or maintain large networks (Anshelevich et al, 2008). Each edge of the network has a cost, and agents try to minimize this cost, according to some connectivity requirements for instance. Characteristics like performance or resilience of resulting networks can be studied (Chun et al, 2004). According to the cost function which is considered, various topologies can be generated, and the control of nodes degree, i.e the number of neighbors per node, appears to be of great importance in order to control the network balance. Recently, several studies addressed the design of peer-to-peer networks. Such networks are dynamic and their growth may alter the efficiency of the services that they provide. For instance, in the case of file sharing applications, bottlenecks may appear as a result of the constant evolution of the overlay. A proper adaptation of the overlay is essential to maintain the quality of service (Ni and Liu, 2004; Hales, 2004). Based on a specific overlay, selfish routing can be addressed using multi-agent resource allocation problems (Gairing et al, 2008; Gibney and Jennings, 1998). In such applications, agents have to assign traffic to one of their links. Generally, traffic of other agents is unknown. Efficient load-balancing can be achieved in this way based on any kind of topology. Several grid computing issues can also be modeled as multi-agent resource allocation problems. For instance, tasks should be uniformly split among different nodes in order to speed up computations (Buyya et al, 2002; Galstyan et al, 2005). Thus, load balancing can be performed using agent-based techniques. Usually, resources of such systems are CPU time.

Supply chain problems are based on a network of facilities, which perform procurement of materials and transformations of these materials into different products, intermediate products as well as finished products (Kaihara, 2003). Facilities can perform different tasks on products. According to the manufacturing process, tasks must be performed in a specific order. The dependencies among tasks that can be performed by facilities represent a social graph. The resources are the time of engine usage. Various criteria can be considered in such a problem. Material flows must be organized in order to maximize the production of facilities, to minimize the cost of transportation and distribution, to respect production delays, ... Some agent-based approaches are proposed to solve the distributed manufacturing scheduling (Shen, 2002; Sycara et al, 1991). The aim of such systems is to maximize the global efficiency. However, no client can be completely neglected and see its products manufactured too late. The Nash product seems to be a welfare notion of interest in such situations. Continuous planning issues under spatial constraints are also addressed (Sahli and Moulin, 2009). Spatially-aware agents are used to solve complex planning problems in real dynamic and large-scale spaces.

Applications based on social networks become more and more popular nowadays. Social networks regroup most of the time people who have elements in common such as preferences, geographic locations, friendships and blood relationships. Such a network represents the social graph on which is based an agent negotiation process. When agents are related, they have common interests and they are able to negotiate their resources. Recently, many Internet applications based on barter appear on the Internet. For instance, in services like `www.homexchange.com` or `www.gchangetout.com`, clients are related according to their preferences. They would like to lend their own house for a given number of weeks, in order to obtain the same number of weeks in another house located in an area corresponding to their wish. The aim of such a barter system is to satisfy all members of the community, which corresponds to an egalitarian problem. Such a system can be easily modeled by means of cooperative agents. All agents enter in the community bringing at least one resource: their house. The aim of all agents is similar. They want to find a house to exchange with their own house for a given vacation period. All agents express preferences on the kind of house they wish, on the location and on the time period. Resources that agents offer and wish from other agents represent connections among the agents, which constitute a social graph. Two agents are related if one of them offers a resource (i.e., an house during an acceptable time period) that interests the other, or if they have close interests. An agent who stays with its own house is a situation associated with a very weak satisfaction. However, when an agent who lends its own house but does not get any in return, this corresponds to the worst situation that is associated with either a negative utility if such values are allowed, or with a null utility otherwise. The aim is to provide an house to every community members during their vacation period.

1.4 Summary

In this chapter, allocation problems have been described and their three main characteristics have been presented.

- **Nature of resources**: Resource properties have an important influence on allocation problems. Depending on them, efficient solving processes in a given context may be completely inefficient in another context. We specifically address allocation problems in which resources are assumed to be *discrete, not divisible, not sharable,* and *static*.

- **Preferences representation**: Agent's preferences also deeply affect allocation problems. We choose to use a cardinal quantitative representation of preferences. Agents express them using an *additive utility function*. Compensatory payments are prohibited in this thesis.

- **Collective evaluation**: From the society's point of view, allocations can be evaluated thanks to social welfare notions and negotiation settings must be designed according by. In this thesis, four main notions are considered: the *utilitarian* welfare, the *egalitarian* welfare, the *Nash* welfare and the *elitist* welfare.

Centralized solving approaches have been described. They are well-adapted in some cases, e.g., for auction problems, but are not applicable when applications are dynamic or when privacy is required by agents. In this context, we discussed distributed methods. These two kinds of approaches do not address the same problems. Based on agent negotiations, distributed methods provide a sequence of transactions to achieve the provided solutions in practice, and they can handle restricted communication possibilities. A suitable design is nevertheless required to achieve allocations as emergent phenomena. The next chapter is dedicated to the negotiation design and provides a description of the different parameters that we choose to consider to set up negotiation processes.

Chapter 2

Distributed Negotiations

In this chapter, multi-agent negotiation problems are described. The combined use of negotiations and multi-agent systems raise several important issues, which are never addressed in centralized approaches. We will discuss issues related to the design of solving processes, focusing on agent behaviors and trying to answer a crucial question: "How do agents need to behave in order to lead negotiation processes to socially optimal solutions?"

The challenges related to the design of negotiation processes are described in Section 2.1, where we explain the difficulties related to such a process. Then, formal definitions of negotiation problems and agents are presented in Section 2.2. Their features are successively detailed in the next sections. Restrictions on agent communications are presented in Section 2.3, which describes different topologies of relationships that can be modeled and have their typical characteristics. Then, different classes of transactions are presented in Section 2.4. Section 2.5 details the decision-making criteria used by agents, i.e., the conditions that transactions must satisfy in order to be performed. Section 2.6 discusses agent behaviors, and presents different methods to model agents' interactions. Finally, we present in Section 2.7 the evaluation of negotiation processes and the metrics used during the experiments that we carried out.

2.1 Challenges

The design of negotiation processes is an essential issue affecting a lot their efficiency. Different techniques based on multi-agent systems have been described in Section 1.3 with an emphasis on their advantages, but a suitable design is required to benefit from them. The challenges related to the negotiation process design can be illustrated by a proposition of (Sandholm, 1998), which was initially written in a task allocation context.

Property 2.1 ((Sandholm, 1998), path). *A sequence of resource purchases (O-contracts) always exists from any resource allocation to the optimal one. The length of the shortest path is at most m (the overall number of resources).*

Proof. The transaction sequence can be constructed by moving a resource one at a time from the agent that initially owns it to the agent that gets it in the globally optimal resource allocation. □

According to Proposition 2.1, a path of *O-contracts*, which corresponds to the purchase of a resource, always exists between any initial allocation to an optimal one. Although this proposition is proposed in the case where communications are not restricted, it is still valid in the context of restricted communication possibilities, if no agent group is isolated from the others. Indeed, the proof is based on the existence of a path between any pair of nodes in the graph, which is always satisfied when the graph is complete. Such a path also exists if the restricted graph is connected. However, the length of the shortest path can be longer than m depending the agents' relationships. Resources may have to go through different bundles in order to achieve an optimal solution. In practice, the main issue is to identify such a path.

In order to find it, agents have to accept any transaction. Resources should then freely circulate among the agents in order to finally be owned by the agent that has it in a globally optimal resource allocation. According to agent-based method principles, the initial allocation evolves thanks to local transactions among agents. However, since agents must accept any transaction, they do not distinguish a profitable transaction from another one. Thus, agents negotiate endlessly and the negotiation process cannot end. Even if an optimal resource allocation is achieved during the solving process, agents continue to negotiate and perform new transactions. This optimal resource allocation is then lost.

Agents must be able to identify profitable transactions. Agents should be autonomous and their decision-making should be based on a local acceptability criterion. This criterion must only be based on the information that agents can get themselves during a negotiation. When no agent can identify an acceptable transaction within its neighborhood, the negotiation process is over. The resource allocation achieved at this moment is the solution provided by the distributed negotiations. This is an important issue in order to ensure the quality of achieved allocation. Negotiation processes must be finite, agents must be able to make their own decision based on local information until no agent identifies acceptable transactions. These features can be achieved through a suitable design of agent behaviors and the choice of an appropriate acceptability criterion.

2.2 Definitions

A resource allocation problem in an agent society can be solved thanks to negotiation processes among agents. Such problems can be distributed using the notion of agent. Instead of maintaining an up-to-date state of the whole system and of all entities' information in a single location, they are distributed inside agents. The notions of *negotiation problem* and *agent*, which are closely related, can be defined as follows:

Definition 2.1 (Negotiation Problem). A negotiation problem is a tuple $\langle \mathcal{P}, \mathcal{R}, \mathcal{T} \rangle$, where $\mathcal{P} = \{1, \ldots, n\}$ is a finite population of n agents, $\mathcal{R} = \{r_1, \ldots, r_m\}$ is a finite set of m resources, and $\mathcal{T}$ corresponds to the set of transactions allowed during the negotiation process.

The transactions that can be allowed are presented later, in Section 2.4. An agent can be defined in a generic way by a resource bundle, a utility function describing its preferences, a list of agents with whom it is able to communicate, a behavior describing how it interacts with other agents and an acceptability criterion related to its decision-making.

Definition 2.2 (Agent). An agent $i \in \mathcal{P}$ is a tuple $\langle \mathcal{R}_i, u_i, \mathcal{N}_i, \mathcal{B}_i, C_i \rangle$, where $\mathcal{R}_i$ is its set of m_i resources, u_i is its utility function, $\mathcal{N}_i$ is the list of n_i neighbors, $\mathcal{B}_i$ defines the agent behavior according to which the agent negotiates, and C_i is its acceptability criterion on which are based its decisions.

Preferences of agent i are defined according to Definition 1.3 by means of additive utility functions. The behavior $\mathcal{B}_i$ describes agent i from an external point of view, whereas the acceptability criterion C_i describes it from an internal point of view. Indeed, the behavior describes how an agent interacts with others while the acceptability criterion corresponds to the conditions that a transaction must satisfy in order to be performed. This criterion thus represents the central condition of the agent decision-making. Behaviors and acceptability criteria are respectively detailed in Sections 2.5 and 2.6.

Two notions must be distinguished. First, a *negotiation* seeks to identify an acceptable transaction among agents. A negotiation is defined by interactions among agents (see Section 2.6). A *negotiation process* seeks to find a path of acceptable transactions, and thus includes many negotiation problems. In this thesis, we always consider a specific negotiation problem based on an agent population $\mathcal{P}$, a set of resources $\mathcal{R}$ which are initially distributed over the population, and a set of transactions $\mathcal{T}$ allowed among agents of the population.

2.3 Social graphs

At the opposite of centralized solving processes, which always assume complete communication possibilities, solving methods based on multi-agent systems can handle the notions of *neighborhood* and *social graph*.

Definition 2.3 (Neighborhood). The neighborhood of agent $i \in \mathcal{P}$, denoted by $\mathcal{N}_i$, is a subset of the population $\mathcal{P}$ with whom it is able to communicate.

$$\mathcal{N}_i \subseteq \left(\mathcal{P} \setminus \{i\}\right), \qquad i \in \mathcal{P}.$$

A graph of relationships, which we call a social graph, can be extracted from the neighborhood of all agents.

Definition 2.4 (Social graph). The social graph $\mathcal{G}$ is a graph of relationships describing the communication possibilities among agents of a population $\mathcal{P}$. In such a graph, nodes represent agents, and an edge between two nodes means that the corresponding agents are able to communicate.

Property 2.2 (Relationship symmetry). *Let e_{ij} be an edge of a social graph $\mathcal{G}$ between two nodes i and j. This edge means that both agents $i, j \in \mathcal{P}$ are directly related. If agent j is a neighbor of agent i, then agent i must also be a neighbor of agent j. More formally:*

$$e_{ij} \in \mathcal{G} \quad \Rightarrow \quad j \in \mathcal{N}_i \ \text{ and } \ i \in \mathcal{N}_j, \qquad i, j \in \mathcal{P}.$$

Such relationships are represented by non-oriented graphs.

The different classes of social graphs can be grouped into three main classes:

- Complete graphs;
- Structured graphs;
- Random graphs.

First, negotiation processes based on complete social graphs can be compared to centralized approaches. Indeed, both of them assume complete communication possibilities among all the agents, and then have similar solving conditions. However, such graphs are only used to carry out comparisons between distributed results and the ones provided by centralized techniques.

Then, graphs with regular topological characteristics belong to the class of structured graphs. For instance, a graph where all agents have the same number of neighbors belongs to this class. Structured graphs also include some specific topologies like grids (Berman et al, 2003), rings or trees.

Finally, unstructured graphs have an irregular topology. Several classes of random graphs exist (Bollobás, 2001), like Erdős-Rényi graphs (Erdős and Rényi, 1959), free-scale graphs or small worlds generated either by preferential attachment or by circle rewiring (Albert and Barabási, 2002). The probability distribution is uniform when Erdős-Rényi graphs are considered. Links between any pair of nodes have the same probability to be generated. In small-worlds, the larger number of neighbors has a node, the larger is the probability to link this node. Such topologies correspond to real-life graphs like the Internet. The algorithms used to generate the classes of social graphs considered in this thesis are detailed in Chapter 3.

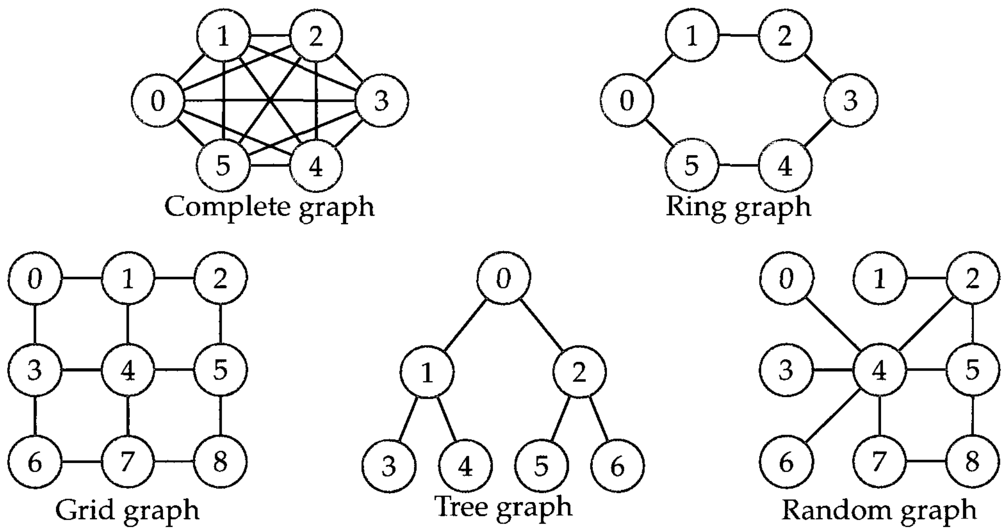


Figure 2.1: Classes of social graphs

Different classes of social graphs are illustrated in Figure 2.1. We will select specific graphs for the experiments in Chapter 3: complete graphs, grids, Erdős-Rényi graphs and small-worlds. These graphs correspond to a representative sample of what can be encountered in most applications. Indeed, their characteristics vary significantly when these graphs are evaluated with the most widely used metrics (Biggs et al, 1986), which are described in the next paragraphs.

The **mean connectivity**, which corresponds to the mean number of neighbors, goes from $n-1$ with complete graphs (where n is the overall number of agents) down to four neighbors in grids. If n_i is the number of neighbors of agent i, it can be evaluated as follows:

$$connectivity = \frac{1}{n} \sum_{i \in \mathcal{P}} n_i.$$

The **clustering coefficient** is a metric that quantifies how well connected is the neighborhood of an agent. The more the neighbors of an agent are directly related, the higher is the clustering coefficient. This coefficient is weak when grids are considered since the four neighbors of an agent are not related directly, whereas the clustering coefficient becomes high when complete graphs are considered since the neighborhood of an agent is completely connected.

$$clustering = \frac{1}{n} \sum_{i \in \mathcal{P}} \frac{2|\{e_{jk}\}|}{n_i(n_i - 1)} \qquad j, k \in \mathcal{N}_i, e_{jk} \in \mathcal{G};$$

with n the overall number of agents, n_i the number of neighbors of agent i, and e_{jk} an edge between two neighbors j, k of agent i.

Another important characteristic is the **mean-shortest path length**. This metric describes the mean closeness among agents. In grids, two agents may be far from each other (e.g., the opposite corners of a grid) whereas in small-worlds, the mean distance between any pair of agents is really small. If $d^{ij}_{\min}$ denotes the shortest path length between two agents $i, j \in \mathcal{P}$, it can be computed as follows:

$$mean\ shortest\ path\ length = \frac{2}{n(n-1)} \sum_{(i,j) \in \mathcal{P}} d^{ij}_{\min}.$$

Only connected graphs are considered in this thesis. In such graphs, a path always exists between any pair of nodes, and its maximal length is $n-1$. If a social graph is not connected, then agents from disconnected parts cannot communicate with each other. Resources can circulate inside a portion of the graph, but cannot move to another one. Hence, these portions can be considered as independent. Thus, a resource allocation problem, which is based on a not connected social graph, can be split into as many independent sub-problems as there distinct portions in the social graph.

Property 2.3 (Not connected graph). *Independently of the objective considered, the solution of an allocation problem based on a non-connected social graph is equivalent to the association of the partial solutions provided by the solving process applied to all the portions of the graph.*

Proof. If a social graph is composed by x portions, there is no path between nodes that belong to different portions. Any pair of agents who do not belong to the same group cannot communicate. Then, no resource can circulate from one of them to another. The solution of the overall problem can be obtained by the union of the solution from each portion. The resource allocation problem can be divided into x independent sub-problems, each one restricted to the population of a portion. The optimal allocations provided by the different solving processes can be merged to constitute the solution of the whole problem. □

2.4 Transactions

During negotiation processes, the resource allocation evolves, step by step, by means of local transactions among agents. The resource traffic is generated thanks to these transactions, which move resources successively from an agent bundle to another one.

Definition 2.5 (Transaction). A transaction is an operation on resources among several agents, which transforms an initial resource allocation A into a new one A'. Agents involved in a transaction are called *participants*, but two agent roles can be distinguished: the *initiator* who starts the negotiation, and the *partners* who are selected by the initiator in its neighborhood.

The definition of a transaction can be based on the offers made by the participants. The number of resources that the participants can offer depends on the allowed transaction class. Indeed, different classes of transactions exist, which are more or less time-consuming, which involve more or less agents, which move more or less resources . . . Three classes of transactions can be distinguished.

- Bilateral transactions, which involve only two agents at a time (also called one-to-one transactions in the literature);
- Multilateral transactions, which involve more than two agents at once according to two different transaction patterns:
  - One-to-Many transactions, where the initiator is involved in all resource operations;
  - Many-to-Many transactions, where any resource operation is allowed among all the participants.

In this section, these three classes of transactions are described. For each class of transactions, the computational complexity, i.e., the number of possible transactions, is determined according to the number of participants and according to the size of their bundle.

2.4.1 Bilateral transactions

Bilateral transactions, also called one-to-one transactions, only involve two agents at a time. They represent the most widely used class of transactions in the literature. Bilateral transactions can be defined in a parametric way using two parameters representing the size of the offers proposed respectively by the initiator and its partner, as illustrated in Figure 2.2.

Definition 2.6 (Bilateral transactions). A bilateral transaction between two agents $i, j \in \mathcal{P}$, denoted by δ_i^j, is initiated by agent i who involves one of its neighbors $j \in \mathcal{N}_i$. It is a pair $\delta_i^j\langle a, b\rangle = (\rho_i^\delta, \rho_j^\delta)$, where the initiator i offers a set ρ_i^δ of a resources ($\rho_i^\delta \subseteq \mathcal{R}_i$) and the selected partner j offers a set ρ_j^δ of b resources ($\rho_j^\delta \subseteq \mathcal{R}_j$).

Let us recall the resource allocations' properties which are satisfied according to the

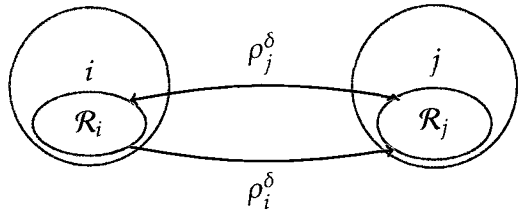


Figure 2.2: Bilateral transactions

nature of the resources that we choose to consider (see Section 1.1.1):

$$\bigcap_{i \in \mathcal{P}} \mathcal{R}_i = \emptyset \quad \text{and} \quad \bigcup_{i \in \mathcal{P}} \mathcal{R}_i = \mathcal{R}.$$

Hence, the intersection of the offers proposed by different agents is always empty. The nature of the considered resources affects the definition and the properties of all transactions. According to the size of the offers proposed by both participants, the possible number of bilateral transactions is restricted.

Property 2.4 (Bilateral transaction complexity). *Let us consider a bilateral transaction $\delta_i^j\langle a, b\rangle$, where the initiator $i \in \mathcal{P}$ owns initially m_i resources and offers a resources ($a \leq m_i$), and where it involves a neighbor $j \in \mathcal{N}_i$ who initially owns m_j resources and offers a set of b resources ($b \leq m_j$).*

The possible number of bilateral transactions of cardinality $\langle a, b\rangle$ between agents i and j is:

$$\#\delta_i^j = \binom{m_i}{a} \times \binom{m_j}{b}.$$

According to Proposition 2.4, the number of possible bilateral transactions grows exponentially with the size of the offers a and b proposed by the participants. Some negotiation policies allow transactions of different cardinality during the same negotiation process. Indeed, even if an agent i owns m_i resources, it may be useful to bound the number of resources it can offer in order to reduce the complexity. Under such conditions, two negotiating agents may offer a resource set of any size from a single resource up to a' and b', which bounds the size of the offered resource sets. We call such negotiation policies "up to

$\langle a', b' \rangle$". The number of possible bilateral transactions can then be determined as follows:

$$\#\delta_i^j = \left(\sum_{x=0}^{a'} \binom{m_i}{x} \right) \left(\sum_{x=0}^{b'} \binom{m_j}{x} \right) - 1.$$

Any bilateral transaction, e.g., defined in (Sandholm, 1998), can be defined by this representation, and the possible number of transactions can be evaluated using Proposition 2.4. Bilateral transactions with specific cardinality are presented in the following paragraphs.

A **gift**, also called O-contract, is a transaction where the initiator offers a single resource and its partner none. The gift of resource $r \in \mathcal{R}_i$ from agent i to agent j is represented by $\delta_i^j\langle 1, 0 \rangle$ with $\rho_i^\delta = \{r\}$ and $\rho_j^\delta = \emptyset$. Only m_i gifts are possible.

A **cluster** transaction, also called C-contract, is a transaction where the initiator offers a set of resources and its partner none. Hence, the cluster of a resources from agents i to agent j is represented by $\delta_i^j\langle a, 0 \rangle$ with $\rho_i^\delta = \{r_1, \ldots, r_a\} \subseteq \mathcal{R}_i$ and $\rho_j^\delta = \emptyset$. Then, $\binom{m_i}{a}$ cluster transactions are possible, which correspond to the number of sets containing exactly a resources in $\mathcal{R}_i$. If the negotiation policy allows agents to offer up to their whole bundle, i.e., if the size of the offers that agents may propose is not bounded, the number of possible clusters is $2^{m_i} - 1$.

A **swap** transaction, also called S-contract, is a transaction where both participants offer each other a single resource. A swap between agents i and j, who respectively exchange resources $r \in \mathcal{R}_i$ and $r' \in \mathcal{R}_j$, is represented by $\delta_i^j\langle 1, 1 \rangle$ with $\rho_i^\delta = \{r\}$ and $\rho_j^\delta = \{r'\}$. In a such case, $m_i \times m_j$ swaps are possible.

Finally, a **cluster-swap** transaction is the general form of bilateral transactions, where both agents offer a subset of their bundle. Then, a cluster swap between agents i and j, who respectively offer a set of a and b resources, is represented by $\delta_i^j\langle a, b \rangle$ with $\rho_i^\delta = \{r_1, \ldots, r_a\} \subseteq \mathcal{R}_i$ and $\rho_j^\delta = \{r'_1, \ldots, r'_b\} \subseteq \mathcal{R}_j$. Thus, $\binom{m_i}{a}\binom{m_j}{b}$ cluster-swaps are possible when agent i exactly offers a resources and agent j exactly offers b resources. If the negotiation policy allows the agents to offer up to their whole bundle, the number of possible cluster-swap transactions is $2^{m_i + m_j} - 1$.

Let us note that cluster transactions include gifts. Cluster-swap transactions, which are the general form of bilateral transactions, contain gifts, clusters and swaps. Figure 2.3 shows the inclusion relationships among the different bilateral transactions.

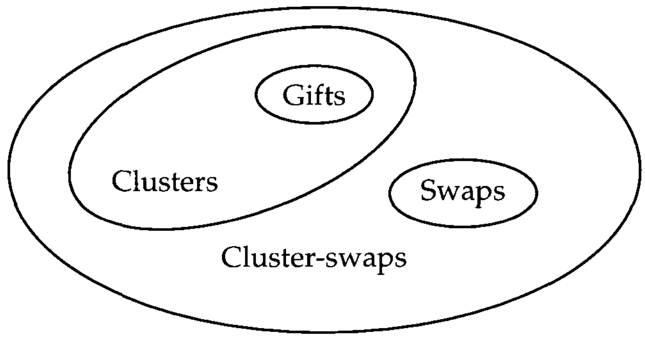


Figure 2.3: Relationships among bilateral transactions

According to the cardinality parameters, these relationships are pretty obvious to prove. Indeed, according to the definition of cluster transactions $\delta_i^j\langle a, 0\rangle$, the offer of the initiator is constrained by $a \leq m_i$. If the initiator only provides a single resource, the transaction δ_i^j corresponds to a gift. Similarly, according to the definition of cluster-swap transactions $\delta_i^j\langle a, b\rangle$, the size of the participant offers are bounded by $a \leq m_i$ and $b \leq m_j$. If both agents only provide a single resource, it corresponds to a swap, if the partner does not provide anything, the transaction corresponds either to a cluster or to a gift. Let us note that the situation in which both agents do not provide any resource is not considered as a transaction.

In this section, all bilateral transactions have been presented. Let us summarize the complexity of the different classes of bilateral transactions in Table 2.1. Let us consider a bilateral transaction δ_i^j between two agents $i, j \in \mathcal{P}$. They own respectively m_i and m_j resources. The total number of possible transactions is defined according to the restriction on the agent offers.

Table 2.1: Summary - Complexity of bilateral transactions

Transaction δ_i^j	Allowed transaction $\mathcal{T}$	Number of possible transactions
Gift	$\langle 1,0 \rangle$	m_i
Cluster	$\langle a,0 \rangle$	$\binom{m_i}{a}$
	up to $\langle a,0 \rangle$	$\sum_{x=0}^{a} \binom{m_i}{x} - 1$
Swap	$\langle 1,1 \rangle$	$m_i \times m_j$
Cluster-swap	$\langle a,b \rangle$	$\binom{m_i}{a}\binom{m_j}{b}$
	up to $\langle a,b \rangle$	$\left(\sum_{x=0}^{a} \binom{m_i}{x}\right)\left(\sum_{x=0}^{b} \binom{m_j}{x}\right) - 1$

2.4.2 Multilateral transactions

Another important class of transactions can be used to modify resource allocations. Indeed, while bilateral transactions only involve two agents at once, multilateral transactions involve more than two agents. During a multilateral transaction, the initiator is able to involve a set of neighbors at once. Two multilateral transaction patterns exist: One-to-many transactions and many-to-many transactions, as described next.

One-to-many transactions

During **one-to-many** transactions, the initiator is able to negotiate simultaneously with a whole subset of its neighborhood. The initiator can offer to each partner a set of resources, and inversely, all partners can offer a resource set to the initiator, as illustrated in Figure 2.4.

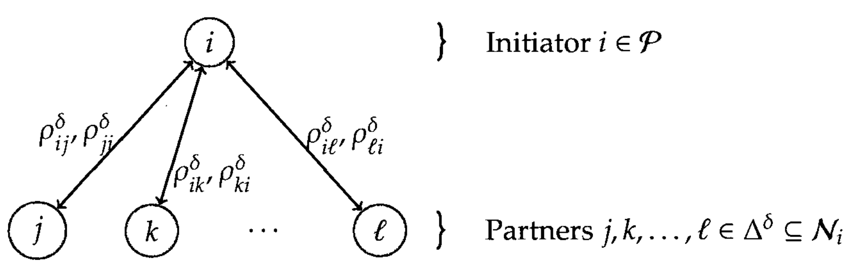


Figure 2.4: One-to-many transaction

The main constraint of this transaction pattern is that the initiator is always involved in any resource move. Resource sets are either offered or received by the initiator. Two partners cannot negotiate with each other. One-to-many transactions can then be formally defined as follows.

Definition 2.7 (One-to-many transactions). A one-to-many transaction δ_i^Δ is initiated by an agent $i \in \mathcal{P}$ and involves a subset of n_δ neighbors $\Delta^\delta \subseteq \mathcal{N}_i$. It is a list of pairs describing the resource sets given and received by all participants. Let $\rho_{k\ell}^\delta$ denote the resource set given by agent k to agent ℓ. The initiator must always be involved in all offers, either to provide or to receive them.

$$\delta_i^\Delta = \{(\rho_{ij}^\delta, \rho_{ji}^\delta) \quad | \quad j \in \Delta^\delta\} \qquad i \in \mathcal{P}.$$

Let us note that, according to Definition 2.3, $j \in \Delta^\delta \Rightarrow i \neq j$ in this definition. Obviously, the resources provided by the initiator are constrained. The initiator cannot give the same resource to different neighbors (due to the nature of the considered resources). Such constraints are required to ensure consistency during a multilateral negotiation. More formally:

$$\begin{aligned}
&\rho_{ji}^\delta \subseteq \mathcal{R}_j, && i \in \mathcal{P}, \quad j \in \Delta^\delta;\\
&\bigcup_{j \in \Delta^\delta} \rho_{ij}^\delta \subseteq \mathcal{R}_i, && i \in \mathcal{P};\\
&\bigcap_{j \in \Delta^\delta} \rho_{ij}^\delta = \emptyset, && i \in \mathcal{P}.
\end{aligned}$$

One-to-many transactions are more complex than bilateral transactions, and then more time-consuming. Indeed, a one-to-many transaction can be considered as several bilateral transactions performed simultaneously.

Property 2.5 (One-to-many transaction split). *A one-to-many transaction δ_i^Δ is equivalent to at most n_δ simultaneous bilateral transactions.*

Proof. According to the definition of a one-to-many transactions δ_i^Δ, the initiator i involves at most n_δ neighbors in the transaction. Since the initiator is always involved, it corresponds to a list of at most n_δ elements. Indeed, any agent $k \in \Delta^\delta$ from the initiator neighborhood who either gives or receives resources leads to the addition of a pair of resource sets $(\rho^\delta_{ij}, \rho^\delta_{ji})$. However, a partner may not be interested in the transaction and hence do nothing. Each pair $(\rho^\delta_{ij}, \rho^\delta_{ji})$ corresponds exactly to a bilateral transaction by definition. □

Property 2.6 (One-to-many transaction complexity). *Let us consider a one-to-many transaction, where agent $i \in \mathcal{P}$ is the initiator. It initially owns m_i resources and involves a set $\Delta^\delta \subseteq \mathcal{N}_i$ of n_δ neighbors. Each neighbor $j \in \Delta^\delta$ initially owns a set of m_j resources. The possible number of one-to-many transactions is:*

$$\#\delta_i^\Delta = (n_\delta + 1)^{m_i} \left| 2 \sum_{j \in \Delta^\delta} m_j \right| - 1.$$

Proof. To count the number of possible one-to-many transactions, we consider the resource allocations themselves. In other words, we determine where resources can be allocated according to their initial owner.

According to Definition 2.7, resources initially owned by a partner of Δ^δ can be either allocated to the initiator i, or stay in the resource bundle of their initial owner. It represents $2 \sum_{j \in \Delta^\delta} m_j$ allocations. The resources owned by the initiator can be allocated either to itself or to any partner, which represents $(n_\delta + 1)^{m_i}$ different allocations. The combination of both parts corresponds to the overall number of possible allocations resulting from a one-to-many transactions. Since the situation in which no modification is performed is not consider as a transaction, we finally deduct the initial allocation. □

Many-to-many transactions

During a **many-to-many** transaction, the initiator i and the set of partners constitute a group where everything is allowed. The main restriction of one-to-many transactions is omitted: The initiator is not anymore necessarily involved in all resource moves. Any agent of this group can offer a set of resources to any other agents of the group, as described in Figure 2.5.

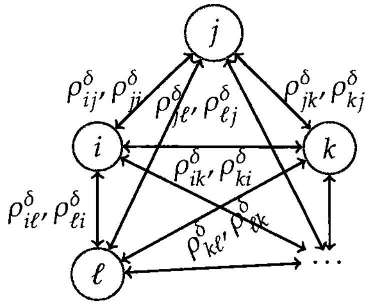


Figure 2.5: Many-to-many transactions

In many-to-many transactions, the role of the initiator is depreciated since any agent of the group can negotiate with the others. However, in practice, since two neighbors are not necessarily directly linked according to the social graph topology, the initiator can be used for the transit of the traded resources. A many-to-many transaction can be formally defined as follows:

Definition 2.8 (Many-to-many transactions). A many-to-many transaction δ_i^Δ is initiated by agent $i \in \mathcal{P}$ and involves a subset of its neighbors $\Delta^\delta \subseteq \mathcal{N}_i$. It can be defined as a list of pairs describing the resource sets offered and received by two agents $j, k \in \Delta^\delta \cup \{i\}$.

$$\delta_i^\Delta = \{(\rho_{jk}^\delta, \rho_{kj}^\delta) | j, k \in \Delta^\delta \cup \{i\}, j < k\}.$$

where ρ_{jk}^δ is the resource set given by agent j to agent k and inversely, ρ_{kj}^δ corresponds to the resource set given by agent k to agent j.

Only constraints ensuring the consistency must be satisfied: A given resource can only be offered to another agent. These constraints depend on the resource nature.

$$\rho_{jk}^\delta \subseteq \mathcal{R}_j, \qquad j, k \in \Delta^\delta \cup \{i\}.$$

Many-to-many transactions are more complex than one-to-many transactions, which are themselves more complex than bilateral transactions. While a one-to-many transaction can be viewed as several bilateral transactions performed at the same time, a many-to-many transaction can be considered as several simultaneous one-to-many transactions.

Property 2.7 (Many-to-many transaction split). *A many-to-many transaction is thus equivalent to at most n_δ simultaneous one-to-many transactions or to $n_\delta(n_\delta - 1)$ simultaneous bilateral transactions.*

Proof. According to the definition of a many-to-many transactions δ_i^Δ, the initiator i involves at most n_δ neighbors in the transaction. $n_\delta(n_\delta - 1)$ pairs of participants can be constituted. Each pair of participants who provide or receive resources corresponds to an element of the list. Each pair $(\rho_{jk}^\delta, \rho_{kj}^\delta)$ exactly corresponds by definition to a bilateral transaction. Then, a many-to-many transaction corresponds to $n_\delta(n_\delta - 1)$ simultaneous bilateral transactions. If we regroup the different pairs of offers according to the agent providing the first offer, it corresponds to several one-to-many transactions. □

Property 2.8 (Many-to-many transaction complexity). *Let us consider a many-to-many transaction δ_i^Δ, where agent $i \in \mathcal{P}$ is the initiator. It initially owns m_i resources and involves a set $\Delta^\delta \subseteq \mathcal{N}_i$ of n_δ neighbors. Each neighbor $j \in \Delta^\delta$ initially owns a set of m_j resources. The possible number of many-to-many transactions is:*

$$\#\delta_i^\Delta = (n_\delta + 1)^{m'} - 1, \qquad \textit{where} \quad m' = \sum_{j \in \Delta^\delta \cup \{i\}} m_j.$$

Proof. As done for one-to-many transactions, the most convenient way to proceed is to consider the resource allocations that can be achieved thanks to a many-to-many transaction, and then where resources can be allocated.

According to Definition 2.8, any resource can be allocated either to the initiator or to an involved agent. m' corresponds to the number of resources available in this restricted population $i \cup \Delta^\delta$. Each resource can be allocated to any agent of the restricted population, so $(n_\delta + 1)$ times. Applied to all resources, it corresponds to the number of possible many-to-many transactions. The initial allocation must also be subtracted since no transaction is performed. □

Many-to-many transactions without restriction on the size of the offered resource sets can be considered as a reallocation of all available resources over the restricted population of involved agents. On a complete social graph, an initiator involving its whole neighborhood in a many-to-many transaction, involves the whole population since all agents are related. In such conditions, many-to-many transactions are equivalent to classical centralized solving methods.

2.5 Acceptability criteria

Agents must locally decide about which actions to perform when several actions are possible. They must be able to determine the best action. Such a decision is based on an acceptability criterion. It strongly influences the negotiation process. Indeed, it highly restricts the set of possible transactions among the agents, by defining the conditions that transactions must satisfy in order to be performed. Accordingly, agents can determine whether or not transactions are profitable. Such criteria can be based on different notions. In this thesis, two main notions are considered: Rationality and sociability. These notions are successively described, and are finally compared in order to emphasize their differences.

Let the criteria defined in this section be illustrated by means of a transaction δ_i^Δ. This transaction changes the initial resource allocation A into a new one A'. Then, let $\mathcal{R}_k$ denote the resource bundle of any agent $k \in \mathcal{P}$ in the allocation A and $\mathcal{R}'_k$ its bundle in A'. The following definition can be restricted to the bilateral case.

2.5.1 Rational criterion

The individual rationality is the most widely used criterion in the literature. It specifies that agents can only accept transactions increasing their individual welfare. It is used especially in the case of selfish agents.

Definition 2.9 (Rational agent). A rational agent only accepts a transaction that increases its own utility value. If the agent $i \in \mathcal{P}$ is rational, an acceptable transaction must satisfy the following condition:

$$u_i(\mathcal{R}_i') > u_i(\mathcal{R}_i), \qquad i \in \mathcal{P}, \quad \mathcal{R}_i, \mathcal{R}_i' \subseteq \mathcal{R}.$$

Definition 2.10 (Rational transaction). A rational transaction δ, initiated by agent $i \in \mathcal{P}$, is a transaction involving rational agents only. Participants accept rational transactions only if the following condition is satisfied:

$$u_j(\mathcal{R}_j') > u_j(\mathcal{R}_j), \qquad j \in \Delta^\delta \cup \{i\}, \quad \mathcal{R}_i, \mathcal{R}_i' \subseteq \mathcal{R}, \quad \Delta^\delta \subseteq \mathcal{N}_i.$$

This criterion can be computed using local information only. Indeed, the current resource bundle, the offer an agent makes and the ones it receives are the only information required. These information are available locally from the agent's point of view. However, the rationality criterion strongly restricts the transaction possibilities. Its use may thus prevent the achievement of socially optimal allocations.

2.5.2 Social criterion

With respect to the social criterion, the welfare of the whole society cannot decrease. Sociability is more flexible than rationality. Social agents are usually qualified as generous.

Definition 2.11 (Social agent). A social agent is an agent who only accepts transactions that increase the welfare of the whole society.

Definition 2.12 (Social transaction). A social transaction δ, which changes the initial resource allocation A to a new one A', is a transaction leading to an increase of the social welfare.

$$sw(A') > sw(A), \qquad A, A' \in \mathcal{A}.$$

The social criterion is centered on the social welfare value, which is a global notion. Its value can only be determined thanks to the welfare of all agents. Agents should then know the resource bundle and the preferences of all agents in the population, in order to determine the value associated with the objective function. Such conditions cannot be satisfied since agents have only local information. The social value of the objective cannot then be locally computed. But, the computation of the exact value of the welfare function is not essential, to know its evolution is sufficient to determine whether or not a transaction penalize the society. Such computations can be restricted to the local environment of agents. If participants to a negotiation consider the remaining population as a constant, the evolution of the social value can be determined on a local basis. The formal definition of social transactions can be specified according to the welfare notion. The expressions of the conditions that transactions must satisfy, once applied to a specific social welfare notion, are detailed in dedicated sections of Chapter 4.

2.5.3 Difference and impact

Two acceptability criteria have been described. Both of them can be used locally to determine whether or not a transaction is profitable. However, these notions are not equivalent. Rational transactions are always social, but at the opposite, social transactions are not necessarily rational.

Example 2.1. Let us illustrate the difference between these notions with an example, using a population of 3 agents $\mathcal{P} = \{0, 1, 2\}$, and a set of 3 available resources $\mathcal{R} = \{r_1, r_2, r_3\}$. The preferences of all agents are described in Table 2.2. The initial resource allocation is $A = [\{r_1\}\{r_2\}\{r_3\}]$. The social objective considered in this example is the egalitarian welfare, which focuses on the individual welfare of the poorest agent.

Table 2.2: Acceptability criteria - Example of agents' preferences

Population $\mathcal{P}$	Resource Set $\mathcal{R}$		
	r_1	r_2	r_3
0	8	6	7
1	9	5	2
2	6	1	3

Table 2.3: Acceptability criteria - Set of possible exchanges

Transaction	Agent utility u_i			Criterion C_i	
	u_0	u_1	u_2	Rational	Social
Initially	8	6	3	-	-
$r_1 \leftrightarrow r_2$	6	9	3	$\times$	$\checkmark$
$r_1 \leftrightarrow r_3$	7	5	6	$\times$	$\checkmark$
$r_2 \leftrightarrow r_3$	8	2	1	$\times$	$\times$

Table 2.3 lists all possible exchanges. It also indicates whether or not acceptability criteria are satisfied. It shows that no rational transaction can be performed. Each time an agent wants to exchange its resource with another one, one participant decreases its utility. Thus, none of them is rational. However, when the social criterion is considered, if the richest agent accepts to loose some utility, the egalitarian welfare can be increased. For instance, let us consider the exchange of resources r_1 and r_3 between agents 1 and 3. This resource swap is not possible if agents are rational, whereas it is acceptable between social agents, as it leads to an increase of the egalitarian welfare value of the whole society.

These criteria influence a lot negotiation processes and it is essential to consider them in the agent decision-making. They restrict the set of possible transactions more or less. The social criterion is more flexible than the rational one and allows agents to perform more transactions. More importantly, both criteria lead to a finite negotiation process.

Property 2.9 (Finite negotiation process). *A negotiation process based on rational transactions or on social transactions end after a finite number of transactions.*

Proof. According to Proposition 1.2, an allocation problem with m resources and n agents has a finite solution space: $|\mathcal{A}| = n^m$. The set of all possible allocations is finite even if its size is exponential.
Any transaction δ, rational or social, always leads by definition to an increase of the social welfare value (independently of the notion considered). Since, during a transaction sequence, it is not possible to return to an allocation previously encountered, the associated social welfare value cannot be greater. Then, no cycle can then appear. Since no cycle

appears, coupled to the finite number of distinct resource allocations, negotiation processes end after a finite number of transactions. □

2.6 Agent interactions

Behaviors define agents from an external point of view. They describe how agents interact with each other, i.e., how they negotiate. During a negotiation, each agent makes and receives offers, and check their acceptability according to its own criterion. If a transaction is acceptable for every participant, it is performed. Otherwise, agents have to decide who has to modify its offer according to their behavior, and thus the negotiation continues.

This section focuses on bilateral negotiations. Multilateral transactions are much more complex to realize. Indeed, the initiator requires more information. In order to identify where resources should be allocated, a lot of information is required, like the partners' preferences. Such a process is memory and time consuming, and some alternative methods should be used. Chapter 5 addresses issues related to multilateral negotiation.

A negotiation between agents can be managed in different ways (Parsons et al, 2003; Rahwan and Larson, 2008; Saha and Sen, 2007). First, participants need to choose the order in which they propose their offers to their partners. According to the minimal concession strategy (Morge et al, 2009), agents always suggest first what is the most advantageous for them. The two participants successively suggest different alternatives until they agree on an acceptable transaction. However, this strategy was initially designed for agents knowing the complete bundle of the other agent. Considering the set of available resources, agents generate the possible set of allocations and sort it according to their preferences. They alternatively suggest an allocation, making more and more concessions until either they agree on an acceptable allocation, or they abort the negotiation.

This minimal concession strategy has to be adapted in order to satisfy the requirements ensuring the autonomy of all agents. In other words, this process must be based on personal information only. Since an agent ignores the bundle of the other agents, it must reason on its own bundle instead of allocations. An agent must prepare the set of offers it can propose, denoted by $L(\rho)$, and orders it according to its own preferences. Then, an agent can propose in the first place the offer the least penalizing for it. Indeed, agents always propose the offer associated with the lowest utility value. For any agent $i \in \mathcal{P}$, the set of offers $L_i(\rho)$ can be generated from its resource bundle $\mathcal{R}_i$ and from the set of allowed transactions $\mathcal{T}$. For instance, if a negotiation process allows only gifts, then the set of alternatives corresponds to the resource bundle, and agents just have to sort it according to their preferences. According to a negotiation policy "up to $\langle 3, 3 \rangle$", which allows agents to propose up to 3 resources, agents have to generate the list of offers whose size is less than or equal to three.

Let us assume that agent $i \in \mathcal{P}$ initiates a negotiation and proposes an offer to one of its neighbors $j \in \mathcal{N}_i$ previously selected. Both offers correspond to a bilateral transaction δ_i^j. If both agents consider this transaction acceptable, it is performed. However, if one participant rejects the offer, three alternatives can then be considered:

- agent i gives up and ends the negotiation;
- agent i changes the selected neighbor;
- agent i changes its offer or asks its partner to modify its own offer.

Based on this set of actions, various behaviors can be designed. Considering what we call a *rooted* behavior, an agent cannot change the selected neighbor during the negotiation. The initiator has to randomly select a member of its neighborhood. In contrast, a *frivolous* behavior allows the initiator to change its partner during the negotiation. The initiator should then shuffle its neighborhood and involve each neighbor successively. A *stubborn* agent only makes one offer. If the transaction is rejected by one participant, the initiator does not want to negotiate again with it. In this case, the initiator only considers the first offer of $L(\rho)$, which corresponds to the least penalizing one. Inversely, considering a *flexible* behavior, the initiator can change the offer that it proposes during the negotiation. In such a case, an agent can propose successively different offers of $L(\rho)$.

Determining the order of these actions is an important issue. For instance, the number of resources that agents offer can vary according to $\mathcal{T}$, and the order in which they consider the size of the possible offers is important. Agents can either regroup in $L(\rho)$ all their offers without considering their cardinality (the number of resources that agents can offer), sort $L(\rho)$ and then start the negotiation, or negotiate successively with the offers regrouped by cardinality. Agents start by proposing successively resource sets of a first kind, and if no acceptable transaction is identified, then they propose offers of an other cardinality. The process continues until all allowed transactions $\delta \in \mathcal{T}$ have been attempted, or until an acceptable transaction is identified.

Example 2.2. Let us consider a single agent $i \in \mathcal{P}$, who owns three resources in its bundle, $\mathcal{R}_i = \{r_1, r_2, r_3\}$. This agent evaluates these resources as follows: $u_i(r_1) = 1$, $u_i(r_2) = 3$ and $u_i(r_3) = 8$. This agent is involved in a negotiation with one of its neighbors. According to the negotiation settings, several transactions are allowed: $\mathcal{T} = \{\langle 1, X\rangle, \langle 2, X\rangle\}$. Agent i can either offer its neighbor a single resource or a set of two resources from its bundle. The order in which offers are proposed is described in Table 2.4 according to the negotiation policy considered.

Table 2.4: Sequence of offers proposed by the agent

Negotiation policies	
All together	By cardinality
r_1	r_1
r_2	r_2
r_1r_2	r_3
r_3	r_1r_2
r_1r_3	r_1r_3
r_2r_3	r_2r_3

The list of offers that agent $i \in \mathcal{P}$ can propose, denoted $L_i(\rho)$, is always sorted according to its preferences u_i, so that it can first propose offers with the lowest utility value. According to the policy "all together", $L_i(\rho)$ is generated at the beginning of the negotiation according to $\mathcal{T}$ and independently of the transaction cardinality. According to the second policy "by cardinality", the list of offers is regenerated for each allowed transaction in $\mathcal{T}$ and the order is modified. The larger is the agent's bundle $\mathcal{R}_i$, the higher are the differences of the two lists.

The first possible behavior is the simplest one. The initiator $i \in \mathcal{P}$ only proposes its offer associated with the lowest utility value (the first element of $L_i(\rho)$) to one of its neighbors $j \in \mathcal{N}_i$. If the transaction is rejected, then the negotiation aborts. This agent behavior is called *rooted stubborn* and is described in Algorithm 2.1. The initiator can only propose a single offer. In all behaviors described in this section, the TEST instructions correspond to acceptability tests, in which agents determine whether or not a transaction is profitable. The main criteria have been described in Section 2.5. The expression to use in such tests depends on the welfare objective considered, and hence is detailed in the corresponding section of the next chapter.

Algorithm 2.1: Rooted and stubborn agent behavior

Input: Initiator i

Output: TRUE if a transaction is performed

```
Lᵢ(ρ) ← generate(𝒯, ℛᵢ) ;                // list of all possible generated offers
Sort Lᵢ(ρ) according to uᵢ ;
ρ ← arg min_{ρ'∈Lᵢ(ρ)} uᵢ(ρ') ;          // selection of the cheapest offer
j ← random(𝒩ᵢ) ;                         // random selection of a partner
Get ρ' from j ;                          // get the offer from the partner
δ ← (ρ, ρ') ;
if TEST then                             // acceptability test
    Perform δ ;
    End the negotiation ;
    return TRUE ;
end
return FALSE
```

According to the second behavior, the initiator $i \in \mathcal{P}$ can only propose its offer ρ_i that is associated with the lowest utility. However, it can successively select different neighbors during the negotiation. If the transaction involving the first partner is rejected, the initiator can select another neighbor to continue the negotiation. Such an agent behavior is called *frivolous stubborn* and is described in Algorithm 2.2. The neighborhood should be shuffled between two negotiations in order to modify the order in which neighbors are considered, otherwise a bias may appear.

Algorithm 2.2: Frivolous and stubborn agent behavior

```
Input: Initiator i
Output: TRUE if a transaction is performed

L_i(ρ) ← generate(T, R_i) ;          // list of all possible generated offers
Sorts L_i(ρ) according to u_i ;
ρ ← argmin_{ρ'∈L_i(ρ)} u_i(ρ') ;     // selection of the cheapest offer
Shuffle N_i ;
forall the j ∈ N_i do                 // sequential selection of neighbors
    Get ρ' from j ;                   // get the offer from the partner
    δ ← (ρ, ρ') ;
    if TEST then                      // acceptability test
        Perform δ ;
        End the negotiation ;
        return TRUE ;
    end
end
return FALSE ;
```

An agent behavior is called *rooted flexible* when the initiator $i \in \mathcal{P}$ can successively propose different offers $\rho_i \in L_i(\rho)$, and it cannot change the selected neighbor. Such a behavior is described in Algorithm 2.3.

Algorithm 2.3: Rooted and flexible agent behavior

Input: Initiator i

Output: TRUE if a transaction is performed

```
L_i(ρ) ← generate(T, R_i) ;          // list of all possible generated offers
Sort L_i(ρ) according to u_i ;
j ← random(N_i) ;                     // partner selection
forall the ρ ∈ L_i(ρ) do
    forall the ρ' ∈ L_j(ρ) do
        δ ← (ρ, ρ') ;
        if TEST then                  // acceptability test
            Perform δ ;
            End the negotiation ;
            return TRUE ;
        end
    end
end
return FALSE ;
```

According to the next behavior, the initiator $i \in \mathcal{P}$ can change partners as well as its offer during a negotiation process. The initiator i proposes each offer $\rho_i \in L_i(\rho)$ to all its neighbors $j \in \mathcal{N}_i$ before changing it. According to such a behavior, if an acceptable transaction exists somewhere in the neighborhood, it will necessarily be identified. Such an agent behavior is called *frivolous flexible* and is described in Algorithm 2.4. The neighborhood should be shuffled between two negotiations in order to modify the order in which neighbors are considered.

Algorithm 2.4: Frivolous and flexible agent behavior

Input: Initiator i

Output: TRUE if a transaction is performed

```
L_i(ρ) ← generate(T, R_i) ;            // list of all possible generated offers
Sort L_i(ρ) according to u_i ;
Shuffle N_i ;
forall the ρ ∈ L_i(ρ) do                                        // flexibility
    forall the j ∈ N_i do                                        // frivolity
        forall the ρ' ∈ L_j(ρ) do
            δ ← (ρ, ρ') ;
            if TEST then                                 // acceptability test
                Perform δ ;
                End the negotiation ;
                return TRUE ;
            end
        end
    end
end
return FALSE ;
```

Agents may also scan their whole neighborhood in order to identify the best transaction to perform. Such agents are called *perfectionist*, and their behavior is illustrated in Algorithm 2.5. In such situations, the initiator $i \in \mathcal{P}$ starts a rooted and flexible negotiation with each neighbor, and only memorizes the best transaction encountered, which is finally performed. The determination of the best transaction between δ and δ' can be adapted according to the agent's acceptability criterion for instance. It can be either the maximization of its own utility or the maximization of the welfare of the population. Such a behavior is more expensive than the others since agents try all possible transactions with their neighbors. Always performing the best transaction does not ensure that optimal allocations are achieved at the end of negotiation processes. Even if the transaction is locally the most interesting one, it may lead resources into a dead-end whereas socially greater allocations might be achieved by allocating these resources to other agents, which are not directly related to the initiator for instance. Negotiation processes among perfectionist agents can be compared to greedy heuristics in the Optimization field.

Algorithm 2.5: Perfectionist agent behavior

```
Input: Initiator i
Output: TRUE if a transaction is performed
L_i(ρ) ← generate(T, R_i) ;                // list of all possible generated offers
Sort L_i(ρ) according to u_i ;
forall the j ∈ N_i do
    forall the ρ ∈ L_i(ρ) do
        forall the ρ'inL_j(ρ) do
            δ' ← (ρ, ρ') ;
            if TEST then                               // acceptability test
                if δ' IS BETTER THAN δ then
                    δ ← δ' ;
                end
            end
        end
    end
end
if δ ≠ ∅ then
    Perform δ ;
    End the negotiation ;
    return TRUE ;
end
return FALSE ;
```

According to flexible behaviors, as described in Algorithms 2.3 and 2.4, the initiator i sequentially proposes all possible offers $\rho_i \in L_i(\rho)$. For each initiator's offer, all the offers ρ_j that can be proposed by its partner $j \in \mathcal{N}_i$ must be attempted. In other words, when the initiator's offer is rejected, it always ask to its partner to propose something else. When it is no more possible, the initiator changes its offer and try to associate it with all possible offers of its partner.

However, when flexible behaviors are considered, negotiations can be managed in a different way. Indeed, each time that a transaction is rejected by one participant, instead of always requesting to the partner to change its offer, participants may determine which one should change its offer using a specific test. This test, called CHANGE TEST in Algorithm 2.6 can be based, for instance, on the difference of utility values between what they get and what they provide. More formally:

$$\text{CHANGE TEST} := u_i(\rho_i^\delta) - u_i(\rho_j^\delta) >^? u_j(\rho_j^\delta) - u_j(\rho_i^\delta)$$

Thus, in all behaviors, instead of a loop based on $\rho' \in L_j(\rho)$, Algorithm 2.6 can be used during a negotiation to decrease the number of attempted transactions.

Algorithm 2.6: Who has to change its offer?

```
. . . ;
indexI ← 0 ;                    // index on the initiator's list of offers
indexP ← 0 ;                    // index on the partner's list of offers
while indexI < m_i && indexP < m_j do
    ρ ← L_i(ρ)[indexI] ;
    ρ' ← L_j(ρ)[indexP] ;
    δ ← (ρ, ρ') ;
    if TEST then                                  // acceptability test
        Perform δ ;
        End the negotiation ;
    else
        if CHANGE TEST then                       // change test
            indexI ← indexI + 1 ;
        else
            indexP ← indexP + 1 ;
        end
    end
end
. . . ;
```

Note that in this thesis, we focus more on the distributed problem solving than on the agent language. The speech acts performed by participants are not explicitly written. Indeed, the initiator can access directly to some information of its partner, such as its offers and its individual welfare. The initiator chooses itself the minimal concession exchange (the least penalizing for both agents). However, this process can obviously split into different speech acts (Guerra-Hernández et al, 2009), as described in Figure 2.6. This figure describes the different speech acts required. Alternatively, each agent makes offers, analyzes the information provided by the other agent, in order to finally determine if

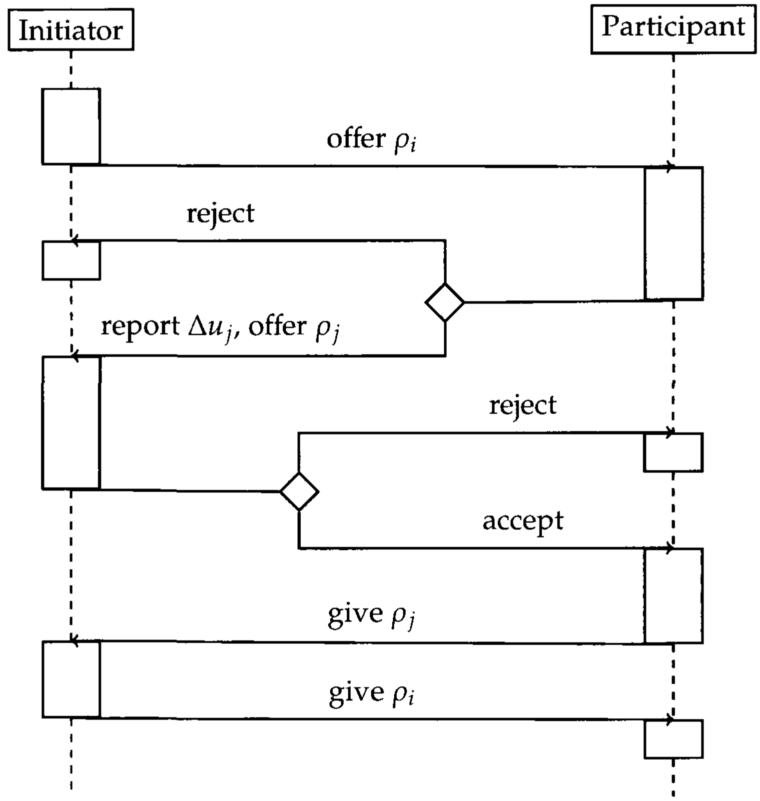


Figure 2.6: Sequence of speech acts

the transaction composed of both offers is acceptable or not. Each agent determines the acceptability of the transaction according to its own criterion. The simplification made in our algorithm is possible since all agents act according to the same behavior and to the same acceptability criterion.

Behavior names defined in this section are abbreviated in the experiments reported in Chapter 4 to improve the understanding of graphs. Table 2.5 summarizes the different behaviors and their abbreviated name.

Table 2.5: Summary - Agent behaviors

	Rooted	Frivolous
Stubborn	fs	rs
Flexible	rf rf full	ff ff full - agent ff full - resource

- **rs**: rooted stubborn behavior;
- **fs**: frivolous stubborn behavior;
- **rf**: rooted flexible behavior based on a specific mechanism to determine who has to modify its offer;
- **rf full**: rooted flexible behavior;
- **ff**: frivolous flexible behavior based on a specific mechanism to determine who has to change its offers;
- **ff full - agent**: frivolous flexible behavior where the initiator favors the partner change;
- **ff full - resource**: frivolous flexible behavior where the initiator favors the offer change.

2.7 Evaluation of negotiation processes

The evaluation of negotiation processes is not an obvious issue. Centralized methods are most often evaluated using the computation time or the quality of the provided solutions (e.g., in the case of heuristics), but negotiation processes can be evaluated using many metrics. Depending on the chosen metric, the performance of negotiation processes may vary. Hence, various metrics should be considered during their evaluation.

2.7.1 Evaluation metrics

A fair evaluation must consider the different aspects of negotiation processes. We propose a set of usable metrics in this section.

First, the **number of performed transactions** indicates the overall number of transactions effectively performed during the whole negotiation process. It corresponds to the length of the transaction sequences required to evolve the initial resource allocations to the provided solutions. The acceptability criterion significantly affects this metric. Indeed, it restricts the set of possible transactions among agents. Negotiations based on a restrictive criterion, like rationality, may end faster than negotiations based on a more flexible criterion like sociability.

The **number of attempted transactions** is the overall number of offers proposed during a negotiation process. Two factors influence this metric: the agents' behaviors and the allowed transactions. When agents are stubborn and/or rooted, they only make a single offer during a negotiation, and then highly limit the number of attempted transactions. However, in the case of flexible and/or frivolous agent behaviors, agents attempt many more offers. In such cases, the set of transactions allowed is an essential parameter. The larger is the number of possible transactions, the larger is the number of attempts, as summarized in Table 2.6. Thus, the number of attempted transactions may increase exponentially.

Table 2.6: Summary - Transaction complexity

Transaction kinds		Number of possible transactions
Bilateral δ_i^j	$\langle a, b\rangle$	$\binom{m_i}{a}\binom{m_j}{b}$
	up to $\langle a, b\rangle$	$\left(\sum_{x=0}^{a} \binom{m_i}{x}\right)\left(\sum_{x=0}^{b} \binom{m_j}{x}\right) - 1$
Multilateral δ_i^Δ	One-to-many	$2(n_\delta + 1)^{m_i}\left(\sum_{j\in\Delta^\delta} m_j\right) - 1$
	Many-to-many	$(n_\delta + 1)^{m'} - 1$ with $m' = \sum_{j\in\Delta^\delta\cup\{i\}} m_j$

Then, the **number of traded resources** indicates the density of the resource traffic when coupled with the number of performed transactions. Bilateral transactions, as well as multilateral transactions, may have a bound on the size of the offers. According to these limits, the number of traded resources during a transaction can vary. For instance, during a bilateral transaction $\delta_i^j\langle a, b\rangle$, a maximum of $a + b$ resources may be traded. However, from the same initial resource allocation, and in order to achieve the same final allocation, a sequence of a gifts $\delta_i^j\langle 1, 0\rangle$ and b gifts $\delta_j^i\langle 1, 0\rangle$ is required.

A negotiation process is a sequence of negotiation steps which are initiated by agents.

Each step corresponds to the identification of an acceptable transaction. The **number of speech turn** corresponds to the number of times a negotiation is initiated, i.e. the number of steps. This metric depends on two parameters: The agent behavior and the allowed transactions. If agents' behaviors are rooted and/or stubborn for instance, they may need a larger number of negotiations to identify an acceptable transaction with one of their neighbors. Similarly, in the case of negotiation processes which are only based on gifts, the number of negotiations required to end the negotiation process is larger for processes based on cluster-swaps for instance.

Finally, the **topological sensitivity** should also be evaluated. Indeed, the topology of social graphs affects a lot the negotiation process. Considering different graph topologies of the same class, negotiation processes starting from the same initial allocation can achieve different allocations. The topological sensitivity can be evaluated thanks to the standard deviation among the social values achieved at the end of negotiation processes. A large deviation means that the negotiation process is very sensitive to the graph topology, and thus the quality of the provided allocation significantly varies according to the initial conditions.

2.7.2 Negotiation efficiency

The efficiency of negotiation processes is an important goal. Indeed, if a negotiation process ends quickly, it might not be interesting if the provided solution is associated with social values which are far from the optimum. The negotiation settings allow the achievement of different kinds of allocations, as illustrated in Figure 2.7.

This figure represents the different allocation sets which are achievable according to the negotiation settings. The largest set corresponds to the whole solution space. Different sets of allocations can be achieved depending on settings, like the set of allowed transactions or the topology of social graphs. In a situation where negotiation processes always lead to optimal allocations, the different solution sets are similar to the optimal one.

In order to evaluate the quality of the provided solutions, we can carry out comparisons

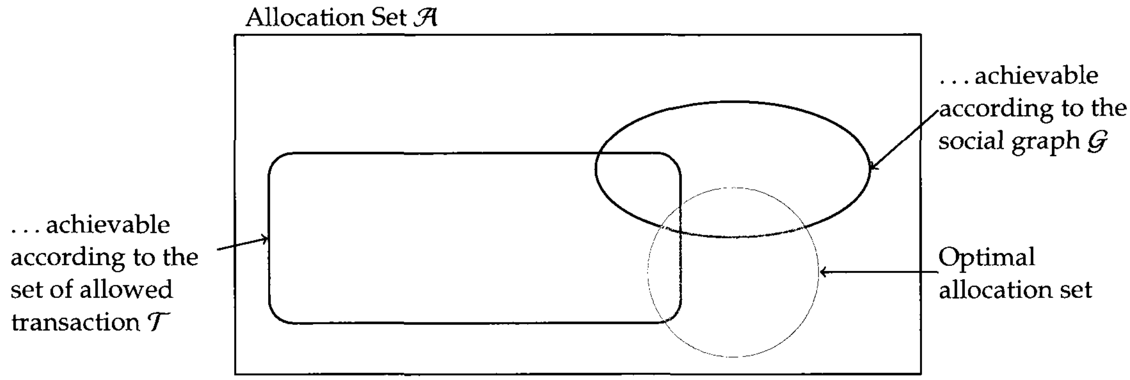


Figure 2.7: Different sets of achievable solutions depend on the negotiation settings

between the values provided by centralized techniques and by distributed negotiations.

Centralized approaches provide social values used as a reference for the comparisons with the results which are provided by distributed negotiations. This social value is called a *global optimum*.

Definition 2.13 (Global optimum). A resource allocation $A \in \mathcal{A}$ is a global optimum if no other resource allocation $A' \in \mathcal{A}$ associated with a greater social value exists.

$$\nexists A' \in \mathcal{A} \quad sw(A') > sw(A) \quad A, A' \in \mathcal{A} \text{ such that } A \neq A'.$$

A global optimum is not dependent on the allowed transactions among agents. Depending on the kinds of allowed transaction, resource allocations corresponding to global optima might not be achievable. Moreover, the optimal social value is unique but several resource allocations can correspond to it.

During a negotiation process, agents negotiate until none of them is able to identify acceptable transactions. The final solution is the resource allocation represented by the state of system at that time. This state can be considered as an equilibrium state for the negotiation process.

Definition 2.14 (Local optimum). A resource allocation $A \in \mathcal{A}$ is a local optimum if no sequence of transactions, belonging to the set of allowed transactions $\mathcal{T}$, leading to a resource allocation associated with a greater social welfare value exists.

$$\forall A' \in \mathcal{A}, \qquad \nexists \delta_i^{\Delta} \quad sw(A') > sw(A) \qquad \delta \in \mathcal{T}, A \in \mathcal{A}.$$

A local optimum is an equilibrium which can not be avoided using transactions from $\mathcal{T}$, with respect to the social graph topology. The closer are the social values associated with the local optimum and with the global optimum, the more efficient are the negotiation processes.

The comparison between the social value achieved by both approaches corresponds to an evaluation of the **price of anarchy** (Gairing et al, 2006; Koutsoupias and Papadimitriou, 2009). This notion is often used to quantify the loss due to the distribution of solving processes (Christodoulou and Koutsoupias, 2005; Roughgarden, 2005). This price of anarchy can be evaluated when the centralized and the agent-based approaches have pretty similar conditions, especially regarding the communication possibilities. This is the case when negotiation processes are based on a complete social graph, which allows to have "similar" solving conditions, and hence to carry out relevant comparisons.

However, restrictions on communication possibilities have a significant impact on negotiation processes. Thus, another notion can be introduced: the **price of the social graph**. This notion quantifies the quality loss due to the restriction imposed on the agent communication possibilities. In this way, negotiation processes based on different social graphs can be compared, and the topology's characteristics favoring the resource traffic or the negotiation efficiency may be identified.

2.7.3 Computation time

The computation time required to carry out negotiation process is also an important evaluation criterion. Its evaluation is quite simple. A solving process starts with the first negotiation, and is over when no agent in the population is able to identify an acceptable transaction.

Since many parameters can influence a negotiation process and its efficiency, the computation time should be determined according to the most time-consuming settings. It should represent an upper bound of the time required to end distributed negotiations when restrictions are made. For instance, a complete social graph maximizes the communication possibilities and the resource traffic. This setting can be considered as the worst topology from a computational point of view, maximizing negotiation opportunities. Agent behaviors also greatly influence the computation time. The worst behavior in terms of computation time is flexible and frivolous. Agents who can change partners as well as offers require more time to negotiate than other agents. Thus, in our experiments, the computation time required by a negotiation process, for a given set of allowed transactions $\mathcal{T}$, is evaluated on complete graphs involving flexible and frivolous agents.

2.8 Summary

In this chapter, distributed solving methods based on agent negotiations have been described. Challenges related to agent-based methods have been discussed and the different parameters defining agents have been successively detailed.

- **Social graphs**: Restricted communications among agents can be modeled using a social graph representing agent relationships. Any graph topology can be handled, but this thesis focuses on *complete graphs*, *Erdős-Rényi graphs*, *grids* and *small-worlds*.

- **Transactions**: Different transaction classes have been presented, from simpler ones like *bilateral transactions* to more complex ones like *multilateral transactions*. Complexity issues have also been discussed.

- **Acceptability criteria:** In order to obtain finite negotiation processes, agents must be able to determine locally if transactions are profitable or not. Two criteria have been defined and characterized: *Rationality* and *sociability*.

- **Agent behaviors**: Agents can negotiate in several different ways. Behaviors define how agents interact with each other. We defined the main features characterizing agent behaviors: *Rooted* or *frivolous*, *stubborn* or *flexible*, *perfectionist*, ...

Finally, we discussed the evaluation of negotiations. Different metrics are presented in order to consider every facet of distributed negotiations: The *number of performed transactions*, the *number of attempted transactions*, the *number of speech turns*, the *number of traded resources* and the *topological sensitivity*. We also discussed the negotiation efficiency by comparing optimal values provided by centralized techniques and by distributed methods.

Simulations must be performed in order to establish the validity of our model of distributed negotiations. This model is characterized by a large number of parameters, and the next chapter describes algorithms required to generate them. Such algorithms are required to ensure the reproducibility of the experiments.

Chapter 3

Experimental Protocol

After theoretical studies of centralized and distributed approaches, experiments and simulations must be performed in order to valid our approach. A large number of parameters can be considered as described in Section 2.7. A precise simulation protocol is required, describing how each parameter is generated, in order to characterize precisely experiments and to ensure their reproducibility. Indeed, each parameter significantly affects the qualities of achieved allocations. This chapter is thus dedicated to the description of the experimental protocol.

According to our definition of agent (see Section 2.2), we need to consider five different parameters. Agent behaviors and acceptability criteria have already been detailed respectively in Section 2.5 and 2.6. The three other parameters must still be specified. Initial allocations and agents' neighborhood are generated using centralized algorithms and then distributed among the agents. Data instances are described and the simulation settings are also characterized.

This chapter is organized as follows. First, the generation of the agents' preferences is described in Section 3.1, i.e., how utility functions are generated. Then, a method to determine initial resource allocations is detailed in Section 3.2, since all resources must be allocated before the beginning of negotiation processes. Section 3.3 is dedicated to the generation of social graphs. For each class of social graphs considered during the experiments, an algorithm of generation is provided. Section 3.4 describes how sequential negotiation processes are managed in practice, with a description of the mechanism distributing the speech turn among agents. Conditions to satisfy in order to detect the end of negotiation processes are also discussed. Finally, the characteristics of data instances and simulation settings are described in Section 3.5.

3.1 Generation of agents' preferences

As described in Section 1.1.2, agents express their preferences using an evaluation function restricted to an additive utility function. Agents' preferences are generated randomly according to a uniform distribution, as described in Algorithm 3.1. During our experiments, the utility value range is $[0, m]$, where m is the total number of resources. Negative utility values can also be used without any impact on the negotiation process efficiency, except when the Nash welfare is considered.

Algorithm 3.1: Generation of utility functions

Input: Agent i, Resource Set $\mathcal{R}$

Output: Utility function u_i of agent i

forall the $r \in \mathcal{R}$ **do**

  $val \leftarrow$ random integer draw in $[0, m]$;

  add (r, val) to u_i ;

end

The size of the range from which utility values are drawn also has an impact on negotiation processes. Indeed, if the range of utility values is not large enough compared to the overall number of resources, a very large number of equivalent optimal solutions may appear. It may then bias the real efficiency of the negotiation processes. The larger is the number of equivalent optimal solutions, the easier is the achievement of one of them. Thus, an inappropriate range value of generation biases the efficiency of negotiations. In order to avoid such a phenomenon, we suggest to generate utility values in the range $[0,m]$, where m is the total number of available resources.

3.2 Generation of initial allocations

A negotiation process starts from an initial allocation, which evolves step by step, thanks to local negotiations among agents. According to Proposition 1.1 describing allocation properties inherent to the resource nature, all resources must be allocated to agents and each resource must be allocated to only one agent. Since there is no reason for an agent to own a resource more than others, the initial resource allocation is generated randomly. Thus, each resource is randomly allocated to an agent according to a uniform distribution, as described in Algorithm 3.2.

Algorithm 3.2: Generation of initial allocations

Input: Agent population $\mathcal{P}$, Resource Set $\mathcal{R}$

Output: Initial allocation A

```
forall the r ∈ R do
    i ← random draw in P ;
    add r to A[i] ;
end
```

Initial resource allocations influence significantly the negotiation efficiency when restricted social graphs are considered. Indeed, these graphs restrict the resource traffic according to their topology. Thus, an agent might never see resources depending on where they are initially allocated. In such a context, initial allocations affect optima that can be achieved. The generation of different graphs are described in the next section.

3.3 Generation of social graphs

Relationships among agents are generally not considered in resource allocation problems. Most of studies on resource allocation problems implicitly assume that the solutions obtained by the proposed methods can always be achieved in practice. According to this assumption, any agent of the population is able to communicate with all the other agents. However, many applications do not satisfy such an assumption, especially when large systems are considered. A social graph must then be defined, representing relationships among agents. The features of different classes of social graphs have a great impact on the efficiency of negotiation processes. This section provides algorithms generating different classes of social graphs, as presented in Section 1.3. They generate non-oriented graphs, but they can be easily adapted to generate oriented graphs, if required.

In our experiments, simulation environments are assumed to be static: Populations and resource sets do not change. Social graphs are generated using a centralized algorithm, and then split and distributed among agents. All algorithms defined in this section, generate and return a social graph $\mathcal{G}$, which is modeled as an ordered list of neighborhoods. The neighborhood of agent $i \in \mathcal{P}$ corresponds the i-th element of the social graph, $\mathcal{G}[i] = \mathcal{N}_i$, as described in Figure 3.1.

Example 3.1. The following example illustrates the representation of a social graph used in this thesis. The relationships among the 5 agents of a population $\mathcal{P} = \{0, 1, 2, 3, 4\}$ are described by the social graph illustrated in Figure 3.1.

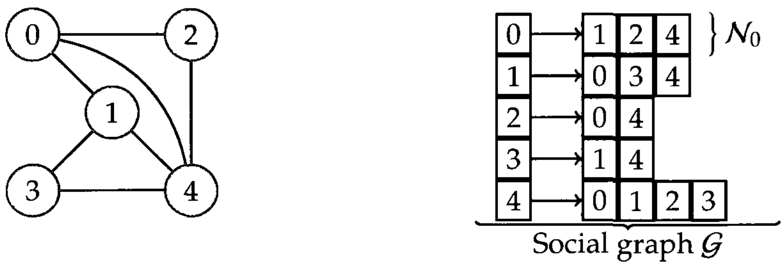


Figure 3.1: A social graph and its representation

The social graph shows that agent 0 is linked to only three other agents. Then, its neighborhood contains three agents: $\mathcal{G}[0] = \mathcal{N}_0 = \{1, 2, 4\}$. Thus, the whole social graph can be defined by: $\mathcal{G} = \{\mathcal{N}_0, \mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3, \mathcal{N}_4\} = \{\{1, 2, 4\}, \{0, 3, 4\}, \{0, 4\}, \{1, 4\}, \{0, 1, 2, 3\}\}$.

Relationships among agents might also be represented by a connection matrix. The social graph $\mathcal{G}$ is a Boolean square matrix of size $n \times n$ (where n is the total number of agents). Each Boolean value $\mathcal{G}[i][j]$ represents the existence of a link between two agents

$i, j \in \mathcal{P}$. However, since we adopt a distributed approach, agents ignore the real size of the population. A representation based on a list of neighbors is thus favored.

The following sections detail how we generate the different classes of social graphs used in the experiments.

3.3.1 Complete graphs

In a complete social graph, each agent of the population can communicate with all other agents, as illustrated in Figure 3.2. Such a social graph is equivalent to the one used in most other agent-based studies or in centralized approaches.

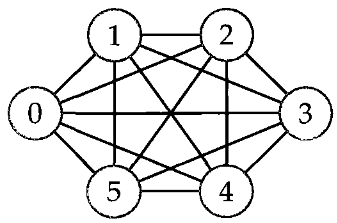


Figure 3.2: Example of complete graphs

A complete social graph is generated as described in Algorithm 3.3.

Algorithm 3.3: Generation of complete graphs

Input: Agent population $\mathcal{P}$

Output: Social graph $\mathcal{G}$

forall the $i \in \mathcal{P}$ **do**
  forall the $j \in \mathcal{P} \setminus \{i\}$ **do**
    add j to $\mathcal{G}[i]$;
  end
end

Complete graphs are only used for comparison purposes. Such a topology cannot be ignored since the efficiency of negotiation processes will be compared to the efficiency of centralized approaches. However, complete graphs have no real interest in the solution of multi-agent resource allocation problem.

3.3.2 Erdős-Rényi graphs

Erdős-Rényi graphs are basically random graphs. Two generation models exist (Erdős and Rényi, 1959; Bollobás, 2001), namely $G(n, M)$ and $G(n, p)$. The first model, $G(n, M)$, is characterized by the number of nodes and the total number of edges required in the graph, whereas the second model, $G(n, p)$, is characterized by the number of nodes and the probability to set an edge between agents of any pair of nodes. According to this model, the probability to set an edge is the same and is independent from the probability to set other edges. This second model $G(n, p)$ is the one that we use to generate the random graphs in our experiments, as illustrated in Figure 3.3.

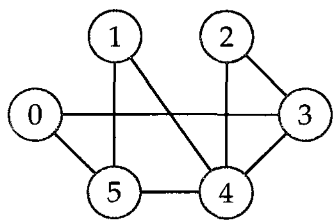


Figure 3.3: Example of Erdős-Rényi graphs

An Erdős-Rényi social graph, which is based on the $G(n, p)$ model, is generated using Algorithm 3.4.

Algorithm 3.4: Generation of Erdős-Rényi graphs

Input: Agent population $\mathcal{P}$, probability p

Output: Social graph $\mathcal{G}$

```
forall the (i, j) ∈ P × P such that i < j do
    val ← random draw in [0, 1] ;
    if val <= p then
        add j to G[i] ;
        add i to G[j] ;
    end
end
```

Let us note that generated graphs are not necessarily connected. If the probability

p is not large enough, some nodes may be isolated. If social graphs are not connected, then multi-agent resource allocation problems can be split into several independent sub-problems, as described in Proposition 2.3, and they can be solved independently.

3.3.3 Grids

In populations where all agents have exactly four neighbors, toric grids can be used to represent their relationships, as illustrated in Figure 3.4. According to the characteristics described in Section 1.3, the grids that we generate tend to have balanced dimension (i.e. a shape close to a square). Given a population of agents, it is quite easy to determine the corresponding dimensions of the grid using a prime number decomposition. Such grids can be generated using Algorithm 3.5.

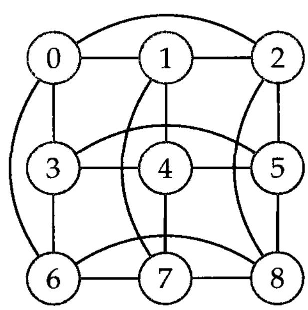


Figure 3.4: Example of grids

Algorithm 3.5: Generation of grids

Input: Agent Population $\mathcal{P}$, lengthGrid l

Output: Social graph $\mathcal{G}$

forall the $i \in \mathcal{P}$ **do**

```
    // determine the 4 neighbors of each agent
    north ← i − l ;
    if north < 0 then
        north ← north + n ;
    end
    west ← i − 1 ;
    if west < 0 then
        west ← west + n ;
    end
    south ← i + l ;
    if south ≥ n then
        south ← south − n ;
    end
    east ← i + 1 ;
    if east ≥ n then
        east ← east − n ;
    end
    add north, south, east, west to G[i] ;
end
```

3.3.4 Small-worlds

Small-worlds describe a wide range of real systems in nature and societies. In our experiments, small-worlds are generated using the preferential attachment model (Albert and Barabási, 2002). New agents joining into the population are connected to the existing agents with a probability proportional to their number of neighbors. An example of small world is illustrated in Figure 3.5.

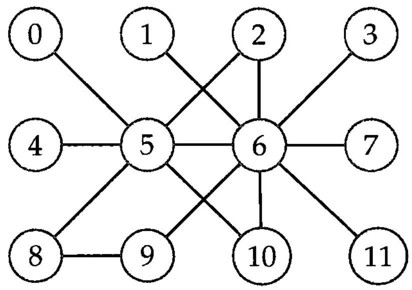


Figure 3.5: Example of small-worlds

Such small-worlds can be generated using Algorithm 3.6.

Algorithm 3.6: Generation of small-worlds

Input: Agent Population $\mathcal{P}$

Output: Social graph $\mathcal{G}$

$total \leftarrow 1$;

forall the $i \in \mathcal{P}$ **do**

    for $j = 0 \rightarrow i$ **do**

        $limit \leftarrow (2 * \mathcal{N}_j)/total$;

        random draw of p ;

        if $p \leq limit$ **then**

            add i to $\mathcal{G}[j]$;

            add j to $\mathcal{G}[i]$;

            $total \leftarrow total + 1$;

        end

    end

end

In this section, the generation of different classes of social graphs has been described. Each of them has different characteristics that prevent negotiation processes to achieve optimal resource allocations, as discussed in Section 1.3.2.

In order to ensure the reproducibility of our experiments, some characteristics of the protocol must still be detailed. Since agents negotiate sequentially, a specific mechanism is used to distribute the speech turn among them. Such a mechanism is described in the next section as well as ending conditions of negotiation processes.

3.4 Negotiation processes

In order to fully define a finite negotiation process, some details must still be given. The first mechanism is the speech turn distribution process, whereas the second is related to the ending conditions of negotiation processes.

Negotiation processes are sequential in this study: Only one agent at a time can initiate a negotiation, according to its behavior. To achieve this, a classical mechanism based on a token is used to decide which agent can initiate a negotiation. The speech turn is uniformly distributed over the population: No agent talks twice unless all agents have talked at least once. Such a distribution is done thanks to a well-known *round-robin* algorithm, which is often used as a task scheduler. The order in which agents receive the token may bias the process. To avoid this phenomenon, the population is shuffled when every agent has initiated a negotiation, i.e., periodically every n negotiations. The distribution process of the token is illustrated in Figure 3.6.

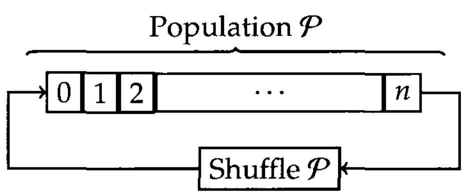


Figure 3.6: Distribution of the speech turns

The last issue is related to ending conditions of negotiation processes. In other words, when can we consider that a negotiation process is terminated? During our experiments, negotiation processes end when no agent in the population can identify acceptable transactions to perform. Thus, negotiation processes can be managed as described in Algorithm

3.7. We assume that the initiator of a negotiation returns the Boolean value TRUE if an acceptable transaction is performed, and FALSE otherwise.

Algorithm 3.7: Negotiation processes

```
Input: Agent population P

transactionDone ← TRUE ;
while transactionDone = TRUE do
    transactionDone ← FALSE ;
    shuffle P ;
    forall the i ∈ P do
        result ← i negotiates ;
        transactionDone ← transactionDone OR result ;
    end
end
```

Ending conditions described here are based on a Boolean criterion. However, depending on agents' behaviors, a different condition may improve the efficiency. In Algorithm 3.7, when no agent is able to identify acceptable transactions, negotiation process end. However, the end of negotiation processes does not necessarily mean that no acceptable transaction exists. For instance, when rooted behaviors are considered, the initiator negotiates only with one neighbor. A negotiation process based on such conditions may end prematurely if every agent has selected a "bad" neighbor. Allowing several negotiation rounds where no agent is able to find acceptable transactions may then improve solutions. These additional rounds may allow agents to select a proper neighbor to negotiate, and then identify acceptable transactions.

This tip must nevertheless be moderated. Its efficiency is conditional to the negotiation cost. Indeed, when simple transactions such as gifts are allowed, allowing additional negotiation rounds is not expensive. However, in the case where complex transactions are allowed, such a tip may become exponentially time-consuming.

3.4.1 Simulation and practice

In practice, it is possible to greatly reduce the computation time. Indeed, negotiation simulations are usually run sequentially. One agent at a time can initiate a negotiation, and once this negotiation is over, another agent is selected to start a new negotiation. However, such a method does not take advantage of the distributed nature of multi-agent systems and of the agent's autonomy. Parallel negotiations can be used to greatly reduce the computation time. In such cases, each agent can only initiate one negotiation at a time, but it can be involved in several negotiations simultaneously. However, synchronization and deadlock issues should be considered. Since agents can negotiate in a concurrent way, consistency must be ensured using specific mechanisms. If an agent is involved in several different negotiations at once, it must not promise the same resource to different partners since resources are not sharable.

Two main different mechanisms are available to synchronize negotiations. The first mechanism applies at the "agent" level whereas the second mechanism applies at the "resource" level.

The first mechanism is the most basic one: The synchronization by neighborhood exclusion. An agent who is already involved in a negotiation cannot be involved in another one at the same time. Once the current negotiation is over, it can then be involved in a new negotiation, as described in Figure 3.7. In this figure, agents 0 and 2 are negotiating as well as agents 1 and 3. These agents are then locked. Agent 4 who looks for a partner cannot either choose agent 2 or agent 1 since they are already busy. It should select agent 5.

This synchronization mechanism is simple and easy to implement. Negotiation processes based on bilateral transaction $\delta_i^j\langle a, 0\rangle$, i.e., where partners do not offer any resource like in gifts or clusters, synchronization mechanisms are useless. Indeed, agents do not wait for an offer of their partner. Since an agent can only initiate one negotiation at a time, no specific synchronization mechanism is required. However, during a negotiation, most of the time agents only offer a small subset of their bundle. Other agents can negotiate with them the unused resources. This idea leads to the second synchronization mechanism.

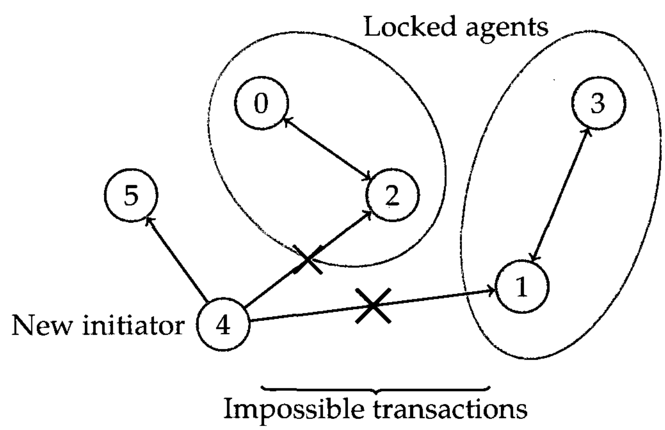


Figure 3.7: Agent-based synchronization mechanism

It is based on the resource exclusion. Instead of waiting that the agent completely ends a negotiation, it is possible to negotiate the resources which are still available as described in Figure 3.8.

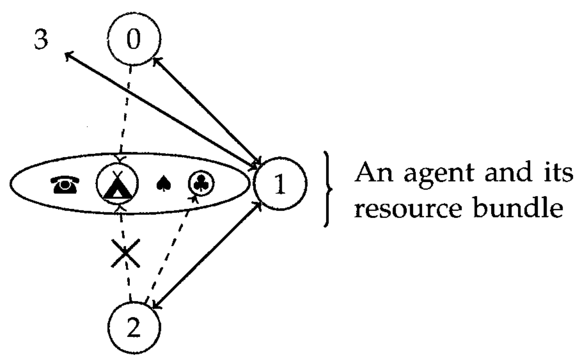


Figure 3.8: Resource-based synchronization mechanism

Several agents can involve a common neighbor in simultaneous negotiations. This agent locks each offered resource. Synchronization mechanisms must be carefully designed in order to avoid deadlocks. These situations arise when several agents wait for the other participants to negotiate.

We have implemented parallel negotiations: Agents are represented by independent threads. Negotiations have been simulated on an homogeneous cluster of computers. Each node of this cluster is a 2.2 GHz AMD Opteron 64-bit processor. It utilizes the InfiniBand network as the means for node-to-node communication and for Input/Output to the cluster file system. The computation time of sequential negotiations have been compared to the computation time of parallel negotiations, involving an increasing number of processor cores. Both synchronization mechanisms have been implemented and tested on numerous simulations, showing a decrease of the computation time with the increase of the number of cores.

3.5 Experimental protocol

Issues related to the evaluation of negotiation processes have been discussed in Section 2.7. The fair evaluation of negotiation processes must be done according to an precise protocol. The purpose of this section is to describe the different settings used. An experiment is characterized by the characteristics of the data instances and of the simulation.

3.5.1 Instance characteristics

An instance is composed of a population of agents who express their preferences over the resource set thanks to a utility function, as described in Section 3.1. Agent neighborhoods are defined by social graphs, generated according to a given class (see Section 3.3). Social graphs and population preferences depend on some parameters that must be set beforehand, as described in Figure 3.9. Parameters are represented by boxes, processes by ellipses, and results by double boxes.

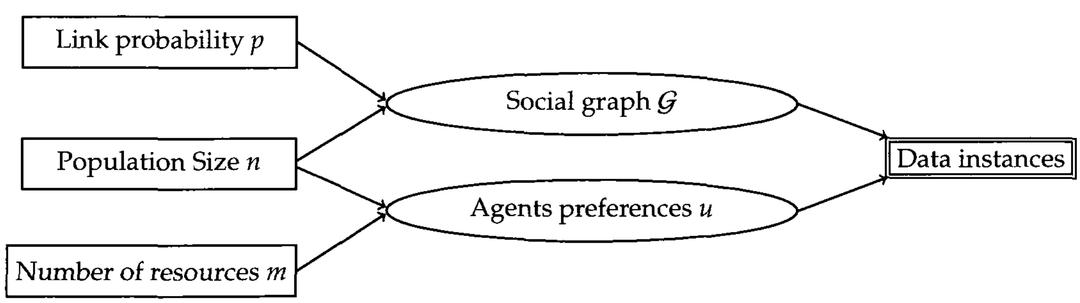


Figure 3.9: Data instance specifications

First, the number of agents n varies from 25 up to 500 agents. The number of available resources is a little dependent on the population size. Indeed allocating 100 resources over a population of 10 agents or over a population of 100 agents has not the same complexity. Instead of characterizing instances by the overall number of resources m, they can be characterized by the mean number of resources per agent $\frac{m}{n}$. In our experiments, $\frac{m}{n} \in \{5, 10, 20\}$. For each pair (n, m), 10 population preferences are generated. 10 social graphs of each class are generated. Hence, for a given link probability and a population size, 31 social graphs are generated: 1 complete graphs, 10 grids, 10 Erdős-Rényi graphs and 10 small-worlds. In the case of graphs from the random family, the link probability p affects the social graph mean connectivity. In this study, p varies from 0.05 up to 1.0. Finally, the association of a social graph with a set of agents' preferences set corresponds to one data instance.

3.5.2 Simulation characteristics

Some simulation settings must still be specified in order to clearly define the experimental protocol. A simulation can be completely specified by four parameters: The agent behavior $\mathcal{B}$, the acceptability criterion C, the class of allowed transactions and finally the social welfare notion sw considered, as described in Figure 3.10.

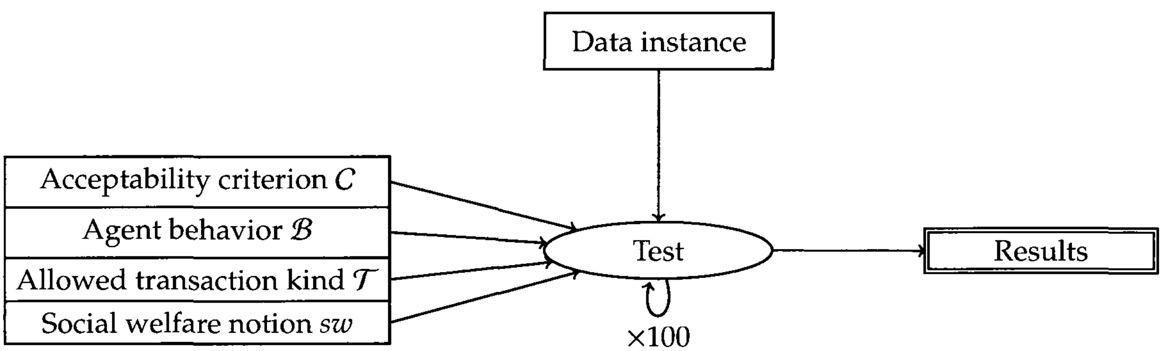


Figure 3.10: Simulation specifications

The four main social welfare notions are considered, namely the utilitarian welfare, the egalitarian welfare, the Nash product and the elitist welfare. Different transactions can be allowed during a negotiation process. Multilateral deals or bilateral deals, such as gifts, swaps or larger exchanges. Two acceptability criteria will be investigated: the individual rationality and the sociability, as defined in Section 2.5. Finally, as described in Section 2.6, various agent behaviors are assessed: Rooted, flexible, frivolous, stubborn, ...

The combination of these four parameters, associated with data instances, defines a simulation. Each simulation will be iterated 100 times from different initial resource allocations in order to evaluate the topological sensitivity as described in Section 3.2.

The description of data instance features and of simulation features shows the very large number of experiments that have been realized, and their diversity. Results related to bilateral negotiations are described and analyzed in Chapter 4, while multilateral issues are addressed in Chapter 5.

Chapter 4

Bilateral Negotiations

Bilateral transactions are the simplest and the most widely used transaction class in the literature. During such negotiations, the agent who initiates a negotiation can only involve one neighbor at a time. These transactions are popular since they require few information. This chapter seeks to find an answer to the following question: "How agents must interact in order to maximize the efficiency of negotiation processes?", according to the social notion considered. This chapter also shows that restricting negotiation processes to bilateral transactions may affect the efficiency of negotiations. Parameters maximizing the negotiation efficiency are identified, i.e., suitable agents' behaviors, allowed transactions and the most adapted acceptability criterion.

This chapter is divided into four sections. Each section is dedicated to a specific social welfare notion. Negotiation processes are analyzed in order to identify simulation features leading agent negotiations to socially optimal allocations, or to socially close allocations when the need arises. Each section is organized as follows. First, centralized approaches are described and algorithms are provided in order to determine the optimal social welfare value. Then, the expression of acceptability criteria are specified according to the objective function considered, and negotiation properties are described. Different facets of negotiation processes are then evaluated according to various metrics. The impact of the different parameters are discussed like agent behaviors, allowed transactions and social graphs. Section 4.1 is dedicated to the utilitarian welfare, Section 4.2 describes egalitarian negotiations, Section 4.3 presents the results related to the Nash negotiations and finally Section 4.4 deals with elitist negotiations.

4.1 Utilitarian bilateral negotiations

The utilitarian welfare is the notion the most widely used in the social welfare theory. This notion is especially used for applications in Economics, as for example the e-trade. An utilitarian objective maximizes the global welfare of the society without any consideration of the individual welfare. First, different centralized approaches are described in order to provide the global optimal welfare value. Such a value can then be used as a reference in order to evaluate the efficiency of utilitarian negotiation processes. The expression of acceptability criteria is then specified when the utilitarian notion is considered. Negotiation properties are also discussed. Finally, negotiation processes are evaluated according to different parameters as described in Section 2.7.

4.1.1 Centralized techniques

Generally, several ways can be used to determine the optimal welfare value. When the utilitarian welfare is considered, two ways are possible. The first one is to model utilitarian resource allocation problems by means of linear programs, which can be solved using any mathematical programming optimizer like CPLEX (ILOG Inc, 1995). The variables of such a model, denoted by x_{ir}, represent the ownership of resource $r \in \mathcal{R}$ by agent $i \in \mathcal{P}$ as follows:

$$x_{ir} = \begin{cases} 1 & \text{if agent } i \text{ owns resource } r \\ 0 & \text{otherwise.} \end{cases} \qquad r \in \mathcal{R}, \quad i \in \mathcal{P}.$$

Then, the determination of the optimal utilitarian welfare value can be formulated as follows:

$$sw_u^\star = \begin{cases} \max \sum\limits_{i \in \mathcal{P}} \sum\limits_{r \in \mathcal{R}} u_i(r) x_{ir} & \\ \text{s.t: } \sum\limits_{i \in \mathcal{P}} x_{ir} = 1 & r \in \mathcal{R} \\ \quad x_{ir} \in \{0, 1\} & r \in \mathcal{R}, \quad i \in \mathcal{P}. \end{cases}$$

The objective function is the maximization of the utilitarian welfare, which can be written as the sum of all agents welfare according to Definition 1.4. Two consistency constraints are required. According to the resource nature, i.e., since resources are neither divisible nor sharable, Boolean variables are considered. However, this model can be adapted easily. For instance, continuous resources could be represented by real variables. The second constraint becomes $x_{ir} \in [0, 1], r \in \mathcal{R}, i \in \mathcal{P}$.

The other way to determine the optimal utilitarian value is to generate an optimal allocation. However, the explicit enumeration of all allocations in order to extract the largest social value is not a scalable approach, as a result of the exponential size of the solution space. However, optimal utilitarian allocations satisfy a specific structural property when additive utility functions express agents' preferences. This property can be used to simplify the generation of an optimal allocation, and then the determination of the optimal utilitarian welfare value. It specifies how to allocate resources over the population in order to maximize the utilitarian efficiency.

Property 4.1 (Utilitarian optimum). *In utilitarian optimal allocations, each resource is allocated to one of the agents who associates the largest utility value with it.*

Proof. Let us make a proof by contradiction. A resource allocation $A \in \mathcal{A}$ is assumed to be a global optimum, in which an agent $i \in \mathcal{P}$ owns a resource $r \in \mathcal{R}_i$. Let us now assume that another agent $j \in \mathcal{P} \setminus \{i\}$ associates a greater utility value with this resource r. More formally:

$$\exists (r, j) \in \mathcal{R} \times \mathcal{P} \quad \text{such that} \quad u_j(r) > u_i(r), \quad i \in \mathcal{P}.$$

If the resource allocation $A' \in \mathcal{A}(A \neq A')$ corresponds to the allocation in which resource r is allocated to agent j, then the following expression is satisfied:

$$\begin{aligned} sw_u(A) &= \sum_{k \in \mathcal{P}} u_k(\mathcal{R}_k) \\ &= u_i(\mathcal{R}_i) + u_j(\mathcal{R}_j) + \sum_{k \in \mathcal{P} \setminus \{i,j\}} u_k(\mathcal{R}_k) \\ &< u_i(\mathcal{R}_i) - u_i(r) + u_j(\mathcal{R}_j) + u_j(r) + \sum_{k \in \mathcal{P} \setminus \{i,j\}} u_k(\mathcal{R}_k) \\ &< sw_u(A'). \end{aligned}$$

The utilitarian welfare value associated with A' is greater than the one associated with A: $sw_u(A) < sw_u(A')$. Then, allocation A cannot be a global optimum since there exists an allocation associated with a greater utilitarian welfare value. Thus, allocations that do not allocate all resources to one of the agents who associates with them the largest utility value are not a global optimum. □

Thus, thanks to this property, optimal allocations can be easily generated when the utilitarian welfare is considered. A simple algorithm can be designed for this purpose. According to Proposition 4.1, such an algorithm has to allocate each resource to an agent who values it the most, as described in Algorithm 4.1.

Algorithm 4.1: Determination of the optimal utilitarian welfare value

Input: Agent population $\mathcal{P}$, Resource set $\mathcal{R}$

Output: $sw_u^\star$ the optimal utilitarian value

forall the $r \in \mathcal{R}$ **do**

    $i \leftarrow \arg\max_{k \in \mathcal{P}} u_k(r)$;   `// Determination of who estimates` r `the most`

    Add r to $A[i]$;   `// Add` r `to agent` i`'s bundle`

end

return $sw_u(A)$;

4.1.2 Utilitarian negotiation properties

The expression of the rationality test does not vary according to the welfare notion considered since it is only based on the agent resource bundle. However, the sociability criterion is based on the chosen welfare notion and its expression can then be specified. It is based on the evolution of the utilitarian welfare value during a transaction. Let us note $A \in \mathcal{A}$ the resource allocation before the bilateral transaction $\delta_i^j\langle a, b\rangle$ and A' the allocation afterwards. Such a transaction involves two agents $i, j \in \mathcal{P}$ who respectively propose offers ρ_i^δ and ρ_j^δ. The resource bundle of any agent $k \in \mathcal{P}$ is denoted by $\mathcal{R}_k$ before the transaction and by $\mathcal{R}'_k$ afterwards ($k \in \{i, j\}$). Any social transaction must satisfy the following expression:

$$\begin{aligned}
sw_u(A) &< sw_u(A') \\
\sum_{k\in\mathcal{P}} u_k(\mathcal{R}_k) &< \sum_{k\in\mathcal{P}} u_k(\mathcal{R}'_k) \\
u_i(\mathcal{R}_i) + u_j(\mathcal{R}_j) + \sum_{k\in\mathcal{P}\setminus\{i,j\}} u_k(\mathcal{R}_k) &< u_i(\mathcal{R}'_i) + u_j(\mathcal{R}'_j) + \sum_{k\in\mathcal{P}\setminus\{i,j\}} u_k(\mathcal{R}'_k) \\
u_i(\mathcal{R}_i) + u_j(\mathcal{R}_j) &< u_i(\mathcal{R}'_i) + u_j(\mathcal{R}'_j) \\
u_i(\mathcal{R}_i) + u_j(\mathcal{R}_j) &< u_i(\mathcal{R}_i) + u_i(\rho_j^\delta) - u_i(\rho_i^\delta) + u_j(\mathcal{R}_j) + u_j(\rho_i^\delta) - u_j(\rho_j^\delta) \\
u_i(\rho_i^\delta) + u_j(\rho_j^\delta) &< u_i(\rho_j^\delta) + u_j(\rho_i^\delta)
\end{aligned}$$

Thus, the utilitarian acceptability criterion is only based on the offers proposed by the participants. During a social transaction, agents who receive resources must associate with them a larger utility value than their initial owner. The initial welfare of participants does not affect the acceptability of a transaction. This expression corresponds to the acceptability test that agents perform to determine whether or not a transaction is profitable when the utilitarian welfare is considered. Hence, the acceptability test, which is represented by the instruction TEST in all behaviors of Section 2.6, can be replaced by:

$$\textsc{test} := \left[u_i(\rho_i^\delta) + u_j(\rho_j^\delta) < u_i(\rho_j^\delta) + u_j(\rho_i^\delta)\right]$$

When the utilitarian welfare is considered, bilateral transactions have some important properties. Acceptable bilateral transactions $\delta_i^j\langle a,b\rangle$ may not be split into a sequence of acceptable bilateral transactions of lesser cardinality. This means that transactions with a large cardinality might be required to achieve socially optimal allocations.

Property 4.2 (Utilitarian transaction split). *Within an utilitarian society where agents express their preferences by means of additive utility functions, it is not always possible to split social bilateral transactions $\delta_i^j\langle a,b\rangle$ between two agents $i,j \in \mathcal{P}$ into a sequence of social bilateral transactions $\delta_i^j\langle a',b'\rangle$ of lesser cardinality ($a \geq a'$ and/or $b \geq b'$).*

Proof. Let us consider a counter example based on a population of two agents, $\mathcal{P} = \{0,1\}$, who negotiate three available resources $\mathcal{R} = \{r_1, r_2, r_3\}$. Their preferences are expressed by utility functions described in Table 4.1. The initial resource allocation is $A = [\{r_1, r_2\}\{r_3\}]$: Agent 0 owns two resources, $\mathcal{R}_0 = \{r_1, r_2\}$, whereas agent 1 only owns resource $\mathcal{R}_1 = \{r_3\}$.

Table 4.1: Utilitarian transaction split - Example of agent preferences

Population $\mathcal{P}$	Resource Set $\mathcal{R}$		
	r_1	r_2	r_3
0	4	6	7
1	10	1	3

Let us consider the transaction $\delta_0^1\langle 2,1\rangle = (\{r_1,r_2\},\{r_3\})$, which changes the initial resource allocation A into another allocation A' ($A, A' \in \mathcal{A}$). During this transaction, agent 0 proposes $\rho_0^\delta = \{r_1, r_2\}$ while agent 1 proposes $\rho_1^\delta = \{r_3\}$. Such a transaction is social since:

$$u_0(\{r_3\}) + u_1(\{r_1,r_2\}) > u_0(\{r_1,r_2\}) + u_1(\{r_3\})$$

Such a transaction leads to an increase of the utilitarian welfare value from $sw_u(A) = 13$ initially to $sw_u(A') = 18$ afterwards. However, this transaction $\delta_0^1\langle 2,1\rangle$ can be split into a sequence of social transactions. Two decomposition patterns can be observed: A swap transaction concatenated with a gift or three successive gifts are the lone possible sequences. Table 4.2 describes the three possible sequences containing transactions of lesser cardinality. In each of them, at least one transaction of the sequence is not acceptable.

Table 4.2: Utilitarian transaction split - List of possible sequences

Split of δ_0^1		Social sequence?
$(\{r_1, r_2\}, \{r_3\})$	$= (\{r_1\}, \{r_3\}) + (\{r_2\}, \emptyset)$	$(\{r_2\}, \emptyset)$ is not social
	$= (\{r_2\}, \{r_3\}) + (\{r_1\}, \emptyset)$	$(\{r_2\}, \{r_3\})$ is not social
	$= (\{r_1\}, \emptyset) + (\{r_2\}, \emptyset) + (\emptyset, \{r_1\})$	$(\{r_2\}, \emptyset)$ is not social

According to Table 4.2, the transaction $\delta_0^1 = (\{r_1, r_2\}, \{r_3\})$ cannot be split into a sequence of acceptable transactions of lesser cardinality. Thus, social bilateral transactions $\delta_i^j\langle a, b\rangle$ cannot always be split into sequences of social transactions of lesser cardinality when the utilitarian welfare is considered. Let us note that, since the utility values are positive, any transaction that cannot be split is locally sub-optimal: A transaction of lesser cardinality achieves a larger utilitarian value. □

4.1.3 Evaluation of utilitarian negotiations

This section is dedicated to the evaluation of the different facets of utilitarian negotiations. First, impacts of restrictions on the transaction cardinality are investigated. Then, the efficiency of negotiations based on different transactions, on different graphs and on different acceptability criteria are studied, using a comparison with the optimal welfare value provided by centralized methods. The impact of the social graph connectivity is then presented. Agent behaviors are evaluated using several metrics to quantify the negotiation process quality. Finally, issues related to the scalability of utilitarian negotiation processes are discussed.

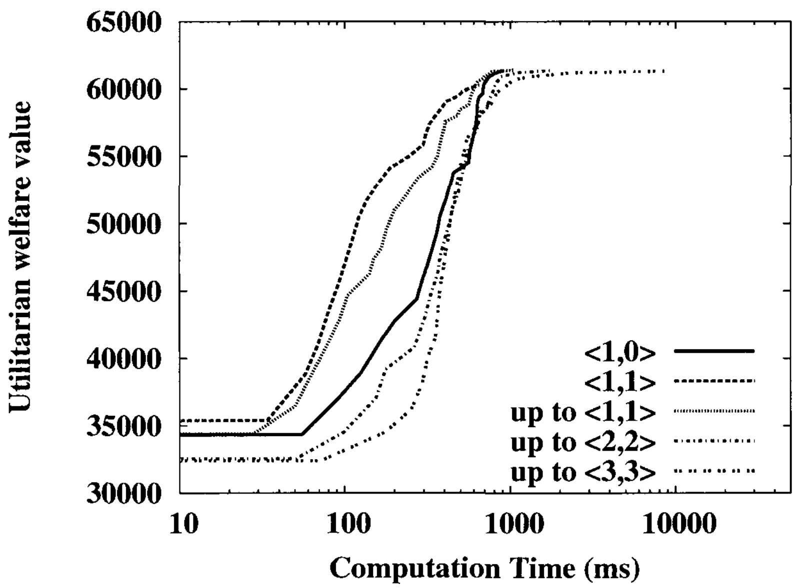
Influence of the transaction cardinality

According to Definition 2.6, a bilateral transaction $\delta\langle a, b\rangle$ is defined using two parameters a and b, which bound the size of agents' offers. Their size influences a lot the resource negotiation process. Proposing large offers during a transaction may be theoretically required to guarantee the achievement of optimal allocations since they cannot always be split. However, allowing such large offers increases exponentially the cost of a negotiation with the size of the participant's bundle. Thus, the following question can be raised: Does the efficiency improvement justifies additional costs induced by the use of large bilateral transactions?

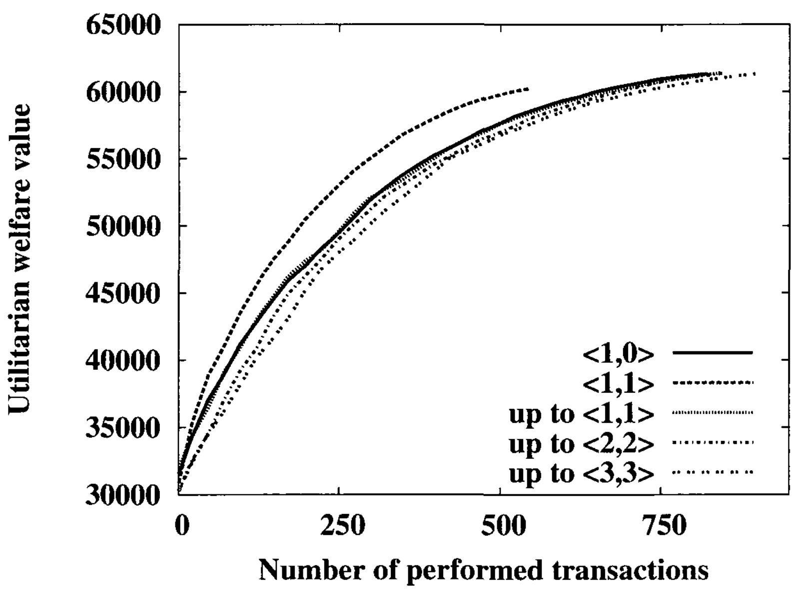
Figures 4.1 show the impact of the transaction cardinality on utilitarian negotiation processes, according to two metrics: The computation time and the number of transactions performed during the whole processes. Negotiation processes are based here on a population of 50 social agents who negotiate 250 resources on complete graphs.

Figure 4.1a represents the utilitarian welfare value evolution according to the computation time, while Figure 4.1b represents its evolution according to the number of performed transactions. On each figure, different transactions are allowed during the negotiation process. The keys characterizing the curves in these graphs represent the transaction cardinality $\langle a, b\rangle$. The curve denoted by "up to $\langle 2,2\rangle$" means that agents can propose from an empty offer to a set of 2 resources. Then, the set of allowed transactions $\mathcal{T}$ can be explicitly written as: $\mathcal{T} = \{\langle 1,0\rangle, \langle 0,1\rangle, \langle 1,1\rangle, \langle 2,1\rangle, \langle 1,2\rangle, \langle 2,2\rangle\}$.

Figure 4.1a shows that, independently of the cardinality of the allowed transactions, all negotiation processes converge towards very close utilitarian welfare values. Large bilateral transactions do not allow the achievement of greater utilitarian allocations whereas they are more time consuming. Negotiation processes based either on $\langle 0,1\rangle$ transactions (gifts), on $\langle 1,1\rangle$ transactions (swaps), or on both transactions ("up to $\langle 1,1\rangle$") end after 1 second while 10 seconds are required when transactions of cardinality "up to $\langle 3,3\rangle$" are allowed for instance. Figure 4.1b shows that the number of performed transactions is almost similar in all cases. Large bilateral transactions do not shorten transaction sequences required to close the negotiation processes. Negotiation processes based on swaps $\langle 1,1\rangle$ end on socially weaker allocations. Since the initial resource distribution cannot be modified, negotiation processes based on swaps end on weaker local optima. Since the utilitarian welfare value achieved is almost similar, independently of the transaction cardinality, the use of large bilateral transactions is not justified due to important additional costs.



(a)



(b)

Figure 4.1: Evaluation of the transaction cardinality impact in terms of computation time in 4.1a and of performed transactions in 4.1b.

Price of social graphs

The efficiency of negotiation processes is an essential feature. This efficiency is evaluated here by a comparison between optimal utilitarian welfare values, provided by centralized algorithms described in Section 4.1.1, and social values provided by agent negotiations. Negotiations are based on a population of 50 agents where 250 resources are available.

Table 4.3 presents the efficiency of negotiation processes based on different sets of allowed transactions, on different acceptability criteria, and on different classes of social graphs. This table shows the proportion of the optimal welfare value that can be achieved. The greater is the proportion, the closer optima are the resulting allocations. Table 4.4 describes the standard deviations observed among the social values provided from different initial allocations. A large standard deviation means a high topological sensitivity. For instance, negotiation processes based on a grid where rational agents negotiate using $\delta\langle 1,1\rangle$ transactions only end on social values representing 79.0% of the optimum with a standard deviation of 1.6%. Depending on the initial resource allocation, the utilitarian welfare value achieved may vary of 1.6%.

Table 4.3: Utilitarian efficiency (%) according to the class of social graphs

Social graph kind	Rational		Social			
	$\langle 1,1\rangle$	up to $\langle 2,2\rangle$	$\langle 1,0\rangle$	$\langle 1,1\rangle$	up to $\langle 1,1\rangle$	up to $\langle 2,2\rangle$
Full	96.6	97.0	100	98.3	100	100
Grid	79.0	81.3	86.2	85.3	86.1	86.1
Erdős-Rényi	94.8	95.0	98.9	97.1	98.9	98.9
Small world	80.8	84.8	91.4	90.0	90.2	90.3

Table 4.4: Standard deviation of the utilitarian efficiency (%) according to the class of social graphs

Social graph kind	Rational		Social			
	$\langle 1,1\rangle$	up to $\langle 2,2\rangle$	$\langle 1,0\rangle$	$\langle 1,1\rangle$	up to $\langle 1,1\rangle$	up to $\langle 2,2\rangle$
Full	0.3	0.2	0	0.2	0	0
Grid	1.6	1.3	0.9	1.1	0.9	0.9
Erdős-Rényi	0.5	0.4	0.1	0.2	0.1	0.1
Small world	2.0	1.3	0.8	1.0	0.8	0.8

When considering complete social graphs, different negotiation strategies always lead to optimal resource allocations. The transactions of weakest cardinality, which achieves optimal allocations, are social gifts, i.e., social $\langle 1,0 \rangle$ transactions. Any negotiation policy that includes social gifts, like "up to $\langle 1,1 \rangle$", "up to $\langle 2,2 \rangle$" or "up to $\langle 3,3 \rangle$" also achieve socially optimal resource allocations. However, their use leads to important additional costs. *The use of social gifts is sufficient to achieve optimal allocations when the utilitarian welfare is considered.* Table 4.3 also shows that, independently of the social graph class, rational negotiation processes always lead to socially weaker allocations than social negotiation processes. The restrictive character of the acceptability criterion affects the resource circulation, and then the quality of the provided solution. The more restricted are social graphs, the weaker is the negotiation efficiency. The combination of a restricted social graph like a grid and the use of rational swaps, which restrict a lot transaction possibilities (since initial resource distributions cannot be modified), leads to the worst social efficiency: Only 79% of the optimal welfare value can be achieved. When grids are considered, social negotiation processes achieve up to 86.2% of the optimum. The weak mean connectivity handicaps the resource traffic and hence the achievement of socially efficient allocations. Negotiation processes lead to allocations associated with up to 98.9% of the optimal welfare value when Erdős-Rényi graphs are considered. Only 91.4% of the optimum is achieved when small-worlds are considered. In an Erdős-Rényi graph, the probability for a link to exist between any pair of nodes is always the same, while in small-worlds, the larger is the number of an agent's neighbors, the higher is the probability to link this agent. Many agents have only one neighbors, and the resource traffic is unequally distributed. Then, bottlenecks, i.e., agents who block the resource circulation, may appear. Swaps are the least efficient transactions, but the difference is generally small. Since the number of resources per agent cannot vary, the resource circulation is very limited. In all cases, the standard deviation observed among the social values achieved remains weak for a given class of social graphs. It means that when the utilitarian welfare is considered, the topology has not a significant impact for a given class. The deviation is higher when rational transactions are considered. Indeed, the rational acceptability criterion holds up the resource traffic, which then influences on the quality of the provided allocations. The more restricted is the resource traffic, the higher is the standard deviation, and thus more important become the initial resource allocation.

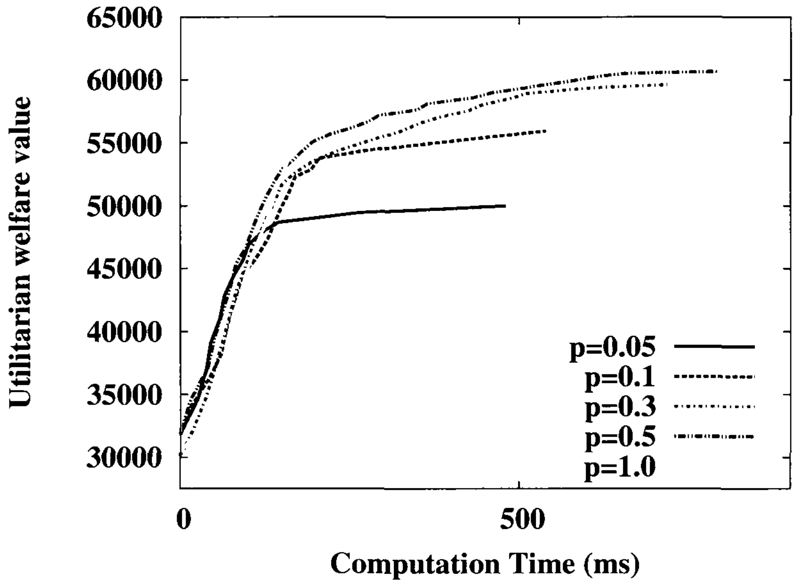
Theorem 4.3. *Within an utilitarian society, where agents express their preferences by means of additive utility functions, negotiation processes based on complete social graphs always converge towards a global optimum using only social* $\langle 1, 0 \rangle$ *transactions.*

Proof. Since the social graph is complete and fully connected, any agent $i \in \mathcal{P}$ can communicate with every other agents $j \in \mathcal{P} \setminus \{i\}$. If a social $\langle 1, 0 \rangle$ transaction containing r can be performed between agents i and j, then $u_j(r) > u_i(r)$ according to the definition of a social transaction. It is always possible to create a sequence of social $\langle 1, 0 \rangle$ transactions leading a resource into the bundle of an agent who associates the largest utility value with it. Applying this process to each resource, and according to Proposition 4.1, the resulting allocation is a global optimum. □

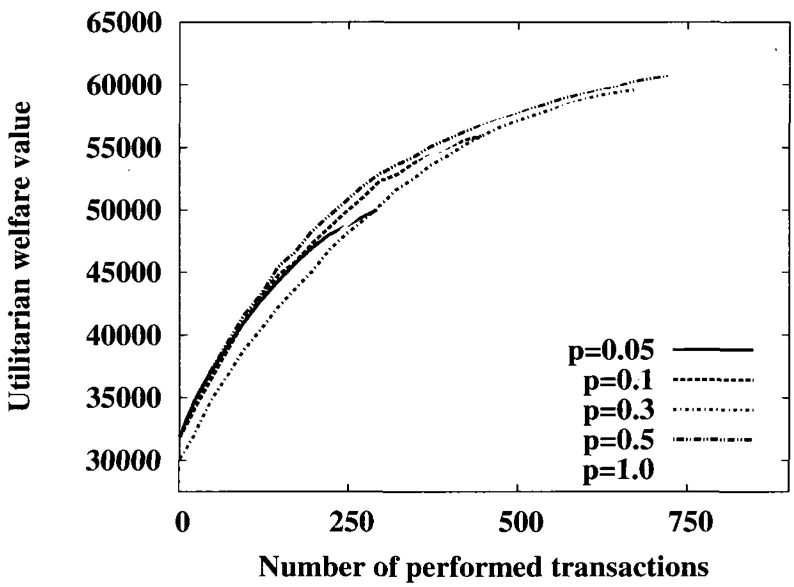
Influence of the social graph connectivity

The social graph topology greatly affects the resource circulation and the negotiation efficiency. The larger are agent neighborhoods, the denser are social graphs, and the easier is the resource traffic. The model of generation for Erdős-Rényi graphs $G(n, p)$ is used to evaluate the impact of the connectivity on utilitarian negotiation processes. The probability p for a link to exist between nodes from any pair can be modified. Note that such a model does not guarantee that the generated graphs are connected.

Figure 4.2 shows the impact of the social graph connectivity on the negotiation efficiency within a population of 50 agents who negotiate 250 resources using social $\langle 1, 0 \rangle$ transactions. Figure 4.2a represents the evolution of the utilitarian welfare value according to the computation time, whereas Figure 4.2b represents its evolution according to the number of performed transactions.



(a)



(b)

Figure 4.2: Evaluation of the mean connectivity impact in terms of computation time in 4.2a and of performed transactions in 4.2b.

These figures show that a weak probability, which corresponds to small agent neighborhoods, leads to short transaction sequences and utilitarian welfare values far from the optimum. For instance, when $p = 0.05$, negotiations end after a sequence of 300 gifts performed in only 0.5 second. However, negotiation processes end on allocations socially far from the optimum. The gradual increase of the probability p leads to longer transaction sequences, to the achievement of larger utilitarian welfare values, and to more time-consuming negotiations. Larger neighborhoods facilitate the resource circulation by offering a larger number of possible transactions to all agents. The impact becomes really significant when $p < 0.3$. Above this value, the resource circulation is sufficient to achieve socially interesting allocations, but below this threshold, social graphs are too restricted, and the flexibility of the social criterion cannot compensate for the restrictiveness of graph topologies.

Influence of agent behaviors

Behaviors define how agents interact with their neighbors, and then how they negotiate. Different behaviors, defined in Section 2.6, can be compared using the metrics presented in Section 2.7.1. In order to evaluate the agents' behaviors, any factor that may alter the comparison should be avoided, like the social graph topology for instance. For this purpose, the negotiation processes which are compared here, are based on complete social graphs, with a population of 50 agents and 250 resources. Agents negotiate using social $\langle 1, 0 \rangle$ transactions only since they are the most efficient transactions.

The name of agent behaviors are abbreviated in Figure 4.3, as described in Table 2.5. Let us recall that "rs" corresponds to rooted stubborn behaviors, while "fs" defines frivolous stubborn behaviors. Flexible behaviors can be either rooted "rf" or frivolous "ff". According to behaviors qualified as full, the initiator makes an exhaustive negotiation with its partner. If agents are moreover rooted, their behaviors correspond to "rf full", whereas when agents are frivolous, a priority can be defined either on offers or on partners. "rf full - agent" denotes agent behaviors favoring partner changes, while "rf full - resource" denotes agent behaviors favoring offer changes.

When rooted stubborn agents negotiate, transaction sequences are short: Only few transactions are performed before the end of utilitarian negotiations. Indeed, around 400 social gifts are required on average to end such processes. Agents always propose first the offer the least penalizing for them. However, since agents are stubborn, they only attempt a single offer, which is always a singleton since all utility values are positive. Thus, according to rooted stubborn behaviors, the number of performed transactions is equivalent to the number of exchanged resources. A large number of speech turns is required to end the negotiation process: Many speech turns are required to communicate with all neighbors for instance. Since agents attempt a single offer per negotiation, the number of attempted transactions is equivalent to the number of speech turns. Thus, when agents interact according to rooted and stubborn behaviors, negotiation processes are quite short, only few offers are attempted and lesser are performed. Such processes generally end on allocations associated with weak social values.

A stubborn but frivolous agent behavior leads to a weak number of performed transactions as well as to a weak number of exchanged resources. Since the initiator can change partners, the number of speech turns required to end such negotiations is weaker than in the case of rooted stubborn behaviors. The number of attempted transactions becomes 10 times higher. Moreover, negotiation processes can achieve socially more interesting allocations, even if they become more time consuming. Both stubborn behaviors are not socially efficient since corresponding negotiation processes end on socially weaker allocations (15% weaker) after similar elapsed time.

Negotiations among flexible agents increase drastically the number of performed transactions. More transactions are performed if the flexible negotiations are "full" (since such behaviors identify all acceptable transactions if some exist in the neighborhood), but the number of exchanged resources are close. If agent behaviors are moreover rooted, the number of speech turns becomes very large. More than 3000 negotiations are required in the case of rooted agents while only 1300 steps are sufficient when agents behave frivolously.

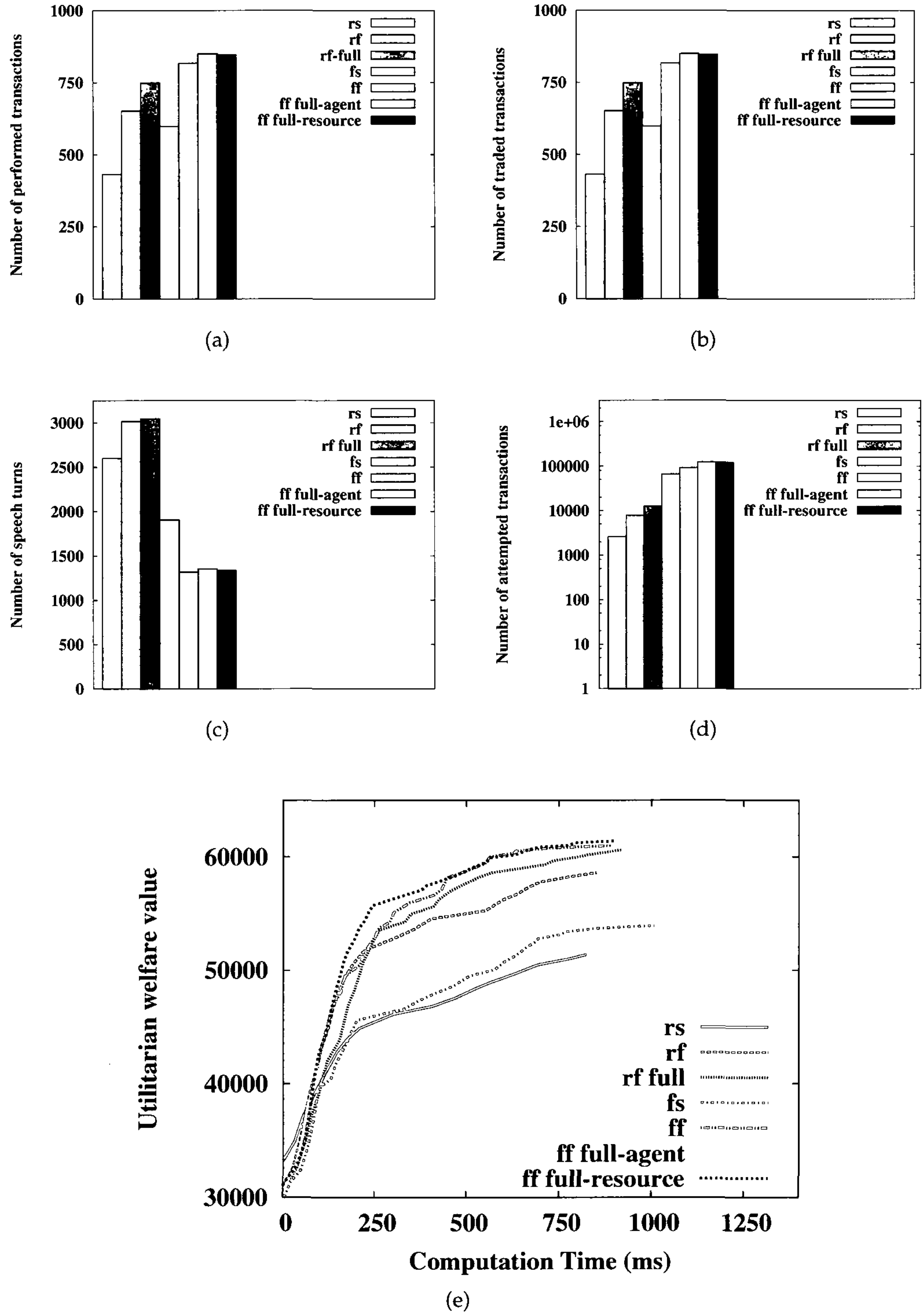


Figure 4.3: Evaluation of the agent behavior impact in terms of performed transactions in 4.3a, in terms of transacted resources in 4.3b, in terms of speech turns in 4.3d, in terms of attempted transactions in 4.3e, and in terms of computation time in 4.3e.

However, frivolous flexible behaviors lead to a very large number of attempted transactions (10 times more). The number of attempted transactions increases exponentially with the mean number of resources per agent. No real difference can be distinguished between the frivolous flexible behaviors. The values of the different metrics are always close. Indeed, on complete graphs, the order on which agents negotiate is not critical, especially when the utilitarian welfare notion is considered.

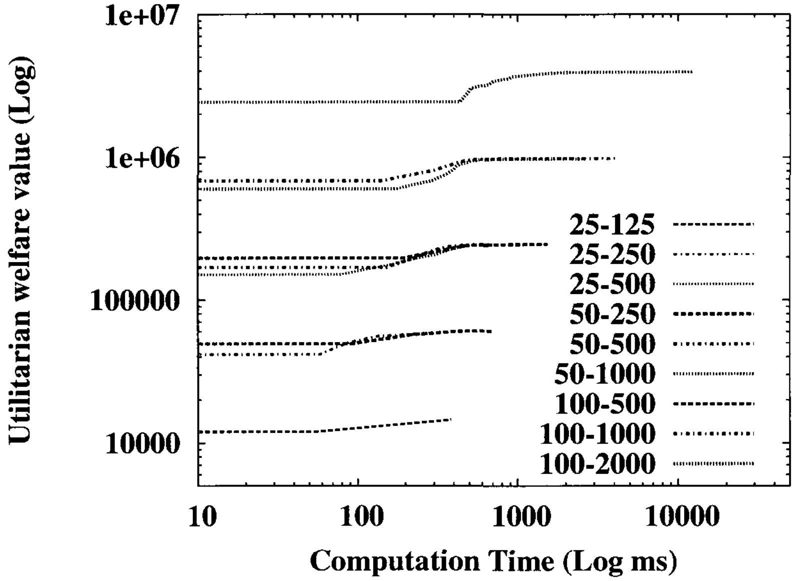
Negotiation scalability

The scalability is also an important issue, which is evaluated according to the conditions described in Section 2.7.3, i.e., on complete social graphs among frivolous and flexible agents.

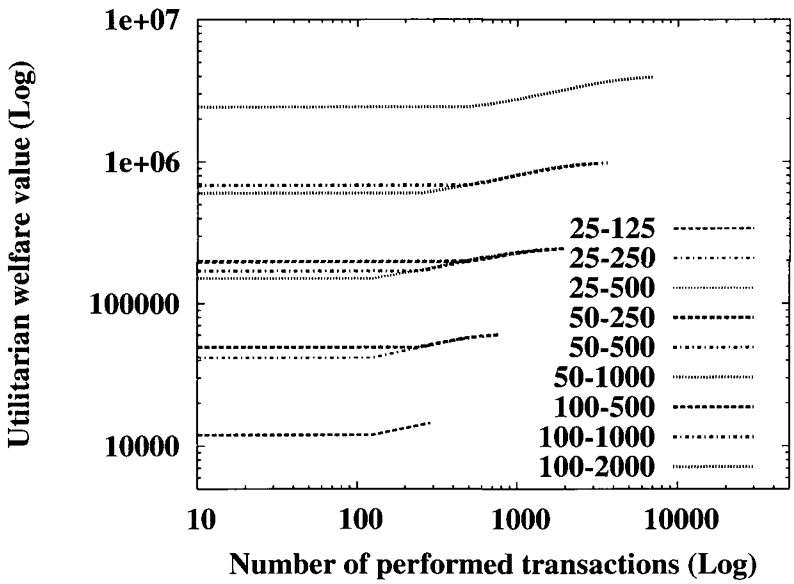
Figure 4.4a represents the evolution of the utilitarian welfare value according to the computation time, on several population sizes, while Figure 4.4b shows its evolution according to the number of performed transactions. The different curves of these figures are characterized by a pair $n - m$ describing the size of the instances, where n is the number of agents and m the overall number of resources. Then, the key "25-125" on Figures 4.4 corresponds to instances populated by 25 agents who are negotiating 125 resources. These graphs underline the impact of the instance size on the observed metrics. Independently of the mean number of resources per agent, the increase of the metric values is almost regular.

Tables 4.5 and 4.6 respectively present the elapsed time that is required to end utilitarian negotiation processes according to several instance sizes and to the number of performed transactions. Each experiment is characterized by the population size n and by the mean number of resources per agent $\frac{m}{n}$. According to these tables, 100 agents owning on average 10 resources each, end utilitarian negotiation processes in 4.1 seconds after a sequence of 3700 social gifts. These tables show that large instances can still be solved in a reasonable time.

Property 4.4 (Utilitarian gift-based negotiation complexity (Endriss and Maudet, 2005)). *During a negotiation process based on social gifts, the number of distinct attempted transactions*



(a)



(b)

Figure 4.4: Evaluation of utilitarian scalability in terms of computation time in 4.4a and performed transactions in 4.4b.

Table 4.5: Utilitarian negotiation scalability - Computation time

Population size n	Mean number of resources per agent $\frac{m}{n}$		
	5	10	20
25	400 ms	550 ms	950 ms
50	625 ms	1.2 s	2.4 s
100	1.7 s	4.1 s	12 s
500	45 s	150 s	450 s

Table 4.6: Utilitarian negotiation scalability - Number of performed transactions

Population size n	Mean number of resources per agent $\frac{m}{n}$		
	5	10	20
25	325	625	1300
50	800	1500	3000
100	1900	3700	7100
500	13500	25000	47500

and the number of transactions that can be performed are both polynomial.

Proof. When the utilitarian welfare is considered, a social transaction $\delta_i^j\langle 1, 0\rangle$ between two agents $i, j \in \mathcal{P}$, in which resource $r \in \mathcal{R}_i$ is offered, is characterized by the relation $u_j(r) > u_i(r)$. Then, during a social gift sequence, the utility value associated with r gradually increases with its successive owners. No social gift allows the return of r to former owners. No cycle of social gifts can then appear. A specific resource r can be transacted at most $n - 1$. Then, the overall number of performed transactions is bounded by $m(n - 1) \sim O(nm)$. □

The demonstration of the complexity in terms of attempted transactions depends a lot on features like the implementation or the distribution of the speech turns. However, if they are uniformly distributed as described in Section 3.4, the proposition can be demonstrated as follows:

Proof. The maximum number of distinct attempted gifts per agent corresponds to $m(n - 1)$. Indeed, in the worst case, an agent may punctually own each resource and tries to give them to everybody. Thus, the number of distinct attempted transactions is bounded by $m(n - 1)^2 \sim O(n^2m)$. □

4.1.4 Conclusion

Centralized approaches are quite trivial when utilitarian problems are considered. Each resource must be allocated to one of the agents who associates with it the largest utility. In distributed agent negotiations, the use of social $\langle 1,0\rangle$ transactions is the most efficient negotiation policy among frivolous and flexible agents.

Best utilitarian negotiation policy

Transaction:	$\langle 1,0\rangle$ (i.e., gifts)
Criterion:	social
Test on δ_i^j:	$u_i(\rho_i^\delta) + u_j(\rho_j^\delta) < u_i(\rho_j^\delta) + u_j(\rho_i^\delta)$
Behavior:	frivolous and flexible

Bilateral transactions are sufficient to achieve socially interesting allocations. The utilitarian welfare notion is flexible enough to favor the resource traffic. The more resources circulate among agents, the easier is the achievement of optimal allocations. Experiments show that large bilateral transactions do not improve the provided solutions, but lead to important additional costs, especially in terms of computation time. Social $\langle 1,0\rangle$ transactions are sufficient to guarantee that optimal allocations are achieved when negotiations are based on complete social graphs. When restricted social graphs are considered, like Erdős-Rényi graphs, grids or small-worlds, social $\langle 1,0\rangle$ transactions cannot guarantee the achievement of a social optimum but lead to socially close resource allocations. The rational acceptability criterion, which is usually used in the literature, restricts a lot the resource circulation and leads utilitarian negotiations to suboptimal allocations, which may be far from the optimum. Utilitarian negotiations based on social $\langle 1,0\rangle$ transactions remain scalable even when large instances are considered. Allocations maximizing the utilitarian welfare can be achieved by a negotiation process among flexible and frivolous agents who negotiate with social gifts, independently of the social graph considered.

4.2 Egalitarian bilateral negotiations

The egalitarian welfare is an important notion, especially when fairness in a society of agents must be achieved. This notion focuses on the welfare of the poorest agent within the population. In this section, centralized approaches are first described in order to estimate the egalitarian optimal value. Egalitarian negotiation issues are then discussed with the specification of the acceptability test expression and the detail of some important properties of agent negotiations. Finally, egalitarian negotiations are evaluated to identify the suitable parameters allowing the achievement of fairness within societies.

4.2.1 Centralized techniques

The identification of the optimal egalitarian welfare value is a difficult problem.

Theorem 4.5 (Egalitarian welfare optimum complexity (Bouveret and Lang, 2005)). *The determination of the optimum egalitarian welfare value is a $\mathcal{NP}$-hard problem.*

Egalitarian resource allocation problems can be formulated by means of a mathematical model using variables x_{ir} describing the ownership of a resource $r \in \mathcal{R}$ by an agent $i \in \mathcal{P}$:

$$x_{ir} = \begin{cases} 1 & \text{if agent } i \text{ owns resource } r \\ 0 & \text{otherwise.} \end{cases} \qquad r \in \mathcal{R}, \quad i \in \mathcal{P}.$$

Then, egalitarian resource allocation problems can be written as follows:

$$sw_e^\star = \begin{cases} \max \min\limits_{i \in \mathcal{P}} \sum\limits_{r \in \mathcal{R}} u_i(r) x_{ir} \\ \text{s.t:} \ \sum\limits_{i \in \mathcal{P}} x_{ir} = 1 \qquad r \in \mathcal{R} \\ \qquad x_{ir} \in \{0, 1\} \qquad r \in \mathcal{R}, \quad i \in \mathcal{P}. \end{cases}$$

The objective is the maximization of the welfare of the poorest agent. Two consistency constraints are also defined. The first one ensures that each resource is allocated to a single agent while the second constraint specifies that all resources are discrete and not sharable. Then, variables x_{ir} correspond to Boolean variables. However, the model can be adapted easily to other resource natures if required. For instance, continuous resources should be represented by real variables. The second constraint becomes: $x_{ir} \in [0,1], \quad r \in \mathcal{R}, i \in \mathcal{P}$.

This model can be solved using any mathematical programming optimizer like CPLEX (ILOG Inc, 1995). However, such a method does not provide an exact solution, and only provides an estimation that can be more or less accurate according to the required gap. This gap is a parameter provided to the solver as an estimation of the distance between returned solutions and optimal ones. The weaker the gap, the more accurate are provided social values, but more time-consuming becomes the solving process. A centralized solving process with a required null gap, is almost similar to the explicit enumeration of all allocations and hence cannot be considered to be scalable.

Heuristics can also be designed to build allocations associated with near-optimal social values. Since egalitarian negotiation processes tend to reduce inequalities, the equal distribution of resources over the population can be considered. One way to proceed is to sequentially allocate to each agent the best remaining resource, as described in Algorithm 4.2. Another way to proceed is to sequentially allocate the current resource to the poorest agent of the population, as described in Algorithm 4.3.

Algorithm 4.2: Estimation of the optimal egalitarian welfare value - 1

Input: Agent population $\mathcal{P}$, Resource set $\mathcal{R}$

Output: $sw_e^\star$ the estimation of the optimal egalitarian value

```
i ← 0 ;
Shuffle(P) ;                                  // Mix the population P
while R ≠ ∅ do
    r ← argmin_{r'∈R} u_i(r') ;    // Determination of the best remaining resource
    Add r to A[i] ;                         // Allocation of resource r to agent i
    R ← R \ {r} ;
    i ← (i + 1)%n ;
end
return sw_e(A) ;
```

Algorithm 4.3: Estimation of the optimal egalitarian welfare value - 2

Input: Agent population $\mathcal{P}$, Resource set $\mathcal{R}$

Output: $sw_e^\star$ the estimation of the optimal egalitarian value

```
forall the r ∈ R do
    i ← argmin_{j∈P} u_j(R_j) ;                // Determination of the poorest agent
    Add r to A[i] ;                         // Allocation of resource r to agent i
end
return sw_e(A) ;
```

In spite of their scalability, these two heuristics have a major drawback affecting the quality of the solutions. They are not really reliable. Indeed, both of them are very sensitive to the order in which agents are considered. Depending on this order, egalitarian welfare values provided by such heuristics may vary a lot.

4.2.2 Egalitarian negotiation properties

To express the social acceptability criterion, we use the definition of the egalitarian welfare (Definition 1.5). The initial resource allocation $A \in \mathcal{A}$ changes into another one A' by means of a social transaction $\delta_i^j\langle a, b\rangle$. Such a transaction involves two agents $i, j \in \mathcal{P}$, who respectively propose the offers ρ_i^δ and ρ_j^δ. The resource bundle of any agent $k \in \mathcal{P}$ is denoted by $\mathcal{R}_k$ before the transaction and $\mathcal{R}'_k$ afterward. According to the social acceptability criterion, an egalitarian transaction must satisfy the following condition:

$$\begin{aligned} sw_e(A) &\leq sw_e(A') \\ \min_{i\in\mathcal{P}}(u_i(\mathcal{R}_i)) &\leq \min_{i\in\mathcal{P}}\left(u_i(\mathcal{R}'_i)\right) \end{aligned}$$

When the egalitarian welfare is considered, the expression of the social acceptability criterion is not a strict inequality. Indeed, depending on the involvement of the poorest agent in the current transaction, the egalitarian welfare value may not increase. If the poorest agent is not involved in the current bilateral transaction δ_i^j, its utility value, which corresponds to the egalitarian welfare value, does not vary since its resource bundle is not modified. Thus, the egalitarian welfare value changes only if the poorest agent of the population is involved.

The expression that social transactions δ_i^j must satisfied can be restricted to only two agents. In such a case, the poorest agent after an egalitarian transaction must be richer than it was before the transaction.

$$\begin{aligned} \min_{i,j\in\mathcal{P}}\left(u_i(\mathcal{R}_i), u_j(\mathcal{R}_j)\right) &< \min_{i,j\in\mathcal{P}}\left(u_i(\mathcal{R}'_i), u_i(\mathcal{R}'_j)\right) \\ \min_{i,j\in\mathcal{P}}\left(u_i(\mathcal{R}_i), u_j(\mathcal{R}_j)\right) &< \min_{i,j\in\mathcal{P}}\left(u_i(\mathcal{R}_i) + u_i(\rho_j^\delta) - u_i(\rho_i^\delta), u_j(\mathcal{R}_j) + u_j(\rho_i^\delta) - u_j(\rho_j^\delta)\right) \end{aligned}$$

In contrast to the utilitarian expression of the social acceptability criterion, which only depends on the traded resources, the egalitarian expression is based on the sets of traded resources transacted as well as on the initial resource bundle of each agent. According to such a criterion, a very rich agent may accept to decrease its own utility for the sake of the whole society. In the case where the poorest agent of the population is not involved in the current bilateral transaction, the social criterion favors the resource circulation, and consequently by the negotiation process can avoid local optima. Indeed, during egalitarian negotiation processes, resources progressively move from richer agents to poorer agents to distribute the richness among agents. The agent's decision making is represented by the instruction TEST in all agent's behaviors described in Section 2.6. This test allows agents to determine whether a transaction is fair or not. It can be written as follows:

$$\text{TEST} := \left[\min_{i,j\in\mathcal{P}}\left(u_i(\mathcal{R}_i), u_j(\mathcal{R}_j)\right) < \min_{i,j\in\mathcal{P}}\left(u_i(\mathcal{R}_i) + u_i(\rho_j^\delta) - u_i(\rho_i^\delta), u_j(\mathcal{R}_j) + u_j(\rho_i^\delta) - u_j(\rho_j^\delta)\right)\right]$$

When the egalitarian welfare is considered, bilateral transactions have some important properties. Social bilateral transactions $\delta_i^j\langle a, b\rangle$, i.e., transactions satisfying the egalitarian acceptability criterion, may not be split into a sequence of egalitarian bilateral transactions of lesser cardinality. This means that transactions of large cardinality may be required to achieve a socially optimal resource allocation.

Property 4.6 (Egalitarian transaction split). *Within an egalitarian agent society, where agents express their preferences by means of additive utility functions, it is not always possible to split social bilateral transactions $\delta_i^j\langle a, b\rangle$ between two agents $i, j \in \mathcal{P}$ into a sequence of social bilateral transactions $\delta_i^j\langle a', b'\rangle$ of lesser cardinality ($a > a'$ and/or $b > b'$).*

Proof. Let us consider a counter example based on a population of two agents $\mathcal{P} = \{0, 1\}$ who are negotiating three available resources $\mathcal{R} = \{r_1, r_2, r_3\}$. Their preferences are expressed by additive utility functions described in Table 4.7. The initial resource allocation is $A = [\{r_1, r_2\}\{r_3\}]$: Agent 0 owns two resources, $\mathcal{R}_0 = \{r_1, r_2\}$, whereas agent 1 only owns a single resource, $\mathcal{R}_1 = \{r_3\}$.

Let us consider the transaction $\delta_0^1\langle 2, 1\rangle = (\{r_1, r_2\}, \{r_3\})$, during which agents 0 and 1 respectively offer $\rho_0^\delta = \{r_1, r_2\}$ and $\rho_1^\delta = \{r_3\}$. Such a transaction is social since agent 0, who

Table 4.7: Egalitarian transaction split - Example of agent preferences

Population $\mathcal{P}$	Resource Set $\mathcal{R}$		
	r_1	r_2	r_3
0	7	4	5
1	5	2	1

is the poorest agent after the transaction δ_0^1 with $u_0(r_3) = 5$, is richer than agent 1, who was initially the poorest with $u_3(r_3) = 1$. More formally:

$$\begin{aligned} \min\left(u_0(\{r_1, r_2\}), u_1(\{r_3\})\right) &< \min\left(u_0(\{r_3\}), u_1(\{r_1, r_2\})\right) \\ u_1(\{r_3\}) &< u_0(\{r_3\}) \end{aligned}$$

Such a transaction leads to an increase of the egalitarian welfare value from 1 to 6 afterwards. Only two decomposition patterns are possible: A swap combined with a gift, or three gifts. But, no $\langle 1, 0\rangle$ transaction is social. Indeed, an agent who gives one of its resources to its partner becomes poorer than the poorest agent before the transaction. Since all possible sequence of transactions of lesser cardinality contains at least one gift, the transaction $\delta_0^1\langle 2, 1\rangle = (\{r_1, r_2\}, \{r_3\})$ cannot be split into a sequence of acceptable transactions.

Hence, an egalitarian bilateral transaction cannot always be split into a sequence of egalitarian bilateral transactions of lesser cardinality. Thus, transactions of large cardinality may be required to achieve socially optimal solutions. □

4.2.3 Evaluation of egalitarian negotiations

The different facets of egalitarian negotiations are successively evaluated in this section. First, the impact of the transaction cardinality on negotiation processes is studied. Then, the fairness of resource allocations achieved using distributed negotiations are compared. The optimal egalitarian value is estimated using a centralized technique. This estimation is then used as reference. The price of the social graph can then be discussed. Next, the impact of the social graph mean connectivity on negotiation processes is presented. Behaviors are compared according to several metrics, in order to identify characteristics allowing the achievement of fair allocations. Finally, issues related to the scalability of egalitarian negotiation processes are discussed.

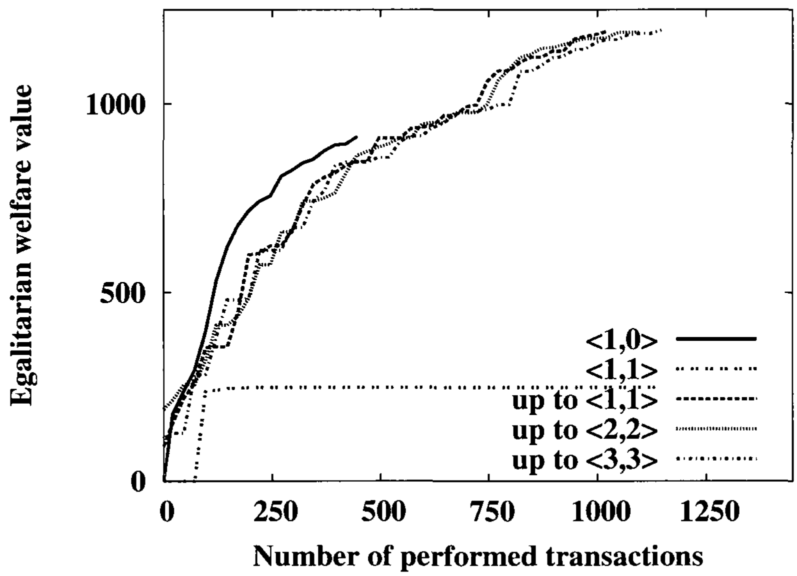
Influence of the transaction cardinality

As defined in Definition 2.6, bilateral transactions $\delta_i^j\langle a,b\rangle$ between two agents $i,j \in \mathcal{P}$ are specified thanks to cardinality parameters (the number of resources that participants i,j can offer) a and b. The size of agents' offers influences the efficiency of negotiation processes. Figure 4.5 shows the influence of the transaction cardinality on the evolution of the egalitarian welfare value during the negotiation processes. Experiments are here based on a population of 50 agents who negotiate 250 resources by means of social transactions. As defined previously, the negotiation policy "up to $\langle 2,2\rangle$" corresponds to $\mathcal{T} = \{\langle a,b\rangle | a \leq 2,\ b \leq 2\}$.

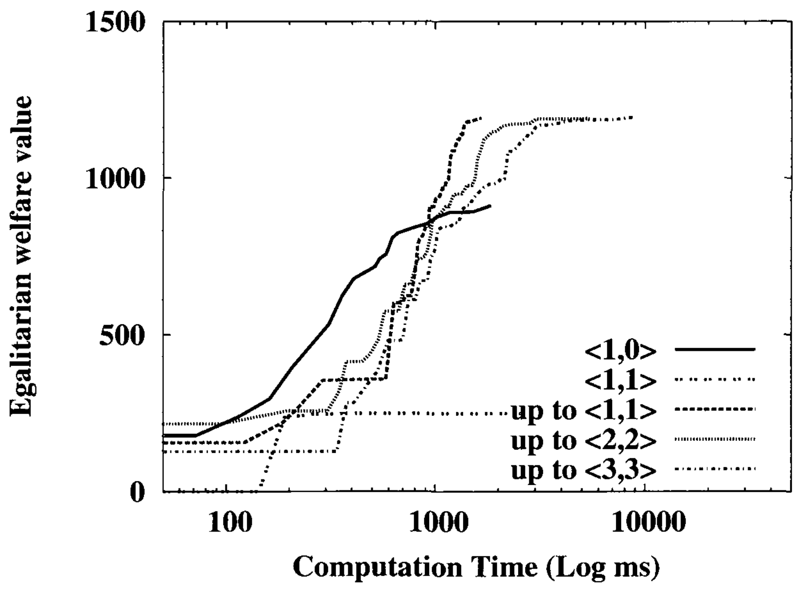
On both figures, several floors can be observed during the evolution of the egalitarian welfare value. These floors characterize specific negotiation periods during which the poorest agent of the population is not involved. As described in the previous section, even if the resources can circulate, no improvement of the egalitarian welfare value might occur.

Figure 4.5a focuses on the number of performed transactions required to end egalitarian negotiation processes, while Figure 4.5b focuses on the computation time. Negotiation processes based on social $\langle 1,0\rangle$ transactions end after only 450 social gifts. Indeed, they lead to shorter transaction sequences, which are less time consuming. However, such negotiation processes end with social values far from the ones achieved by larger bilateral transactions. Nevertheless, such processes might be used when the negotiation speed is the most important objective in spite of a solution of worse quality.

Negotiation processes based on social $\langle 1,1\rangle$ transactions require a large number of performed transactions, which barely improves the social welfare value. Negotiation processes based on $\langle 1,1\rangle$ transactions end on socially very weak allocations. Since such processes are time consuming and inefficient, the use of $\langle 1,1\rangle$ transactions should be avoided.



(a)



(b)

Figure 4.5: Influence of the transaction cardinality according to the number of performed transactions in 4.5a and to the computation time in 4.5b.

The larger is the cardinality of allowed transactions, the more time consuming becomes the negotiation process. The length of transaction sequences is also higher than in the case of $\langle 1,0\rangle$ transactions, but remains very close. Similarly, the egalitarian welfare achieved at the end of egalitarian negotiations is almost identical. The use of a larger set of allowed transactions than $\mathcal{T} = \{\langle 1,0\rangle, \langle 1,1\rangle\}$ is useless since it leads to additional costs in terms of computation time without significant improvement of the solution quality.

Price of social graphs

The efficiency of egalitarian negotiation processes is evaluated thanks to a comparison between the estimation of the optimal egalitarian welfare value, which is provided by centralized methods described in Section 4.2.2, and the welfare value obtained by egalitarian negotiations.

Negotiations are based here on a population of 50 agents where 250 resources are available. All agents interact according to frivolous and flexible behaviors. Different sets of allowed transactions are considered, from $\mathcal{T} = \{\langle 1,0\rangle\}$ to $\mathcal{T} = \{\langle a,b\rangle | a \leq 2,\ b \leq 2\}$, which corresponds to the negotiation policy denoted by "up to $\langle 2,2\rangle$". Since all utility values are positive, no gift can be rational, and then the rational policy allowing both gifts and swaps is equivalent to swaps.

Table 4.8 shows the impact of the social graph topology on the egalitarian negotiation efficiency and Table 4.9 describes the standard deviation among provided egalitarian welfare values.

Table 4.8: Egalitarian negotiation efficiency (%) according to the class of social graphs

Social graph kind	Rational		Social			
	$\langle 1,1\rangle$	up to $\langle 2,2\rangle$	$\langle 1,0\rangle$	$\langle 1,1\rangle$	up to $\langle 1,1\rangle$	up to $\langle 2,2\rangle$
Full	19.3	20.8	78.5	24.1	99.9	99.9
Grid	13.9	14.6	66.2	23.6	80.2	80.6
Erdős-Rényi	17.4	20.2	77.3	23.8	96.1	96.6
Small world	13.1	13.9	63.8	23.4	78.1	78.2

Table 4.8 shows that, generally, negotiations among rational agents achieve unfair

Table 4.9: Standard deviation of the egalitarian efficiency (%) according to the class of social graphs

Social graph kind	Rational		Social			
	$\langle 1,1\rangle$	up to $\langle 2,2\rangle$	$\langle 1,0\rangle$	$\langle 1,1\rangle$	up to $\langle 1,1\rangle$	up to $\langle 2,2\rangle$
Full	62.9	73.9	1.8	28.7	0.3	0.3
Grid	71.3	80.2	4.1	29.6	1.8	1.7
Erdős-Rényi	71.9	76.8	2.2	27.3	6.8	6.5
Small world	73.0	77.5	10.4	27.8	9.4	10.5

allocations. Indeed, independently of the allowed transactions, independently of the social graph topology, rational negotiation processes end quite far from the optimal welfare value. Only 20% of the optimal welfare value is achieved in the best cases. According to Table 4.9, the standard deviation of negotiations among rational agents is very important. In the case of rational negotiations based on small-worlds, egalitarian welfare values that can be achieved may vary by 73%. Initial resource allocations and social graph topologies are the most important factor when rational egalitarian negotiations are considered. Thus, the rationality criterion is definitively not well-adapted to solve egalitarian problems efficiently. It restricts the set of possible transactions too much and throws negotiation processes into local optima. Generosity is hence an essential feature in order to achieve fair allocations.

Even using on complete graphs, no social negotiation policy can guarantee the achievement of egalitarian optima. Whereas social $\langle 1,0\rangle$ transactions are well adapted to the solution of utilitarian problems, they do not suit to the case of egalitarian problems. Only 78.5% of the optimum can be achieved in the best cases. Indeed, after a finite number of transactions, agents can not give any additional resource without becoming poorer than their partners. The exclusive use of gifts is then not sufficient to lead negotiations to socially efficient resource allocations. Negotiations based on social $\langle 1,1\rangle$ transactions lead to severely sub-optimal resource allocations with an efficiency of 24.1% on complete social graphs in the best case. Such a weak efficiency is mainly due to the inherent constraints of swap transactions. Since the resource distribution cannot be modified, a poor agent who has only few resources initially, penalizes a lot the egalitarian negotiation process. When both gifts and swaps are allowed, i.e., when $\mathcal{T} = \{\langle a,b\rangle | a \leq 1,\ b \leq 1\}$, the negotiation efficiency is really close to the optimum. Larger bilateral transactions improve only a little the fairness among agents, but are much more expensive to determine.

Social graphs of weaker mean connectivity like grids lead negotiation processes to socially weaker allocations whereas, when small-worlds are considered, the resource traffic is restricted. The standard deviation is higher when small-worlds are considered. In such cases, according to the generation rules, many agents have only one neighbor, which may penalize egalitarian negotiations. Indeed, if such agents cannot identify an acceptable transaction with their lone neighbor, some resources may be trapped in the bundle of such agents.

Theorem 4.7. *Within an egalitarian society where agents express their preferences by means of additive utility functions, bilateral transactions cannot guarantee the achievement of an egalitarian optimum, independently of the social graph considered.*

Proof. Let us consider a counter-example, based on a population of three agents $\mathcal{P} = \{1, 2, 3\}$ and a set of three available resources $\mathcal{R} = \{r_1, r_2, r_3\}$. The agent preferences are described in Table 4.10.

Table 4.10: Bilateral insufficiency in egalitarian negotiations - Example of agent preferences

Population $\mathcal{P}$	Resource Set $\mathcal{R}$		
	r_1	r_2	r_3
0	2	1	5
1	5	2	1
2	1	5	2

The complete social graph is described in Figure 4.6 with the initial resource allocation $A = [\{r_1\}\{r_2\}\{r_3\}]$. This figure also described the lone egalitarian transaction that would be acceptable.

No sequence of acceptable bilateral transactions can lead to an optimal resource allocation. Indeed, six $\langle 1, 0\rangle$ transactions are possible but none can be performed since they are not social. Indeed, if an agent gives a resource, its bundle becomes empty, and the associated egalitarian welfare value becomes null. Three $\langle 1, 1\rangle$ transactions are possible, but each

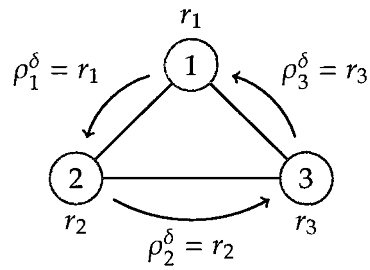


Figure 4.6: Deadlocks in egalitarian negotiations

time the welfare value decreases, meaning that the transaction is not acceptable. Hence, even if the multi-agent system is completely connected, the optimal solution cannot be achieved using only bilateral transactions. Only a multilateral transaction corresponding to three simultaneous gifts is acceptable as described in Figure 4.6.

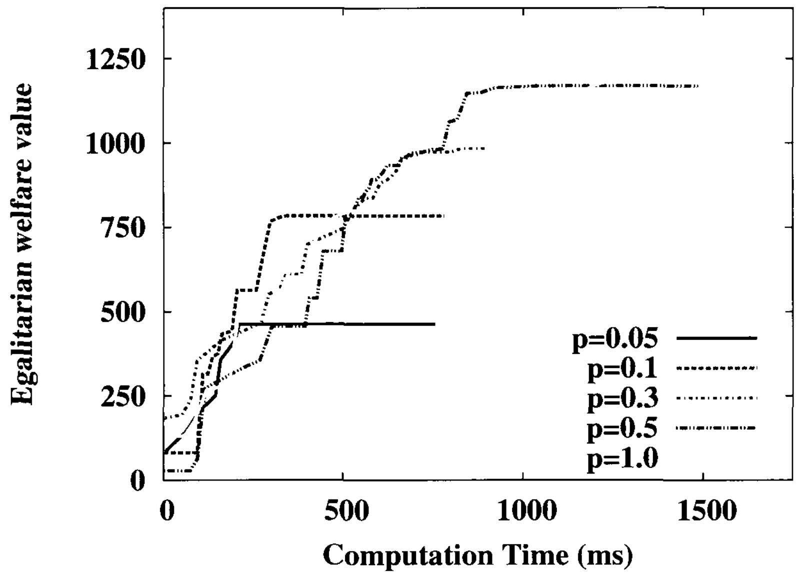
Since bilateral transactions are not sufficient when negotiations are based on a complete social graph, they are also not sufficient when the social graph is restricted. Indeed, in such cases, less transactions are possible, and not acceptable transactions on a complete social graph are still not acceptable on a restricted social graph. □

Influence of the social graph connectivity

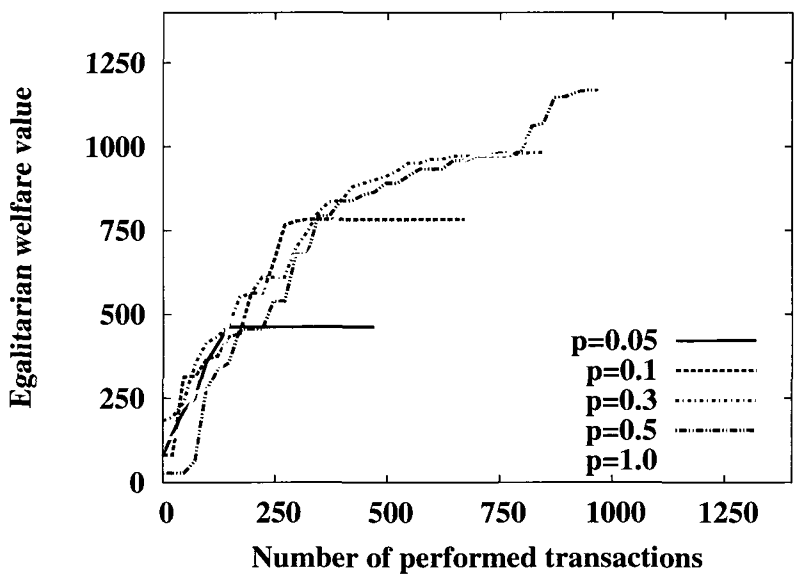
The social graph topology influences a lot the resource circulation as well as the efficiency of negotiation processes. The larger are agents' neighborhoods, the denser are social graphs, and consequently resources can circulate easily. Thanks to the model of generation of Erdős-Rényi graphs $G(n,p)$, which is described in Section 3.3.2, the probability of link generation between two agents can be modified. High probabilities correspond to dense social graphs.

Since rational negotiations can barely identify acceptable transactions, only social negotiations are represented here. Figures 4.7 show the impact of the connectivity. Erdős-Rényi graphs are generated with an increasing probability p from $p = 0.05$ to $p = 1.0$.

Similarly to utilitarian negotiations, Figures 4.7 show that a high probability, which corresponds to a dense social graph, leads to longer sequences of transactions during ne-



(a)



(b)

Figure 4.7: Influence of the mean connectivity on egalitarian negotiations in terms of the computation time in 4.7a and of number of performed transactions in 4.7b

gotiation processes, which achieve moreover a higher welfare value. Larger neighborhoods facilitate the resource circulation by offering larger numbers of possible transactions to all agents. The impact of the connectivity is important only if the probability p of link generation is very low. The impact of the connectivity is not linear, it becomes really significant below $p \leq 0.3$.

Influence of agent's behaviors

Behaviors define how agents interact. The different behaviors defined in Section 2.6, can be compared using metrics presented in Section 2.7.1. In order to evaluate agents' behaviors, any factor that may affect the comparison should be avoided, as for example the social graph topology. For this reason, our experiments with negotiation processes are based on complete social graphs, with a population of 50 agents and 250 resources. Agents only negotiate using social transactions: $\mathcal{T} = \{\langle 1,0\rangle, \langle 1,1\rangle\}$. Both gifts and swaps are allowed since they correspond to the most efficient negotiation policy.

Negotiations between stubborn agents lead to short transaction sequences, where only few resources are exchanged. When agents are also frivolous, the number of exchanged resources is at most 500 whereas negotiations between flexible agents, rooted as well as frivolous, lead to a minimum of 1500 traded resources. Negotiations between stubborn agents are generally fast and end after 1.5 seconds, but lead to weak egalitarian values. When agents are stubborn, the number of performed transactions corresponds to the number of traded resources.

As observed in utilitarian negotiations, egalitarian negotiations between flexible agents, for both negotiation mechanisms (i.e., full or not), lead to close results. The welfare values achieved at the end of negotiation processes are almost identical, but full negotiations are more time consuming. If agents are also frivolous, then the number of attempted transactions is higher but such behaviors also improve the resulting welfare values. Rooted behaviors require a larger number of speech turns compared to frivolous behaviors. The most efficient behavior is the "flexible and frivolous" one, since negotiation processes

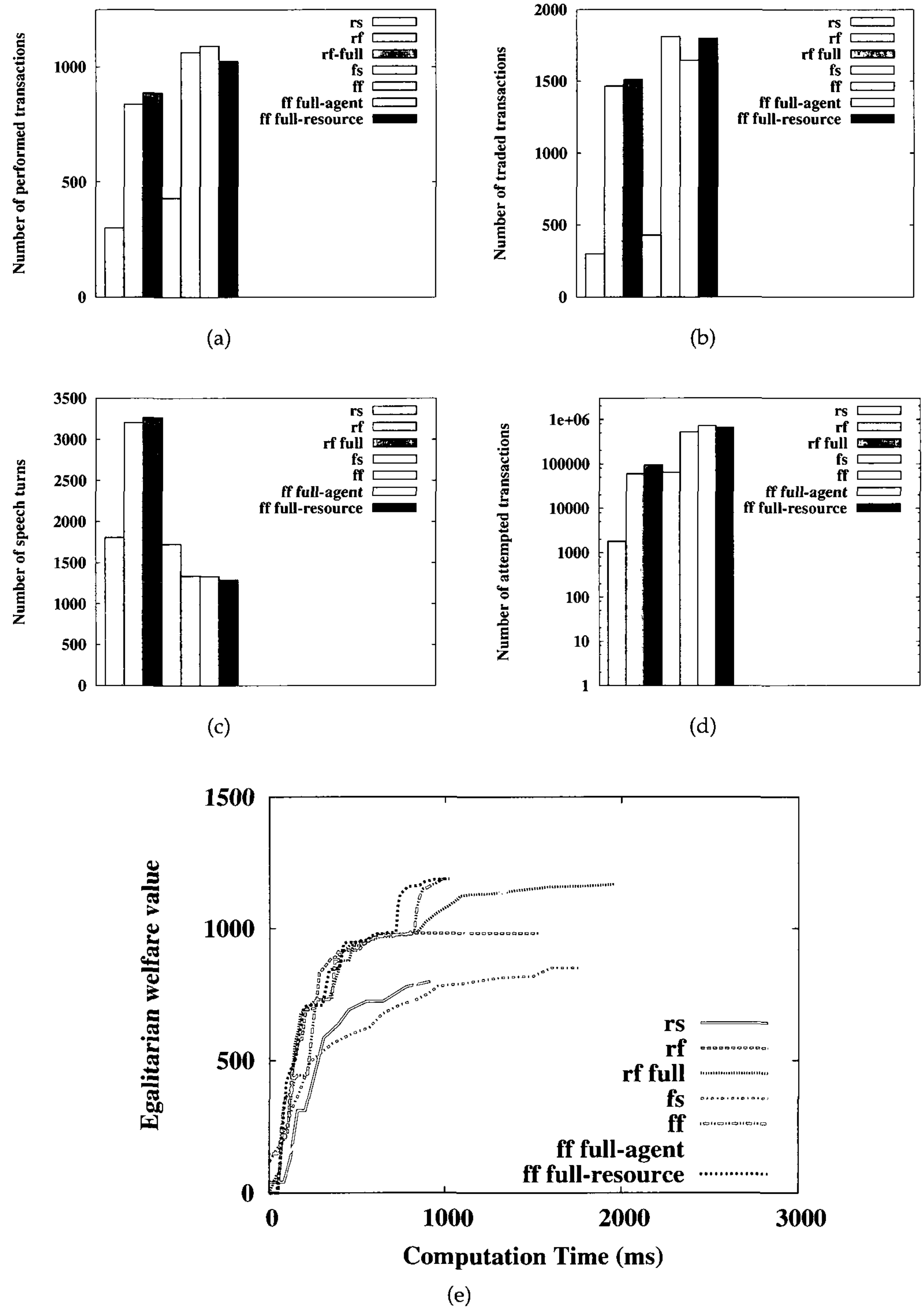


Figure 4.8: Agent behavior impact on egalitarian negotiations in terms of performed transactions in 4.8a, of transacted resources in 4.8b, of speech turns in 4.8c, of attempted transactions in 4.8d and finally in terms of computation time in 4.8e.

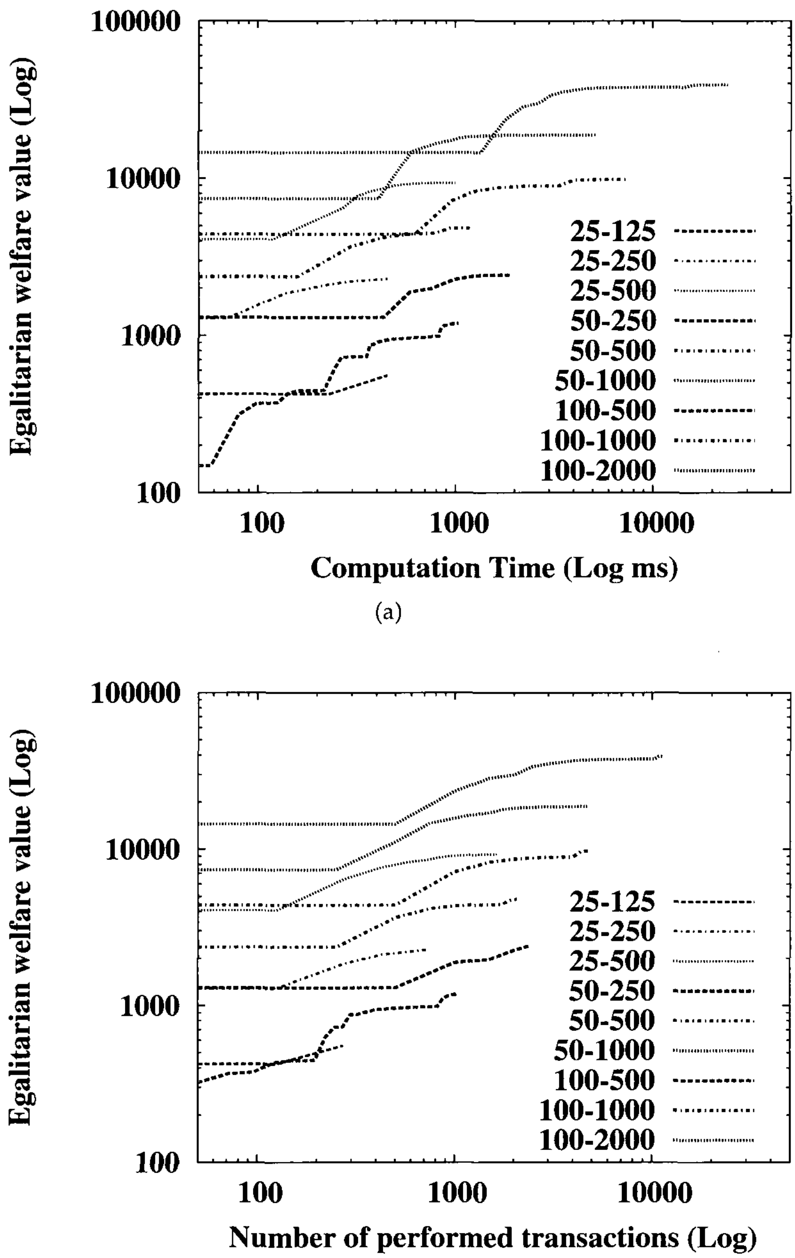
achieve the fairest allocations. The flexibility is the most important characteristic of agent's behaviors when egalitarian negotiations are considered. In order to reduce the inequalities within the agent society, agents must accept to offer any resource of their bundle.

Negotiation scalability

The scalability of egalitarian negotiations is evaluated in terms of performed transactions and in terms of computation time. Negotiation processes are based here on complete social graphs when agents are frivolous and flexible. Such a simulation setting corresponds to one of the most expensive configurations.

Figure 4.9a represents the evolution of the egalitarian welfare value according to the computation time required to end egalitarian negotiations, on several instance sizes. Figure 4.9b shows the evolution of the egalitarian welfare value according to the number of performed transactions. The objective value increases faster at the beginning of the negotiation processes, which then spend a lot of time to lightly improves the solution. Most of the transactions performed during negotiation processes are performed at the beginning of negotiations.

Thus, even if large bilateral transactions improve the solutions, the time required before the end of negotiation processes increases significantly for a small improvement of the solution quality. Negotiations based on social transactions such that $\mathcal{T} = \{\langle 1,0\rangle, \langle 1,1\rangle\}$ seem to be the best compromise between efficiency and scalability. The mean computation times, which are required to end negotiation processes, are presented in Table 4.11, while Table 4.12 details the length of the transaction sequences which are performed. Egalitarian negotiations are more time consuming than utilitarian ones. For instance, a negotiation process between 100 agents who own 20 resources each requires 25 seconds to converge. It is still reasonable compared to the solving time required by a linear program solver as the one described in Section 4.2.1.



(a)

(b)

Figure 4.9: Scalability evaluation of egalitarian negotiations in terms of computation time in 4.9a and of performed transactions in 4.9b.

Table 4.11: Egalitarian negotiation scalability - Computation time

Population size n	Mean number of resources per agent $\frac{m}{n}$		
	5	10	20
25	400 ms	650 ms	1.6 s
50	750 ms	1.5 s	5 s
100	2 s	5.8 s	25 s
500	110 s	342 s	1500 s

Table 4.12: Egalitarian negotiation scalability - Transaction sequence length

Population size n	Mean number of resources per agent $\frac{m}{n}$		
	5	10	20
25	450	800	1900
50	1000	2150	4300
100	2500	5000	11000
500	18500	38000	78000

4.2.4 Conclusion

The determination of optimal allocations is a $\mathcal{NP}$-hard problem even when agent preferences are expressed by additive functions. Either heuristics or linear programs can be used to estimate this optimal welfare value. In distributed agent negotiations, the combined use of social $\langle 1,0\rangle$ and social $\langle 1,1\rangle$ transactions corresponds to the most efficient negotiation policy among frivolous and flexible agents.

Best egalitarian negotiation policy

Transaction:	$\langle 1,0\rangle + \langle 1,1\rangle$ (i.e., gifts and swaps)
Criterion:	social
Test on δ_i^j:	$\min\limits_{i,j\in\mathcal{P}}\big(u_i(\mathcal{R}_i), u_j(\mathcal{R}_j)\big) < \min\limits_{i,j\in\mathcal{P}}\big(u_i(\mathcal{R}_i) + u_i(\rho_j^\delta) - u_i(\rho_i^\delta), u_j(\mathcal{R}_j) + u_j(\rho_i^\delta) - u_j(\rho_j^\delta)\big)$
Behavior:	frivolous and flexible

Egalitarian negotiation processes based on bilateral transactions cannot guarantee the achievement of optimal resource allocations. Generosity is an essential feature required in order to achieve fair resource allocations. The rational acceptability criterion which is commonly used in the literature is in fact very inefficient. The social graph topology also greatly influences the negotiation efficiency. Any characteristic disturbing the resource traffic leads to a decrease of the negotiation efficiency, such as weak mean connectivity and the existence of bottlenecks (e.g., agents with only neighbors). Negotiation processes should be based on social transactions of $\mathcal{T} = \{\langle 1,0\rangle, \langle 1,1\rangle\}$. Indeed, the exclusive use of gifts or swaps is not efficient and the use of transactions of larger cardinality leads to important additional costs for a slight improvement of the welfare value. Even on large instances, egalitarian negotiations remain scalable compared to the solution of optimization models.

4.3 Nash bilateral negotiations

The Nash welfare is an interesting notion that can be viewed as a compromise between the utilitarian and egalitarian welfare notions. This notion favors a decrease of the inequalities within a population as well as an increase of the global welfare of the society. The Nash product is nevertheless barely used due to its computational complexity. Dedicated to the Nash product, this section is organized as follows. Issues related to the estimation of the optimal welfare value are first discussed. Centralized methods face specific difficulties which are described. The expression of the social acceptability criterion is detailed when the Nash welfare is considered. Properties of Nash negotiations are then described. Several aspects of Nash negotiation processes are finally evaluated: the transaction cardinality, the negotiation efficiency and their scalability, the impact of the social graph connectivity and the agents' interactions. We finally discuss the implementation of characteristics in order to efficiently negotiate when using the Nash welfare.

4.3.1 Centralized techniques

The identification of the optimal Nash welfare value is a difficult problem. Indeed, according to Definition 1.6, the Nash welfare notion is not a linear objective.

Theorem 4.8 (Nash welfare optimum complexity (Ramezani and Endriss, 2009)). *The determination of the optimum Nash welfare value is a $\mathcal{NP}$-hard problem.*

Nash resource allocation problems can be formulated by means of a mathematical model using variables x_{ir} describing the ownership of a resource $r \in \mathcal{R}$ by an agent $i \in \mathcal{P}$:

$$x_{ir} = \begin{cases} 1 & \text{if agent } i \text{ owns resource } r \\ 0 & \text{otherwise.} \end{cases} \qquad r \in \mathcal{R}, \quad i \in \mathcal{P}.$$

Then, the Nash resource allocation problem can be formulated as follows:

$$sw_n^\star = \begin{cases} \max \prod_{i \in \mathcal{P}} \sum_{r \in \mathcal{R}} u_i(r) x_{ir} \\ \text{s.t: } \sum_{i \in \mathcal{P}} x_{ir} = 1 \qquad r \in \mathcal{R} \\ \qquad x_{ir} \in \{0,1\} \qquad r \in \mathcal{R}, \quad i \in \mathcal{P}. \end{cases}$$

The objective is the maximization of the Nash product, i.e. the maximization of the product of all agents' utilities. A consistency constraint ensures that each resource is allocated to only one agent. Since resources are discrete and not sharable, Boolean variables are used, but the model can be easily adapted to other resource natures. For instance, continuous resources are represented by real variables, corresponding to quantities of the considered resource: $x_{ir} \in [0,1], \quad r \in \mathcal{R}, i \in \mathcal{P}$.

Such a model cannot be handled in a classical way since the objective function is neither linear, nor convex, nor concave. Theoretically, an estimation could be made thanks to a combination of different optimization techniques. First, a Lagrangian relaxation could be used (Fisher, 2004). This method can solve a system of non linear equations if the objective function is convex. However, the Nash product is not a convex function. A multi-start algorithm has to be combined with this relaxation. Starting from multiple initial solutions may help to avoid local optima when non convex functions are considered (Hickernell and Yuan, 1997). Moreover, since resources are not divisible, an integer solution has still to be found. Indeed, the relaxation changes the Boolean variables of the discrete value set $\{0,1\}$ into reel variables of the continuous value set $[0,1]$. In order to obtain an integer solution, a branch-and-bound algorithm is used. Such an algorithm can be guided by the values provided by the relaxed solution in order to improve the integer solution and to reduce the computation time.

Such a method cannot guarantee the optimality of the resulting solution. Moreover, this method is not really scalable, as a consequence of the non-linearity of the objective and of the exponential solution space. For instance, solving a simple system with 25 agents is equivalent to optimize a sum of products with 25 terms each.

Since such a method is not scalable, we developed some heuristics in order to estimate the optimal Nash welfare value. Since the Nash welfare notion is a compromise between fairness and global efficiency, an estimation of the optimal welfare value can be determined according to different methods based on these characteristics. A first possibility is to consider the fairness of the resource distribution, as described in Algorithm 4.4. This algorithm sequentially allocates the best remaining resources to each agent.

Algorithm 4.4: Estimation of the optimal Nash welfare value - 1

Input: Agent population $\mathcal{P}$, Resource set $\mathcal{R}$

Output: $sw_n^\star$ the estimation of the optimal Nash welfare value

$i \leftarrow 0$;

$Shuffle(\mathcal{P})$; // Mix the population $\mathcal{P}$

while $\mathcal{R} \neq \emptyset$ **do**

  $r \leftarrow \arg\min_{r' \in \mathcal{R}} u_i(r')$; // Determination of the best remaining resource

  Add r to $A[i]$; // Allocation of resource r to agent i

  $\mathcal{R} \leftarrow \mathcal{R} \setminus \{r\}$;

  $i \leftarrow (i+1)\%n$;

end

return $sw_n(A)$;

Another way to estimate the optimal Nash welfare value is to consider, focusing on the global efficiency, as described in Algorithm 4.5. The first step of this algorithm allocates each resource to the agent who values it the most. However, some agents can be neglected and do not get any resource. Such a situation corresponds to a null Nash welfare value. In order to avoid this phenomenon, the algorithm must perform a second step ensuring that all agents own at least one resource and, if the need arises, it tries to pick from an agent who has at least two resources the resource maximizing the local welfare value.

Algorithm 4.5: Estimation of the optimal Nash welfare value - 2

Input: Agent population $\mathcal{P}$, Resource set $\mathcal{R}$

Output: $sw_n^\star$ the estimation of the optimal Nash welfare value

```
;// First step
forall the r ∈ R do
    i ← argmax_{i∈P} u_i(r) ;            // Determination of who values r the most
    Add r to A[i] ;                      // Allocation of resource r to agent i
end
;// Second step
for i ∈ P s.t. m_i = 0 do
    val ← 0 ;
    for j ∈ P s.t. m_i > 1 do            // Determine where to pick up a resource
        r' ← argmax_{r∈R_j} u_i(r')u_j(R_j \ {r'}) ;
        if val < u_i(r')u_j(R_j \ {r}) then
            val ← u_i(r)u_j(R_j \ {r}) ;
            r ← r' ;
            k ← j ;
        end
        Add r to A[i] ;                  // Reallocation of resource r to agent i
    end
end
return sw_u(A) ;
```

In spite of their scalability, these two heuristics have the drawback to affect the quality of provided solutions, and thus are not really reliable. Similarly to the heuristics estimating the optimal egalitarian value, these heuristics are very sensitive to the order in which agents are considered. Depending on this order, the Nash welfare value provided by such heuristics may vary a lot.

4.3.2 Nash negotiation properties

The expression of the social acceptability criterion can be specified when the Nash welfare is considered and the expression of the agents' decision making can be specified. Let us note $A \in \mathcal{A}$ the resource allocation before the bilateral transaction $\delta_i^j\langle a, b\rangle$ that evolves into a new allocation $A' \in \mathcal{A}(A \neq A')$. Such a transaction involves two agents $i, j \in \mathcal{P}$, who respectively propose the offers ρ_i^δ and ρ_j^δ. The resource bundle of any agent $k \in \mathcal{P}$ is denoted by $\mathcal{R}_k$ before the transaction and $\mathcal{R}'_k$ afterward. A social bilateral transaction must satisfy the following condition:

$$
\begin{aligned}
sw_n(A) &< sw_n(A') \\
\prod_{k \in \mathcal{P}} u_k(\mathcal{R}_k) &< \prod_{k \in \mathcal{P}} u_k(\mathcal{R}'_k) \\
u_i(\mathcal{R}_i)u_j(\mathcal{R}_j) \prod_{k \in \mathcal{P}\setminus\{i,j\}} u_k(\mathcal{R}_k) &< u_i(\mathcal{R}'_i)u_j(\mathcal{R}'_j) \prod_{k \in \mathcal{P}\setminus\{i,j\}} u_k(\mathcal{R}'_k) \\
u_i(\mathcal{R}_i)u_j(\mathcal{R}_j) &< u_i(\mathcal{R}'_i)u_j(\mathcal{R}'_j) \\
u_i(\mathcal{R}_i)u_j(\mathcal{R}_j) &< \left(u_i(\mathcal{R}_i) - u_i(\rho_i^\delta) + u_i(\rho_j^\delta)\right)\left(u_j(\mathcal{R}_j) + u_j(\rho_i^\delta) - u_j(\rho_j^\delta)\right)
\end{aligned}
$$

Similarly to the egalitarian interpretation of the social acceptability criterion, the Nash criterion must be based on traded resources as well as on initial resource bundles of both agents. Then, the acceptability test, which is represented by the instruction TEST in all behaviors described in Section 2.6, can thus be replaced by the following expression:

$$
\text{TEST} := \left[u_i(\mathcal{R}_i)u_j(\mathcal{R}_j) < \left(u_i(\mathcal{R}_i) - u_i(\rho_i^\delta) + u_i(\rho_j^\delta)\right)\left(u_j(\mathcal{R}_j) + u_j(\rho_i^\delta) - u_j(\rho_j^\delta)\right)\right]
$$

When the Nash welfare is considered, bilateral transactions have some important properties affecting the negotiation efficiency. An acceptable bilateral transaction $\delta_i^j\langle a, b\rangle$ may not be split into a sequence of acceptable bilateral transactions of lesser cardinality. This means that transactions of large cardinality may be required to achieve socially optimal resource allocations.

Property 4.9 (Nash transaction split). *In a society where agents express their preferences by means of additive utility functions and where the maximization of the Nash welfare is the objective, social bilateral transactions $\delta_i^j\langle a, b\rangle$ between two agents $i, j \in \mathcal{P}$ cannot always be split into a sequence of social bilateral transactions $\delta_i^j\langle a', b'\rangle$ of lesser cardinality ($a > a'$ and/or $b > b'$).*

Proof. Let us consider a counter-example based on a population of two agents, $\mathcal{P} = \{0, 1\}$ who are negotiating the two available resources $\mathcal{R} = \{r_1, r_2\}$. Their preferences are expressed by means of additive utility functions described in Table 4.13. The initial resource allocation is $A = [\{r_1\}\{r_2\}]$: Each agent owns one resource.

Table 4.13: Nash transaction split - Example of agent preferences

Population $\mathcal{P}$	Resource Set $\mathcal{R}$	
	r_1	r_2
0	7	3
1	4	1

Let us consider the transaction $\delta_0^1\langle 1, 1\rangle = (\{r_1\}, \{r_2\})$, during which agents 0 and 1 respectively propose $\rho_0^\delta = \{r_1\}$ and $\rho_1^\delta = \{r_2\}$. This transaction corresponds a social swap since:

$$u_0(\{r_1\})u_1(\{r_2\}) < u_0(\{r_2\})u_1(\{r_1\})$$

This swap increases the Nash welfare value from $sw_n(A) = 7$ of the initial resource allocation to $sw_n(A') = 12$ afterwards. This transaction can only be split into a sequence of two gifts. However, no gift is acceptable. Indeed, any agent who gives its lone resource stays with an empty bundle, which is always associated with a welfare value of 0. In such a case, the Nash welfare value of the whole society is null.

Hence, when the Nash welfare is considered, a social bilateral transaction cannot always be split into a sequence of acceptable bilateral transactions of lesser cardinality. Transactions of large cardinality may thus be required to achieve a socially optimal solution. □

4.3.3 Evaluation of Nash negotiations

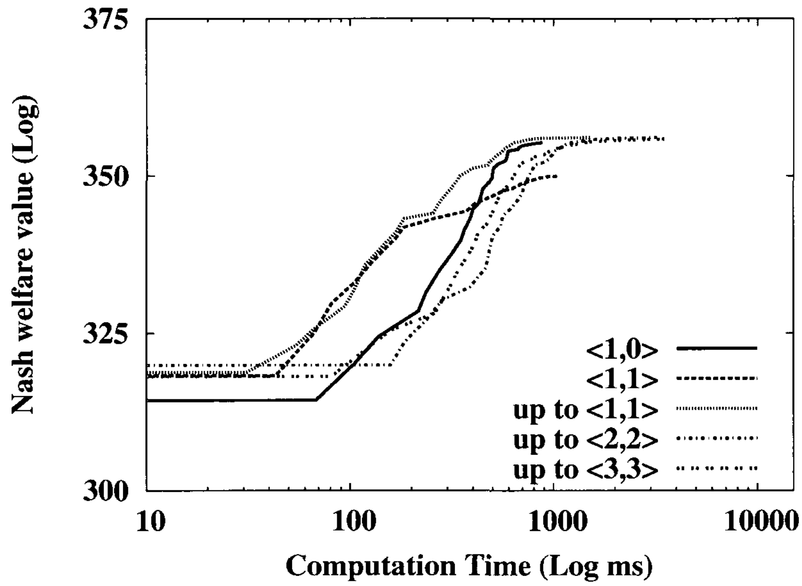
This section is dedicated to the evaluation of the Nash negotiations. In order to determine if large bilateral transactions are required to achieve socially efficient allocations, the influence of the size of agent offers is studied first. Then, the Nash welfare values provided by heuristics are compared to the ones provided by negotiation processes. The impact of the social graph topology is then presented as well as suitable behavior characteristics. Finally, scalability issues are addressed.

Let us first note that it is not convenient to directly work with the Nash welfare values. According to Definition 1.6, the Nash welfare is the product of the individual welfares of all agents. Such values become quickly so huge that it is difficult to deal with them. For instance, a population of 50 agents where each agent estimates all of 250 resources with a positive utility value in the range [1..250] leads to Nash welfare value scale around 10^{60}. Thus, $Log(sw_n(A))$ is used instead of $sw_n(A)$ in the different comparisons, in order to avoid accuracy problems. However, a side effect of the use of Logarithms is the "reduction" of the gap between two welfare values. Indeed, a gap of 0.1% between two Logarithms represents an exponentially larger gap between the values themselves.

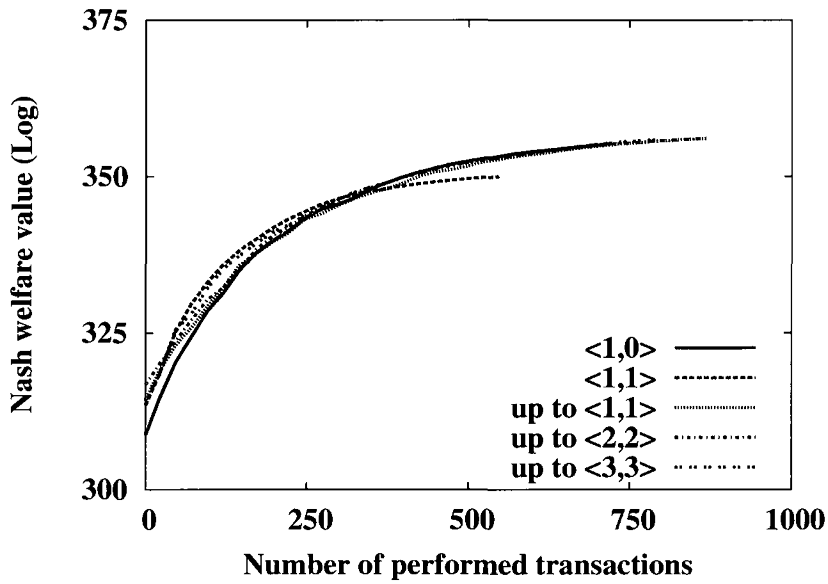
Influence of the transaction cardinality

Bilateral transaction $\delta_i^j\langle a, b\rangle$ between two agents $i, j \in \mathcal{P}$ is characterized by the cardinality of the parameters a and b, describing the size of agent offers. These experiments are based on a population of 50 agents who negotiate 250 resources. Several negotiation policies are used and described using the cardinality parameters. The negotiation policy denoted by "up to $\langle 2, 2\rangle$" means that agents can offer up to two resources during the same transaction. Figure 4.10 shows the evolution of the Nash welfare value during a negotiation process according to the transactions cardinality.

Figure 4.10a shows that the transactions cardinality mainly affects the elapsed time.



(a)



(b)

Figure 4.10: Influence of the transaction cardinality on Nash negotiations in terms of computation time in 4.10a and of performed transactions in 4.10b.

Negotiation processes based on $\langle 1, 0 \rangle$ transactions are less time-consuming and agents perform less transactions in such a case. The larger the allowed transactions, the more time consuming are negotiation processes. However, according to Figure 4.10b that focuses on the number of performed transactions, larger bilateral transactions do not improve the quality of achieved solutions. Negotiations relying only on $\langle 1, 1 \rangle$ transactions require less transactions but also achieve socially weaker allocations. All other negotiation processes end after sequences of transactions of close length. Large bilateral transactions do not seem to significantly improve the Nash welfare value achieved at the end of the negotiation processes. The use of transactions of large cardinality does not justify the important additional costs, and thus the size of the offers should be restricted.

Price of social graphs

When the Nash welfare is considered, the efficiency of the negotiation processes can be evaluated using a comparison with the estimation given by centralized techniques. Several centralized heuristics are described in Section 4.3.1, but only the one providing the largest results is used here. The best results are provided by Algorithm 4.5 which focuses on the global efficiency. It first allocates each resource to the agent who values it the most, and then allocates at least one resource to each agent who gets nothing. Table 4.14 shows the efficiency of negotiations depending on the class of social graphs considered, whereas Table 4.15 shows the standard deviation of the different Nash welfare values, which correspond to the topological sensitivity.

Negotiations are based here on a population of 50 agents where 250 resources are available. All agents interact according to a frivolous and flexible behaviors in every case. Different sets of allowed transactions are considered, from $\mathcal{T} = \{\langle 1, 0 \rangle\}$ to $\mathcal{T} = \{\langle a, b \rangle | a \leq 2,\ b \leq 2\}$. As described in the previous sections, since all utility values are positive, no gift can be rational, and then the rational negotiation policy allowing both gifts and swaps is restricted to swaps.

Table 4.14 shows that some welfare values achieved are greater than 100%. Since

Table 4.14: Nash negotiation efficiency (%) according to the class of social graphs

Social graph kind	Rational		Social			
	$\langle 1,1 \rangle$	up to $\langle 2,2 \rangle$	$\langle 1,0 \rangle$	$\langle 1,1 \rangle$	up to $\langle 1,1 \rangle$	up to $\langle 2,2 \rangle$
Full	99.9	100.1	101.6	100.1	101.7	101.7
Grid	97.0	97.5	99.6	98.2	99.7	99.7
Erdős-Rényi	99.6	99.8	101.4	99.9	101.6	101.6
Small world	97.2	98.0	100.2	98.9	100.4	100.4

heuristics can only give an estimation of Nash welfare values, an efficiency greater than 100% means that the corresponding negotiation processes lead to socially more interesting allocations than the ones provided by the heuristics. As observed in the case of egalitarian negotiations, it is not possible to guarantee that optimal allocations can be achieved using bilateral transactions only.

Rational negotiations generally achieve socially weaker allocations than social negotiations. Two negotiation policies, which are based respectively on $\mathcal{T} = \{\langle a,b \rangle | a \leq 1, b \leq 1\}$ and on $\mathcal{T} = \{\langle a,b \rangle | a \leq 2, b \leq 2\}$, lead to similar results. Allowing gifts and swaps during a negotiation process seems sufficient to achieve socially efficient allocations. Larger transactions do not significantly improve the Nash welfare values achieved while the negotiation cost that increases a lot.

Similarly to the egalitarian case, negotiations based on swap transactions achieve the socially weakest allocations. Since the initial resource distribution cannot be modified, negotiations end quickly on local optima. According to Table 4.15, the standard deviation related to negotiations based on $\langle 1,1 \rangle$ transactions is also higher than for other transactions.

Negotiation processes based on grids leads to the socially weakest allocations. The mean connectivity of the social graphs is an important feature deeply affecting the negotiation efficiency. Relationships among agents are too restricted to allow a suitable resource traffic, and then prevent the achievement of optimal allocations. The comparison between results achieved on Erdős-Rényi graphs and the ones achieved on small-worlds indicates that a large number of agents, leaves of the graph (who have only one neighbor), penalizes a lot the negotiation process.

Table 4.15: Standard deviation of the Nash product (%) according to the class of social graphs

Social graph kind	Rational		Social			
	$\langle 1,1\rangle$	up to $\langle 2,2\rangle$	$\langle 1,0\rangle$	$\langle 1,1\rangle$	up to $\langle 1,1\rangle$	up to $\langle 2,2\rangle$
Full	0.33	0.27	0.06	0.31	0.02	0.02
Grid	0.44	0.40	0.14	0.37	0.14	0.14
Erdős-Rényi	0.33	0.28	0.06	0.32	0.02	0.02
Small world	0.46	0.38	0.13	0.37	0.12	0.12

Theorem 4.10. *When the Nash welfare is considered, within a population of agents who express their preferences by means of additive utility functions, bilateral transactions cannot guarantee the achievement of optimal allocations, independently of the social graph considered.*

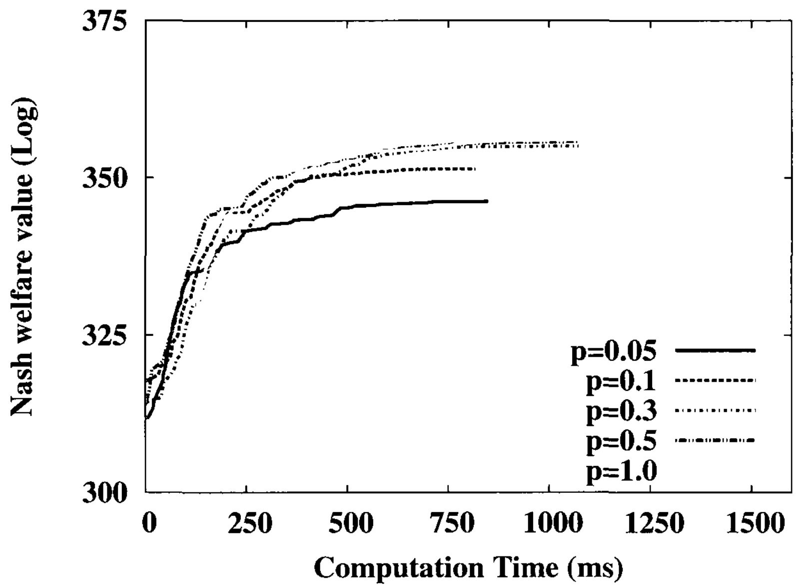
Proof. Similarly to the proof of Proposition 4.7, a counter-example can be generated with different agent preferences, where only a multilateral transactions can solve the problem. □

As for the other welfare notions, negotiations among social agents achieve more efficient allocations compared to rational negotiations usually studied in the literature. Negotiations based on $\mathcal{T} = \{\langle 1,0\rangle, \langle 1,1\rangle\}$ can be considered as the best alternative to achieve socially interesting allocations. Transactions of weaker cardinality are not sufficient whereas larger transactions do not improve the Nash welfare value while their use increases the negotiation cost. However, the exclusive use of bilateral transactions cannot guarantee the achievement of a global optimum, but leads to socially close allocations instead.

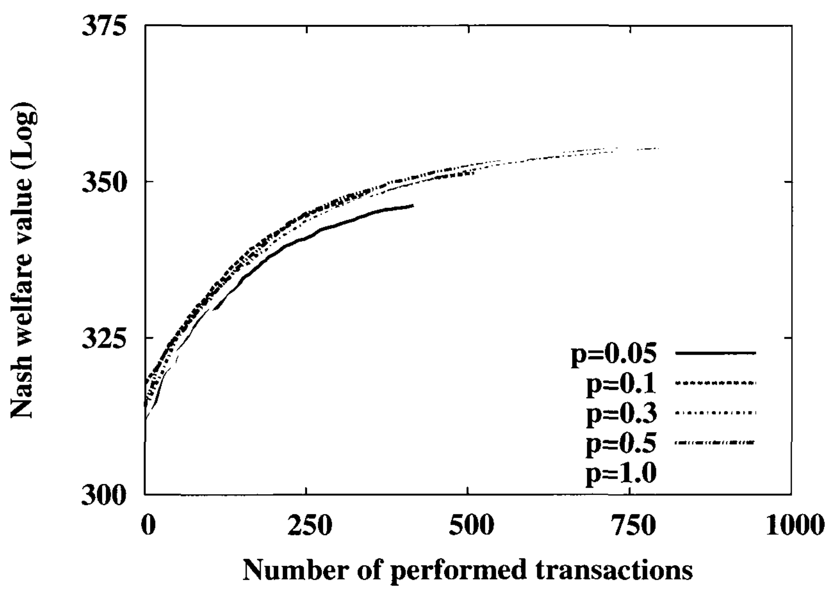
Influence of the social graph connectivity

The social graph topology affects the negotiation efficiency and may prevent the achievement of socially optimal allocations. Agent relationships are represented here by Erdős-Rényi graphs, which is generated thanks to the model $G(n, p)$. The variation of the probability influences on the number of links, and thus the mean neighborhood size.

Figure 4.11a represents the Nash welfare evolution according to the elapsed time, and Figure 4.11b corresponds to the Nash welfare evolution according to the number of



(a)



(b)

Figure 4.11: Influence of the mean connectivity on Nash negotiations according to the computation time in 4.11a and to the number of performed transactions in 4.11b.

performed transactions. They show that denser social graphs lead to longer negotiation processes (with larger number of performed transactions) and to a higher utilitarian welfare value at the end of the negotiation process. The connectivity has an important influence only if the probability p of link generation is very low. The influence of the connectivity is not linear, it becomes really significant below $p \leq 0.3$.

Influence of agent behaviors

Behavior defines how agents negotiate. The different behaviors, which are defined in Section 2.6, can be compared using metrics presented in Section 2.7.1. In order to evaluate agents' behaviors, any factor that may alter the comparison of the different agents' behaviors should be avoided, like the social graph topology. For this reason, simulations of negotiation processes are based on complete social graphs, with a population of 50 agents and 250 available resources. Agents negotiate using social transactions only: $\mathcal{T} = \{\langle 1,0\rangle, \langle 1,1\rangle\}$.

The more restrictive the agents' behaviors, the shorter are the transaction sequences. For instance, stubborn and rooted agents can only perform few transactions during negotiation processes. Since only few offers are attempted, the identification of acceptable transactions is difficult. Resources barely circulate, and then negotiation processes end on socially sub-optimal resource allocations.

When agents are still rooted but flexible, the number of attempted transactions increases as well as the number of performed transactions. A large number of speech turns is required to end Nash negotiation processes. Since agents can propose several offers during a negotiation, the identification of acceptable transactions is easier, which favors the resource traffic and the achievement of socially interesting allocations.

Frivolous agents benefit from their neighborhood during a negotiation. This behavior characteristic increases the potential number of transactions that agents can attempt during a negotiation process. If agents are also stubborn, they only make a single offer during a negotiation. Agents' neighborhoods are large since the social graph is complete, then the

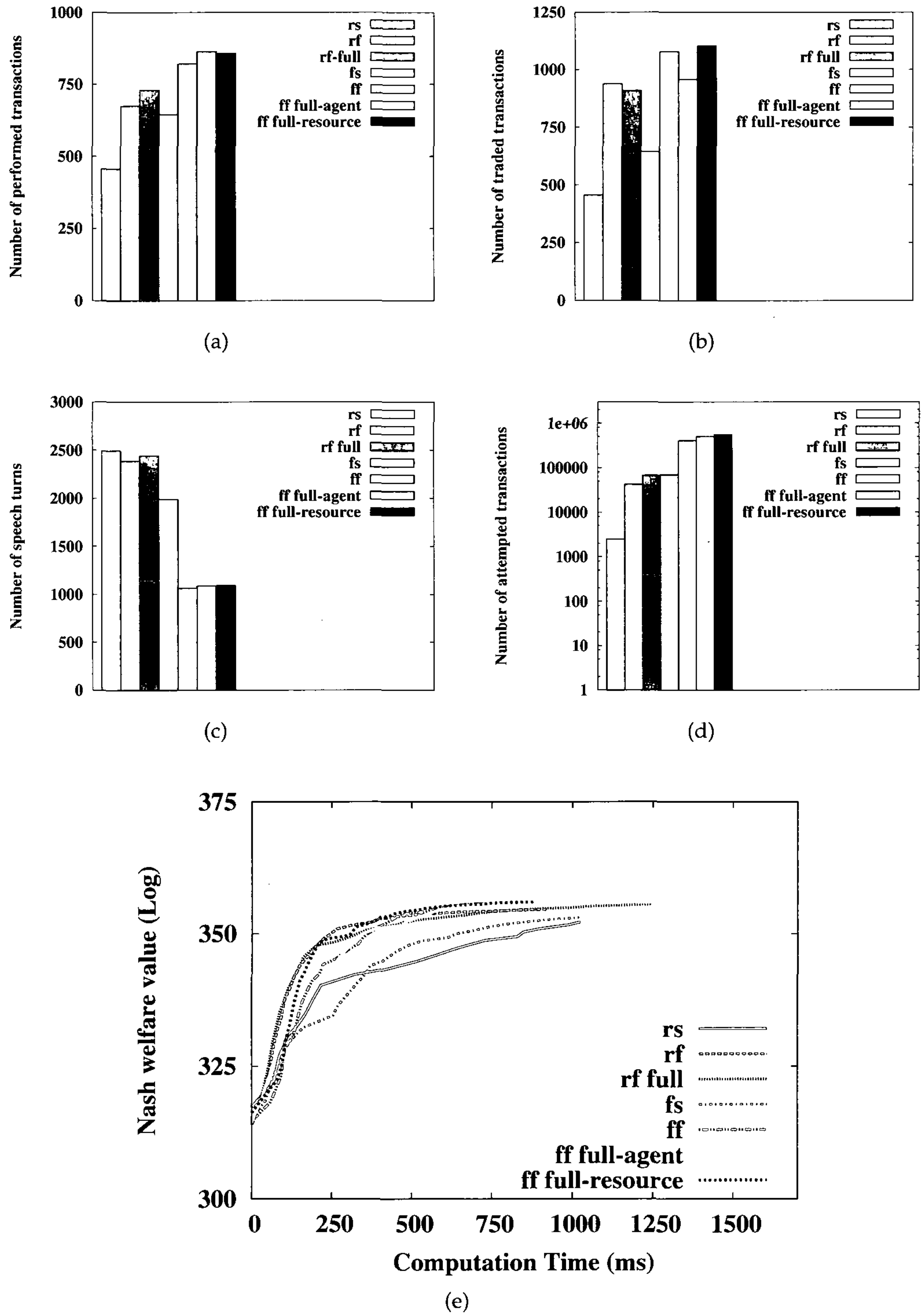


Figure 4.12: Agent behavior impact on Nash negotiations in terms of performed transactions in 4.12a, of transacted resources in 4.12b, of speech turns in 4.12c, of attempted transactions in 4.12d and finally in terms of computation time in 4.12e.

number of offers that agents can attempt is large enough to ensure a sufficient resource traffic. A weak mean connectivity decreases the profit of frivolous and stubborn behaviors.

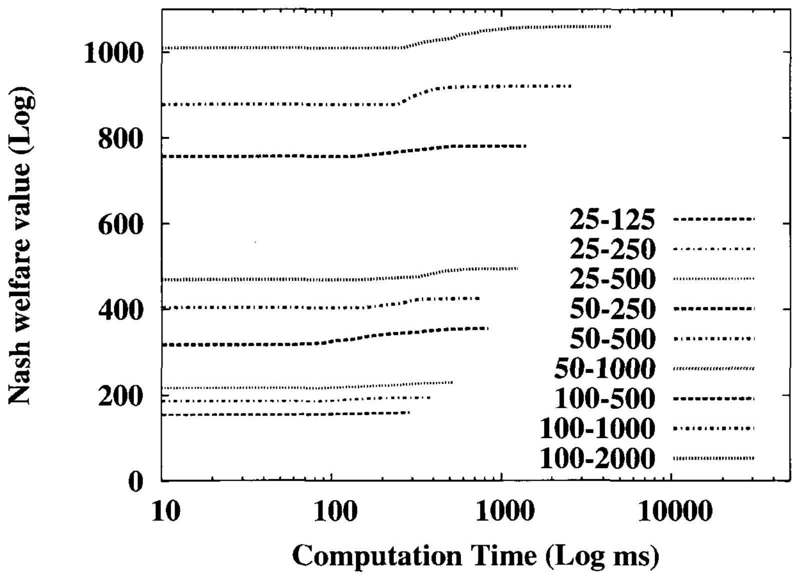
The larger the set of offers that agents can attempt during negotiation processes, the socially greater are generally the achieved allocations. Frivolous and flexible agents maximize the transaction possibilities as well as the resource traffic. A larger number of transactions are performed and more resources are traded. Since agents can benefit from their neighborhood and from their bundle (several offers during the same negotiation can be attempted), only few speech turns are required. The maximization of the resource traffic leads to the achievement of the largest Nash welfare values, independently of the negotiation mechanism used.

Negotiation scalability

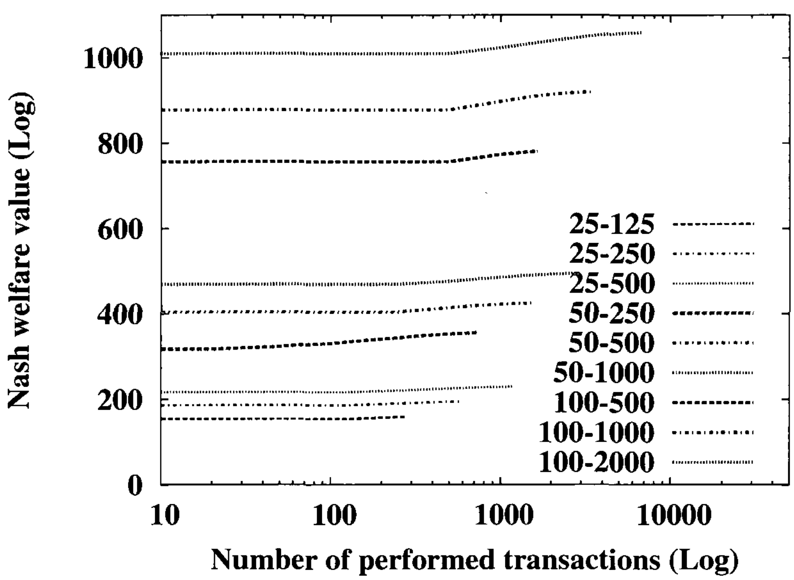
The scalability of Nash negotiations is evaluated according to one of the most time-consuming simulation settings, as described in Section 2.7.3, i.e., on a complete social graph with frivolous and flexible agents, who negotiate using either social gifts or social swaps: $\mathcal{T} = \{\langle 1,0\rangle, \langle 1,1\rangle\}$.

Figure 4.13a represents the evolution of the Nash welfare value according to the computation time whereas Figure 4.13b shows its evolution according to the number of performed transactions. Several instance sizes are used during the experiments, which are characterized by a pair (n, m), where the first element corresponds to the number of agents and the second one corresponds to the number of available resources. Then, the key "25-125" on Figures 4.13 corresponds to instances populated by 25 agents who are negotiating 125 resources.

The mean computation time required to end negotiation processes are presented in Table 4.11, whereas Table 4.12 details the length of the transaction sequence performed. Nash negotiations are a little more time consuming than utilitarian negotiations, but less than egalitarian negotiations. For a given instance, the number of transactions performed does not significantly vary, while the time required to end the negotiation process may



(a)



(b)

Figure 4.13: Evaluation of the Nash negotiation scalability according to the computation time in 4.13a and to the number of performed transactions in 4.13b

significantly vary by a factor greater than 2.

Table 4.16: Nash negotiation scalability - Computation time

Population Size n	Mean number of Resources per Agent $\frac{m}{n}$ 5	10	20
25	340 ms	540 ms	750 ms
50	520 ms	1 s	2 s
100	1.4 s	2.4 s	6.7 s
500	60 s	250 s	600 s

Table 4.17: Nash negotiation scalability - Transaction sequence length

Population Size n	Mean number of Resources per Agent $\frac{m}{n}$ 5	25	50
25	320	600	1300
50	740	1500	3000
100	1700	3300	6900
500	10500	27000	55000

4.3.4 Conclusion

Centralized solving methods quickly face scalability issues. The determination of optimal allocations is a $\mathcal{NP}$-hard problem. Heuristics must be used to estimate this optimal welfare value. In distributed approaches, the combined use of social $\langle 1,0\rangle$ and social $\langle 1,1\rangle$ transactions is the most efficient negotiation policy among frivolous and flexible agents.

Best Nash negotiation policy

Transaction:	$\langle 1,0\rangle + \langle 1,1\rangle$ (i.e., gifts and swaps)
Criterion:	social
Test on δ_i^j:	$u_i(\mathcal{R}_i)u_j(\mathcal{R}_j) < \left(u_i(\mathcal{R}_i) - u_i(\rho_i^\delta) + u_i(\rho_j^\delta)\right)\left(u_j(\mathcal{R}_j) + u_j(\rho_i^\delta) - u_j(\rho_j^\delta)\right)$
Behavior:	frivolous and flexible

Similarly to egalitarian negotiations, bilateral transactions cannot guarantee the achievement of socially optimal resource allocations. The determination of such optima is a difficult task even when centralized approaches are considered. The optimal Nash welfare value can only be estimated in a reasonable time. When negotiations are based on $\mathcal{T} = \{\langle 1,0\rangle, \langle 1,1\rangle\}$, socially efficient allocations can be achieved. Transactions of weaker cardinality are too restrictive and larger transactions do not significantly improve the social welfare value achieved, but increase a lot the negotiation costs. This negotiation policy is flexible enough to ensure a sufficient resource traffic on various classes of social graph topologies. Nash negotiations remain scalable thanks to the limitations on the transaction cardinality even on large instances. Efficient Nash negotiations can be performed among frivolous and flexible agents allowing two transactions: social gifts and social swaps. Although the social graph topology has an important impact, this negotiation policy can always be considered as the best alternative.

4.4 Elitist bilateral negotiations

The elitist welfare is a social welfare notion that aims to maximize the welfare of the richest agent of the population, as defined in Definition 1.7. This notion is commonly used in artificial societies, where a given objective must be achieved at any cost. The objective must be fulfilled, no matter who achieves it. The determination of global optimal elitist welfare values are first presented using centralized techniques. Then, elitist negotiations are discussed with the expression of the elitist acceptability criterion. Elitist negotiations are finally evaluated in order to determine how to negotiate efficiently in elitist societies and how agents should interact in such a context?

4.4.1 Centralized techniques

The identification of the optimal elitist value is a quite simple problem, especially when agents express their preferences using additive utility functions. Since all utility values are positive, optimal elitist allocations satisfy some properties that can be used to simplify the determination of the associated social value. This property specifies how to allocate resources over the population.

Property 4.11 (Elitist optimum). *When the elitist welfare is considered, in socially optimal* *resource allocations, all resources are allocated to one agent, who assigns with them the largest utility value.*

Proof. Let us make a proof by contradiction. A resource allocation $A \in \mathcal{A}$ is assumed to be a global optimum. Then, according to its definition, the following relationship with any other resource allocation $A' \in \mathcal{A}(A \neq A')$ must be satisfied:

$$sw_{e\ell}(A) > sw_{e\ell}(A') \qquad A, A' \in \mathcal{A}, \quad A \neq A'$$

Let us assume that, in the optimal allocation A, agent $i \in \mathcal{P}$ is the richest agent of the population. Then, according to Definition 1.7, the optimal elitist value corresponds to its utility value:

$$\max_{k \in \mathcal{P}} u_k(\mathcal{R}_k) = u_i(\mathcal{R}_i)$$

If agent i does not own all resources, then the allocation A is not a global optimum. Indeed:

$$\exists r \in \mathcal{R}, j \in \mathcal{P} \qquad r \in \mathcal{R}_j, \text{ such that } j \neq i$$

Any allocation A' that allocates this resource r to agent i is associated with a higher elitist value.

$$\begin{aligned} u_i(r) &> 0 \\ u_i(\mathcal{R}_i) + u_i(r) &> u_i(\mathcal{R}_i) \\ sw_{e\ell}(A') &> sw_{e\ell}(A) \end{aligned}$$

Thus, since all utility values are positive, in an elitist optimum, all resources must be allocated to the same agent who estimates the whole resource set $\mathcal{R}$ the most. □

Elitist resource allocation problems can be formulated by means of a mathematical model using Boolean variables x_{ir} describing the ownership of a resource $r \in \mathcal{R}$ by an agent

$i \in \mathcal{P}$:

$$x_{ir} = \begin{cases} 1 & \text{if agent } i \text{ owns resource } r \\ 0 & \text{otherwise.} \end{cases} \qquad r \in \mathcal{R}, \quad i \in \mathcal{P}.$$

Then, elitist resource allocation problems can be formulated as follows:

$$sw^{\star}_{e\ell} = \begin{cases} \max\limits_{i \in \mathcal{P}} \max \sum\limits_{r \in \mathcal{R}} u_i(r) x_{ir} \\ \text{s.t: } \sum\limits_{i \in \mathcal{P}} x_{ir} = 1 \qquad r \in \mathcal{R} \\ \qquad x_{ir} \in \{0, 1\} \qquad r \in \mathcal{R}, \quad i \in \mathcal{P}. \end{cases}$$

Here, the objective is to maximize the utility of the richest agent. Consistency constraints, which are inherent to the resource nature, ensure that each resource is allocated to only one agent.

The other way to determine the optimal utilitarian value is to build an optimal allocation, according to Proposition 4.11. A simple algorithm can then be designed for this purpose. All resources associated with a positive utility value are summed for all agents, and the maximum constitutes the optimal elitist value, as described in Algorithm 4.6.

Algorithm 4.6: Determination of the optimal elitist welfare value

Input: Agent population $\mathcal{P}$, Resource set $\mathcal{R}$

Output: $sw^{\star}_{e\ell}$ the optimal elitist welfare value

$i \leftarrow \arg\max\limits_{j \in \mathcal{P}} u_j(\mathcal{R})$;

$A[i] \leftarrow \mathcal{R}$;

return $sw_{e\ell}(A)$;

In the case where negative values can be associated with resources, this algorithm must be adapted. Indeed, instead of considering the whole set of resources $\mathcal{R}$, each agent must only consider resources that it associates with positive values. The agent that associates with such a resource set the largest welfare obtains this set. The remaining resources can be randomly allocated to others, since they do not affect the elitist welfare value. Algorithm

4.7 describes this process.

Algorithm 4.7: Determination of the optimal elitist welfare value adapted to negative values

---

Input: Agent population $\mathcal{P}$, Resource set $\mathcal{R}$

Output: $sw^{\star}_{e\ell}$ the optimal elitist welfare value

```
maxVal ← 0 ;
maxBundle ← ∅ ;
forall the i ∈ P do
    tmpVal ← 0 ;
    tmpBundle ← ∅ ;
    forall the r ∈ R do
        if u_i(r) > 0 then
            tmpVal ← tmpVal + u_i(r) ;
            Add resource r to tmpBundle ;
        end
    end
    if maxVal < tmpVal then
        j ← i ;
        maxVal ← tmpVal ;
        maxBundle ← tmpBundle ;
    end
end
A[j] ← maxBundle ;
forall the r ∈ R \ {maxBundle} do          // Remaining resources allocation
    i ← random (P \ {j}) ;
    Add r to i ;
end
return sw_el(A) ;
```

4.4.2 Elitist negotiation properties

The expression of the elitist acceptability criterion, defined in Section 2.5, is based on the evaluation of the elitist welfare value evolution between two allocations. Let us note $A \in \mathcal{A}$ the resource allocation before the bilateral transaction $\delta_i^j\langle a, b\rangle$ that changes into a new allocation $A' \in \mathcal{A}(A \neq A')$. This transaction involves two agents $i, j \in \mathcal{P}$, who respectively propose the offers ρ_i^δ and ρ_j^δ. The resource bundle of any agent $k \in \mathcal{P}$ is denoted by $\mathcal{R}_k$ before the transaction and by $\mathcal{R}_k'$ afterward.

An elitist transaction must satisfy the following condition:

$$\begin{aligned} sw_{e\ell}(A) &< sw_{e\ell}(A') \\ \max_{i \in \mathcal{P}} (u_i(\mathcal{R}_i)) &\leq \max_{i \in \mathcal{P}} \left(u_i(\mathcal{R}_i')\right) \end{aligned}$$

Similarly to the egalitarian interpretation of the social acceptability criterion, the elitist interpretation is not a strict inequality when the test is restricted to only two agents. However, the restriction is not as strong as the one imposed in the egalitarian case. Indeed, if the poorest agent of the population is not involved, the egalitarian welfare value cannot vary, but according to the elitist notion, even if the richest agent of the population is not involved, the welfare value may increase. Nothing prevents for instance another agent to become richer than the agent of the population who was the richest before the transaction.

The expression that elitist transactions δ_i^j must satisfied can be restricted to only two agents. In such a case, the richest of the two involved agents after an elitist transaction must be richer than the agent that was richer before.

$$\begin{aligned} \max_{i,j \in \mathcal{P}} \left(u_i(\mathcal{R}_i), u_j(\mathcal{R}_j)\right) &< \max_{i,j \in \mathcal{P}} \left(u_i(\mathcal{R}_i'), u_i(\mathcal{R}_j')\right) \\ \max_{i,j \in \mathcal{P}} \left(u_i(\mathcal{R}_i), u_j(\mathcal{R}_j)\right) &< \max_{i,j \in \mathcal{P}} \left(u_i(\mathcal{R}_i) + u_i(\rho_j^\delta) - u_i(\rho_i^\delta), u_j(\mathcal{R}_j) + u_j(\rho_i^\delta) - u_j(\rho_j^\delta)\right) \end{aligned}$$

According to the elitist criterion, when two agents $i, j \in \mathcal{P}$ negotiate, an agent should give all its resources to its partner in order to maximize the welfare of this agent. It must become richer than the richest agent before the transaction. The agents' decision making, which is represented by the instruction TEST in all behaviors that are described in Section 2.6, is based on this expression:

$$\text{TEST} := \left[\max_{i,j\in\mathcal{P}}\left(u_i(\mathcal{R}_i), u_j(\mathcal{R}_j)\right) < \max_{i,j\in\mathcal{P}}\left(u_i(\mathcal{R}_i) + u_i(\rho_j^\delta) - u_i(\rho_i^\delta), u_j(\mathcal{R}_j) + u_j(\rho_i^\delta) - u_j(\rho_j^\delta)\right)\right]$$

Some properties related to the decomposition of elitist transactions can be established. Similarly to the other cases, elitist bilateral transactions of large cardinality may be essential to achieve optimal resource allocations.

Property 4.12 (Elitist transaction split). *Within an elitist society where agents express their preferences by additive utility functions, social bilateral transactions $\delta_i^j\langle a, b\rangle$ between two agents $i, j \in \mathcal{P}$ cannot always be split into a sequence of elitist bilateral transactions $\delta_i^j\langle a', b'\rangle$ of lesser cardinality ($a > a'$ and/or $b > b'$).*

Proof. Let us consider a counter-example based on a population of two agents, $\mathcal{P} = \{0, 1\}$ who are negotiating the three available resources $\mathcal{R} = \{r_1, r_2, r_3\}$. Their preferences are expressed by means of additive utility functions, described in Table 4.18. The initial resource allocation is $A = [\{r_1, r_2\}\{r_3\}]$.

Table 4.18: Elitist transaction split - Example of agent preferences

Population $\mathcal{P}$	Resource Set $\mathcal{R}$		
	r_1	r_2	r_3
0	5	6	1
1	4	5	5

Let us consider the transaction $\delta_0^1\langle 2, 0\rangle = (\{r_1, r_2\}, \emptyset)$, during which agents 0 and 1 respectively propose $\rho_0^\delta = \{r_1, r_2\}$ and $\rho_1^\delta = \emptyset$. Such a transaction is social since the utility of the richest agent increases from 11 in the initial allocation to 14 in the final allocation:

$$\begin{aligned} \max\left(u_0(\{r_1, r_2\}), u_1(\{r_3\})\right) &< \max\left(u_0(\emptyset), u_1(\{r_1, r_2, r_3\})\right) \\ u_0(\{r_1, r_2\}) &< u_1(\{r_1, r_2, r_3\}) \end{aligned}$$

This transaction can be only split into two successive gifts. However, none of them is acceptable according to the social acceptability criterion. If agent 0 gives one of its resources, then the other agent becomes the richest, but the elitist welfare value of the population is weaker than it was initially. Such gifts are thus not acceptable.

Hence, in elitist societies, bilateral transactions cannot always be split into a sequence of elitist bilateral transactions of lesser cardinality. Transactions of large cardinality may thus be required to achieve a socially optimal solution. □

4.4.3 Evaluation of elitist negotiations

As described in Section 2.7, different facets of negotiation processes are evaluated in this section. First, the impact of the transaction cardinality is investigated in order to determine which transactions are the most suitable to efficiently negotiate according to an elitist objective. Then, the efficiency of different negotiation policies is described according to different classes of social graphs. The impact of the mean connectivity on the negotiation efficiency is presented as well as the weight of the agents' interactions. Finally, issues related to the scalability of elitist negotiation process are discussed.

Influence of the transaction cardinality

According to the elitist acceptability criterion, restrictions can be made on the cardinality of the allowed transactions. Since all utility values are positive, an implicit consequence can be observed: An elitist negotiation process tends to gather all resources into a single agent bundle, who assigns the largest utility value to them. Then, any bilateral transaction $\delta_i^j\langle a, b\rangle$ such that $b > 0$ is contrary to this objective.

According to Proposition 4.12, elitist transactions may not be split in a sequence of elitist transactions of lesser cardinality. Agent $i \in \mathcal{P}$ should then be able to offer its whole resource bundle, without compensation: $\mathcal{T} = \{\langle m_i, 0\rangle\}$. Such a transaction is meaningless when other social welfare notions are considered, but perfectly suits the elitist notion.

Several sizes of offers can be tested. However, the computation time required to end an elitist negotiation processes based on $\delta_i^j\langle 1, 0\rangle$ for instance is exponentially higher ($i, j \in \mathcal{P}$). Even instances of reasonable size (e.g., 50 agents and 250 resources) are not really scalable. Thus, only cluster transactions of maximal size, i.e., $\langle \rho_i^\delta, 0\rangle$ transactions should be used when negotiating.

Price of social graphs

Since only cluster transactions are considered, an acceptability criterion cannot be used anymore. Indeed, since agents only offer their whole resource bundle without compensation, the rational acceptability criterion is improper. Since all utility value are positive, no rational cluster transaction exists. Thus, such an acceptability criterion is unadapted and inefficient when the elitist welfare notion is considered. This acceptability criterion is thus not represented in the following experiments.

Table 4.19 presents the efficiency of elitist negotiation processes based on social $\delta\langle m_i, 0\rangle$ transactions and on different classes of social graphs. Here, 50 agents negotiate 250 resources according to frivolous and flexible behaviors.

Table 4.19: Elitist negotiation efficiency(%) and standard deviation according to the class of social graphs

Social graph	Social efficiency (%) $\langle m_i, 0\rangle$	Standard deviation
Full	100	0
Grid	31.17	26.92
Erdős-Rényi	95.12	11.53
Small world	68.43	66.50

Table 4.19 shows that, when the relationships among agents can be modeled by a complete social graph, negotiation processes based on $\delta\langle m_i, 0\rangle$ transactions, i.e., cluster

transactions of maximal size, always lead to socially optimal resource allocations. When grids are considered, negotiation processes achieve allocations which correspond to only 31.17% of the optimum. The mean connectivity is too weak to ensure a proper resource traffic. Moreover, the achieved elitist welfare value may vary by 26.92%. The large standard deviation reveals the important impact of the topology allocation. Since resources circulate barely, depending on the agent to which they are initially allocated, resources can be trapped somewhere, and then penalize elitist negotiations. Resources remain dispatched over the population, which explains the poor efficiency of negotiations based on grids. Negotiations based on Erdős-Rényi graphs achieve socially efficient allocations. Indeed, 95.12% of the optimal welfare value can be achieved, with a standard deviation of 11.53%. The mean connectivity is higher than in grids, which allows a sufficient resource circulation and result in interesting allocation. However, in the case of small-worlds, only 68.43% of the optimal welfare value can be achieved. Their mean connectivity is really low (on average 6.8 neighbors per agent) and irregular. Most agents have only few neighbors while few agents have many neighbors. This irregularity explains the very large standard deviation which is observed. Depending on the initial resource allocations, many resources cannot change of owners and thus lead negotiation processes into a local optimum.

Theorem 4.13. *Within an elitist society where agents express their preferences by means of additive utility functions, a negotiation processes based on complete social graphs always converge towards a global optimum using cluster transactions of maximal size, i.e.,* $\delta\langle m_i, 0\rangle$ *for any initiator* $i \in \mathcal{P}$.

Proof. Since the social graph is complete and connected, any agent $i \in \mathcal{P}$ can talk with every other agents of the population. The resource traffic is composed by several sets of resources. The number of dispatched resource sets decreases gradually during negotiation processes while their size is gradually growing. Indeed, since $\mathcal{T} = \{\langle m_i, 0\rangle\}$, each time that two agents negotiate, one of them offers its whole resource bundle to the other, who finally owns a larger resource bundle.

It is always possible to create a sequence of cluster transactions that gather all resources into a single agent bundle, which is associated with the largest utility value. The size of the cluster transactions gradually increases during negotiation processes until the whole set of available resources is owned by a single agent. Once all resources are gathered, the whole set of resources can still be offered to agents associating a larger utility value to it. When it is no more possible, according to Proposition 4.11, the resulting allocation is a global optimum. □

Influence of the social graph connectivity

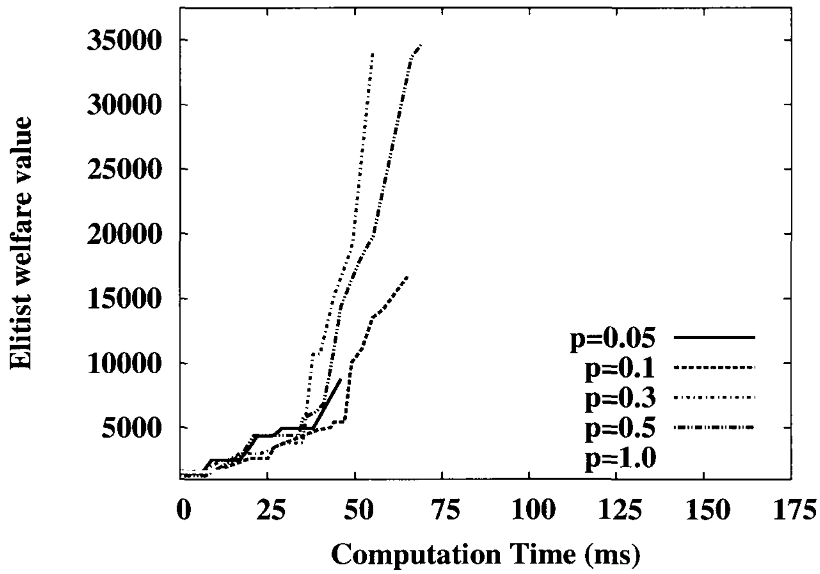
The mean connectivity of social graphs affects the negotiation efficiency since it restricts more or less the transaction possibilities. Considering Erdős-Rényi graphs, the mean connectivity can vary thanks to the probability p of link generation in the model of generation $G(n,p)$. These simulations are based on a population of 50 agents who negotiate 250 resources, carrying out frivolous and flexible behaviors.

As shown in Figure 4.14b, the number of performed transactions does not vary significantly (between 65 and 80). Negotiation processes end after transaction sequences of close length. Figure 4.14a shows on the other hand that the computation time varies from 40 ms to 125 ms. However the elitist welfare value on which negotiation processes end vary a lot. The mean connectivity significantly affects the quality of provided solutions only when the probability is below $p = 0.3$. If the probability of generating a link between two nodes is higher, the efficiency is not affected more than 8%. But if the probability is lower, the elitist welfare value that can be achieved decreases drastically.

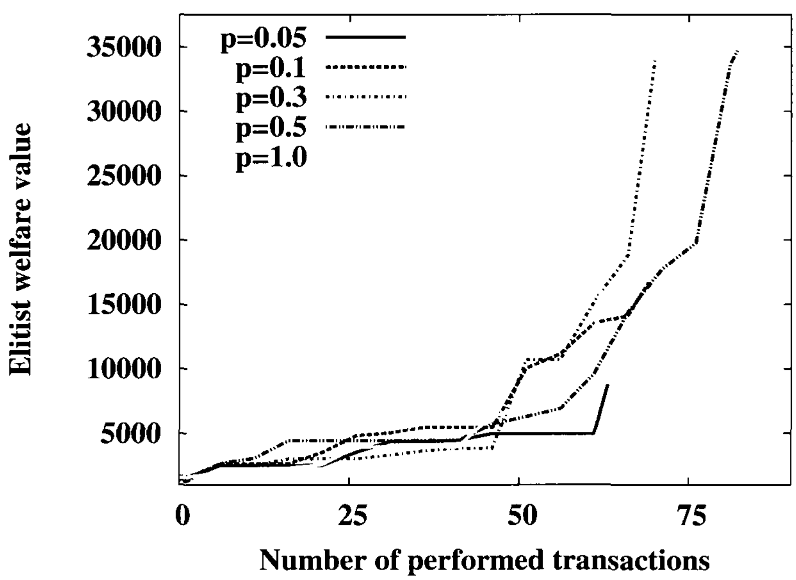
Influence of agents' behaviors

Behaviors define how agents interact, i.e., how they negotiate. The different agent behaviors defined in Section 2.6, can be compared using various metrics as presented in Section 2.7.1. Negotiation processes are based in these experiments on complete social graphs, with a population of 50 agents and 250 available resources. Agents negotiate using social transactions only: $\mathcal{T} = \{\langle m_i, 0\rangle\}$.

Only one transaction is allowed during elitist negotiation processes. As described in



(a)



(b)

Figure 4.14: Social graph connectivity impact on elitist negotiations in terms of computation time in 4.14a and of performed transactions in 4.14b.

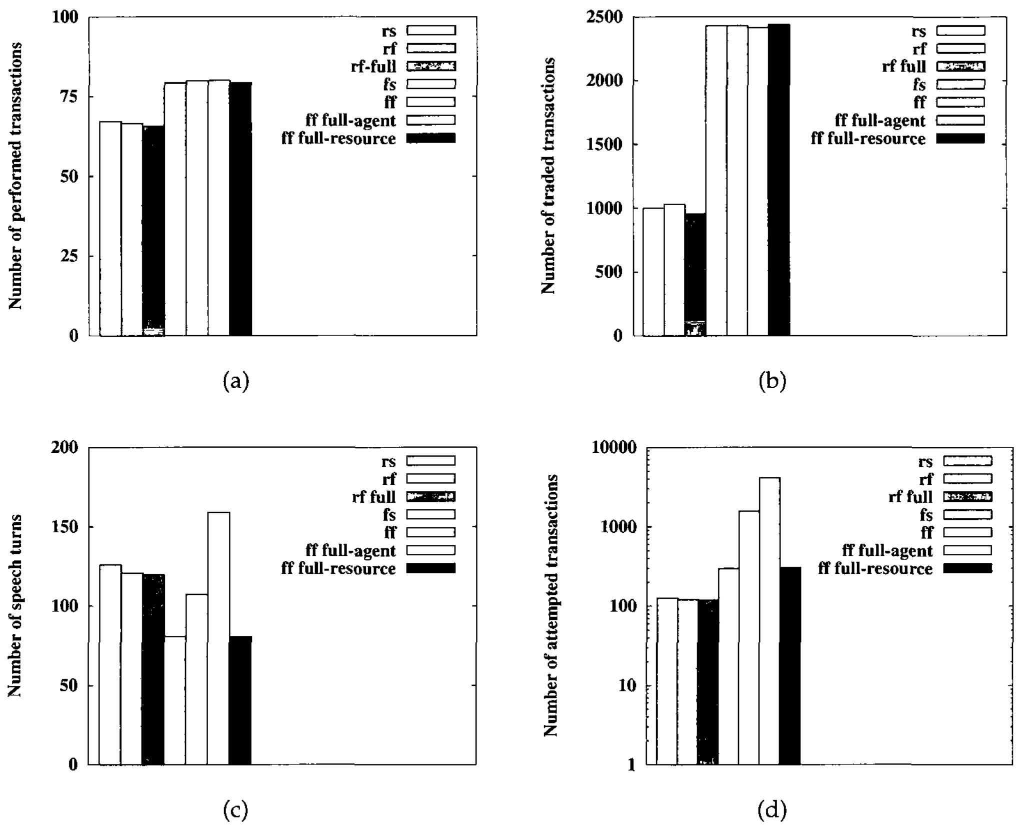


Figure 4.15: Agent behavior impact in terms of performed transactions in 4.15a, of transacted resources in 4.15b, of speech turn in 4.15c, and in terms of attempted transactions in 4.15d.

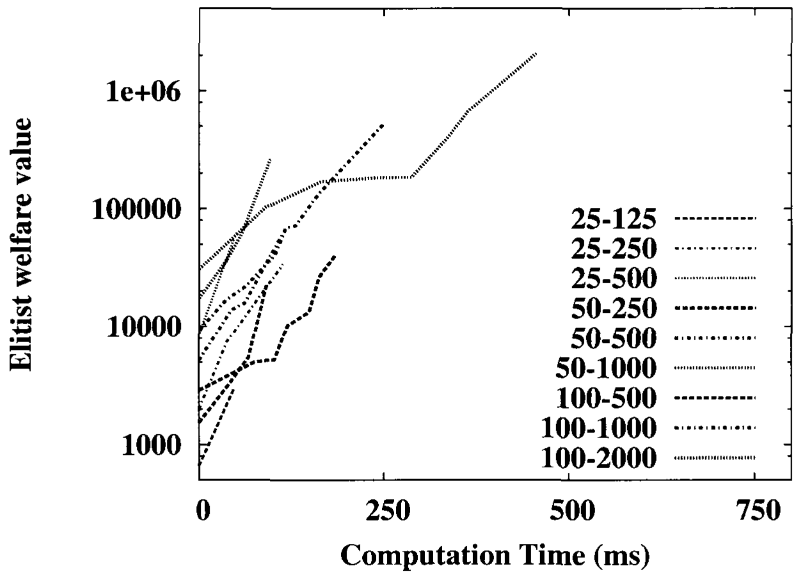
the previous sections, only $\langle m_i, 0\rangle$ transactions are allowed where $i \in \mathcal{P}$ is the initiator. Accordingly, agents are only able to make a single offer. Thus, flexible behaviors have no impact because $L_i(\rho)$, the list of offers that agent i can propose has only one element corresponding to its whole bundle $\mathcal{R}_i$. For this reason, rooted behaviors, either flexible or stubborn obtain almost similar value for all metrics.

Related to the four frivolous behaviors, negotiations end after transaction sequences of identical length, where almost the same number of resources are traded. However, variations can be observed when the number of speech turns or the number of attempted transactions is considered. These variations are simply due to the frivolous character of the agents' behaviors. The order in which neighbors are considered influences the negotiation process. According to this order, the number of attempted transactions varies. In order to achieve socially optimal allocations, agents must be frivolous. They can thus interact with all their neighbors.

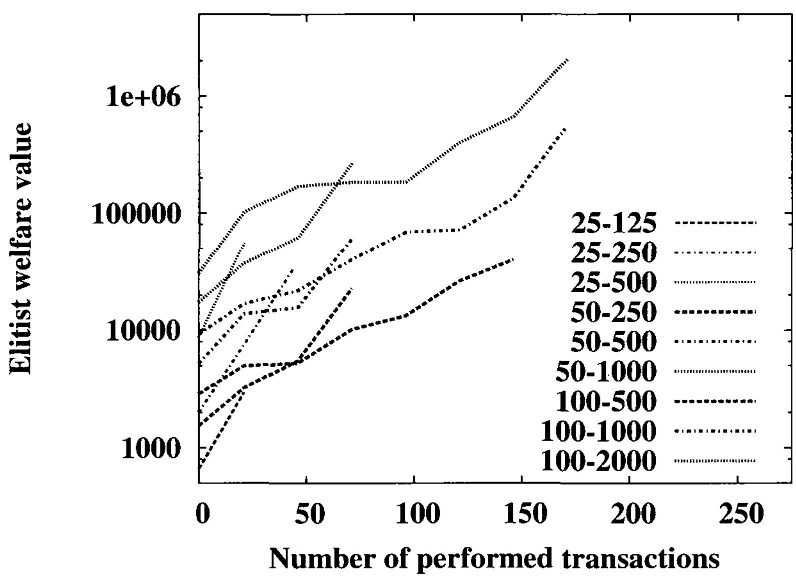
Negotiation scalability

The scalability of elitist negotiations is evaluated when relationships among agents are modeled by means of complete social graphs. Agents interact according to frivolous behaviors. On each figure, several sizes of instances are used. An instance is characterized by a pair $n - m$, where n is the number of agents and m is the overall number of resources, initially distributed in a random way. For instance, the key "100-1000" corresponds to an instance where 100 agents negotiate 1000 resources using only $\langle m_i, 0\rangle$ transactions for any initiator $i \in \mathcal{P}$.

Figure 4.16a represents the evolution of the elitist welfare value according to the computation time, using different instance sizes. Larger instances lead to longer negotiations and to greater elitist welfare values. The different floors that can be observed correspond to transactions between agents who cannot become the richest ones in the population. According to the elitist acceptability criterion, one of the involved agent becomes richer than the richest agent before the transaction. However, if it does not become richer than



(a)



(b)

Figure 4.16: Scalability of elitist negotiations in terms of computation time in 4.16a and in terms of performed transactions in 4.16b

the richest agent of the population, the elitist welfare value does not change, but resources circulate.

Figure 4.16b shows the evolution of the social welfare value according to the number of performed transactions. It reveals that, independently of the mean number of resources per agent, the number of performed transactions if almost the same. Since agents negotiate using only $\langle m_i, 0\rangle$ transactions, agents offer their whole resource bundle independently of their size. The number of resources per agent does not affect the length of transaction sequences, in contrast to negotiation processes based on other social welfare notions, which suffer from an exponential increase.

Table 4.20 presents the computation time required to end elitist negotiation processes according to the instance size, whereas Table 4.21 focuses on the number of performed transactions. These tables show that the mean number of resources per agent has a weak impact on elitist negotiation processes. Such elitist negotiations remain highly scalable even when large instances are considered. Indeed, a negotiation among 500 agents and 10000 agents can be solved in 8.2 seconds, after a sequence of 819 cluster transactions.

Table 4.20: Elitist negotiation scalability - Computation time

Population Size n	Mean number of Resources per Agent $\frac{m}{n}$		
	5	10	20
25	98 ms	113 ms	120 ms
50	132 ms	136 ms	140 ms
100	276 ms	350 s	642 ms
500	1.6 s	3.3 s	8.2 s

Table 4.21: Elitist negotiation scalability - Transaction sequence length

Population Size n	Mean number of Resources per Agent $\frac{m}{n}$		
	5	25	50
25	35	44	46
50	82	84	87
100	162	170	174
500	781	803	819

4.4.4 Conclusion

Centralized techniques are quite trivial when elitist problems are considered. All resources must be allocated to the agent who assigns with them the largest utility to them. In distributed agent negotiations, the use of social $\langle m_i, 0\rangle$ transactions is the most efficient negotiation policy among frivolous and flexible agents.

Best elitist negotiation policy

Transaction:	$\langle m_i, 0\rangle$ (i.e., cluster of maximal size)
Criterion:	social
Test on δ_i^j:	$\max_{i,j\in\mathcal{P}}\big(u_i(\mathcal{R}_i), u_j(\mathcal{R}_j)\big) < \max_{i,j\in\mathcal{P}}\big(u_i(\mathcal{R}_i) + u_i(\rho_j^\delta) - u_i(\rho_i^\delta), u_j(\mathcal{R}_j) + u_j(\rho_i^\delta) - u_j(\rho_j^\delta)\big)$
Behavior:	frivolous

The elitist welfare notion can be used in specific situations where an objective must be achieved independently of the agent who achieves it. The maximization of the welfare of the richest agent may seem inappropriate to human societies, but suits many applications among artificial agents. The rational acceptability criterion is meaningless when the elitist welfare notion is considered since no cluster transaction can be rational. Collaboration among agents is essential during elitist negotiations, which then should be based on social transactions. Only one transaction is required to negotiate efficiently: $\langle m_i, 0\rangle$ transactions, during which an agent offers its whole resource bundle. Since agents can only make one offer, agents do not need to behave with flexibility. Only the frivolity characteristic is useful, to benefit from their neighborhood. When relationships among agents are represented by means of complete graphs, elitist negotiations based on social $\langle m_i, 0\rangle$ transactions always lead to global optima. When the agents' neighborhoods are restricted, the achievement of global optima cannot be guaranteed, but if the agent neighborhood are large enough, socially efficient allocations can still be achieved. The mean number of resources does not affect the scalability of elitist negotiations, while it has an exponential cost when other welfare notions are considered. Elitist negotiations are highly scalable and large instances can be solved quickly.

4.5 Summary

In this chapter, agent negotiations based on bilateral transactions have been evaluated. Different facets of negotiations have been considered: The size of offers that agents can propose, the agent acceptability criterion or their behaviors, social graph classes, the negotiation efficiency according to the social graph, and their scalability. The four main notions of the social welfare theory have been investigated.

Generally, even if they are widely used in the literature, rational negotiations are not efficient when the aim is to maximize the social welfare. To achieve socially efficient allocations, agents must be *generous*. They must accept to loose a little for the sake of the whole society.

Utilitarian negotiations are the most efficient among frivolous and flexible agents, using only social $\langle 1, 0 \rangle$ transactions (i.e., gifts). When complete social graphs are considered, optimal allocations can always be achieved. When restricted relationships are considered, socially efficient resource allocations can still be achieved. Such utilitarian negotiations favor the resource circulation sufficiently in order to be efficient on many classes of graphs. Moreover, they remain scalable even on large instances.

Egalitarian negotiations are the most efficient among frivolous and flexible agents. Two transactions must be used among the agents: Social $\langle 1, 0 \rangle$ transactions and social $\langle 1, 1 \rangle$ transactions are required to achieve fair resource allocations (i.e., either gifts or swaps). Bilateral transactions cannot guarantee the achievement of socially optimal allocations. Such negotiations are more sensitive to topological issues of restricted social graphs. Moreover, any characteristic restricting the resource circulation affects a lot the negotiation efficiency. They remain scalable on large instances in contrast to exact centralized techniques.

Nash negotiations are also the most efficient among frivolous and flexible agents, using both social $\langle 1, 0 \rangle$ transactions and social $\langle 1, 1 \rangle$ transactions (i.e., gifts and swaps). As in egalitarian negotiations, bilateral transactions cannot ensure the achievement of the socially optimal allocations. The efficiency of such negotiation processes is difficult to evaluate since centralized techniques are not easy to handle. Nash negotiations are less sensitive than egalitarian negotiations and also less time-consuming. Indeed, they remain scalable on large instances of population.

Elitist negotiations are efficient among a population of frivolous agents. Agents should use cluster transactions of maximal size, i.e., social $\langle \mathcal{R}_i, 0 \rangle$ transactions. Agents should offer their whole resource bundle if it improves the elitist welfare of the society. Thus, the notion of rationality does not fit at all. Based on complete graphs, negotiations always achieve optimal solutions, but not when restricted communication possibilities are considered. Such negotiations are scalable on large instances, and the computation time is moreover independent of the number of resources available.

Table 4.22: Summary - Efficient bilateral negotiation settings

	Social welfare notions			
	Utilitarian	**Egalitarian**	**Nash**	**Elitist**
Acceptability criterion	social			
Allowed transactions	gifts	gifts and swaps		maximal cluster
Agent behaviors	frivolous and flexible			frivolous

Chapter 5

Multilateral Negotiations

Generally, bilateral transactions are not sufficient to guarantee the achievement of socially optimal allocation. Especially when the social graphs are restricted, the agent communication possibilities are also restricted, which make difficult the resource circulation. In order to favor resource circulation, multilateral transactions might be used. Allowing such transaction classes may increase the possible number of transactions, as described in Section 2.4.2 and hence may improve the quality of provided allocations. However, the determination of such acceptable transactions is not an obvious task, and negotiation processes based on such transactions may face scalability problems.

This chapter addresses multilateral transactions. First, the motivations and the limits using such a transaction class are presented in Section 5.1. This section describes how such transactions favor resource circulation and the achievement of socially more interesting resource allocations, especially when restricted social graphs are considered. Multilateral transactions also suffer from drawbacks that may prevent their practical application. These drawbacks and the limits of the use of multilateral transactions are also discussed. A literature review is presented in Section 5.2, in order to describe different studies on this topic and their characteristics. Section 5.3 is dedicated to the model that we propose to solve the problem of the determination of acceptable multilateral transactions. An introduction to an efficient solving method is presented in Section 5.4. This solving method is applied to our problem in Section 5.5, whereas Section 5.6 evaluates the solutions.

5.1 Motivations and limitations

The determination of acceptable multilateral transactions is not an obvious issue. According to the description of multilateral transactions of Section 2.4.2, an exponential number of possible multilateral transactions can be carried out during each negotiation. This section reviews the pros and cons of using multilateral transactions during negotiation processes. Do their advantages compensate their drawbacks?

5.1.1 Motivations

Multilateral transactions favor the resource traffic during negotiations and hence can facilitate the identification of acceptable transactions. When restrictive acceptability criteria are considered, only few transactions might be acceptable by an agent. For instance, within a population of rational agents, agents only accept transactions increasing their own utility value. Such negotiation processes limit a lot the resource traffic and may only achieve socially sub-optimal allocations. However, during a multilateral transaction, e.g., during a one-to-many transaction δ_i^{Δ}, the initiator can negotiate with a set of its neighbors $\Delta^{\delta} = \{j, k\} \subseteq \mathcal{N}_i$. It can compensate a loss of utility from a part of the transaction involving one neighbor, with the benefit resulting from a part of the transaction involving another neighbor. Thus, the use of multilateral transactions increases the number of acceptable transactions and negotiation processes may avoid local optima. Let us illustrate this phenomenon with an example.

Example 5.1. Let us consider an agent population $\mathcal{P} = \{0, 1, 2\}$ where 5 resources are available, $\mathcal{R} = \{r_1, \ldots, r_5\}$. Agents express their preferences by means of additive utility functions described in Table 5.1. The initial resource allocation is $A = [\{r_1 r_2\}\{r_3\}\{r_4\}]$. The social objective of this system is the maximization of the utilitarian welfare, but examples can be designed for any other notions of the social welfare theory. We also assume that all agents are rational. Hence, they only accept transactions increasing their own welfare.

Table 5.1: Multilateral transaction motivations - Example of agent preferences

Population $\mathcal{P}$	Resource Set $\mathcal{R}$			
	r_1	r_2	r_3	r_4
0	5	7	10	3
1	7	5	4	5
2	2	9	6	4

Agent 0 is the initiator of a one-to-many transaction involving its two neighbors, agents 1 and 2, such that, according to Definition 2.7 of a one-to-many transaction, $\delta_0^{1,2} = \{(\rho^\delta_{01}, \rho^\delta_{10}), (\rho^\delta_{02}, \rho^\delta_{20})\} = \{(\{r_1\}, \{r_3\})(\{r_2\}, \{r_4\})\}$. Agent 0 offers agent 1 the resource set $\rho^\delta_{01} = \{r_1\}$ and receives $\rho^\delta_{10} = \{r_3\}$. Simultaneously, agent 0 offers agent 2 another set of resource $\rho^\delta_{02} = \{r_2\}$ and receives $\rho^\delta_{20} = \{r_4\}$.

This one-to-many transaction is rational and then acceptable since it leads to an increase of the utility of all involved agents, according to Table 5.2, which describes the utility of the involved agents before and after the multilateral transaction.

Table 5.2: Multilateral transaction motivations - Evolution of utility values

Population $\mathcal{P}$	Agent utility	
	Initially	After δ_0^{12}
0	$u_0(\{r_1 r_2\}) = 12$	$u_0(\{r_3 r_4\}) = 13$
1	$u_1(\{r_3\}) = 4$	$u_1(\{r_1\}) = 7$
2	$u_2(\{r_4\}) = 4$	$u_2(\{r_2\}) = 9$

However, the bilateral transactions δ_0^1 and δ_0^2 are not individually rational. Performed simultaneously, these transactions satisfy the rational acceptability criterion, but not individually since one of them, δ_0^3, is not rational. The initiator uses the utility savings from one part to compensate the loss of the other one.

A multilateral transaction can be viewed as a sequence of bilateral transactions at the end of which all involved agents satisfy their own acceptability criterion. This criterion is not necessarily satisfied after each bilateral transaction of the sequence, as described in Figure 5.1. A sequence of acceptable bilateral transactions exists, changing successively the initial resource allocation A_0 to A_1, A_2, A_3 and finally A_4, which corresponds to a socially optimal resource allocation. All bilateral transactions of this sequence are acceptable. An acceptable multilateral transaction can be viewed as a sequence of bilateral transactions, which are not necessarily individually acceptable, but which ensure the acceptability of the achieved allocation. In this figure, the acceptable solution space is connected, but this is not the case most of time, especially when restricted social graphs are considered. In such cases, the use of multilateral transactions can be the only way to guarantee the achievement of socially optimal allocations. Some optimization methods have some similarities. In order to speed up the solving process or to leave local optima, some optimization methods accept to go through non-satisfiable solutions if they are sure that satisfiable solutions can be achieved later on.

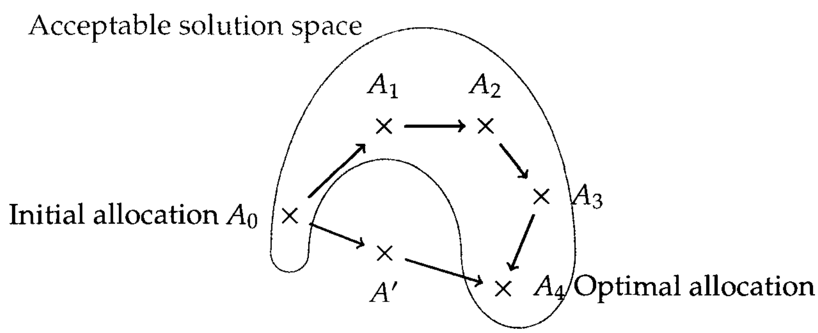


Figure 5.1: Interpretation of multilateral transactions

5.1.2 Limitations

Multilateral transactions have some significant drawbacks that must be considered. The first drawback is related to the quantity of information required to identify an acceptable multilateral transaction. Indeed, an initiator may offer some resources to different neighbors simultaneously. In order to avoid an exponential number of offers that can be proposed to the different neighbors, the initiator can gather information about its neighbors' preferences and their resource bundle. The initiator can then identify an acceptable multilateral transaction. However, to apply such a solving process, all neighbors must accept to reveal their private information to the successive initiators. Without such an information gathering step, it is still possible to identify acceptable transactions, but it may require the explicit enumeration of all possible transactions. It may then compromise the scalability of just one negotiation and then a complete negotiation process may not be scalable. Thus, in order to use multilateral transactions in a scalable way, the information privacy must be sacrificed.

Another important limitation depends on the social welfare notion that is considered. Indeed, each welfare notion does not lead to the same complexity of the solving process. There is no specific difficulty when utilitarian problems are considered. In the case of egalitarian problems, negotiation processes may face scalability problems. Indeed, as described in Section 4.2.1, the exact resolution may be time consuming, but it is still possible. But, when the Nash product is considered, such a method is not scalable. The exhaustive enumeration cannot be avoided to guarantee that no acceptable multilateral transaction exists among the involved agents. Thus, the determination of acceptable multilateral transactions is more or less complex and expensive depending on the social welfare notion considered.

Bilateral transactions cannot guarantee the achievement of optimal allocations, for instance when the egalitarian welfare notion is considered (Proposition 4.7). Some classes of multilateral transactions can solve such problems, e.g., many-to-many transactions, but if the social graph is restricted, even the use of multilateral transactions cannot certify the achievement of socially optimal resource allocations. Proposition 1.3 is still true when the use of multilateral transactions is allowed.

Example 5.2. Let us consider a population of four agents, $\mathcal{P} = \{0, \ldots, 3\}$, and a set of four available resources, $\mathcal{R} = \{r_1, \ldots, r_4\}$ which are initially allocated as follows: $A = [\{r_1\}\{r_2\}\{r_3\}\{r_4\}]$. Their preferences are expressed by means of additive utility functions that are described in Table 5.3.

The relationships among the agents are represented by the social graph illustrated in Figure 5.2. The social graph topology is a chain, where agents know at most 2 neighbors.

Table 5.3: Multilateral transaction insufficiency - Example of agent preferences

Population $\mathcal{P}$	Resource Set $\mathcal{R}$			
	r_1	r_2	r_3	r_4
0	3	1	1	9
1	1	3	1	1
2	1	1	3	1
3	9	1	1	3

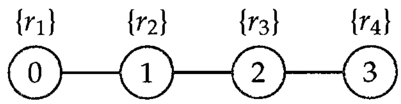


Figure 5.2: Insufficiency of multilateral transactions - Example of social graph

The objective is to maximize the utilitarian welfare. The optimal allocation is $A' = [\{r_4\}\{r_2\}\{r_3\}\{r_1\}]$, which is associated with the welfare value $sw_u(A') = 24$. However, even allowing multilateral transactions, no acceptable sequence leads r_1 into agent 3's bundle and inversely r_4 into agent 0's bundle. Then, the socially optimal resource allocation cannot be achieved, even if multilateral transactions are allowed.

A question can be raised: Since the achievement of optimal allocations cannot be guaranteed by the use of multilateral transactions, are their use justified in spite of their costs? In order to be interesting, an efficient method must be designed to determine acceptable multilateral transactions.

5.2 Related works

Efficient resource allocation is a complex issue and encounters very quickly some scalability difficulties. A formal classification of the transactions is proposed (Sandholm, 1998). The classification starts with bilateral contracts, e.g. gifts, swaps or clusters and ends with multilateral transactions such as the multi-agent contracts and the OCSM contracts that involve more than two agents simultaneously. These multilateral transactions can either be one-to-many or many-to-many transactions according to the resource traffic, as defined in Section 2.4.2. The author showed that the use of multilateral transactions is essential in order to guarantee the achievement of optimal allocations, when the social graph is complete. However, only few works have studied their complexity (Endriss and Maudet, 2005; Friedman and Rust, 1993). Multilateral transactions are difficult to determine and not necessarily scalable.

In order to tackle the scalability issues, different studies suggest to use a restricted set of transactions. First at all, start with the simplest ones, then use more complex ones only when no simpler one is possible (Andersson and Sandholm, 1998). This iterated process leads most of the time to a local optimum.

In the literature, generic models are proposed to solve problems under incomplete information, which are represented by means of *types*, which represents the agents' beliefs. Mechanism design is then used to create interaction rules among the agents of a society. It is a sub-field of microeconomics and game theory that considers how to design the rules of interaction in a game to achieve specific properties, for problems involving multiple agents. Most of the time, systems are populated by selfish agents who can misreport information about their preferences, in order to manipulate the mechanism and increase their profits. By properly designing the interaction rules, it is possible to incite the agents to report truthful information only. Mechanism design recently became a tool in computer science and operational research (Conitzer and Sandholm, 2002; Feigenbaum and Shenker, 2002; Dash et al, 2003), due to distributed systems, like Internet, which have many characteristics of an economy. Such approaches are typically used in applications where agents have limited resources (Kfir-Dahav et al, 2000; Dash et al, 2005) or in load balancing problem (Grosu and Chronopoulos, 2004). These studies are always based on selfish agents (Nisan, 1999). Issues related to faithful mechanism are studied (Dash et al, 2004). Mechanisms are specifically designed to incite agents to report truthful information during negotiations.

Recent studies (Sandholm, 2003) introduce the notion of *Automated Mechanism Design*. This approach tackles mechanism design as an optimization problem and proposes to automatically design the target mechanisms by means of optimization algorithms. The optimization algorithm defines the rules of interaction, hence the agents' behaviors. Integer linear programs are used to model and implement Arrovian welfare functions (Sethuraman et al, 2003). An algorithm for automated mechanism design was proposed in the context of bartering problems (Conitzer and Sandholm, 2004). Only bilateral exchanges are considered without side payments. Two agents are considered with up to 90 types, but the set of possible allocations is restricted to only 30 outcomes. However, the complexity of determining an optimal mechanism grows exponentially with the number of agents involved in the transactions. The efficiency of such a mechanism is also discussed (Jameson et al, 2003).

The determination of acceptable multilateral transactions is modeled in a context of automated mechanism designs. Agents can behave according to different acceptability criteria. Depending on which acceptability notion is considered, the constraints of the model change, as described in the next section.

5.3 Problem statement

This section describes the model that we propose to determine acceptable multilateral transactions. This model is adapted to the automated mechanism design problem.

The definition of agent and utility function must be adapted to automated mechanism design problems. Let us note o the outcome of such problems and $\mathcal{O}$ the set of all possible outcomes. An outcome can be thought as a restricted resource allocation problem. Instead of considering the whole population $\mathcal{P}$, only the agents involved in the multilateral transaction δ_i^{Δ} are considered, $\mathcal{P}' = \Delta^{\delta} \cup \{i\}$, and the set of available resources is restricted to the bundle of the involved agents, $\mathcal{R}' = \bigcup_{i \in \mathcal{P}'} \mathcal{R}_i$. Each agent i is defined by means of:

- A finite set Θ_i of t_i types, where each element $\theta_i \in \Theta_i$ defines preferences that indicate which outcome is preferred in a pair of outcomes;
- A probability distribution P_i on Θ_i: p_i^k is the probability that agent i reports the type θ_i^k;
- An utility function $u_i : \Theta_i \times \mathcal{O} \to \mathbb{R}$ that allows the agent to evaluate an outcome.

The state of the multi-agent system is described by a type profile $\theta = (\theta_1, \ldots, \theta_{n_\delta})$, i.e., a vector composed by the types reported by the n_δ involved agents into the current multilateral transaction δ_i^Δ: θ_j is the type reported by the agent $j \in \mathcal{P}' = \Delta^\delta \cup \{i\}$. The set of possible type profiles is denoted by $\Theta \in \Theta_1 \times \cdots \times \Theta_{n_\delta}$. The optimization problem is defined over the variables g_θ^k where a specific g_θ^k is the probability for the mechanism of choosing the outcome $o_k \in O$ when the types reported by the agents correspond to the type profile θ. A slight abuse of notation is used here, because θ corresponds to a type profile, it is a little improper to use it as an index for the variables.

The type of the mechanism characterizes the design of the mechanism:

- A deterministic mechanism always returns the same outcome. It uses Boolean variables for a specific type profile: $g_\theta^k \in \{0, 1\}$;
- A randomized mechanism returns a probability distribution per type profile. Thus, it uses real variables: $g_\theta^k \in [0, 1]$.

Since all involved agents have an acceptability criterion to satisfy, a set of *acceptability constraints* must be added to the model. When agents are rational, individual rationality constraints are introduced in the model. Some variants of this notion are used in the literature (Mas-Colell et al, 1995).

- **Ex post**: Each agent has to obtain equal or greater utility than initially, after he knows the outcome chosen by the mechanism. If there exists an outcome o_k which gives less than the initial utility of at least one agent for a given type profile θ, the constraints lead the variable g_θ^k to 0. Hence, it is impossible for the designer to return such a mechanism. The constraints are defined as follows:

$$\exists \theta \in \Theta, u_i(\theta, o_k) < u_i(\theta, o_{\text{INIT}}) \quad \Rightarrow \quad g_\theta^k = 0, \qquad i \in \mathcal{P}, o \in O.$$

- **Interim**: Before knowing the outcome chosen by the mechanism, the expected utility of the agents are greater or equal on average than their initial utility. This interim scenario incites the agents to participate in the transaction. However, these constraints do not guarantee the achievement of a greater utility. Indeed, the constraints are about expected utility, i.e., the average utility obtained over the type profile. In most cases, the agents obtain more than initially, but some cases may occur in which they obtain less. There is a part of uncertainty. With this concept, the constraints are defined as follows:

$$\sum_{o_k \in O} u_i(\theta, o_k) g_\theta^k \geq u_i(\theta, o_{\text{INIT}}), \qquad i \in \mathcal{P}', \theta \in \Theta.$$

When agents are not rational but social, sociability constraints must be added to the model. Since such an acceptability criterion must be interpreted according to the social welfare notion considered, the expression of acceptability constraints also depends on it. Such acceptability constraints must ensure that the social welfare value increases with the current multilateral transaction. When the utilitarian welfare is considered, the summation of the utilities obtained by all involved agents must increase:

$$\sum_{i \in \mathcal{P}'} \sum_{o_k \in O} u_i(\theta, o_k) g_\theta^k \geq \sum_{i \in \mathcal{P}'} u_i(\theta, o_{\text{INIT}}), \qquad \theta \in \Theta.$$

When the egalitarian welfare notion is considered, the poorest agent after the transaction must be richer than the poorest agent before:

$$\min_{i \in \mathcal{P}'} \sum_{o_k \in O} u_i(\theta, o_k) g_\theta^k \geq \min_{i \in \mathcal{P}'} u_i(\theta, o_{\text{INIT}}), \qquad \theta \in \Theta.$$

Agents have a set of types that they can report to others. Only one "true" type, denoted $\hat{\theta}_i$, from this set corresponds to the true preference of agent $i \in \mathcal{P}'$. However, in the literature, it is assumed that an agent can misreport its type in order to manipulate the mechanism and to try to achieve a greater utility. Consequently, a set of *incentive compatibility constraints* must be added to the model. According to the kind of equilibrium required among the involved agents, the expression of the constraints may change. For instance, a *Bayes-Nash equilibrium* can be used, meaning that reporting truthful information gives to the agents at least equal or greater utility, assuming other agents report truthful information too. Let $\hat{\theta}$ be the truthful type profile, i.e., in which all agents report truthful information. The constraints can then be written as follows:

$$\sum_{o_k \in O} u_i(\hat{\theta}, o_k) g^k_{\hat{\theta}} \geq \sum_{o_k \in O} u_i(\hat{\theta}, o_k) g^k_{\theta} \qquad \hat{\theta}, \theta \in \Theta, i \in \mathcal{P}'.$$

Another possible equilibrium that can be used would be the *dominant-strategy equilibrium* where reporting truthful information gives to agents at least an equal or a greater utility even if the other agents misreport their types. The constraints for a dominant-strategy equilibrium are stronger than the constraints for a Bayes-Nash equilibrium. In the case of a dominant-strategy equilibrium, the constraints are defined as follows:

$$\sum_{o_k \in O} u_i(\theta_{-i}, \hat{\theta}_i, o_k) g^k_{\hat{\theta}} \geq \sum_{o_k \in O} u_i(\theta_{-i}, \hat{\theta}_i, o_k) g^k_{\theta} \qquad \hat{\theta}, \theta \in \Theta, i \in \mathcal{P}'.$$

where θ_{-i} is an incomplete type profile, without the type reported by agent i. Hence, $u_i(\theta'_{-i} o_k)$ corresponds to the evaluation by agent i of the outcome o_k, when agent i reports truthful information $\hat{\theta}_i$, and other agents report any information θ_{-i} (either truthful or wrong types).

Finally, the objective function must be defined. A social welfare function can be used here, or it is possible to define a specific objective for the initiator. For instance, in a rational society, an initiator may want to maximize it personal profit. In such a case, the objective function can be written as follows:

$$u_i = \sum_{\theta \in \Theta} \sum_{o_k \in O} u_i(\theta, o_k) g^k_{\theta}.$$

When the objective function is either the maximization of the utilitarian welfare notion, or the maximization of the egalitarian welfare, the objective function can respectively be written as follows:

$$sw_u = \sum_{\theta \in \Theta} \left[\sum_{i \in \mathcal{P}} \sum_{o_k \in O} u_i(\theta, o_k) \right] g_\theta^k;$$

$$sw_e = \sum_{\theta \in \Theta} \left[\min_{i \in \mathcal{P}} \sum_{o_k \in O} u_i(\theta, o_k) \right] g_\theta^k.$$

5.4 Introduction to column generation methods

This section is dedicated to the description of a method corresponding to the determination of acceptable multilateral transactions depending on the considered parameters. Unfortunately, the translation into an optimization formulation generates a huge number of variables, even for moderate problem sizes. Efficient solving methods must be used in order to ensure the scalability.

Column generation (Chvátal, 1983) is a way to begin with a small and manageable part of a problem (only few of the variables), and solving this part. The analysis of the corresponding partial solution helps to determine the next part of the problem (one or several variables) to add to the model. This enlarged model can then be solved. Column-wise modeling repeats such a process until it achieves a satisfactory solution to the whole problem.

In formal terms, column generation techniques are solution methods for linear programs with a very large number of variables (e.g. exponential) where constraints can be implicitly expressed with respect to the variables. They are widely used for solving large scale integer programs whose solutions schemes rely on linear programming relaxations. Their general principle is described in the sequel.

5.4.1 The master problem

Let us consider an integer linear program with a huge number of variables, say n, and m constraints such that the relationship $m \ll n$ is satisfied. The optimal solution of this system is denoted by $z^\star$.

$$z^\star = \begin{cases} \min cx \\ \text{subject to:} \quad Ax \geq b \\ \qquad\qquad\qquad x \in \mathbb{Z}^{+n}. \end{cases} \tag{5.1}$$

The classic method to solve integer linear programs corresponds to the so-called *branch-and-bound*. For minimization problems, relaxations are used in order to get lower bounds on the optimal integer solution. The most used relaxation is the linear relaxation, denoted by (5.2), that corresponds to the solution of the following linear program:

$$z^\star_{\text{LP}} = \begin{cases} \min cx \\ \text{subject to:} \quad Ax \geq b \\ \qquad\qquad\qquad x \geq 0. \end{cases} \tag{5.2}$$

The variables x of the linear relaxation (5.2) corresponds to the columns of the matrix A. Thus, variables and columns are used indifferently in the sequel. The optimal solution of (5.2), which can be determined using either the simplex algorithm or the column generation technique, provides a lower bound. Indeed, the following relationship is satisfied:

$$z^\star_{\text{LP}} = \underline{z} \leq z^\star.$$

Nowadays, linear programs are well solved in practice using effective implementations of the simplex algorithm, for example using the CPLEX software (ILOG Inc, 1995). However, when the number of variables is very large, it is better to use column generation methods.

Column generation methods rely on a decomposition of the initial linear program into a restricted linear program and a pricing problem. The restricted linear program is often called *master problem*. It corresponds to a linear program associated with a restricted matrix A', which is a sub-matrix of A. Such a decomposition is illustrated in Figure 5.3. Here is its guiding principle: if an optimal solution of the restricted linear program is identified, this solution may also be optimal for the initial linear program. It can be tested thanks to the signs of the reduced costs of the missing columns, i.e., those are not explicitly considered in the linear relaxation (5.2).

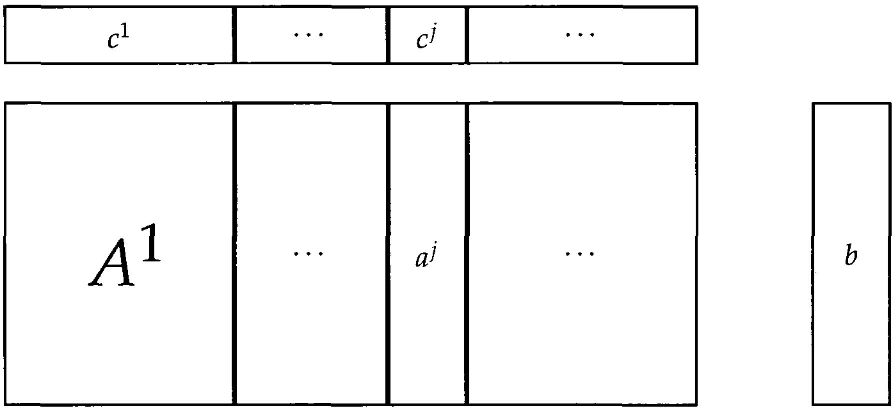


Figure 5.3: Matrix decomposition of a linear program

Column generation techniques proceed as follows: we consider a sub-matrix of dimension $m \times n_1$, from the original matrix, say $A^1 = A'$. For this sub-matrix, we solve the corresponding linear program, and obtain an optimal solution, denoted by x^1_{LP}. Remember that c^1 is a sub-vector of c, which corresponds to the columns of A^1.

$$\begin{cases} \min c^1 x & \\ \text{subject to:} \quad & A^1 x \geq b \\ & 0 \leq x \leq 1 \\ & x \in \mathbb{R}^{n_1}. \end{cases} \quad \overset{\text{CPLEX}}{\longrightarrow} \quad x^1_{\text{LP}} \tag{5.3}$$

A question should then be raised: "Is there a column $a^j \in A \setminus A^1$ such as $\bar{c}_j < 0$?" In other words, can we find a column of the matrix $A \setminus A^1$ for which the reduced cost is negative? There are two possible answers:

- NO: The optimal solution x^1_{LP} of the system (5.3) is optimal for the initial program (5.2).

- YES: Some columns must be added to the matrix A^1 in order to find the optimal solution. In that case, the following system must be solved:

$$\begin{cases} \min c^2 x & \\ \text{subject to:} \quad A^2 x \geq b & \\ \qquad\qquad\quad 0 \leq x \leq 1 & \\ \qquad\qquad\quad x \in \mathbb{R}^{n_2}, & \end{cases} \quad \xrightarrow{\text{CPLEX}} \quad x^2_{\text{LP}} \tag{5.4}$$

  with $A^2 = A^1 \cup \{a^j\}$ such that $\bar{c}_j < 0$.

The process is iterated as long as we are able to find j such as $\bar{c}_j \leq 0$, i.e., a column for which the reduced cost is negative. When iterations are no more possible, the optimal solution for the initial problem (5.2) is achieved.

The column generation technique needs either a feasible solution or an artificial solution in order to start. An artificial solution can be generated in insignificant time: the artificial solution corresponds to a set of columns, which constitute a square matrix with dimension equal to the number of constraints. That number is tiny compared to the number of variables. Moreover, in practice, it is more efficient to add a small set of columns than add columns one by one. Let one important remark be specified: in order to minimize the objective function, the algorithm needs a negative reduced cost, but if the aim is to maximize the objective function, the reduced cost that the algorithm looks for, is then positive.

It remains the following question: "how to determine if whether or not there exists a column with a negative reduced cost?" To answer this question, the so-called *pricing problem* must be solved.

5.4.2 The pricing problem

Consider the current master problem, e.g. equation (5.3). In order to improve the current solution of the master problem, a column a^j associated with a negative reduced cost is sought. For that purpose, the pricing problem has to be solved according to:

$$\begin{cases} \min \bar{c}(a^j) \\ \text{with constraints on the elements of } a^j \text{ to guarantee that } A^2 \subseteq A, \end{cases} \tag{5.5}$$

where $A^2 = (A^1|a^j)$. Indeed, A^2 is the concatenation of A^1 and a^j, i.e. the concatenation of the matrix previously considered in the master problem and the new column.

For the master problem, the reduced cost is defined in the matrix form as follows: $\bar{c} = c - vA$ where v is the vector of the dual variables associated with (5.2). Hence, for column a^j, the expression of the reduced cost becomes:

$$\bar{c}_j = c_j - v^1 a^j = c_j(a^j) - v^1 \cdot a^j, \tag{5.6}$$

where $\bar{c}_j$ is the reduced cost associated with column a^j, and v^1 is the optimal dual vector that is obtained when solving (5.3).

The variables of the pricing problem (5.5) correspond to the m components of vector a^j. The constraints on a^j come from the definition of the constraint matrix of the master problem, which must be a sub-matrix of A.

Solving the pricing problem is equivalent to solving the following problem:

$$\bar{c}^\star = \min\{\bar{c}(a) = c(a) - va : a \in \mathtt{A}\},$$

where $\mathtt{A} = \{a \in \mathbb{R}^m : (A^1|a) \text{ is a sub-matrix of } A\}$.

The pricing problem (5.5) is very often a combinatorial problem which is difficult to solve as it is often $\mathcal{NP}$-complete. However, it is not necessary to solve it exactly at each iteration. Indeed, it is enough to design a heuristic and to use it in combination with an exact algorithm as described in the sequel.

If the heuristic returns $\tilde{c}$ such as $\tilde{c} < 0$, the production of an optimal solution is not important because an improvement of the current solution is possible: the associated column is added to the current constraint matrix and a new iteration is carried out.

However, if the heuristic returns $\tilde{c}$ such as $\tilde{c} \geq 0$, the exact algorithm is called in order to check the sign of the optimal solution of (5.5), denoted by $\bar{c}^{\star}$:

1. $\bar{c}^{\star} = 0$: there is no missing column with a negative reduced cost. Hence the optimal solution of the current master problem is indeed the optimal solution of the initial LP-relaxation.

2. $\bar{c}^{\star} < 0$: an iteration has to be carried out.

The optimal value of the reduced cost $\bar{c}^{\star}$ provided by the exact algorithm cannot be positive. Indeed, in the case where the optimal solution has been identified, the lowest reduced cost is then associated with a column already added to the constraint matrix of the master problem. In this way, the associated reduced cost is zero.

Hence, the heuristic searches after a column with a negative reduced cost, which means that the optimal solution for the pricing problem has not been found. As long as such columns exist, iterations have to be performed. If a positive reduced cost is returned, the exact algorithm is used. Then, if the provided exact solution is still positive, it does not exist a column with a negative reduced cost, and the optimal solution of the current master problem is an optimal solution of the initial LP-relaxation, whereas if the returned exact solution is negative, improvements can be done thanks to additional iterations.

Figure 5.4 summarizes the solving process of column generation techniques. The dotted lines represent the different stages, including the pricing problem.

5.5 Expression of the pricing problem

Three kinds of constraints are required in the model: probability constraints ensuring the consistency, rationality constraints to check the agents' savings, and finally incentive compatibility constraints to prevent the misreport of information. Let $A = (A^{\text{PROBA}}, A^{\text{IR}}, A^{\text{IC}})$ be the constraint matrix of the current master problem. It consists of 3 sets of constraints.

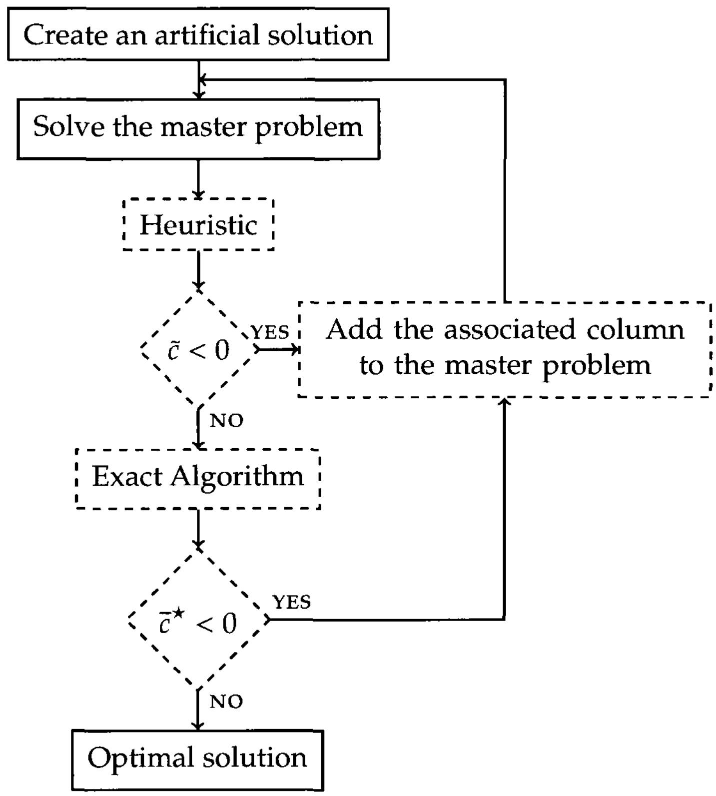


Figure 5.4: Column generation solving process

Firstly, the sub-matrix PROBA comes from the constraints linked with the probability distribution. Secondly the IR sub-matrix comes from the *individual rationality constraints* and, thirdly, the IC sub-matrix comes from the *incentive compatibility constraints*.

Let us denote ξ a generic column of the constraint matrix, which is associated with a variable g_θ^k and composed by various coefficients in each part. The dual vector can be decomposed in a similar way:

$$\xi = \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} \qquad v = \begin{pmatrix} v^{\text{PROBA}} \\ v^{\text{IR}} \\ v^{\text{IC}} \end{pmatrix}.$$

Then the formula of the reduced cost associated with a variable g_θ^k becomes:

$$\bar{c}_{\theta,k} = c_{\theta,k} - v^{\text{PROBA}} \cdot \alpha^{\theta,k} - v^{\text{IR}} \cdot \beta^{\theta,k} - v^{\text{IC}} \cdot \gamma^{\theta,k}.$$

Let us introduce the following x_k and y_θ such as:

$$x_k = \begin{cases} 1 & \text{if } o_k \text{ is selected} \\ 0 & \text{otherwise.} \end{cases}$$

$$y_\theta = \begin{cases} 1 & \text{if } \theta \text{ is the type profile reported} \\ 0 & \text{otherwise.} \end{cases}$$

Consider the first part of the column:

$$(\xi)_{\text{PROBA}}^{\theta,k} = \alpha^{\theta,k}.$$

There are as many constraints related to the probability part as the number of type profiles,

$|\Theta|$. Hence, one index is enough to identify the element of that column part.

$$\alpha_{\theta'}^{\theta,k} = \begin{cases} 1 & \text{if } \theta = \theta' \\ 0 & \text{otherwise.} \end{cases}$$

Hence, the expression of the probability part of the inner product becomes:

$$\begin{aligned} v^{\text{PROBA}} \cdot \alpha^{\theta,k} &= \sum_{\theta' \in \Theta} v_{\theta'}^{\text{PROBA}} \alpha_{\theta'}^{\theta,k} \\ &= v_{\theta}^{\text{PROBA}}. \end{aligned}$$

Then, let us consider the second part of the column:

$$(\xi)_{\text{IR}}^{\theta,k} = \beta^{\theta,k}.$$

The number of individual rationality constraints is $|\Theta| \times n$, the number of type profiles times the number of involved agents. Hence, two indices are required for element identification in this part of column.

$$\beta_{i,\theta'}^{\theta,k} = \begin{cases} u_i(\theta, o_k) & \text{if } \theta = \theta' \\ 0 & \text{otherwise} \end{cases}.$$

Hence, the inner product related to this set of constraints can be expressed as:

$$\begin{aligned} v^{\text{IR}} \cdot \beta^{\theta,k} &= \sum_{i \in \mathcal{P}} \sum_{\theta' \in \Theta} v_{i,\theta'}^{\text{IR}} \beta_{i,\theta'}^{\theta,k} \\ &= \sum_{i \in \mathcal{P}} v_{i,\theta'}^{\text{IR}} \beta_{i,\theta}^{\theta,k} \\ &= \sum_{i \in \mathcal{P}} u_i(\theta, o_k) v_{i,\theta}^{\text{IR}}. \end{aligned}$$

Finally consider the last part of the column, the one related to the incentive compatibility constraints.

$$(\xi)_{\text{IC}}^{\theta,k} = \gamma^{\theta,k}.$$

The number of constraints is $(|\Theta| - 1) \times |\Theta| \times n$. In this case, the elements of this part of a column have to be identified by three indices.

$$\gamma_{i,\theta',\theta''}^{\theta,k} = \begin{cases} u_i(\theta, o_k) & \text{if } \theta = \theta' \\ -u_i(\theta', o_k) & \text{if } \theta = \theta'' \\ 0 & \text{otherwise.} \end{cases}$$

Hence, that part of the inner product becomes:

$$\begin{aligned} v^{\text{IC}} \cdot \gamma^{\theta,k} &= \sum_{i \in \mathcal{P}} \sum_{\theta' \in \Theta} \sum_{\substack{\theta'' \in \Theta \\ \theta'' \neq \theta'}} v_{i,\theta',\theta''}^{\text{IC}} \gamma_{i,\theta',\theta''}^{\theta,k} \\ &= \sum_{i \in \mathcal{P}} \Big[\sum_{\substack{\theta' \in \Theta \\ \theta' \neq \theta}} v_{i,\theta,\theta'}^{\text{IC}} \gamma_{i,\theta,\theta'}^{\theta,k} + \sum_{\substack{\theta' \in \Theta \\ \theta' \neq \theta}} v_{i,\theta',\theta}^{\text{IC}} \gamma_{i,\theta',\theta}^{\theta,k} \Big] \\ &= \sum_{i \in \mathcal{P}} \sum_{\substack{\theta' \in \Theta \\ \theta' \neq \theta}} \Big[v_{i,\theta,\theta'}^{\text{IC}} u_i(\theta, o_k) - v_{i,\theta',\theta}^{\text{IC}} u_i(\theta', o_k) \Big]. \end{aligned}$$

The final expression of the reduced cost is based on the summation of the three parts according to the analytic formula (5.6). Using the expressions of the previous inner products $v^{\text{PROBA}} \cdot \alpha^{\theta,k}$, $v^{\text{IR}} \cdot \beta^{\theta,k}$ and $v^{\text{IC}} \cdot \gamma^{\theta,k}$ and the variables introduced, an expression of the reduced cost associated with variable g_θ^k can be deduced:

$$\bar{c}_{\theta,k} = x_k y_\theta \Bigg(c_{\theta,k} - v_\theta^{\text{PROBA}} + \sum_{i \in \mathcal{P}} \Big[u_i(\theta, o_k) \Big(v_{i,\theta}^{\text{IR}} + \sum_{\substack{\theta' \in \Theta \\ \theta' \neq \theta}} v_{i,\theta,\theta'}^{\text{IC}} \Big) - \sum_{\substack{\theta' \in \Theta \\ \theta' \neq \theta}} u_i(\theta', o_k) v_{i,\theta',\theta}^{\text{IC}} \Big] \Bigg).$$

Constraints have to be added in order to ensure that the new generated column properly defines the constraints of the master problem: the mechanism has to return a probability distribution over the set of possible outcomes O, and each agent is able to report only one type as its true type.

$$\sum_{o_k \in O} x_k = 1;$$

$$\sum_{\theta \in \Theta} y_\theta = 1.$$

5.6 Experimental evaluations

Two main facets can be evaluated in the context of resource allocation problems. The first one corresponds to an evaluation carried out from the agent's point of view, which is related to the cost and the efficiency of identifying an acceptable multilateral transaction in the agent's neighborhood (either one-to-many or many-to-many). The second facet corresponds to an evaluation carried out from the population's point of view, which is related to the efficiency of negotiation processes, especially on restricted social graphs. This second evaluation determines whether or not multilateral transactions significantly improve the efficiency of negotiation processes compared to their costs.

5.6.1 Evaluation of multilateral transactions

Let us consider agent $i \in \mathcal{P}$, who initiates a multilateral negotiation in its neighborhood. It gathers the information that it needs to design the suitable optimization model. The growth of the number of variables (g^k_θ, $\theta \in \Theta, o_k \in O$) according to the neighborhood size is described in Table 5.4. These experiments are characterized by the number of neighbors of the initiator i. All agents have only one type. For instance, when the agent initiator has 3 neighbors involved in the multilateral transaction, and when all of them have 3 resources in their bundle, the optimization model has 3969 variables.

Table 5.5 shows the growth of the number of variables (g^k_θ, $\theta \in \Theta, o_k \in O$) in the optimization model according to the number of types per agent. These experiments are characterized by the number of types per agent and the number of resources per agent. The initiator i involves 3 neighbors in its multilateral transaction δ_i^Δ. For instance, when

Table 5.4: Impact of the neighborhood size on the number of variables explicitly considered

Neighborhood Size	Number of Resources per Agent			
	1	2	3	4
1	1	9	49	225
2	2	36	686	11700
3	3	81	3969	236925
4	4	144	11956	2093400

each agent has 3 different types and 3 resources in its resource bundle, the optimization model contains 107163 variables.

Table 5.5: Impact of the number of types per agent on the number of variables explicitly considered

Number of types per agent	Number of Resources per Agent			
	1	2	3	4
1	3	81	3969	236925
2	24	648	31752	1875400
3	81	2214	107163	6396975
4	192	5184	254016	15163200

The neighborhood sizes and the number of types per agent remain relatively small, but they are sufficient to show the evolution of the problem complexity. The main factor of the growth of the complexity is the number of agents involved in the negotiation: the number of possible exchanges increases exponentially with respect to the number of agents. The number of types per agent has only a secondary impact on the complexity: the number of type profiles increases exponentially with the number of agents and linearly with the possible number of types per agent.

Experiments of this section evaluate the identification of a multilateral transaction, involving different number of participants with resource bundle of different sizes. The different curves on the following figures represents different population sizes.

Figure 5.5 shows the rate of variables explicitly considered by the column generation method, i.e., the variables that have been added to the master problem. This parameter is important since the main drawbacks of automated mechanism design is its huge number of variables. Hence, any method that uses a small percentage of variables helps to apply the automated mechanism design to larger instances, which are relevant in practice.

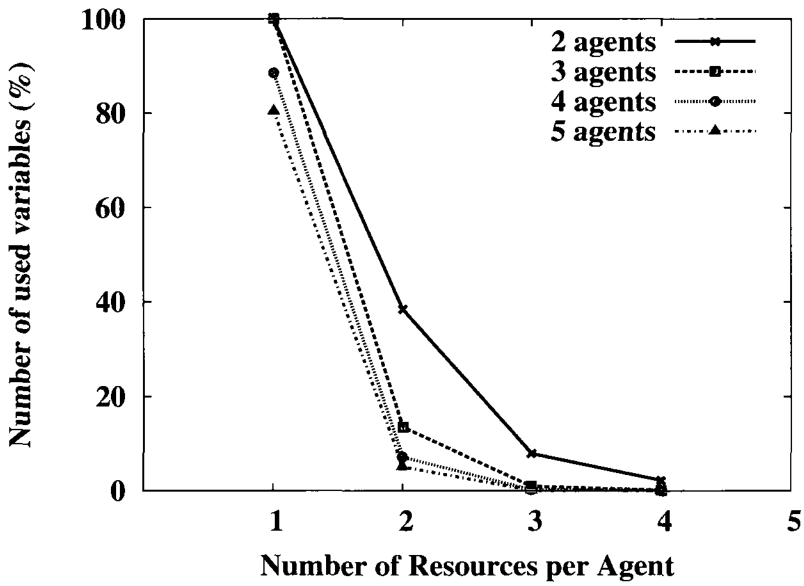
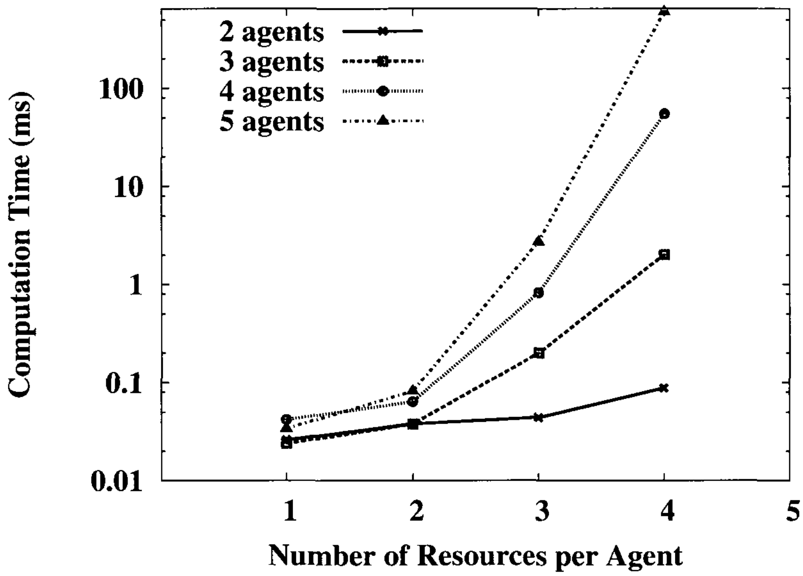


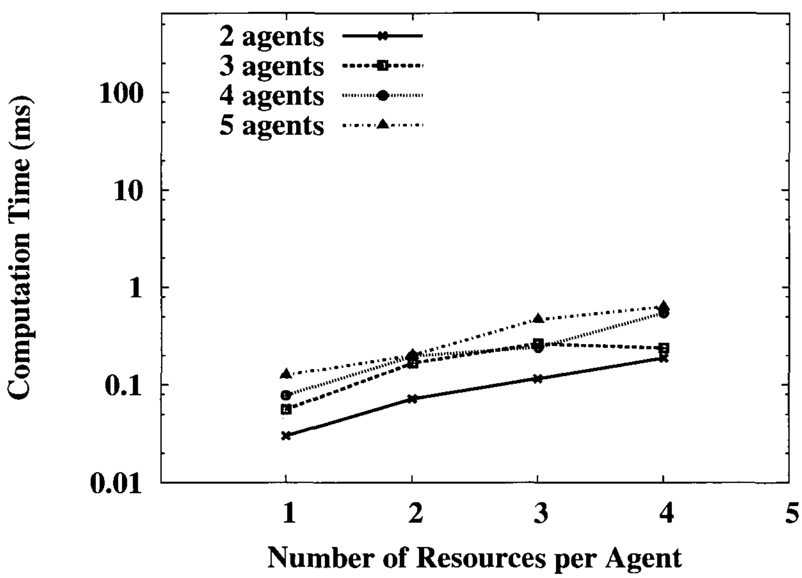
Figure 5.5: Number of variables (%) explicitly that are considered during the solving process

Small instances (e.g., with 2 agents) have a higher percentage of considered variables in consequence of the low number of variables. Instances with only one resource per agent are not meaningful. Indeed, in order to efficiently use the column generation techniques, the constraint matrix must have a specific form: One dimension has to be much smaller than the other ones. However, if the instances are too small, e.g., 2 agents with 1 or 2 resources each, the number of variables becomes similar to the number of constraints. Then, the percentage of considered variables becomes higher. Figure 5.5 shows that the percentage of considered variables drops to less than 1% of the variables that must be considered by the simplex algorithm. This small percentage represents the efficiency of column generation techniques, which can then solve instances quite larger within the same memory space.

Figures 5.6a and 5.6b summarize the computation time respectively for the simplex algorithm and the column generation technique, depending on the number of resources per agent and on the number of involved agents. Each agent has only one type. These figures show that the column generation technique becomes faster than the simplex al-

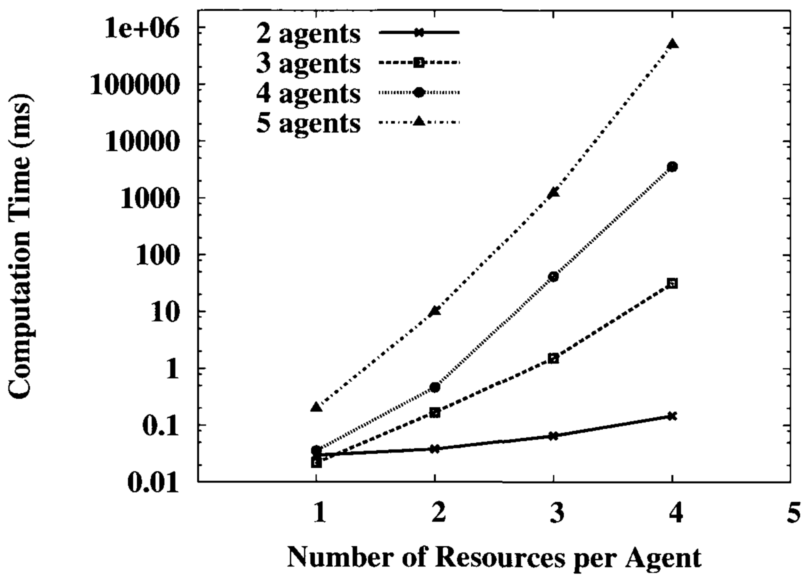


(a)

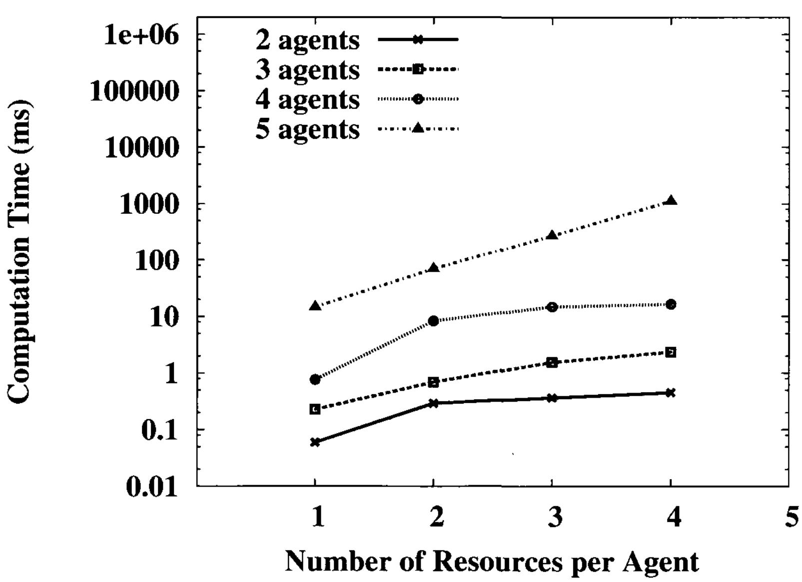


(b)

Figure 5.6: Comparison of the computation time that is required by the simplex algorithm in 5.6a and by the column generation techniques in 5.6b (each agent has only one type).



(a)



(b)

Figure 5.7: Comparison of the computation time required by the simplex algorithm in 5.7a and by the column generation techniques in 5.7b (each agent has two types).

gorithm beyond 2 resources per agent. The computation time of the simplex algorithm increases exponentially while it increases linearly associated with a weak slope in the case of column generation technique. Indeed, the number of agents has a significant impact on the number of variables and the column generation techniques are able to select the best variables to be explicitly considered. In these settings, since only one type per agent is allowed, the number of variables corresponds to the number of possible exchanges among the agents. This number of variables increases exponentially with the number of agents and the number of resources owned by each agent. It could explain the exponential growth of the computation time observed when the simplex algorithm is considered. The column generation technique, even with a straightforward approach to solve the pricing problem, is able to find the few variables associated with a reduced cost of the specified sign (respectively positive for a maximization problem and negative for a minimization problem). They correspond to the exchanges with a positive probability of being selected in the optimal solution.

Figures 5.7a and 5.7b summarize the computation time respectively for the simplex algorithm and the column generation technique, depending on the same features (the number of agents and the number of resources per agent) but when an agent can have 2 different types. The appearance of the curves remains similar as in the previous case of considering only one type per agent. The simplex algorithm computation time grows exponentially whereas the column generation technique results correspond to a linear growth associated with a weak slope. As previously, when the instances have a moderate size, the column generation approach is not really competitive. However, the advantages show up with larger instances with which the column generation approach is significantly better.

We described a scalable method to determine acceptable multilateral transactions. Such a method remains scalable even when large instances are used. It is also possible to bound the number of neighbors that can be involved in a multilateral negotiation in order to ensure the scalability of the approach. Since an efficient method exists, issues related to its effective use within a negotiation process must be addressed.

5.6.2 Evaluation of multilateral negotiation processes

This section is dedicated to the impact of multilateral transactions on the efficiency of negotiation processes. The efficiency of negotiations based on social many-to-many transactions are evaluated by means of a comparison with the global optimal welfare value. This efficiency is compared to the efficiency of negotiation processes based on $\langle 1, 0 \rangle$ transactions, which are the most efficient transactions with utilitarian societies. Experiments are based here on systems populated by 50 agents, who negotiate 250 resources. Similarly to experiments described in Chapter 4, agents negotiate sequentially, initiating the determination of an acceptable multilateral transaction involving its neighborhood.

Table 5.6 shows the efficiency achieved by different negotiation processes, the first ones based on $\langle 1, 0 \rangle$ transactions and the second ones based on many-to-many transactions. For instance, when the social graph that is considered can be represented by a grid, negotiation processes based on social gifts only achieve 86.2% of the optimal welfare value, with 0.9% of standard deviation, whereas negotiation processes based on social many-to-many transactions can achieve 94.7% of the optimal welfare value, with a standard deviation of 0.45%.

Table 5.6: Efficiency (%) of multilateral transactions on negotiation processes depending on the social graph topology

Social graph kind	Social $\delta\langle 1, 0 \rangle$		Social *mtm*	
Full	100	0.0	100	0.0
Grid	86.2	0.9	94.7	0.45
Erdős-Rényi - $p = 0.05$	83.1	1.36	94.3	0.76
Erdős-Rényi - $p = 0.1$	91.3	0.70	99.1	0.76
Erdős-Rényi - $p = 0.2$	95.4	0.38	99.9	0.02
Erdős-Rényi - $p = 0.3$	97.6	0.21	100.0	0.0
Erdős-Rényi - $p = 0.5$	98.9	0.12	100.0	0.0
Erdős-Rényi - $p = 1.0$	100	0.0	100.0	0.0
Small world	91.4	0.78	99.5	0.13

When full social graphs are considered, socially optimal resource allocations can be achieved in both cases. Since social gifts are sufficient to ensure the achievement of optimal allocations, and since many-to-many transactions contain gifts, negotiations based on these transactions lead to optimal solutions.

However, when the considered social graph is restricted, the efficiency of multilateral transactions is larger. When grids are considered, only 86.2% of the optimal welfare value can be achieved when negotiations are based on social gifts, while 94.7% can be achieved with multilateral transactions. Multilateral transactions favor a lot the resource traffic over the population, which limits the impact of the social graph topology.

5.7 Conclusion

Traditionally, negotiation among agents have been limited to the use of bilateral transactions. However, when restricted social graphs are considered, multilateral transactions can favor the resource circulation and then help in order to achieve socially more interesting resource allocations. The lack of scalable methods to efficiently determine acceptable multilateral transactions limits their use in practice. We propose a scalable method to determine acceptable multilateral transactions. These transactions improve the efficiency of negotiation processes, especially when restricted social graphs are considered. They favor the circulation of resources among agents, and hence allow solving processes to leave many local optima, but their use cannot guarantee the achievement of socially optimal allocations. Moreover, their use requires the relaxation of some assumptions that we made initially. Indeed, agents involved in a multilateral transactions must accept to reveal private information, such as their resource bundle or their preferences. This relaxation can be considered as a major drawback depending on the considered application, and consequently it may prevent their use.

Thus, the use of multilateral transactions during agent negotiations leads to socially more interesting allocations without guarantee to achieve optimal solutions. Even if they improve the provided solution, especially when the mean connectivity of the social graph is weak, the use of multilateral transactions requires to give up important assumptions related to the agents' autonomy. Depending on the considered application, the system designer will need to asses their interest.

Conclusion and Further Works

This chapter presents the conclusion of this thesis, coming back on the main issues. Our contributions are also presented with the main results we established. Finally, we describe the limits of this thesis throughout the presentation of interesting further works.

Context

Resource allocation problems have been studied for a long time, usually by means of centralized techniques, which are not well-adapted to variety of applications. Application characteristics like dynamism, privacy or restricted agent communications cannot be handled in scalable ways. These features are essential to many applications. In this thesis, we focus on distributed solving methods based on local agent negotiations. A solving process starts from an initial allocation that evolves step by step, thanks to local transactions between agents. In contrast to centralized techniques, agent negotiations ensure that the provided solutions can be achieved in practice, specifying transaction sequences leading to these solutions.

The objective of this thesis is to design a distributed mechanism based on local transactions leading agent negotiations to socially optimal allocations. In this purpose, we identify four important parameters that must be considered: the transactions, the interactions, the acceptability criteria and the social graphs. We identify the simplest kinds of transactions ensuring the efficiency of the negotiation processes. We also proposed a local acceptability criterion, allowing efficient negotiations among agents, even when restricted communication abilities are considered. We also introduce the notion of social graph, which represents the relationships between agents. In each case, we propose a negotiation setting ensuring the achievement of socially optimal allocations.

Contributions

We successively studied four notions of the social welfare theory in order to identify which agent's behavior can lead negotiation processes to socially optimal resource allocations. Generally, the individual rationality does not achieve socially optimal resource allocations. Indeed, this acceptability criterion is too restrictive, and a new one was proposed, which is based on the evolution of the social welfare value. It allows the achievement of socially interesting allocations. The main results and the efficient negotiation settings that we propose, are summarized in Table 5.8. Each column of this table corresponds to a specific welfare notions.

When the utilitarian problems are considered, social gifts are sufficient to guarantee the achievement of optimal solutions on complete social graphs (Nongaillard et al, 2008b,a). The number of performed transactions and the number of attempted transactions are both polynomial. Agents' behaviors should be frivolous and flexible in order to favor the resource circulation. However, when the social graph is restricted, the achievement of optimal solutions cannot be guaranteed any more. Nevertheless, the utilitarian notion is flexible enough to allow a minimal resource traffic ensuring the achievement of socially interesting allocations. When social graphs are restricted, the most efficient negotiation policy is still based on social gifts (Nongaillard et al, 2009b).

When the egalitarian welfare is considered, it is not possible to guarantee the achievement of optimal solutions using only bilateral transactions, even on complete social graphs (Nongaillard and Mathieu, 2009b,a). However, socially close allocations can be achieved using two transactions. Both social gifts and social swaps are required to ensure efficient egalitarian negotiations. Gifts allow to change the initial resource distribution, and swaps allow improvements on the egalitarian welfare value, which is not possible with gifts. Theoretically, large bilateral transactions are required since they may lead to egalitarian improvements. However their cost does not justify the slight improvement that they bring. Egalitarian negotiations are also sensitive to bottlenecks in the social graph topology. A too weak mean connectivity of agents (with only one neighbor for instance) restrict the resource circulation a lot and hence the efficiency of egalitarian negotiations. These negotiations are more time consuming than in the utilitarian case, but remain scalable even with large instances.

The Nash welfare is a notion for which it is only possible to roughly estimate the optimal welfare value using centralized heuristics (Nongaillard et al, 2009a). Similarly to the egalitarian case, two kinds of transactions are required to negotiate efficiently: Social gifts and social swaps. However, bilateral transactions cannot ensure the achievement of optimal allocations, and Nash negotiation processes may then end on local optima. Nevertheless, such negotiation processes achieve socially closer allocations than the heuristics that we designed. Frivolous and flexible agent behaviors favor the resource traffic and then help in the achievement of interesting allocations. The Nash welfare is more flexible than the egalitarian notion, and is consequently less sensitive to social graph restrictions, even if a very weak mean connectivity handicaps negotiations.

Finally, elitist societies have been studied. This welfare notion is very specific since it only considers the richest agent, neglecting all other agents. The rational acceptability criterion has no meaning here, and the most efficient transaction corresponds to an agent who offers its whole resource bundle. Indeed, any initiator has to give all its resource bundle to its partner if this one becomes richer than the initiator would become receiving the whole resource bundle of its partner. In other words, maximal clusters are the most suitable transactions to efficiently negotiate within elitist societies. When social graphs are complete, it is possible to guarantee the achievement of optimal solutions. However, it is no longer possible when restricted communication possibilities are considered.

Multilateral transactions have been presented with their advantages and drawbacks

(Asselin et al, 2006). Such transactions favor the resource circulation and hence help in the achievement of more interesting allocations. However, when the social graph is restricted, they cannot guarantee the achievement of socially optimal allocations. They are expensive and time consuming to determine, and some assumptions must be relaxed: no information privacy can be considered. All agents in the initiator's neighborhood must report their private information in order for it to determine whether or not an acceptable transaction exists. A scalable method based on an optimization technique has been described, and the efficiency of these transactions has been compared to bilateral ones. Especially when the mean connectivity is weak, multilateral transactions may improve the provided allocations up to 10%, but does this improvement justifies the sacrifice of the agents' autonomy? Only application designers can find an answer to this question. Indeed, depending on the application, the autonomy of the agent cannot be relaxed, and then multilateral transactions cannot be efficiently used.

Further works

In the future, we propose to study **different preference representations**. In this thesis, we studied the efficiency of agents' negotiations according to several welfare notions. However, we always assumed that agents express their preferences by means of additive utility functions. Such a representation is quite restrictive since agents cannot express dependencies among resources. For instance, agents may associate a larger utility value with a set of resources than the simple summation of the utility associated with each of them. Sometimes, resources from a set may have an interest if the agent owns all of them, while individually they have no real value. Other restrictions cannot be expressed, like exclusions: agents may be satisfied or own either a resource or another one, but not if they own both of them. Agents' preferences can be represented by means of weighted propositional formula (WPF), which is a fully expressive representation. Any synergy can be expressed using logic formulas. Each resource is represented using a propositional

	Social welfare notions			
	Utilitarian sw_u	Egalitarian sw_e	Nash sw_n	Elitist sw_n
Centralized Algorithm (on complete graph)	Trivial Allocation of each resource to one of the agents who estimates it the most	$\mathcal{NP}$-hard problem Estimation using linear program	$\mathcal{NP}$-hard problem Accurate estimation quite difficult	Trivial Allocation of all resources to the agents who estimates them the most
Distributed Algorithm (on any kind of graph)	Social criterion Gifts Frivolous and flexible	Social criterion Gifts and swaps Frivolous and flexible	Social criterion Gifts and swaps Frivolous and flexible	Social criterion Clusters Frivolous
Performance	Equivalent	Less memory and time-consuming	Far less time-consuming	More time-consuming
Characteristics	Optimal on complete graphs More than 86% for graph with a very weak connectivity	Bilateral transactions unsufficiency Sensitive to bottlenecks Requires a high mean connectivity	Bilateral transactions unsufficiency Requires a hight mean connectivity Sensitive to graph bottlenecks	Optimal on complete graphs Very scalable Sensitive to the mean connectivity Sensitive to the intial allocation

Figure 5.8: Summary

variable which is true if the agent owns the resource, and false otherwise. Numerical weights represent the relative importance of this formula.

We also propose to study **different models of generation for agents' preferences**. In this thesis, no relationship has been considered between neighbors and their preferences. Indeed, instances have been randomly generated. However, in applications related to social webs, neighbors can be considered as "friends", which have a higher probability to have similar preferences. Then, topologies of social networks are related to agents' preferences. Such an assumption has an important influence on the resource circulation and on the optimal solution. The resource circulation would be easier among agents who express close preferences, which may facilitate the achievement of socially optimal allocations.

Investigations on **different functions for the evaluation of the individual welfare** seem of interests. The welfare of agents depends only on the resources in their bundle. Such evaluation functions are called "free of externality". The agent welfare is independent of the welfare of others. However, in social networks, the notion of group should be considered. Agents belong to different groups or communities. The individual welfare of agents partially depends on the welfare of their group. Thus, agents may be satisfied if any agent of their group owns the resources they wished. Interesting simulations could be performed considering externalities among agents.

Dynamic environments should also be studied. We always assume that negotiations take place in static environments. Indeed, neither the agents' preferences, nor the social graph topology can vary. However, in practice, the utility value associated with a resource may decrease in time. Indeed, an agent may lose interest in some resources while it may express an increasing interest for other resources. As well, the topology of the social graph cannot vary in our approach. However, in dynamic systems, new agents constantly enter with their resources while other ones leave. Arrivals of new agents change the network topology, affecting the resource traffic. With new agents, resources that were blocked somewhere in the system may circulate again, leading the negotiation process to socially more interesting allocations. The dynamic facet of resource allocation problems seems to be of interest to us.

Bibliography

Albert R, Barabási A (2002) Statistical Mechanics of Complex Networks. Reviews of Modern Physics 74(1):47–97

Andersson M, Sandholm T (1998) Contract types for satisficing task allocation II: Experimental results. In: AAAI Spring Symposium Series: Satisficing Models, AAAI Press, USA, California, Stanford University, vol 17, pp 1–7

Andersson M, Sandholm T (2000) Contract Type Sequencing for Reallocative Negotiation. In: ICDCS'00 - Internation Conference on Distributed Computing Systems, IEEE Computer Society, Taiwan, Tapei, vol 0, pp 154–160

Anshelevich E, Dasgupta A, Tardos E, Wexler T (2008) Near-Optimal Network Design with Selfish Agents. Theory of Computing 4:77–109

Arrow K (1963) Social Choice and Individual Values. John Wiley and Sons

Arrow K, Sen A, Suzumura K (2002) Handbook of Social Choice and Welfare, vol 1. Elsevier

Asselin F, Jaumard B, Nongaillard A (2006) A Technique for Large Automated Mechanism Design Problems. In: Proceedings of the IEEE/WIC/ACM international conference on Intelligent Agent Technology (IAT'06), pp 467–473

Bellosta M, Kornman S, Vanderpooten D (2006) An Agent-Based Mechanism for Autonomous Multiple Criteria Auctions. In: IAT'06 - Intelligent Agent Technology, China, Hong-Kong, vol 0, pp 587–594

Berman F, Fox G, Hey A (2003) Grid Computing: Making the Global Infrastructure a Reality. Wiley

Biggs N, Lloyd E, Wilson R (1986) Graph theory. Oxford University Press, USA

Bollobás B (2001) Random Graphs. Cambridge University Press

Bonzon E, Lagasquie-Schiex M, Lang J, Zanuttini B (2009) Compact Preference Representation and Boolean Games. Autonomous Agents and Multi-Agent Systems 18(1):1–35

Boutilier C, Goldszmidt M, Sabata B (1999) Sequential Auctions for the Allocation of Resources with Complementarities. In: IJCAI'99 - International Joint Conference on Artificial Intelligence, Morgan Kaufmann, Sweden, Stockolm, vol 16, pp 527–534

Bouveret S, Lang J (2005) Efficiency and Envy-Freeness in Fair Division of Indivisible Goods: Logical Representation and Complexity. In: IJCAI'05 - International Joint Conference on Artificial Intelligence, UK, Scotland, Edinburgh, pp 935–940

Bouveret S, Endriss U, Lang J (2009) Conditional importance networks: A graphical language for representing ordinal, monotonic preferences over sets of goods. In: IJCAI'09 - International Joint Conference on Artificial Intelligence, USA, California, Passadena, pp 67–72

Brams S, Taylor A (1996) Fair Division: From Cake-Cutting to Dispute Resolution. Cambridge University Press

Buyya R, Abramson D, Giddy J, Stockinger H (2002) Economic Models for Resource Management and Scheduling in Grid Computing. Concurrency and Computation: Practice and Experience 14(13-15):1507–1542

Capera D, Georgé J, Gleizes M, Glize P (2003) Emergence of Organisations, Emergence of Functions. In: AISB'03 - Symposium on Adaptive Agents and Multi-Agent Systems, Society for the Study of Artificial Intelligence and the Simulation of Behaviour, UK, Aberystwyth, University of Wales, pp 103–108

Chakraborty D, Joshi A, Yesha Y, Finin T (2006) Toward Distributed Service Discovery in Pervasive Computing Environments. IEEE Transactions on Mobile Computing 5(2):97–112

Chavez A, Moukas A, Maes P (1997) A Multi-agent System for Distributed Resource Allocation. In: Agents'97 - Autonomous Agents, ACM Press, pp 323–331

Chevaleyre Y, Endriss U, Lang J, Maudet N (2005) Negotiating over Small Bundles of Resources. In: AAMAS'05 - Autonomous Agents and multiagent systems, ACM Press, EU, The Netherlands, Utrecht, pp 296–302

Chevaleyre Y, Dunne P, Endriss U, Lang J, Lemaitre M, Maudet N, Padget J, Phelps S, Rodriguez-Aguilar J, Sousa P (2006a) Issues in Multiagent Resource Allocation. Informatica 30(1):3

Chevaleyre Y, Endriss U, Lang J (2006b) Expressive Power of Weighted Propositional Formulas for Cardinal Preference Modelling. In: KR'06 - Principles of Knowledge Representation and Reasoning, AAAI Press, UK, Lake District, pp 145–152

Chevaleyre Y, Endriss E, Estivie S, Maudet N (2007) Reaching Envy-free States in Distributed Negotiation Settings. In: IJCAI'07 - International Joint Conference on Artificial Intelligence, AAAI Press, India, Hyderabad, pp 1239–1244

Chevaleyre Y, Endriss U, Maudet N (2009) Simple Negotiation Schemes for Agents with Simple Preferences: Sufficiency, Necessity and Maximality. Journal of Autonomous Agents and Multiagent Systems

Christodoulou G, Koutsoupias E (2005) The Price of Anarchy of Finite Congestion Games. In: STOC'05 - ACM symposium on Theory of Computing, ACM Press, USA, Baltimore, pp 67–73

Chun B, Fonseca R, Stoica I, Kubiatowicz J (2004) Characterizing Selfishly Constructed

Overlay Routing Networks. In: INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, vol 2

Chvátal V (1983) Linear Programming. WH Freeman

Clearwater S (1996) Market-Based Control: A Paradigm for Distributed Resource Allocation. World Scientific

Conitzer V, Sandholm T (2002) Complexity of Mechanism Design. In: UAI'02 - Uncertainty in Artificial Intelligence, University of Alberta, Edmonton, Alberta, Canada, Morgan Kauffman, vol 1, pp 103–110

Conitzer V, Sandholm T (2004) An Algorithm for Automatically Designing Deterministic Mechanisms Without Payments. In: AAMAS'04 - Autonomous Agents and MultiAgent Systems, USA, New York, IEEE Press, pp 128–135

Cormas (2001) http://cormas.cirad.fr

Corning P (2002) The Re-Emergence of "Emergence": A Venerable Concept in Search of a Theory. Complexity 7(6):18–30

Coste-Marquis S, Lang J, Liberatore P, Marquis P (2004) Expressive Power and Succinctness of Propositional Languages for Preference Representation. In: KR'04 - Principles of Knowledge Representation and Reasoning, Canada, Whistler, pp 203–212

Cramton P, Shoham Y, Steinberg R (2006) Combinatorial auctions. MIT Press

Dash R, Jennings N, Parkes D (2003) Computational-Mechanism Design: A Call to Arms. IEEE Intelligent Systems 18(6):40–47

Dash R, Ramchurn S, Jennings N (2004) Trust-based Mechanism Design. In: AAMS'04 - Autonomous Agents and MultiAgent Systems, USA, NY, Columbia University

Dash R, Rogers A, Jennings N, Reece S, Roberts S (2005) Constrained Bandwidth Allocation in Multi-sensor Information Fusion: A Mechanism Design Approach. In: Information Fusion, Philadelphia, PA, USA, IEEE Press, vol 2

De Vries S, Vohra R (2003) Combinatorial auctions: A survey. INFORMS - Journal on Computing 15(3):284–309

Deshpande M, Venkatasubramanian N (2004) The Different Dimensions of Dynamicity. In: P2P'04 - Peer-to-Peer Computing, IEEE Computer Society, Switzerland, Zurich, vol 0, pp 244–251

Di Marzo Serugendo G, Gleizes M, Karageorgos A (2006) Self-Organisation and Emergence in Multi-Agent Systems: An Overview. Informatica 30(1):45–54

Doyle J (2004) Prospects for preferences. Computational Intelligence 20(2):111–136

Dunne P (2005) Extremal behaviour in multiagent contract negotiation. Journal of Artificial Intelligence Research 23:41–78

Endriss U, Maudet N (2005) On the Communication Complexity of Multilateral Trading: Extended Report. Autonomous Agents and Multi-Agent Systems 11:01–107

Endriss U, Maudet N, Sadri F, Toni F (2006) Negotiating Socially Optimal Allocations of Resources. Journal of Artificial Intelligence Research 25:315–348

Erdős P, Rényi A (1959) On Random Graphs. Publicationes Mathematicae 6:290–297

Estivie E (2006) Allocation de Resources Multi-Agents: Théorie et Pratique. PhD thesis, Paris Dauphine University

Feigenbaum J, Shenker S (2002) Distributed Algorithmic Mechanism Design: Recent Results and Future Directions. In: DIALM'02 - Discrete Algorithms and Methods for Mobile Computing and Communications, ACM Press, pp 1–13

Ferber J (1999) Multi-Agent System: An Introduction to Distributed Artificial Intelligence. Addison Wesley Longman

Fishburn P (1970) Utility Theory for Decision Making. Storming Media

Fisher M (2004) The Lagrangian Relaxation Method for Solving Integer Programming Problems. Management Science 50(13):1861–1871

Friedman D, Rust J (1993) The Double Auction Market: Institutions, Theories and Evidence. Addison-Wesley

Gairing M, Lücking T, Mavronicolas M, Monien B (2006) The Price of Anarchy for Polynomial Social Cost. Theoretical Computer Science 369(1-3):116–135

Gairing M, Monien B, Tiemann K (2008) Selfish routing with incomplete information. Theory of Computing Systems 42(1):91–130

Galstyan A, Czajkowski K, Lerman K (2005) Resource Allocation in the Grid with Learning Agents. Journal of Grid Computing 3(1):91–100

Ge Z, Figueiredo D, Jaiswal S, Kurose J, Towsley D (2003) Modeling Peer-Peer File Sharing Systems. In: INFOCOM'O3, IEEE Computer Society, USA, California, San Francisco, vol 3, pp 2188–2198

Gibney M, Jennings N (1998) Dynamic Resource Allocation by Market-Based Routing in Telecommunications Networks. Lecture Notes in Computer Science 1437:102–117

Goldstein J (1997) Emergence as a Construct: History and Issues. Emergence 1(1):49–72

Grabisch M (1997) *k*-Order Additive Discrete Fuzzy Measures and their Representation. Fuzzy Sets and Systems 92(2):167–189

Grosu D, Chronopoulos A (2004) Algorithmic Mechanism Design for Load Balancing in Distributed Systems. IEEE Transactions on Systems, Man, and Cybernetics 34(1):77–84

Groves T, Ledyard J (1977) Allocation of Public Goods: A Solution to the" Free Rider" Problem. Econometrica pp 783–809

Guerra-Hernández A, Castro-Manzano J, El Fallah-Seghrouchni A (2009) CTL AgentSpeak(L): A Specification Language for Agent Programs. J Algorithms 64(1):31–40

Hales D (2004) From Selfish Nodes to Cooperative Networks-Emergent Link-Based Incentives in Peer-to-Peer Networks. In: P2P'04 - Peer-to-Peer Computing, IEEE Computer Society, Switzerland, Zurich, pp 151–158

Hemaissia M, El Fallah Seghrouchni A, Labreuche C, Mattioli J (2007) A Multilateral Multi-Issue Negotiation Protocol. In: AAMAS'07 - Autonomous Agents and Multi-Agent Systems, ACM Press, Honolulu, Hawaii, pp 1–8

Hickernell F, Yuan Y (1997) A Simple Multistart Algorithm for Global Optimization. OR Transactions 1(2)

Hofbauer J, Sigmund K (2003) Evolutionary Game Dynamics. American Mathematical Society 40(4):479–520

ILOG Inc (1995) CPLEX User's Manual. ILOG Inc

Jameson A, Hackl C, Kleinbauer T (2003) Evaluation of Automatically Designed Mechanisms. In: UAI'03 - Bayesian Modeling Applications Workshop, Uncertainty in Artificial Intelligence, Mexico, Acapulco

Kaihara T (2003) Multi-Agent Based Supply Chain Modelling with Dynamic Environment. International Journal of Production Economics 85(2):263–269

Kfir-Dahav N, Monderer D, Tennenholtz M (2000) Mechanism Design for Resource Bounded Agents. In: ICMAS'00 - International Conference on MultiAgent Systems, USA, Boston, IEEE Press, pp 309–315

Koutsoupias E, Papadimitriou C (2009) Worst-Case Equilibria. Computer Science Review 3(2):65–69

Krishna V (2002) Auction Theory. Academic Press

Kubera Y, Mathieu P, Picault S (2008) Interaction-Oriented Agent Simulations : From Theory to Implementation. In: ECAI'08 - European Conference on Artificial Intelligence, Greece, Patras, pp 383–387

Lang J (2004) Logical Preference Representation and Combinatorial Vote. Annals of Mathematics and Artificial Intelligence 42(1):37–71

Lehmann D, Muller R, Sandholm T (2006) The Winner Determination Problem. Combinatorial Auctions pp 297–318

Lemaître M, Verfaillie G, Bataille N (1999) Exploiting a Common Property Resource under a Fairness Constraint : A Case Study. In: IJCAI'99 - International Joint Conference on Artificial IntelligenceInter, Morgan Kaufmann, Sweden, Stockolm, vol 1, pp 206–211

Mas-Colell A, Whinston M, Green J (1995) Microeconomic Theory. Oxford University Press

Miranda P, Grabisch M, Gil P (2005) Axiomatic Structure of k-Additive Capacities. Mathematical Social Sciences 49:153–178

Morge M, Mathieu P (2007) Mécanisme de Rétribution pour les Systèmes P2P d'Échange de Fichiers. Ingéniérie des Systèmes d'Information 12(3):61–84

Morge M, Mathieu P, Mancarella P (2009) Assumption-Based Argumentation for the Minimal Concession Strategy of Agents Engaged in Resource Negotiation. In: MFI'09 - Modèles Formels de l'Interaction, EU, France, Lannion, pp 1–10

Moulin B, Chaib-Draa B (1996) An Overview of Distributed Artificial Intelligence. Foundations of Distributed Artificial Intelligence 1:3–55

Moulin H (1986) Choosing from a Tournament. Social Choice and Welfare 3(4):271–291

Moulin H (1988) Axioms of Cooperative Decision Making. Cambridge University Press

Moulin H (2004) Fair Division and Collective Welfare. MIT Press

Ni L, Liu Y (2004) Efficient Peer-to-Peer Overlay Construction. In: ECE'04 - E-Commerce Technology for Dynamic E-Business, IEEE Press, China, Beijing, pp 314–317

Nisan N (1999) Algorithms for Selfish Agents. Lecture Notes in Computer Science 1563(1):1–15

Nisan N (2000) Bidding and Allocation in Combinatorial Auctions. In: EC'00 - Electronic Commerce, ACM Press, USA, Mineapolis, pp 1–12

Nongaillard A, Mathieu P (2009a) A Multi-Agent Resource Negotiation For Social Welfare. In: IAT'09 - Proceedings of the IEEE/WIC/ACM international conference on Intelligent Agent Technology, pp 58–62

Nongaillard A, Mathieu P (2009b) Allocation de ressources et maximisation de bien-être social. In: Actes des Cinquièmes Journées Francophones sur les Modèles Formels de l'Interaction (MFI'09), pp 243–254

Nongaillard A, Mathieu P, Jaumard B (2008a) La négociation du bien-être social utilitaire. In: Chevaillier P, Mandiau R (eds) Actes des 16e Journées Francophones sur les Systèmes Multi-Agents (JFSMA'2008), Cépaduès, pp 55–64

Nongaillard A, Mathieu P, Jaumard B (2008b) A multi-agent resource negotiation for the utilitarian social welfare. In: Artikis A, Picard G, Vercouter L (eds) Proceedings of the 9th International Workshop Engineering Societies in the Agents World (ESAW'2008), Springer, Lecture Notes on Artificial Intelligence, vol 5485

Nongaillard A, Mathieu P, Jaumard B (2009a) A Realistic Approach To Solve The Nash Welfare. In: Demazeau Y, Pavòn J, Corchado J, Bajo J (eds) Proceedings of the 7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS'2009), Springer Verlag, Practical Advances in Intelligent and Soft Computing, vol 55, pp 374–382

Nongaillard A, Mathieu P, Jaumard B (2009b) La maximisation du bien-être utilitaire des sociétés d'agents. Revue d'Intelligence Artificielle 23(5-6):653–674

Parsons S, Wooldridge M, Amgoud L (2003) Properties and complexity of some formal inter-agent dialogues. Journal of Logic and Computation 13(3):347–376

Petcu A, Faltings B, Parkes D (2006) MDPOP: Faithful distributed implementation of efficient social choice problems. In: AAMAS'06 - Autonomous Agents and MultiAgent Systems, ACM Press, Japan, Hakodate, pp 1397–1404

Rahwan I, Larson K (2008) Pareto Optimality in Abstract Argumentation. In: AAAI'08 - Conference on Artificial Intelligence, AAAI Press, USA, Illinois, Chicago, pp 150–155

Ramezani S, Endriss U (2009) Nash Social Welfare in Multiagent Resource Allocation. In: AMEC'09 - International Workshop on Agent-Mediated Electronic Commerce

Rawls J (1999) A Theory of Justice. Belknap Press

Roughgarden T (2005) Selfish Routing and the Price of Anarchy. MIT Press

Saha S, Sen S (2007) An Efficient Protocol for Negotiation over Multiple Indivisible Resources. In: IJCAI'07 - International Joint Conference on Artificial Intelligence, India, Hyderabad, pp 1494–1499

Sahli N, Moulin B (2009) EKEMAS, an Agent-Based Geo-Simulation Framework to Support Continual Planning in the Real-word. Applied Intelligence 31(2):188–209

Sandholm T (1998) Contract Types for Satisficing Task Allocation I : Theoretical Results. In: AAAI Spring Symposium: Satisficing Models, AAAI Press, USA, California, Stanford University, vol 99, pp 68–75

Sandholm T (2002) Algorithm for optimal winner determination in combinatorial auctions. Artificial Intelligence 135(1-2):1–54

Sandholm T (2003) Automated Mechanism Design: A New Application Area for Search Algorithms. Lecture Notes in Computer Science pp 19–36

Sen A (1970) Collective Choice and Social Welfare. Holden-Day

Sen A (1995) Inequality Reexamined. Oxford University Press

Sen A (1997) Choice, Welfare and Measurement. Harvard University Press

Sen S (1996) Reciprocity: a Foundational Principle for Promoting Cooperative Behavior Among Self-Interested Agents. In: ICMAS'96 - International Conference on Multi-Agent Systems, AAAI Press, USA, California, Menlo Park, pp 322–329

Serugendo G, Gleizes M, Karageorgos A (2006) Self-Organisation and Emergence in MAS: An Overview. Informatica 30(1):45–54

Sethuraman J, Piaw T, Vohra R (2003) Integer Programming and Arrovian Social Welfare Functions. Mathematics of Operations Research 28(2):309–326

Shen W (2002) Distributed Manufacturing Scheduling using Intelligent Agents. IEEE Intelligent Systems 17(1):88–94

Sycara K, Roth S, Sadeh N, Fox M (1991) Resource Allocation in Distributed Factory Scheduling. IEEE Expert 6(1):29–40

Uckelman J, Chevaleyre Y, Endriss U, Lang J (2009) Representing utility functions via weighted goals. Mathematical Logic Quarterly 55(4):341–361

Weibull J (1997) Evolutionary Game Theory. MIT Press

Wellman M, Doyle J (1992) Modular Utility Representation for Decision-Theoretic Planning. In: AIPS'92 - Artificial Intelligence Planning Systems, Morgan Kaufmann, USA, Maryland, College Park, pp 236–242

Woolridge M (2001) Introduction to Multiagent Systems. John Wiley & Sons