Università degli Studi di Verona

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea Magistrale in Ingegneria e Scienze Informatiche

Tesi di Laurea Magistrale

Stable Coalition Formation in the Energy Market Using Graphical Model Based Approach

Candidato:
Filippo Bistaffa
Matricola vr098282

Relatore:
Alessandro Farinelli

Correlatore:
Meritxell Vinyals

Anno Accademico 2010–2011

Contents

List of Figures

List of Algorithms

Abstract – One of the main problems in the production and distribution of energy is the inefficiency of its storage on a large scale. User energy consumption usually follow a profile which provide local maxima during some well known time periods (usually around 7:00 and 18:00), so ensuring an adequate amount of energy in these particular moments is a critical problem, since the production methods – *i.e.* power stations – are slow to fire up and slow down and cannot follow properly the demand of the network.

This thesis aims at studying a method of forming groups of energy users – called *coalitions* – in a way that minimizes such peaks. Having a flatter profile let the formation buy the energy in a smarter way, since more of it can be retrieved through the long-term market – the *forward market* – which has cheaper prices and grants a monetary gain to the single users.

Since the number of possible ways to arrange this coalitions is exponential in the number of users, a brute force attack would be infeasible, as trying every possible solution would require too long for real-world sized problems. However, the dimension of the space of solutions can be smartly lowered considering additional constraints, which state if a particular result can be considered valid or not. Social networks come as a great help in the formulation of such constraints, providing knowledge-based relations among users that can be exploited in the coalition formation procedure.

Such networks can be perfectly expressed as graphs, which can be analyzed with many formal methods; in particular, we'll focus on *graphical models*, a powerful tool that will be used as base for a message passing algorithm able to compute optimal and stable coalition in a distributed way.

Chapter 1

Introduction

In the field of the generation and distribution of electrical energy, one of the main difficulties faced by producers is the impossibility to store large amounts of energy in an efficient way. Since the demand by the consumers is not constant – rather, it usually has high peaks in some periods of the day, typically in the morning and in the evening – matching the demand of energy in these particular moments is a critical engineering issue. In all current electricity grids this balance is achieved by varying the supply-side to continuously match demand. The amount of demand required on a continuous basis is usually carried by the baseload stations owing to low cost generation, efficiency and safety. However, these stations are slow to fire up and cool down, so they are not able to match the peakload periods that exceeds this baseload demand, which is then satisfied by using expensive, carbon-intensive, peaking plant generators. Although only running when there is high demand, these peaking plant generators are responsible for the most part of consumers' electricity bill and pollutant emissions.

To address this issue, current approaches envision demand-side peak-shaving strategies such as real-time pricing or profile's based tariffs to encourage consumption such that the peaks on demand are *flattened*. A flattered demand results in a more efficient grid not only with lower carbon emissions but also with lower prices for consumers. In this thesis, the problem of how the grid efficiency can be further improved from a social perspective is investigated. In particular, we explore the idea of allowing consumers to join into coalitions with other consumers with complementary energy needs. Then, a group of consumers can act in the market as a single virtual consumer with a flattened demand for which it gets much better prices. Specifically, having a regular energy demand profile has a twofold positive effect: on the one hand it allows customers to buy the a consistent amount of energy on a forward cheaper market thus saving on the electricity bill. On the other hand a flatter energy demand is crucial to reduce the need of peaking plant generators thus minimizing pollutant emissions.

1.1 Contributions

In more details, this thesis makes the following contributions:

- **Implementation of a novel approach to stable coalition formation [71]** – A novel approach to this problem is presented, describing a GDL message-passing algorithm based on the work of Vinyals *et al.*, which allows users to organize in *optimal* and *stable* coalitions in a distributed way. In this context, optimality is intended as the maximization of the collective value for the entire system, also referred as *social welfare*. Actually, the presented schema is the first in literature to deal with the goals of optimality *and* stability at the same time, which are usually treated separately. In particular, we will describe a novel graphical model representation scheme for *coalitional games* defined over graphs, and exploit this representation to devise the first decentralized algorithm leading to the formation of stable coalition structures. We remark that this technique, build upon well-known message passing algorithms from the GDL family, represents an important extension of previous seminal works [27], showing how to treat general problems with cyclic graphs.

- **Application of the implemented approach to a real-world case study: the energy market** – The mentioned theoretical notions and solution techniques have been applied to a real-world case study: the energy market. We will focus on a scenario in which a group of energy customers, assumed to be organized in a social network, have a demand of energy supplies (*e.g.* electrical power), and need to buy them in a way that minimizes their costs, forming *coalitions*. As will be pointed out, this setting has a natural modelization in *coalitional game theory* in which players are considered to be selfish, thus setting us in a non-cooperative environment. Real energy customers have little interest in increasing system gain, as their main concern is to maximize their own utility, with the simple goal of having a cheaper bill. Therefore, the concept of *stability* assumes a key role, since it is not possible to impose a solution "from above", as it would be immediately rejected by the agents. Rather, the goal is to find an appropriate distribution of payments that incentives single energy customers to maintain the current structure, as any deviation would worsen their payoff, making energy more expensive.

- **Implementation of a GUI for simulating coalition formation in the energy market** – The mentioned algorithm has been implemented realizing a Java simulator, which has allowed us to verify the correctness and the effectiveness of the theoretical hypotheses.

Moreover, the demonstration application permits the user to interact with the coalition formation process and have detailed informations about the optimal outcome and the possible gains that can be obtained. The platform generates a set of nodes, one per energy consumer, and allows the user to modify the network by adding/removing links in an easy way. Each node has an energy profile loaded from real data characterizing the domestic electricity market and usage patterns of households in the United Kingdom. Once the coalition formation scenario is set, the simulation starts the GDL message-passing algorithm, organizing energy consumers into stable optimal coalitions. Upon convergence, energy consumers in the same coalition are colored with the same color, showing detailed informations about the gain obtained if the suggested solution is used. Finally, the interface also allows to visualize the energetic profiles of coalition members. Each chart plots a consumer energy profile and the joint coalition energy profile. The GUI offers the user an effective way of restarting simulations after reconfiguring the network topology, testing how the existence or the nonexistence of a particular link affects the emerging coalitions and consumers gain.

This application has been the subject of a demo paper submitted at *11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)* [14] and is available for download as an executable JAR file at `http://profs.sci.univr.it/~farinelli/energySCF.jar`. Moreover an illustrative video, describing the simulator and the related topics can be viewed at `http://www.youtube.com/watch?v=FT25oETMkfw`.

- **Definition of several metrics to evaluate coalitions in the energy market** – The problem of a proper evaluation of coalitions is covered by means of a complete analysis apt to appropriately choose a metric to estimate how the considered coalition is suitable to users' demands. As pointed out before, "good" coalition are groups whose joint energy profile is flattened, because more energy can be bought in a cheaper way on the forward market, so the metrics described have the primary objective of quantifying this feature, attempting to assign a smaller value to a profile with a peaky behavior, while encouraging the formation of coalitions whose profiles are more regular. Moreover, another eligible property of the metric is *non-superadditivity.* Intuitively, a setting is said to be superadditive if it is always profitable for two teams to join forces and form an unique coalition, so there's no reasonable incentive for the agents to go alone. In the considered context, it would imply putting every agent in the same group, making everyone buying energy together, which is just not reasonable.

Furthermore, in real-world environments the complex dynamics and the increased coordination costs of bigger groups make the setting intrinsically non-superadditive. Specifically, two particular metrics have been analyzed: the *load diversity factor*, which captures the "smoothness" of a given energetic profile, and the *user price factor*, which gives an estimation of the final cost of the electricity bill, assuming a *baseload* amount of energy is bought on the long term market at a cheaper price, obtaining the remaining part on the day-ahead market.

- **Empirical evaluation** – The simulator has been used to make an extensive set of empirical experiments (covered in details in Section 1.3), which have confirmed the correct behavior of the GDL message-passing algorithm. Various aspect of the considered problem has been analyzed, inspecting how different metric configurations and graph topologies affect the coalition formation process, in terms of size of coalitions and average gain obtained by the agents. In particular, the variation of the prices in the forward and day-ahead market and its influence on the experimental results is addressed.

1.2 Algorithm outline

From a general point of view, the presented GDL-based algorithm works in two phases: assuming a pre-existing hierarchy among the consumers, agents at lower levels progressively communicate their demands, in terms of expected payments they require in order to join a given coalition. These demands are treated as *messages* exchanged among agents, used to choose the appropriate coalition structure satisfying everyone's requests (if possible). In general cyclic graphs, this schema is iterated multiple times (linear in the number of agents): intuitively, at each step one single agents fixes his decisions and leaves the computation, which subsequently runs on a smaller problem.

The algorithm exploits the structure of the problem by means of a particular graph, which models the connections among agents and define constraints over the considered problem. Specifically, *social networks* capture the knowledge relations existing between users and restrict the number of possible coalition structures to investigate, significantly improving the computational complexity of the process. In particular, consumers may not want to join coalitions with unknown consumers for which they do not have any source of trust regarding their reported profiles or their capacity to meet their payment obligations. In contrast, if the social network is used to restrict coalition membership, customers join coalitions of *friends of friends*, thus being sure that someone they know directly is always involved.

Upon convergence, the algorithm returns a *stable* outcome (if existent), in a sense that users are not incentivized to deviate from the found equilibrium, since it already rewards them with the best payment they can individually achieve. Any alteration of the proposed coalitional structure is not convenient for the agents, as it would represent an actual loss in terms of obtained value. This is a fundamental issue to be addressed, since in real-world settings the agents interacting are intrinsically *selfish*, thus they are only interested in maximizing their own utility. More formally, the proposed solution can be proved to belong to the *core* of the game, defined as the set of all stable solutions.

1.3 Experimental analysis

As mentioned above, a comprehensive set of empirical experiments has been conducted to test the performance of the presented coalition formation technique. In particular, the analysis focuses on two fundamental features of the solutions: the *average user gain* and the *coalition structure*. Intuitively, *average user gain* refers to the effective gain users have obtained adopting the coalition organization proposed by the given outcome, with respect to a naive solution where only single coalitions are formed. In particular, to investigate the sensitivity of the coalition formation process with respect to the underlying network topology, we have evaluated our model on three different network models (*random*, *scale-free* and *small-world* networks). Moreover, each topology has been parametrized using different density levels, representing the amount of connections with respect to the number of nodes. Furthermore, the above mentioned features have been interpreted studying the correlation with various metric configuration, simulating different price situations and considering different approaches adopted by the consumers with respect to the risk.

We remark that, as an additional sanity check, all the results have been tested with an alternative technique, verifying the coherence of obtained solutions. In particular, the same coalition formation problem has been formalized using linear constraints and solved using *IBM ILOG CPLEX Optimization Studio*, which currently represents the *state of the art* in linear programming optimization.

1.4 Results

This work gave many promising results: the presented method is effectively able to compute coalitions, granting various benefits both for single energy consumers, who can save on their electricity bills, and for the entire system, in an ecological perspective.

Due to the reduction of peaks in the aggregate energy profiles, the demand of peaking plant generators – responsible of most of the pollutant emission in the energy production – can be limited, with a promising environmental effect. In addition, the metrics proposed (especially the *user price factor*) have proven to be very effective in the tests, being able to effectively capture the quality of good solutions and generating an interesting variety of results. Finally, based on these results, we performed an analysis on how different market configurations and input graphs topologies affect the coalition formation process.

1.5 Thesis outline

This thesis is structured in the following way: Chapter 2 gives a brief outline of the related work which previously addressed similar topics in literature. Chapter 3 formally defines the theoretical basis of coalitional game theory, which are used in Chapter 4 to extensively describe the mentioned GDL message-passing algorithm. This technique is then applied to a real-world case study: the energy market, as depicted in Chapter 5, in which we also describe the demonstration application. Chapter 6 reports and discusses the extensive set of experiments we performed to analyze the various aspects of the considered problem. Finally, Chapter 7 concludes the work and highlights possible directions of future work in this area.

Chapter 2

Related Work

In this chapter we'll briefly discuss the coalition formation activities undertaken when agent in a multiagent system come together to form teams. This problem is known as *coalition structure formation*, which can be divided in two main categories: *coalition structure generation* activities refer to the context in which agents are collaborative and willingly agree to cooperate to fulfill the agenda of a single designer; on the other hand we have *coalition formation activities by selfish rational agents*, in which agents are selfish entities whose interest is only maximizing personal interests.

We'll now discuss both scenarios, describing the main properties and the techniques that can be applied to solve the problems [22]. All topics are presented in an informal way, leaving specific concepts and formal definitions to Chapter 3, where everything will be defined precisely.

2.1 Coalition Structure Generation

The context of the *coalition structure generation* is entirely based on a very strong hypothesis: the agents will be considered to be fully cooperative and to work together with the common goal of maximizing the gain for the entire system.

For example, the whole environment could be "owned" by a single designer, following its will and making the performance of single agents less important than the *social welfare* of the system (considered to be the sum of the values of single coalitions).

Obviously, finding the optimal solution is straightforward if the considered game is *superadditive*: when this happens, it is always profitable for two teams to join forces and form an unique coalition, so there's no reasonable incentive for the agents to go alone; instead, the *grand coalition* will always form, grouping all players.

A formal definition of *superadditivity* and related notions will be given in Section 3.3.

Nevertheless, it will be shown that many important environments (included the energy market domain) are not superadditive. Therefore, the problem of finding the best partition of agents in teams in non-superadditive settings is referred as *coalition structure generation*, which has been extensively studied in literature.

Most of these approaches assume that the game is represented by the list of possible coalition together associated to a value – representing how "good" that solution is – or by an oracle. However we remark that algorithms for coalition structure generation have been recently developed, taking advantage of specific succinct representations (synergy coalition groups [50], skill games [9], or games with a constant number of agent types [7]).

Note that an enumeration-based approach is not possible in this particular field, since the number of coalition structures is exponential in the number of agents n, and it's huge even comparing it to the number of coalitions, which is 2^n. This dimension is represented by the *Bell number* B_n [11], which can be shown to be greater than n^n. This implies that the direct enumeration would require super-polynomial time considering the naive representation of a game, leading the scientific community to change the approach for the research of solutions.

2.1.1 Dynamic Programming

The use of *dynamic programming* for the generation of coalition structure [59] can be explained very well considering the notion of *superadditive cover*, in which the value of each coalition is the social welfare of the best coalition structure that the member of the considered group can form. It can be shown that finding the optimal coalition structure can be effectively obtained via standard dynamic programming techniques.

Computing the optimal social welfare value $v^*(C)$ is possible by progressively computing the value of smaller coalitions, starting from the 1-sized ones (whose optimal value is simply the value of the coalition itself, since there's an unique coalition structure). For bigger coalitions, one proceeds to split the considered one into two disjoint non-empty groups C' and C'', selecting the one that maximizes $v^*(C') + v^*(C'')$. Note that these coalitions are smaller, meaning that their optimal values have already been computed. The results is finally given by the comparison of this sum with the entire coalition $v(C)$, since the non-empty disjoint assumption prevents considering the whole group.

The running time of this algorithm is exponential in the number of agents n, since the number of possible ways of possible of size k is given by $\binom{n}{k}$, and for each one there are $2^{k-1} - 1$ possible ways to split it into two non empty coalitions. Nevertheless, this is a lot smaller than n^n, meaning that this method avoids searching the entire solution space. However, dynamic programming is still too slow for real-world applications.

2.1.2 Anytime Algorithms

Sandholm [60] showed that is possible to find a coalition structure that is within some *provable bound* from the optimal one through an *approximation algorithm.*

The idea is to establish a graph organizing all possible coalition structures resulting from a game with n agents in n levels, such that level i contains all the solution formed exactly by i coalitions. Thus, at the first level we always find the grand coalition, since it is the only configuration containing one coalition, while the n-th level has the solution formed by the singletons.

Two nodes of this particular graph are connected if it is possible to obtain one configuration from the other only by splitting one coalition in two. For example, there will be an edge between configurations $\{\{1\},\{2\},\{3\}\}$ and $\{\{1\},\{2,3\}\}$. Given this constraint, only nodes from adjacent levels will be connected.

One interesting – provable – property of this particular graph is that every possible coalition appears in the first two levels; obviously this doesn't hold for coalition structures.

Exploring the entire graph doesn't provide any advantage over the naive enumeration approach, but if we restrict the search to the first two levels we can find a solution whose value is within a factor of $\frac{1}{n}$ from the optimal.

So we can begin our research starting from the bottom and keep track of the best solution for $i = 1$ and $i = 2$; if we have more computational time to spend, we refine our solution starting from the current one and proceeding with a breadth-first strategy, always saving the best available result. This process can be stopped as soon as the available running time for the research is over, or else we have visited all the remaining parts of the graph.

Experimentally, this approach has been shown to perform quite well, and it has the advantage to be *anytime* – *i.e.* it guarantees to provide incrementally better solution if given more time or computational resources. Note, however, that exploring the first two levels is not trivial and quite expensive: 2^{n-1} coalition structures are contained only in level 2.

Some refinements and improvements were made by Dang and Jennings [26], and Rahwan [56], who replaced the coalition structure graph and applied particular search techniques, such as *branch and bound.*

Interestingly, the aforementioned dynamic programming approach suits very well to the presented graph representation: it first evaluates every possible movement in the graph, and then moves from a node upward until an optimal one is reached, after which no further computation is beneficial. Not every edge in the graph is needed to find the best structure, as long as there's a path leading to it. This observation is the basis of the work proposed by Rahwan and Jennings [54], who improved these approaches letting the algorithms use less time and memory.

A further step was the presentation of *hybrid improved dynamic programming* [55], which adds a preliminary step at the beginning of the algorithm to prune the less promising parts of the space search, making the subsequent parts significantly faster than both of its ingredients.

A different way to combine anytime algorithms and dynamic programming is proposed by Service and Adams [62], whose approach relies on an approximation scheme, *i.e.* a family of algorithms with the running time based on the desired approximation ratio.

2.2 Stable Coalitions

The topics discussed so far implicitly assume that *cooperative* agents can be grouped into teams by a central computational unit, which aims to form coalitions that maximize the social welfare. This is not the case in many real-world problems, in which agents' only interest is to maximize their own utility – that is why they are referred as *selfish.* Though they accept to work with others, their primary objective is to get as much as possible out of some kind of contract. One important application, that will be covered in detail in Chapter 5, is the energetic market, in which the single users may be interested in the improvement of the *social welfare* (*e.g.* for ecological purposes), but their main goal is the minimization of the cost they have to pay for the energetic contract.

One fundamental notion related to the field of coalition formation with selfish agents is the concept of *core*, which we will formally defined in Section 3.4.1. The core represent the set of *stable* solutions, *i.e.* all the outcomes for which the users are not interested in deviating from the given structure, since their own demands are already satisfied. Optimal solutions out of this particular set are not interesting, since they would be altered as soon as possible by the selfish players of the game.

Thus, in this section we'll cover coalition formation activities that deal with this kind of agents.

2.2.1 Bargaining Approach

In some formulations, agents can establish some explicit negotiations as part of a *bargaining process* in order to form groups. When *coalition bargaining procedures* happen, the game can be seen as a (non-cooperative) one [37, 45, 52]. A wide variety of examples has been provided in literature [13, 24, 30, 33, 51, 53, 61, 74], we'll briefly examine some representative examples.

Okada [51] proposed a model in which at every round of bargaining a proposer chosen randomly makes an offer for a given coalition and a linked payoff vector. If the proposal is accepted, the mentioned coalition abandons the negotiation process. At every round, the value of the remaining coalitions is lowered by a fixed value.

Okada then characterizes *subgame-perfect equilibria (SPE*[1]*)*, showing that, if the SPE are considered to be *stationary* and the proposer is chosen randomly and uniformly at each round, every proposal made in round one is immediately accepted by the interested agents with no delay. The stationary hypothesis is responsible of the property that every proposal and response done during a certain round is based only on the active set of agents in that particular instant, and is uncorrelated to the past history; with this assumption, the equilibrium proposals and responses of the players are considered the solution to a payoff maximization problem. Note that, as shown by Chatterjee [24], if a *fixed* proposal order is given (discarding the random and uniform hypothesis), a delay of agreement in equilibrium *can* effectively happen.

Moldovanu and Winter [13] approach changes the order of proposals, which is based upon responders' replies; the first agent to refuse an offer becomes the next *initiator*, which can propose a coalition and a payoff vector to its members, but can also pass the initiative to another player. Similarly as seen before, when an offer is accepted, the given coalition abandons the game, but there's no discount of values over time. Moldovanu and Winter focus on SPE too, showing that if the bargaining strategy is an *ordered independent equilibrium (OIE)*, the resulting payoff must be in the core. On the other hand, if the considered game has subgames with non-empty cores, then for each payoff vector there exists an OIE with the same payoff

As pointed out, the concept of *core* is very important in the field of bargaining procedures, since it has been shown that many other bargaining procedures mentioned in literature converge to outcomes in the core under certain hypotheses [13, 30, 33, 53, 61, 74].

2.2.2 Dynamic Coalition Formation

If the context we're dealing with does not permit considering a sequential bargaining process, the aforementioned techniques cannot be applied. Political and market alliances in a "fluid" environment provide an example. Coalition structures can be modified due to a wide variety of internal and external factors – agents abandoning or joining the system, fluctuation in the computational and communication ability of the partners – making the whole scenario *dynamic*.

A first approach to this topic has been made by Shehory and Kraus [63], who considered a (non-bargaining) three-stage process, in which the values of potential coalition are computed in the first stage in a distributed way. A particular protocol takes care of this, ensuring that every agent knows the capabilities of the potential partners and determining how players may approach each other.

[1] SPE is a strategy profile whose restriction to any subgame following any history in the game is in equilibrium *w.r.t.* that particular subgame [45, 52]

Subsequently, a greedy phase follows, in which every agent chooses from the pre-computed values the one that best suits its own needs. Finally, these decisions are broadcasted to the system, notifying every user with the benefits of forming certain coalitions.

Since this method is exponential in the number of agents, particular heuristics can be used to reduce the amount of communication messages (*e.g.* bound the size of each coalition by a small constant).

Arnold and Schvalbe [4] extended the study of coalitional stability in dynamic coalition formation, making use of a Markov process model. At each stage, the state is represented by a particular coalition structure and a associated allocation of payoffs, or *demands*. Then, with a probability of γ, every agents decides if it wants to move from the current configuration and try to obtain a better payoff, in which case the behavior is described by a "non-cooperative best-reply-rule": a player decides to change its situation if its expected payment under ideal condition exceed its current payoff, and its demand is the highest possible fulfilling feasibility constraints. The process follows some sort of "evolution" by the agents adjusting their coalitions and demands as long as changes are feasible. The presented process is shown to have at least one state of equilibrium – Arnold and Schvalbe showed that if the core isn't empty, exactly one of them exists for every core allocation – which will be reached with certainty.

Konishi and Ray [40] presented similar approach, in which the change from the current state (with consequent coalition structure and payoff modifications) is done only if it's profitable for *every* member of the group. Everyone of them agrees about the probability with which state transition may occur. Augustine [5] focused on the dynamics of coalitional games with submodular[2] characteristic functions under three natural profit-sharing models, in which agents are divided into groups and are paid according to a pre-defined profit-sharing scheme. They may want to deviate from this configuration myopically, aiming to increase their payoff.

Sycara [44, 68] adopted a task-oriented approach in the multiagent system community, developing the Reusable Environment for Task Structured Intelligent Network Agent (RETSINA) framework. RETSINA distributed system supports heterogeneous agents to join and leave the environment dynamically, which may cooperate and work together in order to execute tasks.

2.2.3 Dealing with Uncertainty

All the approaches described so far make the implicit assumption that the values of coalitions are known with certainty, but in real situations a variety of parameters can make this hypothesis unreasonable to be made.

[2] A function is said *submodular* if and only if $f(A \cup B) + f(A \cap B) \leq f(A) + f(B)$

Players can be unsure about the capabilities of the partners – either in terms of computational and communication power – or if they intend to use them, there can be unexpected obstacles and failures, and so on.

Uncertainty about resources

Ketchpel [38] started dealing with these topics specifying a two-agent protocol to permit coalition formation in the face of uncertain rewards. Kraus [43] focused on a scenario where the players have incomplete information about each other, and have to establish a bargaining process in order to decide how to share a joint resource. Every agent keep track of partners' responses to offers and consequently update their beliefs, defining a type – chosen in a finite predefined list – for each member of the system, which has its own utility function based on the resource usage. Since every resource is considered shareable among a maximum of two users, the negotiations in any given period are *bilateral*: one user is currently holding the resource during the protocol execution, while the other is waiting to have access to it.

Shehory and Kraus [63, 64] also dealt with problem in which the capabilities of single users are not fully known. An estimation is required by every agent, which is based upon the information sent by their potential partners. In [64] an algorithm for payoff allocation is presented, assuming a non-cooperative game placed in a non-superadditive environment. Shehory [65] developed a similar approach using the RETSINA framework to implement a coalition formation algorithm using agent collaboration. This work focuses on maximizing social welfare, thus it does not cover payoff allocation issues.

Kraus [41, 42] also focused on coalition formation in the "Request for Proposal" domain. In this particular formulation, agents are forced to complete a set of tasks, each comprised of sub-tasks – each of which must be performed by a different user. The effort taken by an agent to execute a sub-task may be unknown to others, but they *do* know the overall payoff associated to the complete task. The allocation of the single sub-tasks is done by an auction, in which agents use heuristics to form teams in order to bid. In [41] a democratic division of total payoff is assumed, which does not necessarily encourage single agents declaring the actual effort they spent in the performance of their sub-tasks. The subsequent work [42] covers various payoff allocation strategies, going from a simple proportion with agents' costs to a *compromise-based* approach, where the agents may be willing to receive lower payoffs as long as they join a coalition. This approach performed well in the tests, showing that compromises facilitate the formation of successful groups, but some improvements are still possible: agent are expected to behave exactly as listed in the paper and deviation are not considered; also iterative coalition formation is not addressed: once a team is formed, it's excluded from negotiations and cannot be split.

Banneker and Sen [10] analyzed a scenario with uncertainty regarding payoffs. The process of coalition formation or payoff allocation is not covered. Rather, the author focus on the problem of "coalition selection", faced by an user which has incomplete information about the advantage of joining a particular coalition, and has to take a decision after a fixed number of iteration involving interaction with it. The proposed solving mechanism make use of a particular multinomial probability distribution, which summarizes the information gained by the agents over possible payoffs for joining the coalition, also providing an heuristic to solve ties.

Blankenburg [15] realized a coalition formation algorithm that consider *trust* values which are progressively updated by single agents, exchanging their private approximations regarding task costs and coalition valuations. Some cryptographic techniques are used to encourage truthful reports of valuations. However, this interesting approach is rather resource-intensive: it requires extensive inter-agent communication and relies on the computation of optimal coalition structures and stable solutions (which are computationally intensive tasks).

In the formalization proposed by Chalkiadakis [21], every user has a *type*, which is the argument of the function defining the values of coalitions, and a set of (non-probabilistic) *beliefs* regarding each other's types. As in real world, these beliefs don't need to be consistent among agents, thus two players can have different beliefs about another agent. The authors provide a definition of the core for this setting, providing simple examples with beliefs leading to a non-empty core solution, but they not cover the mechanism that allows agents to update beliefs according to the performance of formed groups.

Stochastic cooperative games

Suijs [66, 67] gave a great contribution to the formalization of the aforementioned topics, introducing *stochastic cooperative games (SCGs)*. The description of such games is formed by a set of agents, a set of coalitional actions and a function assigning a (finite) random variable to each of them, which represents the expected payoff to the coalition when the given action is taken. It is important to note that, in this particular setting, the agents are unsure about coalitional payoffs, and hence about their own one. This issue is covered by Suijs by using *relative* payoff shares, *i.e.* allowing agents to provide relative demands on the fractional share of the resulting payment. These papers give the theoretical basis for games with payoff uncertainty, but do not explicitly cover the coalition formation process. Moreover, the provided model does not support the possibility of incomplete information about partners, assuming *common expectations* regarding coalition values. This particular hypothesis is dropped in the theory of Bayesian cooperative games, described in the following section.

Bayesian cooperative games

Many authors focus on *Bayesian cooperative/coalitional games*, including Myerson [8, 47], Ieong and Shoham [35] and Chalkiadakis [18–20, 23].

The latter has been developed directly starting from the cited work of Suijs, introducing many important features, such as *agent type uncertainty*. More formally, Chalkiadakis focused on a particular class of *Bayesian coalition formation problems (BCFPs)*, where every player has *private probabilistic beliefs* about other members' *types* (or *capabilities*). This uncertainty then determines coalitional values' uncertainty, which represents the potential gain users can obtain from a particular coalition formation process.

As an example, let's imagine a group of professionals collaborating to complete a common project, *e.g.* carpenters and electricians coming together to build a house; every contractor has trade-specific skills about their types, for example characterizing their capabilities from incompetent to highly skilled. The overall payoff for the whole work obviously depends on the quality of the result, which is function of the ability of single workers and the potential synergies and conflicts among them. Agents don't know certainly the types of potential co-workers, so the possible coalition structures and values are determined by a probability distribution based on single agents' beliefs, which influence the formation process and the stability of the results. Every team must also decide which project undertake – for instance they may choose a simple house or an high-rise building – and such decision is heavily influenced by the actual types of worker belonging to the given coalition, but also by uncertainty about partner types as well as payoff division. Stability is treated *given the agents' beliefs*, *i.e.* a player shouldn't have the incentive to change the current state of the coalition, because every alternative configuration isn't expected to give a better payoff with respect to the one it currently hopes to receive. Thus, many alternative forms of stability can be defined, depending on the level of agreement required among single players' beliefs. The authors list three variants of *Bayesian Core (BC)* in [18], a *Weak Bayesian Core*, a *Strict Bayesian Core* and a *Strong Bayesian Core*, depending on the strictness one wants to enforce (looser definitions are super-sets of the stricter ones).

Given this model, Chalkiadakis [18, 19] also presents a *dynamic process with random proposers*, showing its convergence to the *Strong BC*, which can be also seen as an extension of the work of Arnold and Schwalbe [4]. The coalition formation process is covered through an heuristic algorithm [18, 20], using an iterative approach which updates beliefs during a bargaining process, in attempt to simulate a *perfect Bayesian Equilibrium (PBE)* [45].

Ieong and Shoham [35] introduced the notion of *worlds* in which games may be set. Every agent is assumed to partition the (private) information gained about the given world in *information sets*, each of which represents a world itself, indistinguishable from others from the agent's point of view.

Core-related topics are addressed defining an *ex-ante*, an *ex-interim* and an *ex-post* core – *i.e.* a set of stable outcomes before the disclosure of the world, after the world is drawn (referred to the awareness of each player regarding information sets), and after the true world is revealed, respectively.

ex-interim core is strictly related to the one introduced by Chalkiadakis, but the main concern is about the grand coalition; several other differences exist between the two presented approach, such as the possibility to consider an infinite level of mutual agent modeling – given partners' payoffs – in Ieong and Shoham's core.

2.3 Coalition generation on graphs

An alternative and very interesting approach in the field of coalition formation in characteristic function games is based on considering additional constraints that limits the total number of feasible coalitions, which are formalized with the help of a graph.

The idea of having coalitions whose potential membership is restricted by some kind of graph is an old one, since it naturally reflects many real-life situations. Work in *network formation*, in particular, following the seminal work of Myerson [48], has attempted to solve the problem of *progressively building* stable coalitional structures in networks, through the addition and removal of links among nodes [36]. That line of research, however, focused mostly on non-cooperative aspects of the coalition formation problem – for instance, by modeling the problem as a bargaining game or some other type of game in extensive form.

In *cooperative* environments, starting with the seminal work of Deng and Papadimitriou in [28], there has been work on graph-inspired *representations* for coalitional games. Such representations include Ieong and Shoham's marginal contribution nets [34], Bachrach *et al.*'s hypergraph-based representation to tackle coalition structure generation in skill games [9], and Brafman *et al.*'s work on identifying succinct coalitional game representations to model multiagent *planning* problems [16]. Moreover, there has been some work on *cooperative solution concepts* in graph-restricted games [17]. However, that work has *not* for the most part focused on the concept of the core, neither has it attempted to address the question of how core-stable coalitions emerge.

One exception is the notable work of Demange [27]. She proved that when the graph restricting a game is a tree there always exists an element in the core; and, moreover, presented a process that identifies a coalition structure and a payoff allocation that lie in the core. This work is fundamental in the context of this thesis, as the GDL algorithm presented in Chapter 4 can be seen as a natural extension, providing a a decentralized algorithm defined over a novel graphical representation of the coalition formation problem.

In a clear distinction to Demange's approach, the algorithm works on a graphical model representation which, while being a junction tree of the original graph, is nevertheless a tree whose nodes are *not* agents – but, rather, variable and function nodes of a factored graph. Thus, a clear mapping between the solution of such graphical model and the characteristic function game stable outcome is provided. Indeed, by extending GDL our approach draws a clear connection between graphical models and stable coalition formation. Importantly, GDL can be employed for optimal inference in general graphs by making use of well known tree decomposition techniques (*e.g.*, by compiling the original problem into a junction tree).

Thus, the field of coalition generation over games on graphs, despite of being poorly investigated, is a very promising one, leading to a convenient reduction of the computational complexity required by algorithms. In particular, the work covered by this thesis provides a clear intuition on how to address the problem of constructing core elements in graphs containing cycles.

Chapter 3

Stable Coalition Formation

The goal of this chapter is to formally introduce the theoretical concepts of cooperative game theory, used as basis for the work discussed later on in this thesis. We will first introduce the notion of *characteristic function game*, presenting some interesting special types of such games and discussing the standard solution concepts.

3.1 Characteristic function games

A *game* in the context of game theory is the abstract mathematical representation of a scenario in which some players – called *agents* – interact. These *agents* are usually referred as natural numbers, contained in a set $N = \{1, 2, 3, \ldots, n\}$. N also represents the *grand coalition* of the game, in a sense that it contains every player in the game; thus, a *coalition* is simply a subset of N. Note that, in common sense, this term refers to a group of people with some kind of commitment to a common task, but this assumption is not made in the following sections.

Definition 1 (Characteristic function game)**.** *A characteristic function game G is given by a pair (N, v), where $N = \{1, 2, 3, \ldots, n\}$ is a finite, non-empty set of agents, and $v : 2^N \to \mathbb{R}$ is a characteristic function, which maps every coalition $C \subseteq N$ to a real number $v(C)$, which is usually referred as the value of the partition C.*

This definition does not give any actual indication to the nature of v, which can be freely derived depending on the given scenario. In addition, note that v assign a value to the *whole* coalition, thus no explicit indication is given on how this value should be divided among single agents. The problem of finding proper partitions of the coalitional value is a fundamental topic in cooperative game theory, and assumes a central role in the field of *coalition formation activities by selfish rational agents*, in which agents are selfish entities whose interest is only maximizing personal interests.

An implicit assumption also made in characteristic function games is the possibility to share $v(C)$ amongst the members in C in any way the members of C choose. Formally, one refers to this property as *transferable utility games (TU games).* For example, a game in which the value of coalition can be represented by money belongs to this category.

Input representation

Due to the computational approach we focus on in this thesis, we must wisely analyze how these games have to be represented as inputs to computer programs. The obvious way suggested by Definition 1 is to explicitly list every coalition $C \subseteq N$, specifying the associated value $v(C)$. However, this representation is of size $\Omega(2^n)$, *i.e.* exponential in the total number of agents in the game, thus not convenient for practical use unless we are dealing with very small instances.

As shown by Megiddo [46], it is possible to model real-life interactions as cooperative games maintaining an efficient bound with respect to the size of the input representation, using some *implicit* forms of specification whose encoding size is $poly(n)$.

Positive payments

One usual assumption widely made in the game theory is to consider the value of the empty coalition $\emptyset$ as 0.

Moreover, is it possible to assume the value of each coalition as non-negative (*i.e.* agents form groups to make profit) or non-positive (*i.e.* agents form groups to share costs). In the current chapter the former scenario will be mostly used, even though all the property and definition can be easily adapted to the latter one.

In fact, one of the *characteristic function* – or *metric* – presented in Section 5.4, specifically the User Price Factor, assigns a negative real number to every coalition, due to its cost-related connotation.

The value assigned by the metric is a simulation of the cost the agents in the given team would pay if they bought the needed energy in a “smart” way in the *forward market.*

3.2 Outcomes

A key concept for cooperative game theory is the concept of *outcome.* An outcome of a characteristic function game is formed by two parts:

- A partition of the n agents into coalitions, called *coalition structure.*
- A *payoff vector*, which distributes the value of each coalition among its members.

Definition 2 (Coalition structure)**.** *Given a characteristic function game $G = (N, v)$, a coalition structure over N is a collection of non-empty subsets $CS = \{C^1, \dots, C^k\}$ such that:*

- $\bigcup_{j=1}^{k} C^j = N$
- $C^i \cap C^j = \emptyset$ *for any* $i, j \in \{1, \dots, k\}$ *such that* $i \neq j$

These two properties actually define a *partition* of the N set, in a sense that every agent must appear in one and exactly one coalition.

Definition 3 (Payoff vector)**.** *A vector $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ is a payoff vector for coalition structure $CS = \{C^1, \dots, C^k\}$ over $N = \{1, \dots, n\}$ if*

- $x_i \geq 0$ *for all* $i \in N$
- $\sum_{i \in C^j} x_i \leq v(C^j)$ *for any* $j \in \{1, \dots, n\}$

The second property is ofter referred as a *feasibility* requirement, as it does not enforce to distribute the *entire* amount of $v(C^j)$ among the members of coalition C^j.
Rather, one could require a stronger *efficiency* constraint, imposing that no amount of payment is wasted and *all* is partitioned among the players, *i.e.* $\sum_{i \in C^j} x_i = v(C^j)$ for any $j \in \{1, \dots, n\}$.

Definition 4 (Outcome)**.** *An outcome of G is a pair (CS, x) where CS is a coalition structure over G and x is a payoff vector for CS. Using the notation $x(C)$ we denote the total payoff $\sum_{i \in C^j} x_i$ of a coalition $C \subseteq N$ under x.*

Abusing this notation, we will write $v(CS) = \sum_{C \in CS} v(C)$, defining the concept of *social welfare* of the coalition structure CS.

3.3 Notable subclasses

Characteristic function games can be classified in a wide variety of subclasses, depending on the further properties that are required to hold. We'll now present two notable examples – *monotone* games and *superadditive* games – that are particularly interesting for the following sections of this thesis.

3.3.1 Monotone games

In many environments adding an agent to a given coalition can only increase – more formally, not decrease – the overall value of that particular group. This property is usually referred as *monotonicity*:

Definition 5 (Monotonicity)**.** *A characteristic function game $G = (N, v)$ is said to be monotone if it satisfies $v(C) \geq v(D)$ for every pair of coalition $C, D \subseteq N$ such that $C \subseteq D$.*

Many common types of games are monotonic, but it would be too restrictive to consider monotonicity as an universal property: for example, coordination and communication cost required after the addition of a new agent to a coalition may, in fact, reduce the total value, even when there is an increased productivity.

3.3.2 Superadditive games

A related and stronger property that is also fulfilled by many practical cases is *superadditivity*: when it holds, it is always profitable for two teams to join forces and form an unique coalition.

Definition 6 (Superadditivity)**.** *A characteristic function game $G = (N, v)$ is said to be superadditive if it satisfies $v(C \cup D) \geq v(C) + v(D)$ for every pair of disjoint coalition $C, D \subseteq N$.*

If the valued are assumed to be non-negative, superadditivity implies monotonicity, since $v(C) \leq v(D) - v(D \setminus C) \leq v(D)$, but the converse is not always true. As an example, consider a characteristic function that grows logarithmically with the coalition size, *i.e.* $v(C) = \log |C|$.
In real-world cases, non-superadditivity may be the result of anti-trust or anti-monopoly laws, which may prohibit large companies from working together at the expense of the market.
Given the aforementioned properties, if a game is superadditive there's no reason for the players to form multiple coalitions, since they can gain as much as possible joining together and form the *grand coalition*. Thus, the best outcome is of the form $(\{N\}, x)$ where x satisfies $\sum_{i \in N} x_i = v(N)$.
It is important to note that every non-superadditive game can be turned into superadditive computing, for every coalition, the maximum value that can be gained by splitting into smaller groups. Formally, given a game $G = (N, v)$, a new game $G^* = (N, v^*)$ can be defined by setting:

$$v^*(C) = \max_{CS \in \mathcal{CS}_C} v(CS)$$

for every coalition $C \subseteq N$, where $\mathcal{CS}_C$ denotes the space of all coalitions over C. The game so defined is referred as the *superadditive cover* of G, which can be easily shown as superadditive even if G is not. Intuitively, the value of a group C in G^* is the best amount the agents in C can obtain if they are free to choose their partners. Superadditive cover can be used to ease the study in terms of *stability*, as we will show in the following sections.

3.4 Solution concepts

Every coalition structure and payoff vector that respect the given definitions are valid outcomes of a particular game, but not all of them are equally desirable and have the same probability to occur. For example, a solution that assigns the entire value to just one agent, even if all the members of the group contributed equally to the team work, is not very appealing for the rest of the coalition, with respect to a solution that fairly distributes the value.

In general, a given outcome can be evaluated according to two sets of criteria: *fairness* and *stability.* The former refers to the ability of the given solution to represent every agent's contribution in terms of individual payments, while the latter focus on the incentives for the players to stay on the current coalition structure, rather than deviating from it.

3.4.1 The core

We will now introduce an important concept referred as the *core*, which plays a key role in the work covered by this thesis and in the field of *coalition formation activities by selfish rational agents* in general.

Consider a characteristic function game $G = (N, v)$ and an outcome (CS, x), recalling that $x(C)$ refers to the total payoff of C under x. If $x(C) < v(C)$ for some $C \subseteq N$, the agents in C would be incentivized to leave the current coalition structure and form a group on their own, distributing the the extra payoff earned from the deviation by sharing it equally among new players, *i.e.* setting:

$$x'_i = x_i + \frac{v(C) - x(C)}{|C|}$$

The considered outcome (CS, x) is said to be *unstable*, since some agents are incentivized to change it. The set of stable outcomes, where no subset of players has interest in deviating the current configuration, is called the *core of G* [31].

Definition 7 (Core)**.** *The core $\mathcal{C}(G)$ of a characteristic function game $G = (N, v)$ is the set of all outcomes (CS, x) such that $x(C) \geq v(C)$ for every $C \subseteq N$.*

It is easy to prove that stability implies efficiency, in a sense that any outcome in the core – therefore stable – maximizes the *social welfare*, *i.e.* the total payoff of all players. Thus, solutions in the core are more likely to occur when the game is played.

Theorem 1. *If an outcome (CS, x) is in the core of a characteristic function game $G = (N, v)$, then $v(CS) \geq v(CS')$ for every coalition structure $CS' \in \mathcal{CS}_N$*

The core of superadditive games

We have seen that for superadditive games the notion of outcome is only related to the payoff vector, since the best coalition structure is always the *grand coalition.* Is it possible that this constraint makes some outcomes in the core to be dropped, *i.e.* can the core of a superadditive game contain some outcomes where the grand coalition is not formed?
Theorem 1 imposes that if a game is *strictly superadditive* – *i.e.* $v(N) > v(CS)$ for any coalition structure CS that consists of two or more coalition – the aforementioned situation can not happen.
On the other hand, if $v(N) = v(CS)$ for some $CS \in \mathcal{CS}_N$, some outcomes in the form (CS, x) may actually be in the core. Anyway, for any such outcome it holds that $(\{N\}, x)$ is in the core as well, since:

$$\sum_{i \in N} x_i = \sum_{C \in CS} \sum_{i \in C} x_i = \sum_{C \in CS} v(C) = v(N)$$

so x is actually a payoff vector for N, proving the previous statement. Thus, without loss of generality, all outcomes where the grand coalition does not form can be safely ignored, because an equivalent (*i.e.* with the same payoff vector) grand coalition outcome must be in the core.
For superadditive games the notation can be simplified, defining the core as the set of vectors x such that:

- $x_i \geq 0$ for all $i \in N$
- $x(N) = v(N)$
- $x(C) \geq v(C)$ for all $C \subseteq N$

We remark that this definition holds only for superadditive games. If it is not the case, enforcing the grand coalition form may cause a loss of stability. Nevertheless, there's an important result [32] linking the concept of core of a game to its superadditive cover, which permits to view the stable outcomes as payoff vectors of the grand coalition without loss of generality.

Theorem 2. *A characteristic function game $G = (N, v)$ has a non-empty core if and only if its superadditive cover $G^* = (N, v^*)$ has a non-empty core.*

Many works in literature simply assumes superadditivity, justifying this hypothesis by saying that replacing the game with its superadditive cover is always possible. However this approach has been proved to be problematic by Aumann and Dreze [6] if cross-coalition transfers are not allowed. This *modus operandi* is also problematic since computing the characteristic function of the game G^* can be particularly difficult from a computational point of view.

Chapter 4

Coalition Formation with GDL

This chapter presents the work of Vinyals *et al.* [71] that stands as theoretical base for the real-world application presented in Chapter 5 (the main contribution of this thesis), where these algorithms will be applied to the domain of the energy market. Specifically, the goal is to allow a group of consumers to access and buy energy from the market acting as a single virtual consumer, but the formation of particular coalitions. Furthermore, the set of energy coalitions formed are required to be: (i) *optimal*, *i.e.*, the found outcome maximizes the collective value; and (ii) *stable* [52], *i.e.* no selfish agent can be incentivized to deviate from the given solution. As an additional restriction, the result has to fulfill some membership constraints represented by a graph, which models the existing knowledge relations.

In particular, we will now describe a novel graphical model representation scheme for *coalitional games* defined over graphs, and exploit this representation to devise decentralized algorithms leading to the formation of core-stable coalition structures. For games defined over trees, we propose an algorithm that allows agents to identify a stable outcome. That is, an optimal coalition structure along with a payoff allocation such that the resulting pair is in the core of the game. Similarly, for arbitrary graphs, an algorithm that either outputs a core element – or, alternatively, detects core emptiness – has been developed. Both algorithms build upon well-known message passing algorithms from the GDL family [58, 72] which has been extended to this setting. The aforementioned scenarios have a natural modelization in coalitional games, formally defined in Chapter 3. More importantly, a new representation for coalitional games over graphs will be introduced, allowing the execution of *decentralized* algorithms which return coalition structures that meet the properties mentioned above. Actually, the presented schema is the first in literature to deal with the goals of optimality *and* stability at the same time, which are usually treated separately.

4.1 Related concepts

We will now present some background notions required by the following sections, dealing with classical concepts of game theory and coalition formation literature. Some definition are slightly modified with respect to the ones given in the previous chapter, in order to adapt them to the context of coalition formation over graphs.

4.1.1 Coalitional games on graphs

In addition to standard definition of characteristic function games, we will now assume that feasible coalitions are determined by a graph G, with two fundamental properties:

- Each node of the graph represents an agent
- A coalition S is allowed to form if and only if every two agents in S are connected by some path in the subgraph induced by S.

We denote the set of agent nodes in G by $A(G)$. Given a set of agents $A \subseteq A(G)$ we also denote G_A as the subgraph of G induced by A and $G_{\setminus A}$ as the subgraph of G induced by all the agents $A(G)$ excluding those in A.

Definition 8 (Coalitional game on a graph). *A coalitional game CG on a graph G is a tuple $\langle A, v, F(G) \rangle$ where:*

- *$A = \{1, 2, 3, \ldots, n\}$ is a finite, non-empty set of agents.*
- *$F(G)$ is the set of all feasible coalitions – i.e. coalitions permitted to form given G.*
- *v is the characteristic function, defined for all coalitions in $F(G)$.*

Definition 9 (Coalition structure on a graph). *Given a game on a graph $\langle A, v, F(G) \rangle$, a coalition structure $CS = \{S_1, \ldots, S_k\}$ is a set of feasible ($\forall S \in CS : S \in F(G)$), exhaustive ($S_1 \cup \ldots \cup S_k \supseteq A$) and disjoint ($\forall S, S' \in CS : S \cap S' \cap A = \emptyset$) coalitions with respect to agents in A.*

While the restriction of CS to be exhaustive and disjoint with respect to A differs from the traditional in the literature, it is required later on the formalization of algorithms and proofs, to consider games in which the graph G contains agents nodes not in A ($A \subset G$). Observe that, if this is the case, this definition allows these *ghosts* agents to be in more than one coalition or in no coalition at all; while if $A = G$, the traditional setting persists. Also, all solution-related concepts – including the notions of payoff allocation, stability and core – are assumed to be defined in the classical way.

4.1.2 GDL message-passing algorithm

GDL is a general algorithm based on the message-passing paradigm, which permits to compute an objective function by the exploitation of the factorization of a global function into the combination of local ones. GDL is a widely used algorithm – in fields that go from computer vision to information theory – thanks to its ability of unifying a family of well-known techniques (*e.g.* Viterbi's, Pearl's belief propagation or Shafer-Shenoy algorithms to name a few).

Consider a function F, that depends on N variables, $\mathcal{X} = \{x_1, \dots, x_n\}$, and is defined as the combination of M factors $\mathcal{F} = \{f_1, \dots, f_m\}$ such that $F(X) = \bigotimes_{f \in \mathcal{F}} f_m(X_m)$ where $X_m \subseteq \mathcal{X}$ are the variables in the domain of f_m and $\bigotimes$ stand for the combination (called *joint*) operator. This global function can be encoded using a particular type of graphical model, called factor graph, a bipartite graph composed of two kinds of elements: variable nodes ($\mathcal{X}$) and function nodes ($\mathcal{F}$). Then the objective function is to find the assignment of variables in $\mathcal{X}$ that maximize the global function $X^* = \arg\max_{f \in \mathcal{F}} f_m(X_m)$.

To obtain optimality and convergence, GDL arranges the objective function in a junction tree. A junction tree for $\langle \mathcal{X}, \mathcal{F} \rangle$ is a tree of cliques that can be represented as a triple $\langle \mathcal{C}, \Psi, \mathcal{S} \rangle$ where:

- $\mathcal{C} = \{X_{C_1}, \dots, X_{C_n}\}$ is a set of cliques, where each clique X_{C_i} is a subset of variables $X_{C_i} \subseteq \mathcal{X}$.
- $\Psi = \{\psi_1, \dots, \psi_m\}$ is a set of potentials, one per clique, where a potential ψ_i is defined as the combination of a set of functions $\mathcal{F}_i \subseteq \mathcal{F}$, $\psi_i(X_{\psi_i}) = \bigotimes_{f \in \mathcal{F}_i} f(X_f)$.
- $\mathcal{S}$ is a set of separators, where a separator $Sep_{ij} \in \mathcal{S}$ is an edge between clique X_{C_i} and X_{C_j} containing their intersection, *i.e.* $Sep_{ij} = X_{C_i} \cap X_{C_j}$.

In addition, two more properties needs to be satisfied: *covering*, *i.e.* each potential domain is a subset of the clique to which it is assigned ($X_{\psi_i} \subseteq X_{C_i}$) and each function $f \in \mathcal{F}$ is included in exactly one potential, and *running intersection*, imposing that if a variable x_i is in two cliques X_{C_i} and X_{C_j}, then it must also be in all cliques on the path between them.

The main goal of GDL is to exploit the distribution of cliques to compute the objective function factored among them. Thus, GDL establishes a message-passing protocol for cliques, to share information about the variables in their scope.

In the single-vertex message-passing version – focused on computing the objective function at only one clique X_{C_i} – the execution is run over a rooted junction tree, in which each edge is directed toward the target clique X_{C_i}.

Running the *single-vertex GDL algorithm* over a rooted junction tree $\langle \mathcal{C}, \Psi, \mathcal{S} \rangle$, requires that each clique X_{C_i} exchanges a message $\mu_{i \to p}$ with its clique parent X_{C_p}, when, for the first time, it has received messages from all its children:

$$\mu_{i \to p}(Sep_{ip}) = \max_{X_{C_i} \setminus Sep_{ip}} \psi_i(X_{\psi_i}) \otimes \bigotimes_{j \in X_{Ch_i}} \mu_{j \to i}(Sep_{ji}) \tag{4.1}$$

where X_{Ch_i} stands for index of cliques's children of X_{C_i} in the rooted junction tree.

Once every children has sent its message, each clique X_{C_i} can proceed with the computation of its *state function* $s_i(X_{C_i})$ (also known as belief or *knowledge function*) as:

$$s_i(X_{C_i}) = \psi_i(X_{\psi_i}) \otimes \bigotimes_{j \in X_{Ch_i}} \mu_{j \to i}(Sep_{ji}) \tag{4.2}$$

When the algorithm has completed the execution, each state function $s_i(X_{C_i})$ (with $X_{C_i} \in \mathcal{C}$) represents its knowledge with respect to the variables $X^*_{C_j}$, reminding that $s_i(X_{C_i}) = \max_{X \setminus X_{C_i}} \bigotimes_{j \in D_i \cup \{i\}} \psi_j(X_{C_j})$, where D_i stands for the index of the descendants of X_{C_i} in the rooted junction tree $\langle \mathcal{C}, \Psi, \mathcal{S} \rangle$.

Finally, the assignment formed by optimal values for the local variables can be inferred by running the *value propagation* phase, recursively applying:

$$X^*_{C_j} = \arg\max_{C_j, Sep_{jp} = Sep^*_{jp}} s_j(X_{C_j}) \tag{4.3}$$

where p stands for the parent of X_{C_i} in the directed junction tree and Sep^*_{jp} stands for the values of Sep_{jp} variables already inferred on cliques up X_{C_i} in the rooted junction tree.

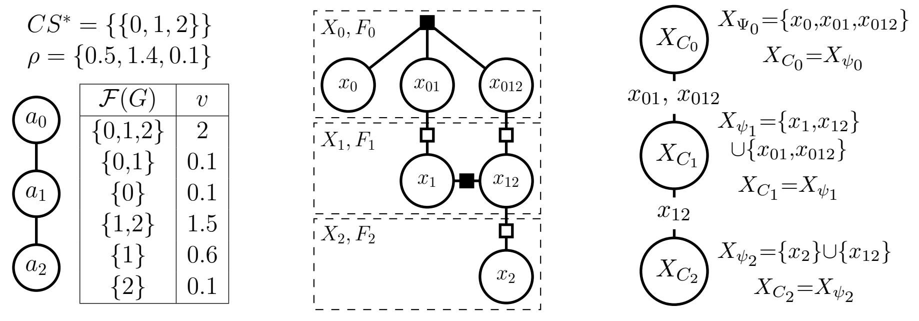
As a last step, $\bigcup_{X_{C_j} \in \mathcal{C}} X^*_{C_i}$ computes the optimal solution X^*.

4.2 Problem representation

We will now define a novel representation of a coalitional game on a graph G based on a factor graph, which gives a smart model of the interactions among agents in G, by arranging them in a pseudotree PT.

Definition 10 (Pseudotree)**.** *A pseudotree PT of a graph G over a set of agents A is a rooted tree with agents A as nodes and the property that for any pair of agents $i, j \in A$ if exists any path between them in G composed of agents not in A then i, j are on the same branch*[1] *in PT. In the case $A(G) = A$, this property reduces to: any two agents that share an edge in G are on the same branch in PT.*

[1] A branch stands for the path between a leaf node and the root node in PT



(a) Game over a tree (b) Representation (c) γ-Junction tree

Figure 4.1: Example of a) a game on an acyclic graph; b) a representation of (a); and c) a junction tree of (b).

Figure 4.1(a) shows an example of a game on an acyclic graph (which already represents a pseudotree rooted in agent a_0), its representation (Figure 4.1(b)) and the final junction tree (Figure 4.1(c)). Note that any pseudotree of G will form a line between agents in $A(G)$ because there's an edge in G between any pair of agents in $A(G)$ and hence, should be placed in the same branch. We refer to the set of agents' nodes in PT with $A(PT)$.
Then, given any agent $a_i \in A(PT)$, we denote Ch_i as its children, P_i as its parent, An_i as its ancestors, D_i as its descendants and PT_i as the subtree rooted at a_i in PT. Thus, in Figure 4.1(a), $Ch_1 = D1 = \{2\}$, $P_1 = 0$, $An_1 = \{0\}$ and PT_1 is a tree rooted at a_1 composed of agents a_1, a_2 sharing an edge between them. The defined pseudotree imposes an ordering among variables, which is propagated to the corresponding agents in A. Thus, given a game on a graph $CG = \langle A, v, F(G)\rangle$ and a pseudotree of G over A, it's possible to partition the set of feasible coalitions into $|A|$ disjoint sets $\{\mathbf{S_i}|a_i \in A\}$ – one per agent – where the set of coalition $\mathbf{S_i}$ contains all the feasible coalitions that include agent a_i but no agent up a_i in PT, $\mathbf{S_i} = F_{\{i\}}(G_{\setminus An_i})$.

Definition 11 (Required coalition)**.** *Given $CG = \langle A, v, F(G)\rangle$ a game on a graph and a pseudotree PT of G over A, we define the set of required coalitions for a coalition $S \in \mathbf{S_i}$, $Req(S)$, as $Req(S) = \bigcup_{j \in Ch_i} Req(S, j)$, while $Req(S, j)$ is recursively defined as:*

$$Req(S,j) = \begin{cases} \emptyset & \textit{if } S \cap A(PT_j) = \emptyset \\ S' \cup \bigcup_{k \in Ch_j} Req(S \setminus S', k) & \textit{otherwise} \end{cases} \tag{4.4}$$

where $S' = \arg\max_{\{S'' \in \mathbf{S_j} | S'' \subseteq (S \cap A(PT_j))\}} |S'' \cap S|$, that is the coalition in $\mathbf{S_i}$, strictly composed of variables in $S \cap A(PT_j)$, with maximum intersection with S.

Intuitively, the concept of *required coalition* is that an agent a_i willing to form one of its local coalition $S \in \mathbf{S_i}$ ensures the – exclusive – participation of agents in S down a_i in the tree ($S \cap D_i$), not directly through them but by means of the set of required coalitions $Req(S)$.
As an example, if agent a_0 wants to form the coalition $\{0, 1, 2\}$, it won't be necessary for him to negotiate directly with a_2, he can delegate this action to a_1, which will be responsible to form the coalition $\{1, 2\}$ (local to a_1). Ideally, one agent a_i has to negotiate the formation process only with its children Ch_i, allowing a more succinct representation of the problem, saving computational resources.

Next, the aforementioned concept will be summarized to give a novel representation of a coalitional game in terms of factor graph, which exploits this idea to efficiently capture the dependencies that emerge among agents, giving feasibility constraints to the formation of coalitions by means of a graph.

Definition 12 (Representation)**.** *Given $CG = \langle A, v, F(G)\rangle$ a game on a graph and a pseudotree PT of G over A, we define a factor graph representation of CG as $R(CG, PT) = \langle \mathcal{X}, \mathcal{F}\rangle$ where:*

- *$\mathcal{X} = \{X_1 \cup \ldots \cup X_{|A|}\}$ is a set of binary variables, one per feasible coalition, that are partitioned in $|A|$ disjoint sets, one per agent. Analogously to the concept of local coalitions, given an agent $a_i \in A$, its set of local variables X_i contains all the coalitions variables that include agent a_i but no agent up a_i in PT. Formally, $X_i = \{x_S | S \in F_{\{i\}}(G_{\setminus An_i})\}$*
- *$\mathcal{F} = \{F_1 \cup \ldots \cup F_{|A|}\}$ is a set of functions that are partitioned in $|A|$ disjoint sets, one per agent. Given an agent a_i, its set of local functions F_i is composed of:*
  - *$\{f_v(x_S) | x_S \in X_i\}$, a set of value functions, one per variable in X_i, where a function $f_v(x_S)$ returns the value of coalition S when $x_S = 1$, i.e. $f_v(x_S = 1) = v(S)$.*
  - *$f_u(X_i)$, a unique function that controls that one and only one of the variables X_i set to 1:*

$$f_u(X_i) = \begin{cases} 0 & \textit{if } \sum_{x_S \in X_i} x_S = 1 \\ -\infty & \textit{otherwise} \end{cases} \tag{4.5}$$

  - *A set of functions that capture the dependencies between each variable $x_S \in X_i$ and its set of requiring variables $X_S^{\setminus r}$ where $X_S^{\setminus r}$ stands for all the variables corresponding to coalitions that require $S(X_S^{\setminus r} = \{x_{S'} | S \in Req(S')\})$.*
    Formally, for each $x_S \in X_i$, the following two sets of variables are included:

- $\{f_r(x_S, x_{S'})|x_{S'} \in X_S^{\setminus r}\}$, *a set of require functions, one per requiring variable of* x_S. *Given a requiring variable* $x_{S'} \in X_S^{\setminus r}$, *the require function* $f_r(x_S, x_{S'})$ *enforces that* $x_{S'}$ *is activated only if its requested variable* x_S *is also activated, subtracting the value of* x_S *in this case.*

$$f_r(x_S, x_{S'}) = \begin{cases} -\infty & \textit{if } x_{S'} = 1 \wedge x_S = 0 \\ -v(S) & \textit{if } x_{S'} = 1 \wedge x_S = 1 \\ 0 & \textit{otherwise} \end{cases} \tag{4.6}$$

- $\{f_b(x_{S'}, x_{S''})|x_{S'}, x_{S''} \in X_S^{\setminus Req}\}$ *a set of blocking functions, one per each pair of variables that requested the same variable* x_S. *Intuitively, the set of blocking functions control that a most one of the coalitions variables that require* x_S *activates.*

$$f_b(x_S, x_{S'}) = \begin{cases} -\infty & \textit{if } x_S = 1 \wedge x_{S'} = 1 \\ 0 & \textit{otherwise} \end{cases} \tag{4.7}$$

Given the above definition of representation $R(CG, PT) = \langle \mathcal{X}, \mathcal{F} \rangle$ of a CG, its optimal solution is defined as $X^* = \arg\max_X \sum_{f \in \mathcal{F}} f(X) = \mathcal{F}(X)$. To obtain the optimal outcome from X^*, *i.e.* optimal coalition structure CS^*, we will next define a mapping Ω that maps any assignment of variables X in $R(CG, PT)$ to a coalition structure in CG.

Definition 13 (Mapping Ω). *Given* $R(CG, PT) = \langle \mathcal{X}, \mathcal{F} \rangle$ *a representation of a game on a graph,* Ω *is a function that maps any assignment for any set of variables* $X \subseteq \mathcal{X}$ *into a coalition structure* CS *composed of all coalitions* S *activated in* X *–* $x_S = 1 \in X$ *– for which it does not exist any other coalition that contains all agents in* S *activated in* X, *i.e.* $\nexists x_{S'} = 1 \in X$ *such that* $S \subset S'$.

The correctness of Ω, *i.e.* the representation of $R(CG, PT)$ where $CG = \langle A(G), v, F(G) \rangle$, is granted by Theorem 3 (which will not be proved here), which states that the optimal solution of $R(CG, PT)$ identify the optimal coalition structure in CG, $\Omega(X^*) = CS^*$.

Theorem 3. *Given a game on a graph* $CG = \langle A, v, F(G) \rangle$ *and a pseudotree* PT *of* G *over* $A(G)$, *the representation* $R(CG, PT) = \langle \mathcal{X}, \mathcal{F} \rangle$ *is correct. Thus, the following property holds:*

$$\forall_{X|\mathcal{F}(X) \neq \infty} \big(\Omega(X) \in \mathbf{CS} \wedge \mathcal{F}(X) = v(\Omega(X))\big)$$

Using this representation $CG = \langle A, v, F(G) \rangle$, we can now process it through the GDL message-passing algorithm presented in Section 4.1.2, allowing agents to obtain the optimal outcome for CG. Since GDL requires an input in a junction-tree form, we'll now define a particular one $R(CG, PT)$.

Definition 14 (Junction tree γ). *Let γ be a function that given a game on a graph $CG = \langle A, v, F(G)\rangle$ and a pseudotree PT of G over A maps them to a junction tree $\gamma(CG, PT) = \langle \mathcal{C}, \Psi, \mathcal{S}\rangle$, where:*

- $\Psi = \{\phi_i | a_i \in A$ *contains one potential per agent in A where $\psi_i(X_{\psi_i})$ is defined as the combination of functions F_i (F_i defined as in Definition 12).*

$$X_{\psi_i} = \bigcup_{f \in F_i} X_f = \{X_i \cup \bigcup_{x_S \in X_i} X_S^{\setminus r}\}$$

- $\mathcal{C} = \{X_{C_i} | a_i \in A\}$ *contains one clique per agent in A, where $X_{C_i} = X_{\psi_i} \cup \bigcup_{j \in Ch_i} X_{C_i} \setminus X_j$.*

- $\mathcal{S}$ *is a set of separators that contains one separator Sep_{ij} per pair of cliques X_{C_i} and X_{C_j} such that a_j is parent of a_i in PT. As in definition, a separator Sep_{ij} includes the intersection of cliques X_{C_i}, X_{C_j} and hence, in this case $Sep_{ij} = X_{C_i} \setminus X_i$.*

In these definitions we assume that $\gamma(CG, PT)$ distributed the cliques in a way such that each agent $a_i \in A$ is assigned to a single clique X_{C_i}. Thus, the GDL message-passing scheme among cliques is indeed a message-passing scheme among agents in PT. Therefore, the properties given so far allow us to obtain the optimal solution X^* of $R(CG, PT)$ and hence, by mapping Ω, the optimal coalition structure in $CG = \langle A, v, F(G)\rangle$.

More formally, once the execution of GDL is over, the *state function* of each agent $a_i \in PT$ recovers the value of the optimal coalition structure $CS^{*,i}$ of a subgame CG^i:

$$\max_{X_{C_i}} s_i(X_{C_i}) = \max_{X_{C_i}} \max_{\mathcal{X} \setminus X_{C_i}} \bigotimes_{j \in A(PT_i)} \psi_i(X_{C_i}) = v\left(CS^{*,i}\right) \tag{4.8}$$

where $CS^{*,i}$ stands for the best coalition structure that agent a_i and agents down in PT can form, using a subset of coalitions of CG that do not contain any agent up a_i in PT.

4.3 Algorithms

4.3.1 Stable coalition formation in trees

The *SCF-Trees*, outlined in Algorithm 3, has three phases: *preprocessing*, *demand propagation* and *offer propagation*.

First, agents start with a preprocessing phase (line 2) that compiles the problem into a junction tree of the compact representation (see section 4.2) to be used in the following two phases. In the preprocessing phase, agents arrange the graph into a pseudotree PT.

In Figure 4.1, the tree of the game in Figure 4.1(a) is rooted at a_0. Then each agent a_i creates one binary variable x_S, along with a function with its value v_S, for each possible coalition that a_i can join with agents down the tree. For example, in Figure 4.1 a_0 creates three variables, namely x_0, x_{01}, x_{012}. Notice that the set composed of all these variables is X_i (e.g. $X_0 = \{x_0, x_{01}, x_{012}\}$). Then, each agent a_i waits for its parent's message that contains a set of tuples where each tuple is composed of a coalition up the tree x_S that require a set of agents S' in $A(PT_i)$. In an acyclic graph, these variables are singly composed of all parent's variables that contain a_i (X_{pi}). Then, for each variable $x_S \in X_{pi}$, each agent a_i creates a require function between x_S and $x_{G_{S'} \cap G_{A(PT_i)}}$.

Thus, in Figure 4.1, a_1 after receiving $\{\langle x_{01}, 1\rangle, \langle x_{012}, \{1, 2\}\rangle\}$ from a_0 creates two require functions, namely $r(x_{01}, x_1)$ and $r(x_{012}, x_{12})$. Then, each agent a_i communicates to each of its children $a_j \in Ch_i$ a message that contains for each set of variables that include a_j, $x_S \in X_{ij}$, a tuple with x_S and the set of agents from S reachable from a_j, $S \cap A(PT_j)$. The intuition is that each agent a_j would act as a mediator negotiating the payment demanded by agents down PT_i to join a coalition x_S with a_i.

Finally, each agent computes its local function f_i as the combination of: (i) function u_i that controls that one and only one of X_i coalition variables is activated (set to 1); (ii) value functions $\vec{v}$; and (iii) require functions $\vec{r}$ (note that in acyclic graphs the set of blocking functions is empty). $Sep_{ji} = \{x_S \in X_i | S \ni j\}$. When an agent a_i receives the set of coalitions Sep_{pi} from its parent, a_i creates a *require* relation for each $x_S \in Sep_{pi}$ and the coalition $x_{G_S \cap G_{T_i}} \in X_i$ (that is the coalition composed of a_i and all agents down a_i in T also included in S). Hence, when *preprocessing* is over, each agent a_i has computed:

- X_i, the set of its local variables.
- $\bigcup_{x_S \in X_i} X_S^{\setminus r}$, the set of all the variables of a_p that includes a_i.
- f_i, its local function.

After this first processing phase is over, *SCF-Trees* runs two phases:

- A *demand propagation phase* (`DemandPropagation`), in which agents exchange demand messages up the tree.
- An *offer propagation phase* (`ValuePropagation`), in which agents exchange offer messages down the tree.

When executing the *demand propagation* protocol, each agent a_i waits until receiving a *demand* message from each of its children $a_j \in Ch_i$ (lines 1-3 of Algorithm 1).

Algorithm 1 `DemandPropagation`

```
Each $a_i$ knows $\langle a_p, X_i, Ch_i, \psi_i \rangle$ and runs:
1: for all $a_j \in Ch_i$ do
2:     Wait for the demand message $d_{j \to i}(Sep_{ji})$ from $a_j$;
3: end for
4: $p_i(X_{C_i}) = \psi_i(X_{\psi_i}) \otimes \bigotimes_{j \in Ch_i} d_{j \to i}(Sep_{ji})$;
5: $\rho_i = \max_{X_{C_i}} p_i(X_{C_i})$;
6: if $a_i$ is not the root then
7:     $d_{i \to p}(Sep_{ip}) = \max_{X_i} p_i(X_{C_i}) - \rho_i$;
8:     Send $d_{i \to p}(Sep_{ip})$ to $a_p$;
9: end if
```

The *demand* message that a_j sends to its parent a_i contains for each of the coalition variables x_S in Sep_{ji} the amount required for a_j and agents down a_j in PT to join coalition S. For example, in Figure 4.1(b), agent a_0 waits until receiving the *demand* message from a_1 that contains a function over variables in their separator $\{x_{01}, x_{012}\}$. Upon receiving all *demand* messages, each agent a_i computes its *payment function* p_i as the combination of function ψ_i, that combines its local utility and restrictions for variables X_{ψ_i}, and the *demand* messages from the children, that subtract the amount required for agents down a_i (line 4). Then, a_i computes its *payment* ρ_i as the highest payment a_i can get on any of its variable configurations (which stand for local a_i's coalitions). After that, if agent a_i has a parent in PT (lines 6-8), a_i sends a message to its parent a_p that summarizes its *payment function* p_i over all possible configurations of variables in Sep_{ip}, subtracting its own *payment* ρ_i. The result of this summarization is for each coalition variable x_S in Sep_{ip} the payment required from agents in $A(PT_i)$ to join S. Thus, in Figure 4.1(b), a_1 summarizes its payment function p_1 over variables $\{x_{01}, x_{012}\}$ (filtering out $X_1 = \{x_1, x_{12}\}$). Then, agents proceed to execute the value propagation phase by executing the *value propagation* protocol (whose pseudocode is depicted in Algorithm 2).

During the *value propagation* protocol, each agent a_i waits until receiving a *value* message from its parent a_p (lines 1-3). Such message specifies whether a_p is willing to pay the amount requested by agents in $A(PT_i)$ so that they will join a coalition $x_S \in Sep_{ip}$. For example, in Figure 4.1(b), a_1 waits until receiving a message from a_0 with its decision with respect to coalitions x_{01}, x_{012}. Then, a_i computes the best coalition it can join given the decision of its parent a_p (line 4). If a_p decided to create a coalition $x_S \in Sep_{ip}$ and thus is willing to pay agent the amount requested by a_i and other agents (S') down the tree to join S, then the best coalition for a_i is $x_{S'}$. In contrast, if a_p does not activate any variable $x_S \in Sep_{ip}$, a_i will select the best coalition a_i can join that includes itself and some agents down a_i in the tree. Finally, agent a_i sends an *value* message to each of its children $a_j \in Ch_i$ that contains which coalition $x_S \in Sep_{ij}$ a_i accepted to create, if any (line 5-7).

Algorithm 2 ValuePropagation

```
Each $a_i$ knows $\langle a_p, p_i(X_{C_i}), Ch_i, \{Sep_{ji}\}\rangle$ and runs:
if $a_i$ is not the root then
    Wait for $Sep^*_{ip}$ from $a_p$;
end if
$X^*_{C_i} = \arg\max_{X_{C_i}, Sep_{ip}=Sep^*_{ip}} p_i(X_{C_i})$;
for all $a_j \in Ch_i$ do
    Send $Sep^*_{ji} \leftarrow X^*_{C_i} \cap Sep_{ji}$ to $a_j$;
end for
```

Algorithm 3 SCF-Trees

```
Each $a_i$ knows $\langle v, F_{\{i\}}(G)\rangle$ and runs:
Preprocessing phase
Pseudotree arrangement: run token based mechanism that arrange agents into a pseudotree $PT$ – At completion, $a_i$ knows $a_p, Ch_i, PT_i$;
Junction tree arrangement: run message-passing algorithm that arrange agents into a junction tree $\gamma$ – At completion, $a_i$ knows $X_i, \psi_i$;
Demand propagation phase
$\langle p_i(X_{C_i}), \rho_i, \{Sep_{ji}\}\rangle \leftarrow$ DemandPropagation$(a_p, X_i, Ch_i, \psi_i)$;
Value propagation phase
$X^*_{C_i} \leftarrow$ ValuePropagation$(a_p, p_i(X_{C_i}), Ch_i, \{Sep_{ji}\})$;
```

4.3.2 Stable coalition formation in graphs

The aforementioned algorithm can be extended to the case of general graphs, by the execution a n-iteration of the tree-based case. The process starts from the transformation of the input to obtain a pseudotree through a DFS search.

Pseudotree construction through DFS

SCF-Graphs operates on a variable ordering which is given by a DFS (Depth-First Strategy) arrangement of the problem graph. Formally, DFS is a search strategy that progresses by expanding the first child node of the search tree that appears and thus going deeper and deeper until a goal node is found, or until it hits a node that has no children. Then the search backtracks, returning to the most recent node it hasn't finished exploring. In particular, a DFS traversal of the graph is done using Algorithm 4, which distributedly computes, for each agent a_i, its parent p_i, its children ch_i, its pseudoparents pp_i and its pseudochildren pch_i, assuming the list of adjacent nodes ngh_i is provided. In addition, heuristics can be used to generate a specific kind of pseudotree, since multiple – corrected – ones can result from the same input graph, depending on the ordering used to iterate over adjacent nodes. In this context, we are interest in the generation of DFS arrangement with the lowest induced width [12].

Algorithm 4 DFS traversal on a graph

```
$nv_i \leftarrow ngh_i$
if $a_i \neq root$ then
    $\langle token, sender \rangle \leftarrow$ Wait_Token()
    $p_i \leftarrow sender$
    $pp_i \leftarrow (token \cap ngh_i) \setminus p_i$
    $nv_i \leftarrow nv_i \setminus (pp_i \cup p_i)$
    for all $a_j \in pp_i$ do
        $pch_j \leftarrow pch_j \cup a_i$
        $nv_j \leftarrow nv_j \setminus a_i$
    end for
end if
$token \leftarrow token \cup a_i$
while $nv_i \neq \emptyset$ do
    $a_j \leftarrow pop(nv_i)$
    $ch_i \leftarrow ch_i \cup a_j$
    Send_Token($token$,$a_j$)
    $\langle token, sender \rangle \leftarrow$ Wait_Token()
end while
$token \leftarrow token \setminus a_i$
if $a_i \neq root$ then
    Send_Token($token$,$p_i$)
end if
```

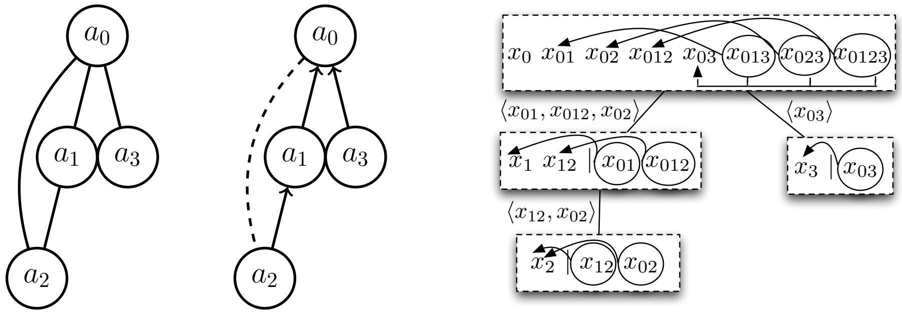
Definition 15 (Induced width)**.** *An ordered graph is a pair (G, d), where G is a undirected graph and $d = (X_1, \ldots, X_n)$ is an ordering of the nodes. The width of a node in an ordered graph is the number of neighbors that precede it in the ordering. Thus, the induced width is the width of the induced ordered graph obtained as follows: for each node, from last to first, in d, its preceding neighbors are connected to a clique. The induced width of a graph, is the minimal induced width over all ordering. The induced width is also equal to the treewidth of a graph.*

In a centralized setting, the most common heuristics for this problem are the following: the *maximum degree* [69] and the *min-fill heuristic* [39]. The latter does not produce general pseudotree orderings (much less DFS ones), and is difficult to implement in a distributed setting because it would require coordination at each step between all the remaining agents.

In what follows we only cover the *maximum degree* heuristic, because it is the most suitable for the requirements of real application described in Chapter 5. This heuristic simply consists in considering the set of adjacent, not yet visited nodes nv_i as an ordered list instead of a set, which goes from the most connected node to the least one.

Execution of *SCF-Graphs*

In this section we introduce the *SCF-Graphs* algorithm (whose pseudocode is outlined in Algorithm 5) that allow agents to self-organize into stable coalition structures or detect the core emptiness on any graph-restricted coalitional game. *SCF-Graphs* requires a preprocessing phase, in which agents compile the game $CG = \langle A(G), v, F(G) \rangle$ into a pseudotree PT and generate leading coalitions and requiring mappings as described in Section 4.2. As a result of this preprocessing each agent a_i in SCF-Graphs starts knowing its parent (a_p) and children (Ch_i) in the PT as well as its leading coalitions L_i, the requiring mapping Req_i and the characteristic function v. Building on this, each agent a_i initializes its local decision problem (line 1) by creating its set of local decision variables X_i, including leading and requiring sets, and its value function V_i.


a_0
a_1
a_3
a_2
a_0
a_1
a_3
a_2
x_0 x_{01} x_{02} x_{012} x_{03} x_{013} x_{023} x_{0123}
$\langle x_{01}, x_{012}, x_{02} \rangle$
$\langle x_{03} \rangle$
x_1 x_{12} | x_{01} x_{012}
x_3 | x_{03}
$\langle x_{12}, x_{02} \rangle$
x_2 | x_{12} x_{02}

(a) G (b) PT (c) Clique tree.

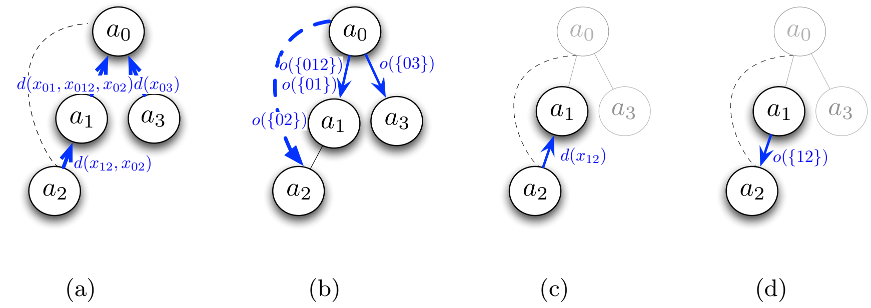
Figure 4.2: Example of (a) a graph with a cycle (G); (b) a pseudotree PT of G; and (c) a clique tree of PT.

Then, *SCF-Graphs* runs into three phases:

- *Demand propagation* phase
- *Main* phase
- *Offer propagation* phase

during which agents exchange two kinds of messages: (i) *demand messages*, to exchange demands through tree-edges for requiring coalitions whose activation is required by some agent up to PT; and (ii) *offer messages*, to propagate offers through tree- and pseudo- edges for required coalitions to agents down PT.

In what follows we describe the main phases of the *SCF-Graphs* algorithm using the trace in Figure 4.3 of an execution of *SCF-Graphs* over the the pseudotree arrangement of the CG cyclic graph game in Figure 4.2(a).



(a) (b) (c) (d)

Figure 4.3: Messages exchanged at different steps of the execution of *SCF-Graphs* over the clique tree in Figure 4.2.

Each agent a_i starts the *demand propagation phase* waiting until receiving a *demand* message from each of its children $a_j \in Ch_i$ (line 3). Figure 4.3(a) depicts the demand messages exchanged during this phase. Thus, agent a_1 waits until it receives the *demand* message from a_2 that contains a function over feasible configurations of variables $X_{21} = \{x_{12}, x_{02}\}$ before computing the demand message for its parent a_0. To compute its demand for its parent, each agent needs first to compute its *payment function* and its *expected payment* (Procedure `demand_propagation`). Each agent a_i computes its payment function, $P_i(X_i^P)$, as the combination of its value function, $V_i(X_i)$, and the (last) *demand* messages from its children that subtract the amount required for agents down a_i.

Thus, in Figure 4.3(a), agent a_1 will combine its value function $V_1(X_1)$ with the demand received from a_2. Notice that the payment function depends on a set of variables X_i^P, that is the set of variables of a_i (X_i) and the set of variables in the scope of any demand received from a child $a_j \in Ch_i$ (X_{ji}). Thus, in Figure 4.3(a), the payment function of a_1 will depend not only on its variables X_1 but also on some alien coalition variable, namely x_{02}, that is propagated from a_2. Then, a_i computes its expected payment ρ_i as the highest payment he can get on any of its leading coalitions, that is for any configuration of a variable in X_i^L. Thus, in Figure 4.3(a), a_1 computes the highest payment he gets among variables x_1,x_{12}. After that, if agent a_i is not the root, a_i sends a message to its parent a_p that summarizes its payment function P_i over all possible configurations of variables in X_{ji} (filtering out its leading variables X_i^L) and subtracts its expected payment ρ_i. Thus, in Figure 4.3(a), a_1 summarizes its payment function P_1 over variables $\{x_{01}, x_{012}, x_{02}\}$ (filtering out its leading variables $X_1^L = \{x_1, x_{12}\}$). The result of this summarization is, for each coalition variable x_S in X_{ip}, the demand that agent a_j will need to satisfy to agents in $A(PT_i)$ to ensure their participation in S.

In case of an alien variable $x_S \notin X_i$, this payment is not simply the amount demand by agents down PT_i but considers the potential loss of a_i when restricting to coalitions that does not contain any agent in S (the maximum offer a_i is going to offer to agents in S to join). Thus, in Figure 4.3(a), the demand of agent a_1 to its parent a_0 for x_{02} is the maximum between the amount demanded from a_2 to join this coalition and the potential loss of a_1 from moving to a coalition $(\{1\})$ that does not include a_2.

Once the *demand propagation* phase is over, agents start the *main phase* (lines 5-10) in which each agent a_i iteratively checks (line 5): (C_1) if he has received an offer for all requiring coalitions $(X_i^L \neq X_i)$; and (C_2) if its payment depends on some alien coalition, $(X_i^P \neq X_i)$. If any of these two conditions is not satisfied, it means that agents needs to either receive new demands from children (that get rid of alien variables) or offers from their ancestors (that make an offer for some requiring coalition), so agent a_i waits for messages. Thus, in Figure 4.3(a)-(d), after the initial demand propagation phase, agent a_3 waits until receiving an offer message from its parent a_0, whereas agent a_1 needs to wait for a message from its parent a_0 and a new demand from its child a_2 that gets rid of the alien coalition x_{02}. When agent a_i receives an offer message over a coalition S, he adds requiring variable x_S to its set of leading variables X_i^L (given the offer, now a_i is the leader of coalition S) and updates its value function and optimal decision over S. Additionally, a_i checks if the offer messages comes from its parent in the pseudotree, in which case the agent marks itself as the new root. Finally, independently if a new offer or demand is received the agent re-runs the demand_propagation procedure recomputing its payment function, its payment and the demand for its parent. When $C_1 \wedge C_2$ is satisfied (notice that for the root they are always satisfied), each agent a_i starts the *offer propagation phase* (lines 9-39). For example, in Figure 4.3, a_0, as root, starts the offer propagation phase just after the initial demand propagation phase. First, each agent a_i computes its optimal local configuration x_i^* given the already inferred decision of its ancestors An_i over requiring coalitions (line 9) and retrieves the optimal coalition S_i^* corresponding to this optimal configuration (line 10). Next, each agent locally checks for the emptiness of the core (lines 11-12). The core is detected as empty if a_i detects that its payment conditioned to the decision of ancestors $(P_i(x_i^*))$ is less than the best payment he can get considering all its local configurations (ρ_i). Finally, each agent a_i can start computing the offers of coalitions requiring some agent down the tree. For each of its leading coalitions $S \in L_i$, the offer of agent a_i for S is distributed among independent offers for S's required coalitions $(Req_i(S))$. Each agent a_i computes the offer of a coalition S for a required coalition S' as the value of its payment function for S minus the $a_i's$ payment (notice that this includes demands from all agents) canceling the difference between the joint demand for $x_{S'}$ and coalitions still not offered $(x_{S' \cap o})$ and the joint demand of coalitions still not offered (x_o).

Thus, in Figure 4.3(b), a_0 computes the offer of $\{0, 1\}$ for $\{0, 1\}$ (the only requiring coalition of $\{0, 1\}$ in X_1 is itself) as the value of its payment function for $\{0, 1\}$ ($P_0(x_{01})$) minus its payment (ρ_0) whereas subtracting the demand from coalition $\{0, 1\}$. Notice that whenever S is a basic coalition the amount offered to S is just the value of agent a_i for S minus a_i's payment. In contrast, when the coalition is composite the offer is distributed among more than one required coalitions. For example, in Figure 4.3(b), the offer of a_0 for its leading coalition $\{0, 1, 3\}$ is distributed among its requiring coalitions $\{0, 1\}$ and $\{0, 3\}$. After computing offers, each agent a_i sends for each of its basic coalitions $\{S \in L_i | Req_i(S) = \{S\}\}$, an offer message to the agent in $S \setminus \{i\}$ with higher position in PT (lines 36-39). At the end of the algorithm (line 41), agents run a distributed procedure[2] to propagate the core status across the graph (emptiness of the core is propagated through the whole graph).

In this way, *SCF-Graphs* builds an optimal and stable solution, if one exists, otherwise the emptiness of the core is detected and propagated among all agents, as stated by Theorem 4.

Theorem 4. *Given a game on a graph $CG = \langle A(G), v, F(G) \rangle$, if the core of CG is not empty, the outcome produced by SCF-Graphs belongs to the core of CG; otherwise SCF-Graphs outcomes the optimal coalition structure of CG detecting the emptiness of the core.*

[2] Any distributed algorithm used for convergence detection can be used for that purpose.

Algorithm 5 SCF-Graphs

```
Each $a_i$ knows $\langle a_p, X_i, \psi_i(X_{\psi_i}), Req\_i, \rho_{min} \rangle$
$\texttt{EMPTY\_CORE}_i \leftarrow false$
$\forall x_S \in X_{\psi_i} : o(x_S) \leftarrow 0$
Wait for a demand message from each child $a_j \in Ch_i$ $(d_{j \to i}(Sep_{ji}))$
$\langle p_i(X_{C_i}), \rho_i \rangle \leftarrow \texttt{demand\_propagation()}$
while $X_{C_i} \neq X_{\psi_i} \vee X_i \neq X_{\psi_i}$ do
    handle_messages()
    $\langle p_i(X_{C_i}), \rho_i \rangle \leftarrow \texttt{demand\_propagation()}$
end while
$X_i^* = \arg\max_{X_i, X_i^*} p_i(X_i)$
$S_i^* = \gamma(X_i^*)$
if $p_i(X_i^*) \neq \rho_i$ then
    $\texttt{EMPTY\_CORE}_i = true$
end if
for all $x_S \in X_i$ do
    if $Req_i(x_S) = \emptyset$ then
        $\texttt{OFFER}_i \leftarrow p_i(x_S = 1, X_i = 0) - \rho_i - \sum_{j \in Ch_i} d_{j \to i}(x_S = 1, X_i = 0)$
        if $S_i^* = S$ then
            $\texttt{OFFER}_i \leftarrow \texttt{OFFER}_i + \rho_{min} \cdot |S \setminus \{a_i\}|$
        end if
        $o(x_S) \leftarrow max(o(x_S), \texttt{OFFER}_i)$
    else
        $X_{\setminus o} \leftarrow Req_i(x_S)$
        for all $x_{S'} \in Req_i(x_S)$ do
            $X_{\setminus o} \leftarrow X_{\setminus o} \setminus \{x_{S'}\}$
            $\texttt{h}_i = \sum_{j \in Ch_i} d_{j \to i}(x_S = 1, x_{S'} = 1, X_{\setminus o} = 1, X_i = 0)$
            $\texttt{k}_i = \sum_{j \in Ch_i} d_{j \to i}(x_S = 0, x_{S'} = 0, X_{\setminus o} = 1, X_i = 0)$
            $\texttt{OFFER}_i \leftarrow p_i(x_S = 1, Req_i(x_S) = 1, X_i = 0) - \rho_i - \texttt{h}_i + \texttt{k}_i$
            if $S_i^* = S$ then
                $\texttt{OFFER}_i \leftarrow \texttt{OFFER}_i + \rho_{min} \cdot |S' \setminus \{a_i\}|$
            end if
            $o(x_{S'}) \leftarrow max(o(x_{S'}), \texttt{OFFER}_i)$
        end for
    end if
end for
$\rho_i \leftarrow \rho_i - \rho_{min} \cdot |S_i^* \setminus \{a_i\}|$
for all $x_S \in X_i | Req_i(x_S) = \emptyset$ do
    Let $a_j$ be the agent in $S \setminus \{a_i\}$ with highest level
    $o_{i \to j} \leftarrow o_{i \to j} \cup \langle x_S, o(x_S = 1), x_S^* \rangle$
end for
Send $\texttt{OFFER}_i$ messages to respective agents
$\texttt{EMPTY\_CORE}_i \leftarrow \texttt{propagate\_core\_status(EMPTY\_CORE}_i\texttt{)}$
if $\texttt{EMPTY\_CORE}_i = true$ then
    $\rho_i \leftarrow \infty$
end if
```

Algorithm 6 `handle_messages`

```
Wait_for_messages()
for all For each message received do
    if offer $o_{j \to i}(Sep_{ji})$ message received then
        if $a_j = a_p$ then
            $a_i \leftarrow root$
        end if
        for all $\langle x_S, o(x_S), x_S^* \rangle \in o_{j \to i}$ do
            $X_i \leftarrow X_i \cup x_S$
            $o(x_S = 1) \leftarrow o(x_S)$
            $X_i^* \leftarrow X_i^* \cup (x_S = x_S^*)$
        end for
    end if
end for
```

Algorithm 7 `propagate_core_status`

```
Wait from EMPTY_CORE$_j$ from each child $a_j \in Ch_i$
if $\exists k$ : EMPTY_CORE$_k = true$ then
    EMPTY_CORE$_i = true$
end if
if $a_i \neq root$ then
    Send EMPTY_CORE$_i$ to $a_p$
    Wait from message from $a_p$
    EMPTY_CORE$_i \leftarrow$ EMPTY_CORE$_p$
end if
Send EMPTY_CORE$_i$ to all children $Ch_i$
```

Algorithm 8 `demand_propagation`

```
$p_i(X_{C_i}) = \psi_i(X_{\psi_i}) \otimes \bigotimes_{j \in Ch_i} d_{j \to i}(Sep_{ji}) \otimes \bigotimes_{x_S \in X_i^{\backslash r}} o(x_S)$
$\rho_i = \max_{X_i} p_i(X_i)$
if $a_i \neq root$ then
    $d_{i \to p}(Sep_{ip}) = \max_{X_i} p_i(X_{C_i}) - \rho_i$
    Send $d_{i \to p}(Sep_{ip})$ to $a_p$
end if
```

Chapter 5

Application to the Energy Market

In this chapter we will show how the aforementioned theoretical notions and solution techniques – especially the GDL-based message-passing algorithm – can be applied to a real-world case study: the energy market. We will focus on a scenario in which a group of energy customers have a demand of energy supplies (*e.g.* electrical power), and need to buy them in a way that minimizes their costs, forming *coalitions*. We assume that energy customers are organized in a social network. The social network models the knowledge relationships between energy customers and restricts the structure of possible coalitions they can form. This network can be trivially represented by a graph with certain particular properties, formally discussed in Section 5.2.

It is easy to see that this setting has a natural modelization in *coalitional games on graphs* defined in Section 4.1.1, in which players are considered to be *selfish*, thus setting us in a non-cooperative environment. As said before, real energy customers have little interest in increasing system gain, as their main concern is to maximize their own utility, with the simple goal of having a cheaper bill. Therefore, the concept of *stability* introduced in Section 3.4.1 assumes a key role, since it is not possible to impose a solution "from above", as it would be immediately rejected by the agents. Rather, the goal is to find an appropriate distribution of payments that incentives single energy customers to maintain the current structure, as any deviation would worsen their payoff, making energy more expensive.

5.1 The Problem

In all current electricity grids this balance is achieved by varying the supply-side to continuously match demand. The amount of demand required on a continuous basis is usually carried by the baseload stations owing to low cost generation, efficiency and safety.

However, these stations are slow to fire up and cool down, so they are not able to match the peakload periods that exceed this baseload that require, in contrast, the use of expensive, carbon-intensive, peaking plant generators. Although only running when there is high demand, these peaking plant generators are responsible of most part of consumers electricity bill.

Along this line, the vision of the *smart grid* includes demand-side peak-shaving strategies such as real-time pricing or profile's based tariffs to encourage consumption such that the peaks on demand are *flattened* [1]. A flattered demand results in a more efficient grid not only with lower carbon emissions but also with lower prices for consumers. Hence, some works [57,70] focused on techniques that flatten individual consumer demand by automatically controlling home domestic or micro-storage devices. Unluckily, since each consumer independently optimizes its own consumption, the effectiveness of this approach has a clear limit on the consumer's restrictions and comfort (*e.g.* it will be unavoidable to get a consumption peak in the non-working hours of consumers).

Against this background, the problem of how the grid efficiency can be further improved from a social perspective is investigated. In particular, we explore the idea of allowing consumers to join into coalitions with other consumers with complementary energy needs. Then, a group of consumers can act in the market as a single virtual consumer with a flattened demand for which it gets much better prices. As part of the smart grid community, electricity consumers have already access to smart meters that allow them to monitor its (load) energy profile[1] in an hour-day basis (Figure 5.1). Thus, the energy profile of a customer a_i can be represented as a vector $E_i = \{e_i^1, \ldots, e_i^T\}$ where e_i^t is the amount of energy consumed at time slot t and T is the length of the considered interval.

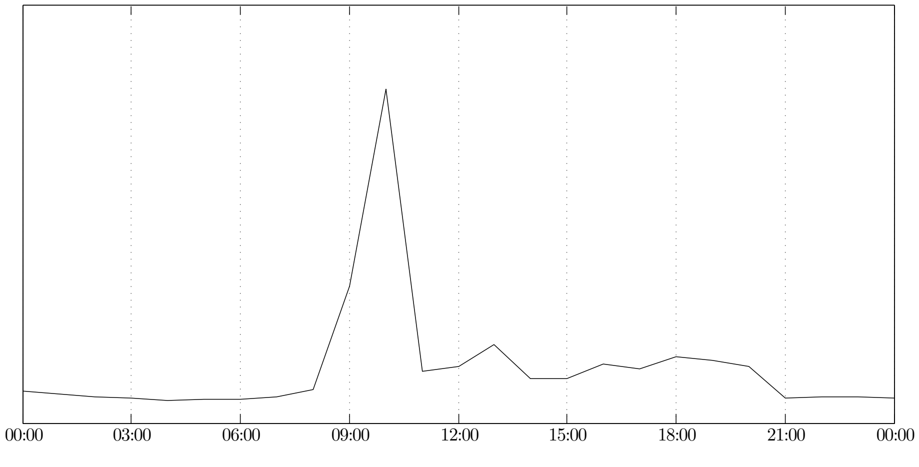


Figure 5.1: Typical energetic profile over 24 hours

[1] The load energy profile is a graph of the variation in the electrical load versus time.

5.2 Social networks

Given the huge recent success of social networks (*e.g.* at the time of writing Facebook has more than 500 millions users), consumers can use them as free interaction tools to self-organize into energy coalitions.

Social networks not only provide a way of interaction among energy consumers but also restrict coalition membership by reflecting realistic barriers to the formation of certain coalitions. In particular, consumers may not want to join coalitions with unknown consumers for which they do not have any source of trust regarding their reported profiles or their capacity to meet their payment obligations. In contrast, if the social network is used to restrict coalition membership, customers join coalitions of *friends of friends*, thus being sure that someone they know directly is always involved.

Social networks have been used in many different fields of study, such as biology, communication studies, economics, information science, organizational studies, and sociology. Given this wide variety of topics and formalization, we will focus only on two categories of social networks, which have been deeply studied and can be modeled through mathematical procedures: *scale-free networks* and *small-world networks*[2].

5.2.1 Scale-free networks

Definition 16 (Scale-free network)**.** *A scale-free network is a network in which the degree distribution follows a power law, at least asymptotically. That is, the fraction $P(k)$ of nodes in the network having k connections to other nodes goes for large values of k as:*

$$P\left(k\right) \sim ck^{-\gamma}$$

where c is a normalization constant and γ is a parameter whose value is typically in the range $2 < \gamma < 3$, although it may lie outside these bounds.

Scale-free networks are believed to be the model of many real-world networks, including World Wide Web links, biological networks, and social networks, although the scientific community is still discussing these claims as more sophisticated data analysis techniques become available [25].

Many models have been proposed as mechanisms to explain conjectured power law degree distributions in real networks, such as *preferential attachment*, which refers to the process according to which the likelihood of connecting to a node depends on the node's degree. For example, a web page will more likely include hyperlinks to popular documents with already high degrees, because such highly connected documents are easy to find and thus well known.

[2] In the experimental cases, these topologies will the tested against *random networks*, which will be used as a benchmark

This mechanism is the base of the model proposed by Barabàsi and Albert [3], who studied the growth of social network and provided a simple algorithm to generate one that follows the properties of scale-free networks. Starting from a network of m_0 initial and connected[3] nodes, with $m_0 \geq 2$, new nodes are added to the network one at a time. Each new node is connected to m existing nodes with a probability that is proportional to the number of links that the existing nodes already have. Formally, the probability p_i that the new node is connected to node i is:

$$p_i = \frac{k_i}{\sum_j k_j}$$

Note that m is a parameter of the generator, which determines the final *density* – number of edges divided by the number of nodes – of the graph. In some formulation, *exactly* m edges are added at each step, in which case the density can be computed *a priori*; otherwise, a random number in the set $\{1, \ldots, m\}$ is chosen, making the number of edges in the final graph unpredictable. The resulting network can be shown to be scale-free, with a degree distribution of:

$$P(k) \sim k^{-3}$$

The example network shown in Figure 5.2 has 20 nodes and has been generated with the Barabàsi-Albert model, adding a maximum of 2 nodes per step.

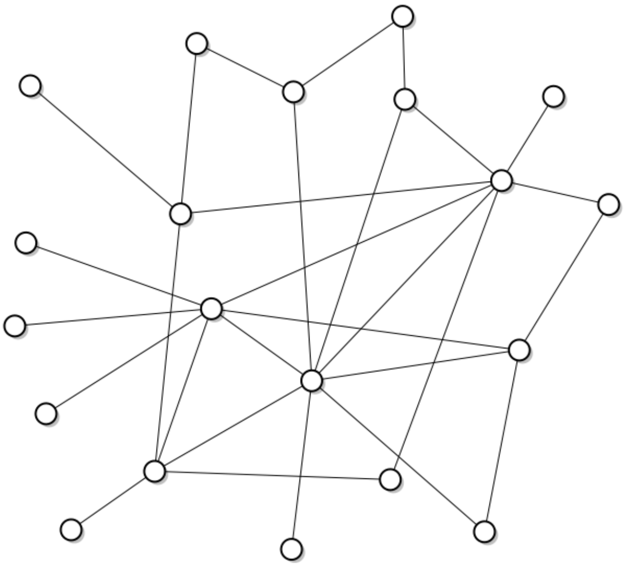


Figure 5.2: Scale-free network example

[3] If a node is not connected, it will always remain disconnected from the rest of the network. For example the initial network could be a tree of m_0 nodes.

5.2.2 Small-world networks

An alternative model to characterize social network is provided by *small-world networks*, whose topology is formed in a way such that most of the nodes are not neighbors of one another, but most of them can be reached from every other by a small number of hops or steps.

Definition 17 (Small-world network)**.** *A small-world network is a network where the typical distance L between two randomly chosen nodes (the number of steps required) grows proportionally to the logarithm of the number of nodes N in the network, that is:*

$$L \propto \log N$$

In the context of social interactions, the formation of small-world networks is the result of strangers being linked by a mutual acquaintance. Many empirical graphs are well-modeled by small-world networks. Social networks, wikis[4] such as Wikipedia, and gene networks all exhibit small-world characteristics.

This kind of networks were deeply studied by Watts and Strogatz [73], who identified two structural parameters used to classify the features of a particular network, namely the *clustering coefficient*, a measure of degree to which nodes tend to cluster together, and the *average shortest path length.* They also noticed that in purely random graphs build with the Erdős-Rényi model [29], both these parameters are low, while the clustering coefficient measured in many real-world networks significantly higher than expected, maintaining a small average shortest path length.

These are the main features enforced by the Watts-Strogatz model: Figure 5.3 shows an example of a small-world network of 20 nodes generated with this algorithm.

The typical ring-like structure is the consequence of the initial ring lattice of n nodes, connected to k other ones ($\frac{k}{2}$ per side, imposing that k must be even), which is subject to a *rewiring* process, which replaces some edges with a probability given by the parameter β.

The aforementioned properties make cliques, and near-cliques, very likely to form in this type of graphs. In addition to this, there are several other features that small-world networks tend to have: typically, there is an overabundance of *hubs* – nodes in the network with a high number of connections (known as high degree). These hubs serve as the common connections mediating the short path lengths between other edges. By analogy, the small-world network of airline flights has a small mean-path length (*i.e.* between any two cities you are likely to have to take three or fewer flights) because many flights are routed through hub cities.

[4] A *wiki* is a website whose users can add, modify, or delete its content via a web browser using a simplified markup language or a rich-text editor.

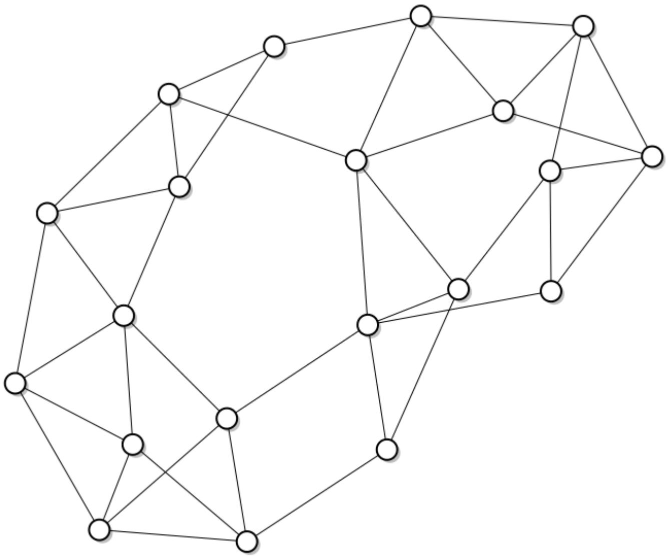


Figure 5.3: Small-world network example

5.3 Buying in the Energy Market

In economic terms, electricity is a *commodity*[5] capable of being bought, sold and traded. Thus, we refer as *energy market* as the system that allows to buy (through bids), sell and make short-term trades related to energy, generally in the form of financial or obligation swaps. Bids and offers use supply and demand principles to set the price. Long-term trades are contracts similar to power purchase agreements and generally considered private bi-lateral transactions between counterparties. Wholesale transactions (bids and offers) in electricity are typically cleared and settled by the market operator or a special-purpose independent entity charged exclusively with that function. Market operators do not clear trades but often require knowledge of the trade in order to maintain generation and load balance.

Electricity is by its nature difficult to store and has to be available on demand. Consequently, unlike other products, it is not possible, under normal operating conditions, to keep it in stock, ration it or have customers queue for it. Furthermore, demand and supply vary continuously.

There is therefore a physical requirement for a controlling agency, the *transmission system operator*, to coordinate the dispatch of generating units to meet the expected demand of the system across the transmission grid. If there is a mismatch between supply and demand the generators speed up or slow down causing the system frequency (either 50 or 60 hertz) to increase or decrease.

[5] From the economic point of view, a *commodity* is the generic term for any marketable item produced to satisfy wants or needs.

If the frequency falls outside a predetermined range the system operator will act to add or remove either generation or load. In addition, the laws of physics determine how electricity flows through an electricity network. Hence the extent of electricity lost in transmission and the level of congestion on any particular branch of the network will influence the economic dispatch of the generation units.

The scope of each electricity market comprises the transmission grid (or network) that is available to the wholesalers, retailers and the ultimate consumers in any geographic area.

5.3.1 Wholesale electricity market

A *wholesale electricity market* exists when competing generators offer their electricity output to retailers. The retailers then re-price the electricity and take it to market. While wholesale pricing used to be the exclusive domain of large retail suppliers, increasingly markets are beginning to open up to end-users. Large end-users seeking to cut out unnecessary overhead in their energy costs are beginning to recognize the advantages inherent in such a purchasing move, moreover, recently consumers have started buying electricity directly from generators too.

Buying wholesale electricity is not without its drawbacks (market uncertainty, membership costs, collateral investment), however, the larger the end user's electrical load, the greater the benefit and incentive to make the switch. Wholesale markets usually offer two different approaches for the buying strategies users can adopt: a short-term market, where buyers can obtain relatively small quantities of energy, usually intended to be spent in 24 hours, and a long-term market, in which larger amounts of energy can be bought for a longer interval of time. In common terminology, one usually refers respectively to *day-ahead market* and *forward market.*

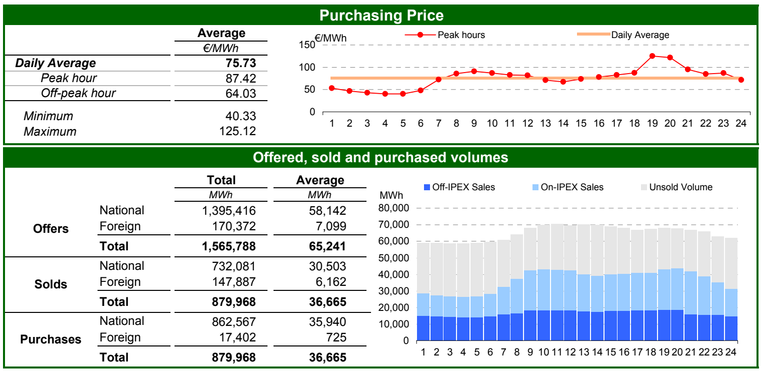
The system price in the day-ahead market is in principle determined by matching offers from generators to bids from consumers at each node to develop a classic supply and demand equilibrium price, usually on an hourly interval, and is calculated separately for subregions in which the system operator's load flow model indicates that constraints will bind transmission imports.

On the other hand, forward market tends to be more risky, due to an high volatility[6] consequence of the complexity of the system, possible peak demand and supply shortage; thus, as a common financial rule, prices are usually cheaper on this market. Users can access prices in real-time, simply accessing wholesale markets websites [2], which offer detailed reports and data regarding the current prices of short and long term market, together with information about purchased volumes of energy (Figure 5.4).

[6] Prices vary significantly over a relatively small period of time

Electricity customers and power producers tend to protect themselves from forward market instability buying energy in an aggregate form, thus forming the appropriate groups – or *coalitions* – satisfying everyone's demands is a fundamental task.

Long-term market offers a wide variety of contracts available to the investors, however the simplest and most commonly used form is represented by *fixed price forward contracts*, in which parts agree to trade a good – *e.g.* electric energy – at a specified future time with a price agreed today. Thus, the importance of having a flatter, more predictable usage profile becomes clear, since it allows groups of customers to buy larger stocks at a smaller price, minimizing the wasted amount of energy, which is also important from an ecological point of view. Thus, for a correct evaluation of a given solution, a precise, formal method that associates a coalition with its quality is needed. We will now introduce some of the metrics defined for this purpose, which have been adopted in the experimental tests.



Purchasing Price

	Average
	€/MWh
Daily Average	**75.73**
Peak hour	87.42
Off-peak hour	64.03
Minimum	40.33
Maximum	125.12

Offered, sold and purchased volumes

		Total	Average
		MWh	*MWh*
Offers	National	1,395,416	58,142
	Foreign	170,372	7,099
	Total	**1,565,788**	**65,241**
Solds	National	732,081	30,503
	Foreign	147,887	6,162
	Total	**879,968**	**36,665**
Purchases	National	862,567	35,940
	Foreign	17,402	725
	Total	**879,968**	**36,665**

Figure 5.4: Example of a report with price profile and volume exchange data

5.4 Metrics

Another fundamental problem addressed in this thesis is the choice of a appropriate metric to evaluate how the considered coalition is suitable to users' demands. From a formal point of view, the metric simply represents the *characteristic function* of the considered game (as defined in Definition 1). As pointed out before, "good" coalition are groups whose joint energy profile[7] is "flattened", because more energy can be bought in a cheaper way on the forward market, so the metrics described hereafter have the primary objective of quantifying this feature, attempting to assign a smaller value to a profile with a peaky behavior, while encouraging the formation of coalitions whose profiles are more regular.

[7] Analogously to single customer coalitions, we represent the joint energy profile of a coalition S as a vector $E_S = \{e_S^1, \ldots, e_S^T\}$ where $e_S^t = \sum_{i \in S} e_i^t$.

In addition, another eligible property of the metric is non-superadditivity: as described in Section 3.3.2, in superadditive games the optimal coalition structure is always given by the *grand coalition*. In this context, it would imply putting every agent in the same group, making everyone buying energy together, which is just not reasonable. Furthermore, in real-world environments the complex dynamics and the increased coordination costs of bigger groups make the setting intrinsically non-superadditive.

We will now present some metrics used in the actual implementation of the GDL algorithm presented in Chapter 4 and adopted for the empirical test described in Chapter 6, specifically the *load diversity factor* and the *user price factor*.

5.4.1 Load diversity factor

The first approach used in the definition of a metric is to focus on the "smoothness" of the aggregate profile obtained summing the energetic profiles.

The *load diversity factor* of a coalition S ($LDF(S)$) is computed as:

Definition 18 (Load diversity factor)**.**

$$LDF\left(S\right) = \frac{\sum_{i \in S} max_t E_i}{max_t E_S} + |S| - 1 \tag{5.1}$$

where E_i refers to the energy profile of the user i and E_S refers to the joint energy profile of coalition S.

It is easy to show that the quantity computed by the fractional term is always ≥ 1; intuitively, if the sum of the peaks of single users (given by the numerator) is close to the maximum of the joint profile, this value is lower, meaning that the maximum values of the individual profiles are positioned at the same time. Since this is the exact situation we want to avoid, the load diversity factor needs to be maximized, making it suitable for the aforementioned applications.

The additional $|S| - 1$ is due to the fact that, if we only consider the fractional term, the LDF of a single coalition is exactly 1, while, in general, for coalition of n users it is $< n$, making more profitable for players to go always alone.

Since we want to avoid this situation, encouraging the formation of groups, we reward each coalition with an additional value equal to the number of single coalitions avoided. For example, if agent 1 joins agent 2 in the coalition $\{1, 2\}$, one single coalition is avoided, *i.e.* an empty coalition must be considered, assumed to have a value of 1. In general, it is easy to see that if a coalition S of n users is formed ($|S| = n$), $n - 1$ single coalitions are avoided, increasing the value of S of $|S| - 1$.

From the test performed, load diversity factor allows to form coalition with a good quality for the system, but the value computed has no correlation with the final cost paid by the members of the group. Moreover, the interpretation of the empty coalitions is not clear and cannot be rigorously justified.

For these reasons, a different metric has been developed, the *user price factor*, which effectively captures the monetary gain users can obtain in the formation of coalition to buy from the energy market.

5.4.2 User price factor

The value of a coalition S, $v(S)$, is the total payment that the set of consumers need to carry out to cover the demand of their joint energy profile. As described before, we consider that customers buy directly their electricity in two different markets: the forward market and the day-ahead market. In the forward market, consumers in a coalition S buy in advance the fix continuous amount of energy of their joint energy profile, $base(S)$, for a better price. The amount of energy that exceeds this baseload, $peak(S)$, is bought in the day-ahead market.

Definition 19 (User price factor). *The user price factor of a coalition S $(UPF(S))$ is computed as:*

$$UPF\left(S\right) = -base(S) \cdot p_F - peak\left(S\right) \cdot p_{DA} - k\left(S\right) \tag{5.2}$$

where p_F and p_{DA} are the unit energy price in the forward and the day-ahead market respectively.

Since $p_F < p_{DA}$, the flattered the energy profile, the most a coalition of consumers can buy in the forward market and the lower the payment of the coalition. The term $k\left(S\right)$ is intended to represent the negative contributions due to the additional costs necessary to manage a bigger coalition, such as increased coordination costs and more complex dynamics inside the coalition. Thus, $k(S)$ must be proportional to the size of the coalition, and in this work it is assumed $k(S) = (|S| - 1) \cdot \frac{p_F}{T}$, where T is the length of the considered time interval.

From the tests shown in Chapter 6, user price factor has performed well, allowing the formation of mid-sized coalition, thus avoiding superadditivity. We remark this is simply the result of reasonable considerations, consequence of the real-life behavior of big groups of agents.

It is important to note that Definition 19 does not specify how the actual quantity $base(S)$ is computed, allowing the users to choose from different strategies for the transactions with the forward market. For example, a team could adopt a completely risk-free strategy, buying from the long term market only the quantity of energy it is absolutely sure to consume; in this case, $base(S) = min_t E_S \cdot T$.

Rather, a more risky approach can be used: a coalition might choose to buy more energy from the forward market than the continuous quantity it is sure to consume (represented by the minimum). Thus, a coalition S buys more energy in the forward market but some of this energy is not expected to be consumed. Figure 5.5 shows an example of this strategy: the dark amount represents the safer base load, corresponding to the minimum of the profile; then if the light amount is bought, following a more risky approach, more energy will be obtained at a cheaper price, but the marked quantities are not expected to be consumed by the coalition and hence, will be wasted. Although buying more energy than the expected consumption, this strategy can be perfectly rational for agents, in order to get better prices, if the amount of risk taken is directly proportional to the ration between the price of the energy in the forward market and the price in the day-ahead market. Thus, the base load to buy in the forward market to maximize the coalition gain is the maximum amount such that at least the fraction $\frac{p_F}{p_{DA}}$ is expected to be consumed.

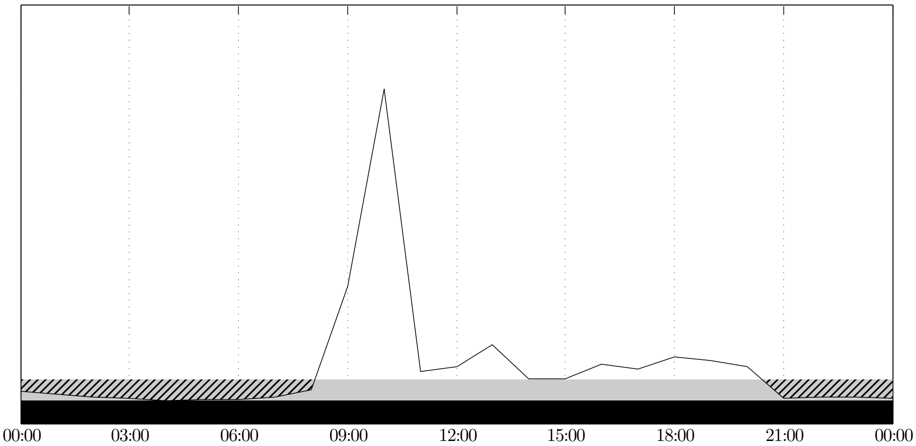


Figure 5.5: Example of different base loads for the same energy profile

Algorithm 9 shows how to easily compute this quantity assuming the profile E is discretized in m time parts, thus considering it as an array of m elements. Then, after ordering the m elements in descending order, the base load is the element in position $m \cdot \frac{p_F}{p_{DA}}$, multiplied for the length of the time interval. Coherently, if $p_F = p_{DA}$, then the algorithm returns as baseload the minimum quantity that is expected to be consumed.

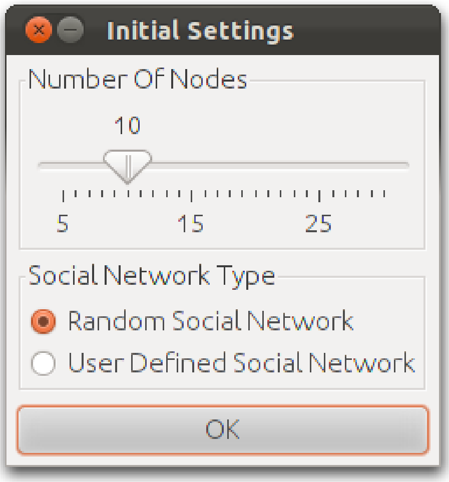
Algorithm 9 Computation of a risky baseload

```
Sort $E_S$ in descending order
$n \leftarrow \lfloor m \cdot \frac{p_F}{p_{DA}} \rfloor$
$base(S) \leftarrow E_S^n \cdot m$
```

5.5 Demonstration Application

To have an interactive visual tool for coalition formation in the energy market, we developed a platform that allows energy consumers to organize into stable energy profile coalitions. The demonstration application is written in Java and permits the user to interact with the coalition formation process and have detailed informations about the optimal outcome and the possible gains that can be obtained.

The demonstration starts by asking the user the number of energy consumers for the simulation (Figure 5.6). Moreover, the user can choose between creating the social network randomly, or, alternatively, create a user defined social network from scratch.


Initial Settings

Number Of Nodes

10

Social Network Type

● Random Social Network

○ User Defined Social Network

Figure 5.6: Initial dialog of the simulator application

In both cases, the platform generates a set of nodes, one per energy consumer, and allows the user to modify the network by adding/removing links in an easy way. Each node has an energy profile loaded from real data characterizing the domestic electricity market and usage patterns of households in the United Kingdom.

Once the coalition formation scenario is set, the simulation starts the *SCF-Graphs* algorithm that organizes energy consumers into stable optimal coalitions. Upon convergence, energy consumers in the same coalition are colored with the same color. For example, observe that in Figure 5.7, the coalition $\{0, 1, 3, 4, 5, 6\}$ was formed, (the grouped agents are all colored in red), whereas consumer 2 is on its own. The interface also highlights the pseudotree used by the algorithm (showing bolded edges) with the associated root node. On the right lower corner, the application also shows the average user gain – that is the gain that represents the consumer assigned payoff with respect to the value of its individual energy profile.

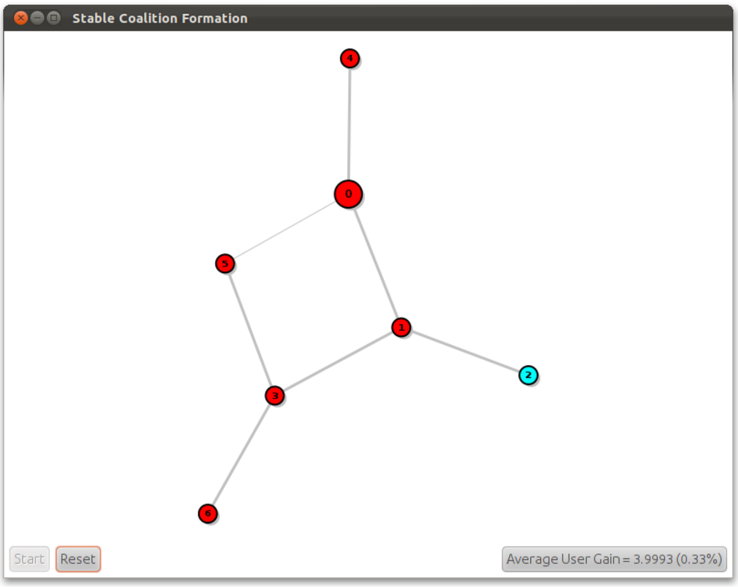

Stable Coalition Formation
4
0
5
1
2
3
6
Start
Reset
Average User Gain = 3.9993 (0.33%)

Figure 5.7: The simulator main interface

By clicking on a node, the GUI displays statistical data related to the specific energy consumer such as its coalition, the coalition's value and the individual payment that the consumer contributes to the coalition. Consumers payments are set in such a way that consumers do not have any incentive to deviate. Finally, the interface also allows to visualize the energetic profiles of coalition members (see Figure 5.8). Each chart plots a consumer energy profile, delimited by a red line, and the joint coalition energy profile, delimited by a blue line. The difference between the joint and the individual profile is filled in the same color used to mark the considered partition (red in the example). The GUI offers the user an effective way of restarting simulations after reconfiguring the network topology, testing how the existence or the nonexistence of a particular link affects the emerging coalitions and consumers gain.

As a simulator, this platform provides to the users with a proof of concept of what we can do already today as energy consumers in order to get cheaper and greener energy. Furthermore, it presents the decentralized coalition formation problem among energy users to the community as an exciting real-world domain for the applicability of multi-agent technology. This application has been the subject of a demo paper submitted at *11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)* [14] and is available for download as an executable JAR file at `http://profs.sci.univr.it/~farinelli/energySCF.jar`. Moreover an illustrative video, describing the simulator and the related topics can be viewed at `http://www.youtube.com/watch?v=FT25oETMkfw`.

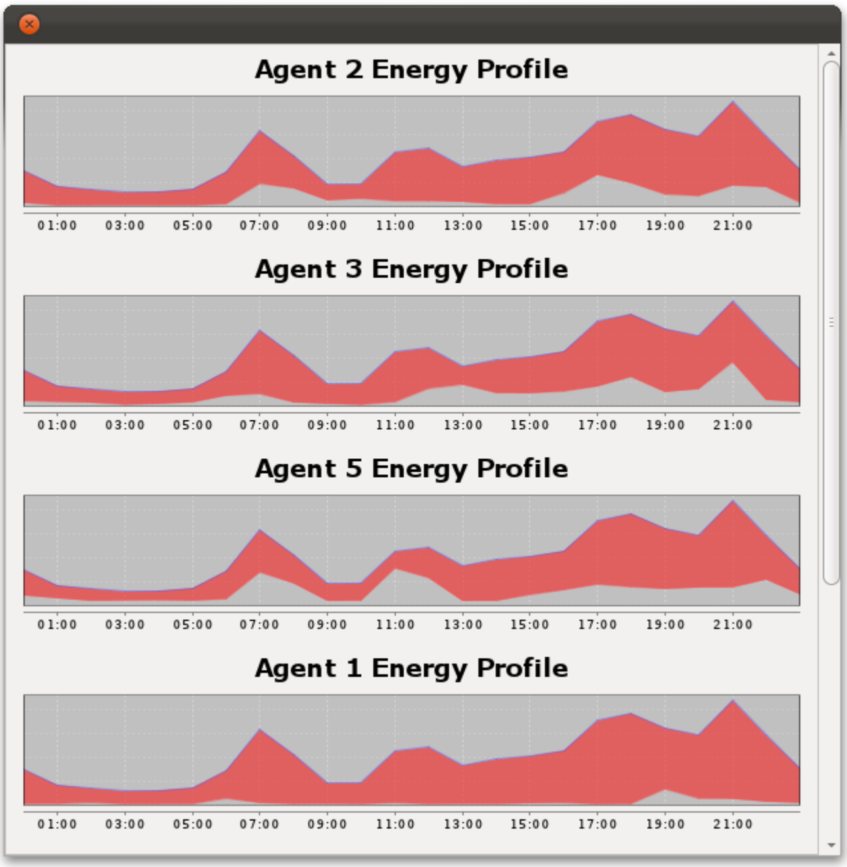


Figure 5.8: The coalition energy profile inspector

Chapter 6

Results

In this chapter we will present the results of the empirical experiments conducted to test the quality of the presented coalition formation technique. In particular, two fundamental features of the solutions are addressed: the *average user gain* and the *coalition structure*. Intuitively, *average user gain* refers to the effective gain users have obtained adopting the coalition organization proposed by the given outcome, with respect to a naive solution where only single coalitions are formed.

Definition 20 (Average user gain)**.** *The average user gain of set of agents A with the payoff vector x is computed as:*

$$AUG(A, x) = \frac{\sum_{i \in A}(x_i - v(\{i\})}{|A|} \tag{6.1}$$

where x_i refers to the individual payment of agent i, and $v(\{i\})$ refers to the value of the coalition with agent i alone. In addition, the average percent user gain refers to the percentual gain obtained by users w.r.t. the average value of single coalitions ($APUG(A, x) = \frac{AUG(A,x) \cdot |A|}{\sum_{i \in A} v(\{i\})} \cdot 100$).

On the other hand, the *coalition structure* analysis focuses on the size of the computed coalitions, in terms of minimum, maximum and average dimension of the groups.

6.1 Problem generation

To investigate the sensitivity of the coalition formation process with respect to the underlying network topology, we evaluate our model on three different network models. Moreover, for each network topology we analyze networks with different density levels. Formally, the density of a graph is defined as the ratio between the number of links and the number of agents in the graph ($\frac{|E|}{|A|}$).

In more detail, in our experiments we test our model on the following network configurations:

Random Networks: Graphs are created by *randomly* adding a number of links d for each agent. Densities used in this case are : $d = 1$ (low), $d = 2$ (medium) and $d = 3$ (high).

Scale Free Networks: Graphs are created by using an implementation of the Barabàsi-Albert model. At each step, a node is added and attached to d neighbors using a biased random selection giving more chance to a node if it has a high degree. Graphs are generated using three different densities: ($d = 0.92$, low), ($d = 1.75$, medium) and ($d = 3.17$, high).

Small-World Networks: Graphs are created by following the Watts and Strogatz model. This model generates a ring of graph where each node is connected to its k nearest neighbors in the ring ($\frac{k}{2}$ on each side, which means k must be even). Then it process each node of the ring in order following the ring, and "rewiring" each of their edges toward the not yet processed nodes with randomly chosen nodes with a rewiring probability of 0.1. Graphs are generated using three different values for parameter k: $k = 2$ ($d = 1$, low), $k = 4$ ($d = 2$, medium) and $k = 6$ ($d = 3$, high).

Notice that whereas scale free and small-world networks are known to capture some characteristics of social networks, random networks constitute a more synthetic model for our domain. All experiments are run using networks of 12 nodes. For each instance, the energy profile of each node is randomly selected from a real dataset composed of energy profiles characterizing the real domestic electricity consumption of 5000 households in the United Kingdom. Each energetic consumer has been monitored for a time period of a month (December 2009), measuring a the value of power consumption every 30 minutes, for a total of 48 daily time slots. The initial data contained some corrupted entries, due to a problem with the sensor, so, before running any experiment, the data has been filtered keeping only meaningful measurements.

6.2 CPLEX verification

To test the correctness of the algorithm presented in Chapter 4 and its JAVA implementation, all the results have been tested with an alternative technique, to check that the solution obtained are coherent. The same coalition formation problem has been formalized using linear constraints and solved with Dantzig simplex algorithm [49], implemented with *IBM ILOG CPLEX Optimization Studio*. We remark that CPLEX implementation currently represents the *state of the art* in linear programming optimization, thus its running times are not comparable with the test case.

6.3 Market's parameters

For the hereafter described experiments, *user price factor* has been used for compute the coalitional value. As described in Section 5.4.2, user price factor has two important parameters: p_F, the price of the electricity in the *forward* market, and p_{DA}, referred to *day-ahead* market (although the price of electricity in the day-ahead market varies on each time slot, we consider here that p_{DA} is calculated by averaging the hourly price of a day). Moreover, in addition to the market prices, another sensitive parameter is the choice of the strategy to decide which is the base amount to buy in the forward market (defined through fraction parameter p). Then, in our experiments, we explore three different market conditions:

- M_1: $p_F = 70$, $p_{DA} = 80$, $p = 1$
- M_2: $p_F = 70$, $p_{DA} = 80$, $p = 0.875$
- M_3: $p_F = 1$, $p_{DA} = 2$, $p = 0.5$

Notice that whereas in M_1 agents follow a risk-free strategy ($p = 1$), in M_2 and M_3 agents takes the maximum much risk that gives them a positive expected gain ($p = \frac{p_F}{p_{DA}}$). Regarding market prices, in M_1 and M_2 prices used are those of current electricity markets in Italy [2] whereas M_3 explores a different scenario in which buying in the forward market is more incentivized with better prices. As a consequence, the maximum relative percent gain[1] an agent can get in M_1 and M_2 is of 12.5% whereas in M_3 is of 50%.

6.4 Results

Using the different configurations explained in section above, we evaluate our model by performing repeated simulations (50 instances per graph configuration) and analyzing the following features.

Consumer's gain

Figure 6.1 show the results for 12 agents on a random, scale-free and small-world networks in the three different market scenarios respectively. We also plotted the standard error of the mean as a measure of the variance in each graph. Only data regarding instances with non-empty core are plotted. Observe that in all configurations, although the average percent user gain is increased with density, this increment is not significant. Moreover, the average percent user gain is much higher (around 10%) in M_3 market than not in M_1 and M_2 (around 1%).

[1] The *maximum relative percent gain* is the difference between the cost of buying all demand in the day-ahead market and the cost of buying all demand in the forward market divided by the cost of buying all demand in the day-ahead market $100 \cdot \frac{p_{DA}-p_F}{p_{DA}}$.

Core emptiness

Table 6.1 shows the percentage of instances under each configuration for which the core was detected as empty. Notice that in all network topologies, the number of instances for which the core is empty increases with the density of the network. These results are coherent with the well-known results that any acyclic network (which has by definition the lowest density) is guarantee to have a non-empty core [27]. As we increase the density the number of cycles also increase and results show that the probability of core emptiness is higher. Regarding different network topologies, we observe that the number of instances with core emptiness is higher in scale free networks, where the links are concentrated on hubs, than not on random and small-world networks, where each node in average have the same degree. Finally, we also observe that the number of instances with core empty is much higher on M_1 and M_2 than not in M_3. Although we need to perform a deeper analysis on these results, they lead to the hypothesis that the more the distance of prices in the market the less the probability of having an empty core in the coalitional game.

Topology	**Density**	% **Empty Core**		
		M_1	M_2	M_3
Random	Low	8%	0%	0%
	Medium	50%	26%	6%
	High	56%	44%	10%
Scale-Free	Low	0%	0%	0%
	Medium	52%	22%	2%
	High	46%	38%	12%
Small-World	Low	8%	6%	2%
	Medium	46%	18%	8%
	High	46%	48%	6%

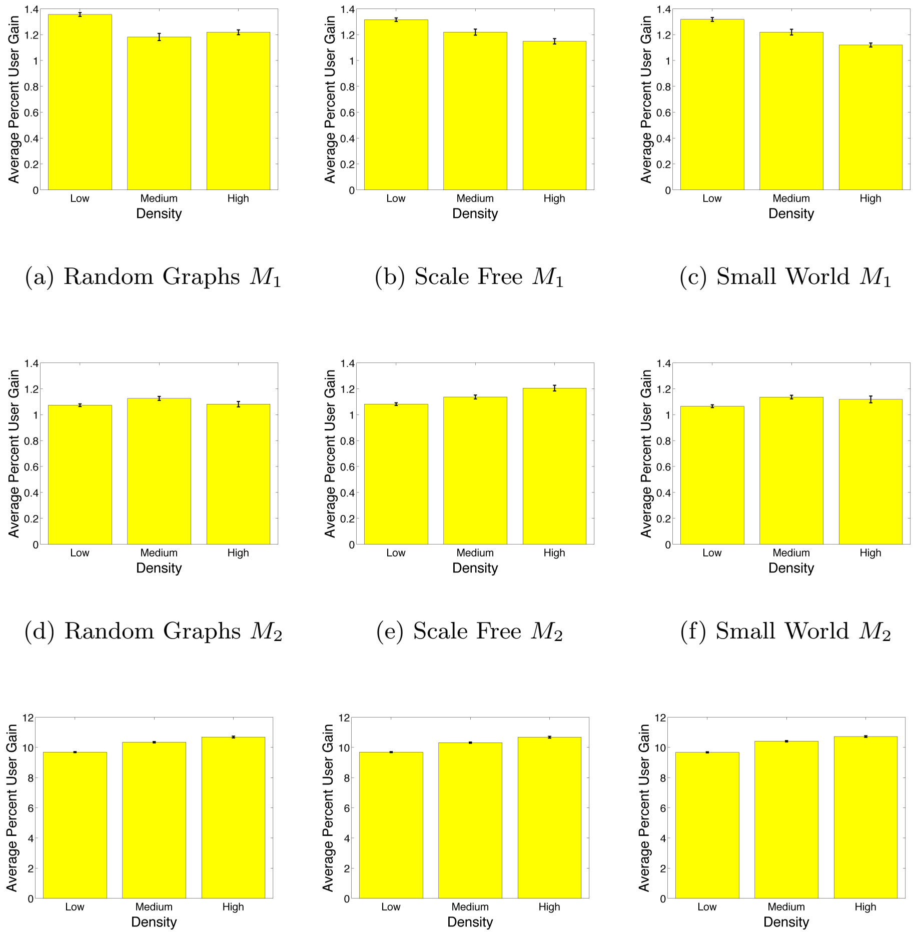
Table 6.1: Percentage of instances with empty core under different configurations.

Structure of energy coalitions

In this section we analyze the structure of the energy coalitions obtained in the experiments. For each configuration, we plot the mean of the minimum, average and maximum size of coalitions formed. Figure 6.2 plots the results for networks of 12 agents on a random, scale free and small-world networks in two different market scenarios. We also plotted the standard error of the mean as a measure of the variance in each graph. Market scenario M_3 is omitted because in this case we detected that the gran coalition was always formed in all tested instances.

Since the difference between p_{DA} and p_F in M_3 is relatively high, buying more energy in the forward market is very profitable, leading to the formation of bigger coalitions. In contrast we observe that for markets M_1 and M_2, the market conditions lead to coalitions of middle size in all network structures. We also observe that as we increase the density of the network, more coalitions of middle size are composed whereas in low densities agents tend to compose larger coalitions.

6.5 Plots

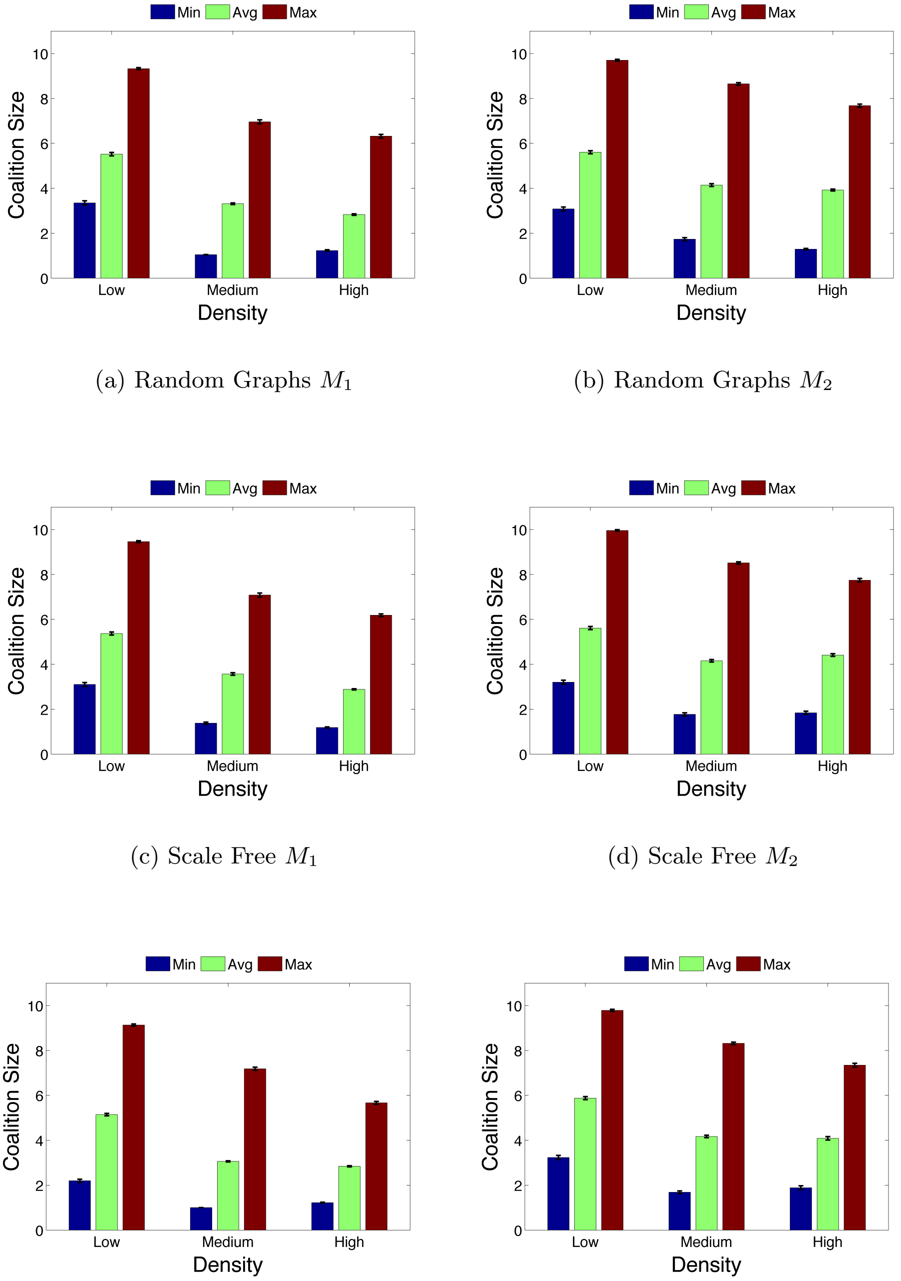


(a) Random Graphs M_1 (b) Scale Free M_1 (c) Small World M_1

(d) Random Graphs M_2 (e) Scale Free M_2 (f) Small World M_2

(g) Random Graphs M_3 (h) Scale Free M_3 (i) Small World M_3

Figure 6.1: Graphs showing the average percent gain of consumers on different topologies and densities under market conditions M_1 (a)-(c), M_2 (d)-(f) and M_3 (g)-(i).



(a) Random Graphs M_1

(b) Random Graphs M_2

(c) Scale Free M_1

(d) Scale Free M_2

(e) Small World M_1

(f) Small World M_2

Figure 6.2: Graphs showing the minimum, average and maximum size of coalitions formed on different topologies and densities under market conditions M_1 (a)-(c), M_2 (d)-(f).

Chapter 7

Conclusions and Future Work

In this thesis we have shown how the stable coalition formation can be effectively applied to a real-world application such the purchase of energy on the wholesale electrical marked.

Users with complementary consumption profiles can join together in a common group, forming a *coalition* whose aggregate energy demand is more regular and flattened. Thus, more of this energy can be retrieved from the long term market, which provides large amounts of energy at a cheaper prices, granting the single agents a real monetary advantage. Also, the request for expensive, carbon-intensive peaking plants generators is reduced, providing a great benefit for the environment.

To help the solving process, the space of research has been reduced by means of a *social network*, which models the social interactions among the users. Social networks not only provide a way of connecting energy consumers but also restrict coalition membership by specifying realistic constraints to the formation of certain groups. In particular, consumers may not want to join coalitions with unknown consumers for which they do not have any source of trust regarding their reported profiles or their capacity to meet their payment obligations.

The work here presented is very promising and has given good results, as pointed out by the empirical tests. The provided solutions can effectively lower the cost single users have to pay for their electricity bill, with a promising effect from the ecological point of view, due to the limited pollutant emissions.

Moreover, the considered approach is suitable to be applied to real-world scenarios, since it assumes to deal with selfish agents, a typical feature of real energy customers. We remark that the pursuit of *both* optimality and stability has never been covered in literature, while this work addresses the two problematics together.

Nevertheless, there are many possibilities for future development in this direction. The theoretical representation defined in Chapter 4 provides a smart intuition in the context of cooperative games on graphs, but can be further improved to allow a better scalability with respect to the dimension of instances used as inputs for the problem. At the current state, our technique can solve general graphs of 12-13 nodes, with a density of 2.

In addition, the computational complexity of the approach can be lowered (at the expense of the accuracy) adopting approximated or non-complete solving techniques, which may provide a good – but not optimal – solution in a reasonable amount of time, even for bigger instances than the ones tractable at the moment.

Focusing on the implementation, a very promising direction of improvement may be represented by the adoption of thread-intensive techniques provided by the GPU computing. Due to the high parallelization characteristics of the routine used in the coalition formation process, the use of a massive multi-threading implementation (such as *CUDA* or *OpenCL*) would grant a great speed-up, though the theoretical computational complexity would not be affected.

Finally, the metrics proposed in Section 5.4 (especially the *user price factor*) have proven to be effective in the tests, but alternative forms of evaluation of the coalitions are not to be excluded, for example considering features other than the simple price paid by the customers or managing the risks with different approaches.

In conclusion, the work of this thesis has given a great contribution in the field of stable coalition formation theory, especially considering the approach of new and interesting problematics in the energetic domain and providing a novel and brilliant solving technique, which may be the starting point of many future developments.

Bibliography

[1] Grid 2030. A national vision for electricity's second 100 years. Washington Department of Ecology (2003), http://www.oe.energy.gov/DocumentsandMedia/Electric_Vision_Document.pdf

[2] GME – gestore mercati energetici / wholesale market. Italia (2012), http://www.mercatoelettrico.org/En/Default.aspx

[3] Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. Reviews of Modern Physics 74, 47–97 (Jan 2002)

[4] Arnold, T., Schwalbe, U.: Dynamic coalition formation and the core. Journal of Economic Behavior & Organization 49(3), 363–380 (2002), http://linkinghub.elsevier.com/retrieve/pii/S016726810200015X

[5] Augustine, J., Chen, N., Elkind, E., Fanelli, A., Gravin, N., Shiryaev, D.: Dynamics of profit-sharing games. In: IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011. pp. 37–42 (2011)

[6] Aumann, R.J., Drèze, J.H.: Cooperative games with Coalition Structures. International Journal of Game Theory 3, 217–237 (1975)

[7] Aziz, H., de Keijzer, B.: Complexity of coalition structure generation. In: Proceedings of 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan, May 2-6, 2011, Volume 1-3. pp. 191–198 (2011)

[8] B., R., Myerson: Virtual utility and the core for games with incomplete information. Journal of Economic Theory 136(1), 260 – 285 (2007), http://www.sciencedirect.com/science/article/pii/S0022053106001438

[9] Bachrach, Y., Meir, R., Jung, K., Kohli, P.: Coalitional structure generation in skill games. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010 (2010)

[10] Banerjee, B., Sen, S.: Selecting partners. In: Proceedings of the fourth international conference on Autonomous agents. pp. 261–262. AGENTS '00, ACM, New York, NY, USA (2000), http://doi.acm.org/10.1145/336595.337478

[11] Bell, E.: Exponential numbers. The American Mathematical Monthly 41(7), 411–419 (August 1934), http://links.jstor.org/sici?sici=0002-9890%28193408%2F09%2941%3A7%3C411%3AEN%3E2.0.CO%3B2-2

[12] Benhamou, F. (ed.): Principles and Practice of Constraint Programming - CP 2006, 12th International Conference, CP 2006, Nantes, France, September 25-29, 2006, Proceedings, Lecture Notes in Computer Science, vol. 4204. Springer (2006)

[13] Benny, M., Eyal, W.: Order independent equilibria. Games and Economic Behavior 9(1), 21–34 (April 1995), http://ideas.repec.org/a/eee/gamebe/v9y1995i1p21-34.html

[14] Bistaffa, F., Farinelli, A., Vinyals, M., Rogers, A.: Decentralized stable coalition formation among energy customers in the smart grid. Demo Paper, submitted at 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)

[15] Blankenburg, B., Dash, R.K., Ramchurn, S.D., Klusch, M., Jennings, N.R.: Trusted kernel-based coalition formation. In: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems. pp. 989–996. AAMAS '05, ACM, New York, NY, USA (2005), http://doi.acm.org/10.1145/1082473.1082623

[16] Brafman, R.I., Domshlak, C., Engel, Y., Tennenholtz, M.: Transferable utility planning games. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010 (2010)

[17] van den Brink, R.: On hierarchies and communication. Tinbergen Institute Discussion Papers 06-056/1, Tinbergen Institute (Jul 2006), http://ideas.repec.org/p/dgr/uvatin/20060056.html

[18] Chalkiadakis, G.: A bayesian approach to multiagent reinforcement learning and coalition formation under uncertainty. Ph.D. thesis, Toronto, Ont., Canada, Canada (2007), aAINR39416

[19] Chalkiadakis, G., Boutilier, C.: Bayesian reinforcement learning for coalition formation under uncertainty. In: AAMAS. pp. 1090–1097. IEEE Computer Society (2004)

[20] Chalkiadakis, G., Boutilier, C.: Coalitional bargaining with agent type uncertainty. In: Veloso, M.M. (ed.) IJCAI. pp. 1227–1232 (2007)

[21] Chalkiadakis, G., Elkind, E., Jennings, N.R.: Simple coalitional games with beliefs. In: Boutilier, C. (ed.) IJCAI. pp. 85–90 (2009)

[22] Chalkiadakis, G., Elkind, E., Wooldridge, M.: Computational Aspects of Cooperative Game Theory. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers (2011)

[23] Chalkiadakis, G., Markakis, E., Boutilier, C.: Coalition formation under uncertainty: bargaining equilibria and the bayesian core stability concept. In: Durfee, E.H., Yokoo, M., Huhns, M.N., Shehory, O. (eds.) AAMAS. p. 64. IFAAMAS (2007)

[24] Chatterjee, K., Dutta, B., Ray, D., Sengupta, K.: A noncooperative theory of coalitional bargaining. Review of Economic Studies 60(2), 463–77 (April 1993), `http://ideas.repec.org/a/bla/restud/v60y1993i2p463-77.html`

[25] Clauset, A., Shalizi, C.R., Newman, M.E.J.: Power-law distributions in empirical data. SIAM Reviews (Jun 2007), `http://arxiv.org/abs/0706.1062`

[26] Dang, V.D., Jennings, N.R.: Generating coalition structures with finite bound from the optimal guarantees (2004)

[27] Demange, G.: On group stability in hierarchies and networks. Journal of Political Economy 112(4), 754–778 (August 2004), `http://ideas.repec.org/a/ucp/jpolec/v112y2004i4p754-778.html`

[28] Deng, X., Papadimitriou, C.: On the complexity of cooperative solution concepts. Mathematics of Operations Research 19(2), 257–266 (1994)

[29] Erdős, P., Rényi, A.: On random graphs 1. Publ. Math. (Debrecen) 6, 290 (1959)

[30] Evans, R.: Coalitional bargaining with competition to make offers. Games and Economic Behavior 19(2), 211–220 (May 1997), `http://ideas.repec.org/a/eee/gamebe/v19y1997i2p211-220.html`

[31] Gillies, D.B.: Solutions to general non-zero-sum games. In: Tucker, A.W., Luce, R.D. (eds.) Contributions to the Theory of Games IV, pp. 47–85. No. 40 in Annals of Mathematics Studies, Princeton University Press, Princeton (1959)

[32] Greco, G., Malizia, E., Palopoli, L., Scarcello, F.: On the complexity of the core over coalition structures. In: IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011. pp. 216–221 (2011)

[33] Hart, S., Mas-Colell, A.: Bargaining and value. Econometrica 64(2), 357–80 (March 1996), http://ideas.repec.org/a/ecm/emetrp/v64y1996i2p357-80.html

[34] Ieong, S., Shoham, Y.: Marginal contribution nets: A compact representation scheme for coalitional games. In: Proc. of the ACM Conference on Electronic Commerce, 2005. p. 193 (2005)

[35] Ieong, S., Shoham, Y.: Bayesian coalitional games. In: Fox, D., Gomes, C.P. (eds.) AAAI. pp. 95–100. AAAI Press (2008)

[36] Jackson, M.O.: A survey of models of network formation: Stability and efficiency. Game Theory and Information 0303011, EconWPA (Mar 2003), http://ideas.repec.org/p/wpa/wuwpga/0303011.html

[37] Kalai, E.: Game theory: Analysis of conflict : By roger b. myerson, harvard univ. press, cambridge, ma, 1991. 568 pp. Games and Economic Behavior 3(3), 387–391 (August 1991), http://ideas.repec.org/a/eee/gamebe/v3y1991i3p387-391.html

[38] Ketchpel, S.: Forming coalitions in the face of uncertain rewards. In: In Proc. of AAAI94. pp. 414–419 (1994)

[39] Kjaerulff, U.: Triangulation of graphs – algorithms giving small total state space. Technical Report R9009 Department of Mathematics and Computer Science Aalborg University Denmark (R-90-09) (1990), http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.50.9834

[40] Konishi, H., Ray, D.: Coalition formation as a dynamic process. Journal of Economic Theory 110(1), 1 – 41 (2003), http://www.sciencedirect.com/science/article/pii/S0022053103000048

[41] Kraus, S., Shehory, O., Taase, G.: Coalition formation with uncertain heterogeneous information. In: Proceedings of the second international joint conference on Autonomous agents and multiagent systems. pp. 1–8. AAMAS '03, ACM, New York, NY, USA (2003), http://doi.acm.org/10.1145/860575.860577

[42] Kraus, S., Shehory, O., Taase, G.: The advantages of compromising in coalition formation with incomplete information. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2. pp. 588–595. AAMAS '04, IEEE Computer Society, Washington, DC, USA (2004), http://dx.doi.org/10.1109/AAMAS.2004.259

[43] Kraus, S., Wilkenfeld, J., Zlotkin, G.: Multiagent negotiation under time constraints. Artificial Intelligence 75(2), 297 – 345 (1995), http://www.sciencedirect.com/science/article/pii/000437029400021R

[44] Liu, J.S.: Coordination of Multiple Agents in Distributed Manufacturing Scheduling. Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (April 1996)

[45] Mas-Colell, A., Whinston, M., Green, J.: Microeconomic theory. Oxford University Press (1995), http://books.google.it/books?id=KGtegVXqD8wC

[46] Megiddo, N.: Computational complexity of the game theory approach to cost allocation on a tree. Mathematics of Operations Research 3(3), 189–196 (1978)

[47] Myerson, R.: Cooperative games with incomplete information. In: Glycopantis, D., Yannelis, N.C., Yannelis, N.C., Kehoe, T.J., Cornet, B. (eds.) Differential Information Economies, Studies in Economic Theory, vol. 19, pp. 481–505. Springer Berlin Heidelberg (2005), 10.1007/3-540-26979-7_25

[48] Myerson, R.B.: Graphs and Cooperation in Games. Mathematics of Operations Research 2(3), 225–229 (1977), http://dx.doi.org/10.2307/3689511

[49] Nash, J.C.: The (dantzig) simplex method for linear programming. In: Computing in Science and Engg., vol. 2, pp. 29–31. IEEE Educational Activities Department, Piscataway, NJ, USA (2000)

[50] Ohta, N., Conitzer, V., Ichimura, R., Sakurai, Y., Iwasaki, A., Yokoo, M.: Coalition structure generation utilizing compact characteristic function representations. In: Gent, I.P. (ed.) CP. Lecture Notes in Computer Science, vol. 5732, pp. 623–638. Springer (2009)

[51] Okada, A.: A noncooperative coalitional bargaining game with random proposers. Games and Economic Behavior 16(1), 97–108 (September 1996), http://ideas.repec.org/a/eee/gamebe/v16y1996i1p97-108.html

[52] Osborne, M.J., Rubinstein, A.: A course in game theory. Levine's Bibliography 814577000000000225, UCLA Department of Economics (May 2009), http://ideas.repec.org/p/cla/levrem/814577000000000225.html

[53] Perry, M., Reny, P.J.: A noncooperative view of coalition formation and the core. Econometrica 62(4), 795–817 (July 1994), http://ideas.repec.org/a/ecm/emetrp/v62y1994i4p795-817.html

[54] Rahwan, T., Jennings, N.R.: An improved dynamic programming algorithm for coalition structure generation. In: In Proc. of the 7th Int. joint Conf. on Autonomous Agents and Multi-agent Systems (AAMAS. pp. 1417–1420 (2008)

[55] Rahwan, T., Jennings, N.: Coalition structure generation: Dynamic programming meets anytime optimisation. Proceedings of the TwentyThird AAAI Conference on Artificial Intelligence 2008 p. 156–161 (2008), http://eprints.ecs.soton.ac.uk/16112/

[56] Rahwan, T., Ramchurn, S.D., Jennings, N.R., Giovannucci, A.: An anytime algorithm for optimal coalition structure generation. J. Artif. Intell. Res. (JAIR) 34, 521–567 (2009)

[57] Ramchurn, S.D., Vytelingum, P., Rogers, A., Jennings, N.R.: Agent-based control for decentralised demand side management in the smart grid. In: Proceedings of 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan, May 2-6, 2011, Volume 1-3. pp. 5–12 (2011)

[58] Rogers, A., Farinelli, A., Stranders, R., Jennings, N.R.: Bounded approximate decentralised coordination via the max-sum algorithm. Artificial Intelligence 175(2), 730–759 (February 2011), http://eprints.ecs.soton.ac.uk/21664/

[59] Rothkopf, M.H., Pekec, A., Harstad, R.M.: Computationally manageable combinatorial auctions (1998)

[60] Sandholm, T., Larson, K., Andersson, M., Shehory, O., Tohmé, F.: Coalition structure generation with worst case guarantees (1999)

[61] Serrano, R., Vohra, R.: Non-cooperative implementation of the core. Social Choice and Welfare 14(4), 513–525 (1997), http://ideas.repec.org/a/spr/sochwe/v14y1997i4p513-525.html

[62] Service, T.C., Adams, J.A.: Constant factor approximation algorithms for coalition structure generation. Autonomous Agents and Multi-Agent Systems 23(1), 1–17 (July 2011)

[63] Shehory, O., Kraus, S.: Methods for task allocation via agent coalition formation. Artif. Intell. 101(1-2), 165–200 (1998)

[64] Shehory, O., Kraus, S.: Feasible formation of coalitions among autonomous agents in nonsuperadditive environments. Computational Intelligence 15(3), 218–251 (1999), http://dx.doi.org/10.1111/0824-7935.00092

[65] Shehory, O., Sycara, K., Jha, S.: Multi-agent coordination through coalition formation. In: Singh, M., Rao, A., Wooldridge, M. (eds.) Intelligent Agents IV Agent Theories, Architectures, and Languages, Lecture Notes in Computer Science, vol. 1365, pp. 143–154. Springer Berlin / Heidelberg (1998), http://dx.doi.org/10.1007/BFb0026756, 10.1007/BFb0026756

[66] Suijs, J., Borm, P.: Stochastic cooperative games: Superadditivity, convexity, and certainty equivalents. Games and Economic Behavior 27(2), 331 – 345 (1999), http://www.sciencedirect.com/science/article/pii/S0899825698906728

[67] Suijs, J., Borm, P., Waegenaere, A.D., Tijs, S.: Cooperative games with stochastic payoffs. European Journal of Operational Research 113(1), 193 – 205 (1999), http://www.sciencedirect.com/science/article/pii/S0377221797004219

[68] Sycara, K., Pannu, A., Williamson, M., Zeng, D., Decker, K.: Distributed intelligent agents. IEEE Expert: Intelligent Systems and Their Applications 11, 36–46 (December 1996), http://dx.doi.org/10.1109/64.546581

[69] Tarjan, R.E., Yannakakis, M.: Addendum: Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. SIAM J. Comput. 14(1), 254–255 (1985)

[70] Tompros, S.L., Mouratidis, N.P., Draaijer, M., Foglar, A., Hrasnica, H.: Enabling applicability of energy saving applications on the appliances of the home environment. IEEE Network 23(6), 8–16 (2009)

[71] Vinyals, M., Farinelli, A., Rogers, A., Jennings, N.: A message-passing approach to computing stable coalitions on graphs, under submission

[72] Vinyals, M., Rodriguez-Aguilar, J., Cerquides, J.: Constructing a unifying theory of dynamic programming dcop algorithms via the generalized distributive law. Autonomous Agents and Multi-Agent Systems 22, 439–464 (2011), http://dx.doi.org/10.1007/s10458-010-9132-7, 10.1007/s10458-010-9132-7

[73] Watts, D.J., Strogatz, S.H.: Collective dynamics of “small-world” networks. Nature 393(6684), 440–442 (Jun 1998), http://dx.doi.org/10.1038/30918

[74] Yan, H.: Noncooperative selection of the core. International Journal of Game Theory 31, 527–540 (2003), http://dx.doi.org/10.1007/s001820300137, 10.1007/s001820300137