Instituto Superior de Engenharia do Porto

Six Thinking Hats for Electricity Markets Simulation

Adapting Meeting Tools to Agent Decision

João Costa Borges Barreto

Dissertation to obtain the Master of Science degree in

Computer Science

Specialization in

Knowledge-based and Decision Support Technologies

Supervisor: Isabel Cecília Correia da Silva Praça Gomes Pereira, PhD

Co-Supervisor: Tiago Manuel Campelos Ferreira Pinto, MSc

Porto, Outubro 2012

“Para a Elisabete”

Abstract

In almost all industrialized countries, the energy sector has suffered a severe restructuring that originated a greater complexity in market players' interactions. The complexity that these changes brought made way for the creation of decision support tools that facilitate the study and understanding of these markets.

MASCEM – "Multiagent Simulator for Competitive Electricity Markets" arose in this context providing a framework for evaluating new rules, new behaviour, and new participants in deregulated electricity markets. MASCEM uses game theory, machine learning techniques, scenario analysis and optimisation techniques to model market agents and to provide them with decision-support.

ALBidS is a multiagent system created to provide decision support to market negotiating players. Fully integrated with MASCEM it considers several different methodologies based on very distinct approaches.

The Six Thinking Hats is a powerful technique used to look at decisions from different perspectives. This tool's goal is to force the thinker to move outside his habitual thinking style. It was developed to be used mainly at meetings in order to "run better meetings, make faster decisions".

This dissertation presents a study about the applicability of the Six Thinking Hats technique in Decision Support Systems, particularly with the multiagent paradigm like the MASCEM simulator. As such this work's proposal is of a new agent, a meta-learner based on STH technique that organizes several different ALBidS' strategies and combines the distinct answers into a single one that, expectedly, out-performs any of them.

Keywords:

Artificial Intelligence; Decision Support Systems; Electricity Markets; Multiagent Simulation; Machine Learning

Resumo

Nas últimas décadas, em quase todos os países industrializados, os mercados de energia sofreram um processo de reestruturação com o intuito principal de aumentar a competitividade do sector.

Denominada de liberalização, esta alteração trouxe várias mudanças, nomeadamente à estrutura das companhias de energia – onde antes uma única empresa era verticalmente responsável por todo o ciclo de fornecimento, desde a produção ao fornecimento, passando pela distribuição, da sua divisão resultam agora várias empresas que se dedicam exclusivamente a uma destas três actividades. Este novo aspecto mais horizontal do mercado gera situações concorrenciais, beneficiando o consumidor final que passa a poder escolher o seu fornecedor de energia.

Porém a liberalização dos mercados de energia traz consigo uma maior complexidade ao funcionamento da rede, e à comercialização de energia. Além da complexidade, as alterações introduziram um maior grau de imprevisibilidade e incerteza, forçando os intervenientes a repensar as suas estratégias e atitudes. Existem vários modelos de mercado, cada um com as suas próprias regras e diferentes desempenhos; gera-se assim diferentes necessidades de prever o comportamento deste mercado:

- Será do interesse dos reguladores detectar, atempadamente, falhas nas regras do mercado,

- Aos agentes de mercado interessa tirar partido desta nova estrutura para que possam aumentar os seus lucros.

O emprego de ferramentas de simulação é uma forma muito adequada para encontrar ineficiências de mercado ou de apoio à decisão dos intervenientes no mercado; nestas ferramentas o paradigma multiagente revela-se formidável para o trabalho dado que pode representar naturalmente as várias partes envolvidas num sistema dinâmico e adaptativo. Algumas ferramentas relevantes neste domínio são EMCAS [Koritarov, 2004], AMES [Li e Tesfatsion, 2009], e MASCEM [Praça *et al.*, 2003], [Vale *et al.*, 2011a].

O simulador MASCEM – "Simulador Multiagente para Mercados de Electricidade Competitivos" surgiu neste âmbito servindo de ferramenta de apoio às entidades intervenientes nos mercados de energia que lidam com a necessidade de melhor compreender o comportamento, a evolução das relações comerciais e os mecanismos destes mercados. A estrutura deste simulador permite a avaliação do comportamento do mercado aquando da introdução de novas regras e novos participantes nos mercados liberalizados de energia. Este simulador usa teoria de jogos, técnicas de aprendizagem, análise de cenários e técnicas de optimização para modelar agentes de mercado que agem de uma forma dinâmica. Este sistema também possui histórico das interacções entre os agentes e de mercado pelo que pode suportar as decisões de cada um dos agentes de acordo com as suas características e objectivos.

O MASCEM reproduz dois ambientes de mercado distintos – Mercado de Bolsa e Negociação de Contratos Bilaterais. No Mercado de Bolsa, existe um operador responsável pela gestão do início e do fecho de cada dia de negociações, e pela determinação do preço de mercado para o dia.

Sobre este simulador foi desenvolvido o sistema ALBidS, cujo principal objectivo é criar uma ferramenta adaptativa, com faculdades de aprendizagem que ofereça uma maior eficácia no apoio à decisão às entidades de mercado. O ALBidS oferece suporte às decisões feitas no Mercado de Bolsa realizando previsões do preço de mercado e propondo valores de licitação que determina serem adequados; estas decisões são sustentadas pelas suas capacidades de análise de contexto e de análise de histórico. O ALBidS integra várias estratégias de decisão, cada uma delas abordando o problema de diferentes formas, para que conjuntamente possam contribuir para a melhor decisão.

Edward de Bono [de Bono E., 1985] criou a técnica "Seis Chapéus do Pensamento" como um instrumento para ajudar a ver uma decisão por perspectivas diferentes. O objectivo desta ferramenta é obrigar uma pessoa a sair do seu estilo habitual de raciocínio. Foi desenvolvida para ser usada primariamente em reuniões para que estas decorram mais rapidamente tendo como resultado decisões melhores.

Este método apresenta-se como o oposto do pensamento argumentativo, conflituoso; nele é pedido a cada participante que alterne a sua forma de pensar, evitando sempre o confronto ou a crítica não construtiva; isto permite a exploração total do assunto em discussão.

Esta dissertação apresenta um estudo sobre a aplicabilidade da técnica Seis Chapéus do Pensamento em Sistemas de Apoio à Decisão, nomeadamente, com o paradigma multiagente do simulador MASCEM.

Para tal foi necessário traçar o plano de abordagem, verificando se seria necessário fazer alterações ao formato de resposta ou às interacções entre agentes. Neste plano de abordagem, determinou-se que as decisões feitas pelo ALBidS seriam vistas como resultado de uma reunião com vários intervenientes, cada um com um método diferente de abordar o problema – considerou-se desnecessário criar uma estrutura em que cada interveniente abordaria o problema de várias formas. Cada uma destas formas distintas de pensar foi associada a uma estratégia diferente do ALBidS. Foi também necessário criar uma estrutura deliberativa, presente no método, que pegasse nas ideias postas "na mesa" e determinasse a decisão final da reunião.

Um contributo importante deste trabalho está na proposta da combinação de vários algoritmos, tentando assim conseguindo um resultado melhor que os resultados individuais.

Palavras-chave: Inteligência Artificial; Sistemas de Apoio à Decisão; Mercados de Energia; Simulação Multiagente; Machine Learning

Acknowledgements

Any work of a considerable size such as this could not have been completed without the collaboration of others.

My first words are to my wife, for all the support and strength she gave me in this entire endeavour. Without her I could have not accomplished this work.

A very special word to my supervisor Dr Isabel Praça, for all the guidance, for accepting my idea for this work, and for believing in me when I am sure others would have not. Thank you for your posture and attitude, they made me feel comfortable enough to err, to think loudly and to bring out all my ideas no matter how silly they were. I am positive that lots of ideas in this work only bloomed because there was space to prune out the bad ones. Thank you for all the times you waited for me to our meetings because my work schedule only allowed us to meet late in the afternoon. You also made me believe that I can achieve one degree more, thank you.

To Tiago Pinto, words cannot express how thankful I am for all that you've done for this thesis; your insightful view on how to develop this work led to better and faster results than I ever expected.

Tiago Miguel, your hard work was fundamental for this thesis. I cannot thank you enough for all your efforts and commitment. My most humble gratitude to you.

To GECAD, a place I got accustomed to in only four days, I felt home there, working day and night to finish this project. Special thanks to Gabriel Santos, who helped me pass the waiting times, the boring times, and the bad times, always pushing me into believing that I could do it.

To my family and especially to my dear nephews whose special dates I had to miss, so that I could finish this work on time.

Finally to my friends who had to bear with me and my frustrations and my absence.

Index

Figure List

Table List

Acronyms

STH	Six Thinking Hats
MAS	Multiagent system
MASCEM	Multiagent Simulator for Competitive Electricity Markets
AI	Artificial Intelligence
DisComp	Distributed Computing
DAI	Distributed Artificial Intelligence
OAA	Open Agent Architecture
ICL	Interagent Communication Language
DG	Distributed Generation
ALBidS	Adaptive Learning Strategic Bidding System
VPP	Virtual Power Players
2E	Efficiency/Effectiveness
SA-QL	Simulated Annealing Q-Learning
NN	Neural Network
SA	Simulated Annealing
TS	Tabu Search
PSO	Particle Swarm Optimization
GA	Genetic Algorithm
WPMP	Wholesale Power Market Platform
SML	Simple Metalearner
WML	Weighted Metalearner
RLA	Reinforcement Learning Algorithm

1 Introduction

Ever since the 80s of the 20th century, the electricity industry has been facing an important new challenge – a market environment is replacing the traditional centralized-operation approach thus creating a more competitive environment.

This deregulation, often accompanied by privatization processes, brought many changes, for example where many electricity companies used to be responsible for all production cycle, they are now split into several companies each focusing exclusively on generation, transmission or distribution. The change also gives a more horizontal nature to the energy market giving the consumers a greater role in the market, hitherto unable to pick their energy supplier.

The new market is also a more complex and unpredictable one, forcing interveners to rethink their strategies and behaviour. Several market models exists, with different rules and performances creating the need to foresee market behaviour, regulators want to test the rules before they are implemented and market players need to understand the market so they may reap the benefits of a well-planned action. The employment of simulation tools is a very adequate way to find market inefficiencies or to provide support for market players' decision; the Multiagent systems paradigm is formidable for the job, as it can naturally represent several constituents interacting in a dynamic, adaptive system. Some relevant tools in this domain are EMCAS [Koritarov, 2004], AMES [Li and Tesfatsion, 2009], and MASCEM [Praça *et al.*, 2003], [Vale et al., 2011a].

In [Pinto T., 2011] in order to complement MASCEM simulator with new strategies, learning and adaptability, a new system was proposed: ALBidS – Adaptive Learning Strategic Bidding System. This new system implements several new strategies and behaviours along with those originally implemented in MASCEM.

This work aims to treat these two systems as if they were simulating a meeting to provide suggestions of which are the best actions for the supported player to perform. As such it would be interesting to study how far it is possible to adapt a tool for group discussion and individual thinking such as De Bono's Six Thinking Hats to these tools; also this method's application will allow us to study the outcome of combining diverse strategies.

2 Electricity Markets Simulation

2.1 Energy Markets

Around the world, the electricity industry experienced major changes in the structure of its markets and regulations. This transformation is often called the deregulation of the electricity market. The industry is becoming competitive; a market environment is replacing the traditional centralized-operation approach; this allows for market forces to drive electricity's price [Praça *et al.*, 2003].

The restructuring made the market more complex, challenging all conceptual models that previously dominated the sector [Pinto *et al.*, 2009a], as the complexity grows so does the difficulty of making an informed suitable decision; as such the intervenient entities need to rethink about their behaviour and market strategies.

This event gave way for the usage of new software tools – such as price-based unit commitment and price-bidding tools—to support new market activities. Also, it create the necessity to think about new modelling approaches that simulate how electric power markets might evolve over time and how market participants might react to the changing economic, financial, and regulatory environment in which they operate [Praça *et al.*, 2003]. Simulation and Artificial Intelligence techniques may be very helpful under this context.

All over the world, companies and governments met new challenges in the area of generation, transmission, distribution, and retail of electrical energy; the former vertically operated public utilities no longer regulate the market, leaving competition to form the price. Electricity markets are, however, a special case of a commodity market, due to the difficulty on storing electrical energy and to the need of a constant balance between generation and load.

Benefits on this free market approach are directly related to the efficiency of itself, the definition of the market structure implies rules and regulations which should not encourage strategic behaviours that diminish market performance. An electricity market's main objectives are to ensure the system's secure and efficient operation and to decrease the cost of electricity through competition.

By its own nature, electrical energy generation is a distributed problem; while traditionally electricity was produced in a small number of power plants, the system now also includes several new distributed sources, mainly renewable (see sections 2.1.7 and 2.3.1). This system is, naturally, much more difficult to control, since it includes many more power generation plants, and the generation itself is more unpredictable due to the difficulty in forecasting the energy production of some renewable sources (e.g. wind and photovoltaic).

2.1.1 Regulatory Models

The market environment typically consists of a pool (symmetric or asymmetric), as well as a floor for bilateral contracts. A balancing market is also necessary. These markets also, usually, include a market operator and a system operator. The market operator is responsible for the correct functioning, and initiating of the market; it manages the pool using a market-clearing tool to establish market price and the set of accepted bids for every negotiation period. The system operator is responsible for the management of the transmission grid and also analyses every established contract for technical feasibility (from the power system point of view).

2.1.2 Spot Market

The spot or day-ahead market is a daily basis functioning market [Pinto T., 2011], where players negotiate electric power for each hour, or half hour of the following day. Such markets are structured to consider production fluctuations as well as differences in production costs of distinct units.

In this market, each participating entity must present their selling or buying proposals for each of the 24 or 48 periods of a day. These proposals or bids are typically composed by a tuple *(power, price)*, with different meanings, whether they come from buyers or sellers, respectively: *power* stands for amount of power to be bought or sold, and *price* is the maximum accepted price or minimum selling price.

When the negotiation is finished, a economic dispatch for each period is set by the market operator. At the end of each period the market operator uses a market-clearing tool establishing the market price – a unique price that will be applied to all transactions of this period.

2.1.2.1 Symmetric and Asymmetric Pools

In pools, the most common type of negotiation is a standard uniform auction. If only suppliers can compete in the pool, it’s called an asymmetric market. If both suppliers and buyers can compete, it’s a symmetric market (similar to a double auction).

In an asymmetric market, the suppliers present their bids, and the market operator orders them starting with the lowest price and moving up. The consumers reveal their needs to set up the demand. Once the market operator knows the demand, it accepts the suppliers’ bids starting from the lowest and accepts as many as are necessary to fill the demand. The market price—to be paid to all accepted suppliers—is that of the last accepted bid (the one with the highest price). In a symmetric market, suppliers and consumers both submit bids. The market operator orders the selling and demand offers: selling bids start with the lowest price and move up, and demand bids start with the highest price and move down. Then, the proposed bids form the supply and demand step curves, and the point at which both curves intersect determines the market price, paid to all accepted supplier and consumers. The bids of every supplier offering prices lower than the established market price and every consumer offering prices higher than the market price will be accepted. Figure 1 depicts both cases.

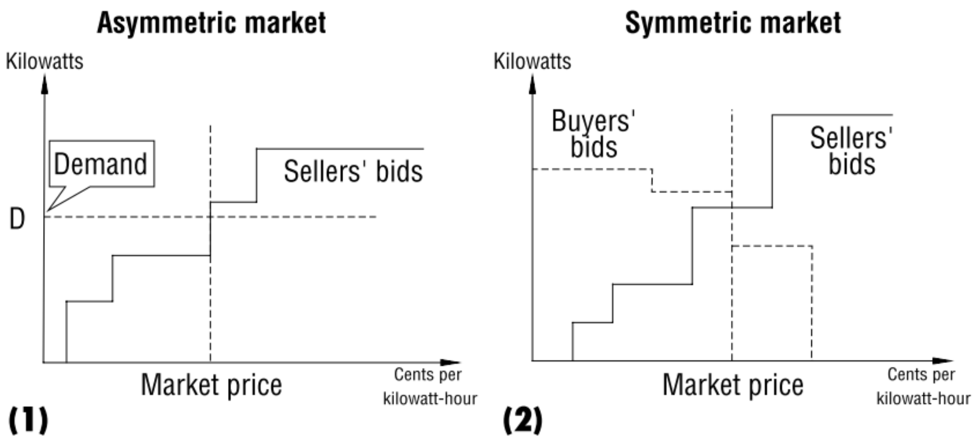


Figure 1 – Asymmetric and Symmetric Markets, from [Praça *et al.*, 2003]

2.1.3 Complex Market

The complex market [Santos *et al*., 2011] provides the opportunity for the presentation of restrictions that allow players to leave the market if they are not respected, meaning that players are not interested in participating unless those conditions are met. In addition, to meet the requirements of the simple day-ahead pool, the complex market includes at least one of the following conditions: *Indivisibility*, *Charge Gradient*, *Minimum Income* and *Scheduled Stop*. Some complex conditions – non-technical – are also used by market agents as strategies for achieving the highest possible profit.

2.1.3.1 Complex Conditions

The *Indivisibility* condition allows setting a minimum value of operation in the first offer of each period. Below this value, the participation of the production unit on the market is not possible. This condition applies to generating units that cannot work below a technical limit.

The *Charge Gradient* condition refers to the ramping up and down of plants; it allows establishing the maximum difference between the initial and the final power, between periods, for a production unit. This allows avoiding abrupt changes between consecutive periods (resulting from technical impossibility of achieving such changes).

The *Minimum Income* condition is used to ensure that the production unit does not enter the market if it cannot obtain a minimum amount in Euros (€), in the total of all periods, plus a variable fee per transacted kWh. This restriction depends on the sales strategy of each agent.

The *Scheduled Stop* condition is used in situations when the production unit has been withdrawn for not meeting the condition of required Minimum Income. This condition ensures that the production stopping is not done abruptly, rather undertaking a scheduled stop in a maximum time of 3 hours, avoiding production to immediately decrease to zero, from the last period of one day to the first period of the next. This is done by accepting the first offer of the first three periods as a simple offer, with the sole condition that the offered power is decreasing in each period, to smooth the production decrease until it gets to zero.

The market operator must assure the economical dispatch taking into account the specified conditions, which may imply the renegotiation of the period or day in matter, depending on the possible removal of entities that have presented competitive bids but whose complex conditions were not satisfied. In day-ahead market, only seller agents may present complex conditions.

2.1.4 Bilateral Contracts

Bilateral contracts are negotiable agreements between sellers and buyers (or traders) about power supply and receipt [Praça *et al.*, 2003]. The bilateral contract model is flexible; negotiating parties can specify their own contract terms. These negotiations are direct and made outside the spot market, this provides opportunities for reaching some advantageous agreements from an economic perspective, but also from a spatial one, when negotiating with players that offer benefits resulting from their location [Pinto T., 2011].

Bilateral contracts are established through requests for proposals distributed by buyers or traders — the demand agents. If a demand agent chooses to participate in the bilateral market, it will first send a request for electricity with its price expectations to all the sellers in the simulated market. In response, a seller analyses its own capabilities, current availability, and past experience. The seller must be sure that it's feasible to deliver energy to the buyer's location. So, it must get the network operator's feedback before reaching agreement with the demand agent. If the seller can make an offer to the requested parameters, it formulates a proposal and sends a message to the demand agent. The demand agent evaluates the proposals and accepts or rejects the offers [Praça *et al.*, 2003].

2.1.5 Balancing Markets

The purpose of balancing markets is to serve short-term operational security of supply (security of grid operation) to deal with imbalance settlement [Morais *et al.*, 2008].

The consideration of complex conditions (2.1.32.1.3.1) is essential for the balancing market [Santos *et al.*, 2011].

The balancing market's goal is to take care of the necessary adjustments on the viable daily program and the last final hourly program, correcting possible deviations from forecasted production or consumption [Vale et al., 2011a]. It is, therefore, a complementary platform to the day-ahead market. Although only sellers can present complex conditions to the spot market, in the balancing market, both sellers and buyers may present complex conditions. Another important issue is that sellers may become buyers and buyers may become sellers on the balancing market. That is also a new subject to be explored by market players when defining strategies for bid definition.

2.1.6 Complementary Markets

Electric energy is one of the most commonly used forms of energy [Pinto T., 2011]. With the shortage perspective of the non-renewable resources and the increasing usage of electric energy, it becomes imperative to develop new methods of energy production, investing in technologies that contribute to a more energetically rational way of living.

The verified growth of the investment on distributed generation, namely in wind and photovoltaic technologies, has been creating new opportunities for the promoters that own such technologies. Besides the selling of electrical energy to the system operators or in energy markets, the promoters can develop their activity in other markets, such as the Carbon Market, the Green Certificates emission, or the selling of water steam and hot water, among others. An alternative potential business is the integration with industries as livestock, the treatment of municipal solid waste, cork, in order to significantly reduce investment and/or operation costs [Pinto T., 2011].

These market mechanisms are complementary to the electric market, originating a more dynamic and alternative global market. The complementarity between such different types of markets creates the opportunity for players to improve their negotiating approaches, considering the investments in different markets.

The increasing complexity brought by the conception of such a diversity of market types resulted in high changes concerning the relationship between the electricity sector entities. It also resulted on the emergence of new entities, mostly dedicated to the electricity sector and electricity energy trading management. In what regards the commercial transactions, the analysis of different market mechanisms and the relationship between market entities becomes crucial. Namely in the case of Portugal, where the Iberian market, in partnership with Spain, has materialized not long ago, there are many aspects to analyse, improve, and even redefine. All market participants develop interactions among them, needing information systems for that purpose. As the observed context is characterized as being of significant adaptation and change, the need for decision support tools directed to this markets' analysis is also accentuated [Pinto T., 2011].

2.1.7 Virtual Power Players

The increase of distributed generation (DG) has brought about new challenges in electricity markets and in DG units operation and management. Despite the favourable scenario to DG growth, there are important aspects to consider, both of economic and technical nature. Issues such as the dispatch ability (namely in wind and photovoltaic technologies), the participation of small producers in the market and the high cost of maintenance require further attention. Virtual Power Producers are composed of multi-technology and multi-site heterogeneous production entities, which can enable overcoming some of these problems. They can also aggregate consumers and other energy resources such as storage, becoming Virtual Power Players (VPP) [Oliveira *et al.*, 2009], [Praça *et al.*, 2008].

2.2 Multiagent Systems

In the mid to late 1970s, a new paradigm on Artificial Intelligence (AI) research was born – Distributed Artificial Intelligence (DAI). This area of study evolved from two areas: Artificial Intelligence itself and Distributed Computing (DisComp). Multiagent systems can be seen as "systems in which several interacting, intelligent agents pursue some set of goals or perform some set of tasks" [Weiss G., 1999a]. This field of study gained widespread and recognition around mid-1990s, however, since then this field has grown enormously [Wooldridge M., 2009a].

Supporting this growth are reasons such as [Wooldridge M., 2009a], [Wooldridge M., 2009b] the belief that this paradigm is suitable to properly exploit the possibilities offered by DisComp, particularly the Internet; the continual cost reduction in computing capability and the ease of creating systems that are network-capable. There is, however, much more to MAS than this; [Wooldridge M., 2009a] defines them as a "*natural metaphor for understanding and building a wide range of what we might crudely call artificial social systems*". Following this social system thread of thought, it is viable to think of an agent as a computer system that made to act on behalf of a human.

2.2.1 Applications

Agent technology applications [Jennings and Wooldridge, 1995] range from simple systems (e.g. Microsoft's TIP WIZARD) to very large, interoperable expert systems or databases (e.g. ARCHON [Jennings N., 1994]) involved in the control or management of such complex systems as electricity distribution and supply, particle accelerators, and more. According to [Jennings and Wooldridge, 1995], we can identify three kinds of agents: "*gopher*" simple agents that can only execute straightforward tasks based on pre-specified rules and assumptions; sophisticated "*service-performing*" agents that execute high-level tasks at user's demand; and "*predictive/proactive*" agents which volunteer information or services to a user, without being asked.

2.2.2 Definition of Agency

Although there is not a universally accepted definition for an agent, most researchers would agree that an agent is a self-contained problem solving entity (implemented in hardware, software or a mixture of the two) exhibiting some or all of the following properties [Jennings and Wooldridge, 1995], [Weiss G., 1999a], [Weiss G., 1999b]:

- Autonomy – an agent's behaviour depends partially on its own experience. In other words, an agent controls, to some extent, their own behaviour and act without the intervention of humans and/or other systems. They also have a degree of control over their own internal state;

- Social/Interaction ability – Agents should have, a mean by which they can communicate with other agents, and an internal mechanism for deciding when and which social interactions are appropriate. Agents may be affected by external entities in pursuing their goals and executing their tasks. This interaction may occur indirectly though the environment in which they are embedded or directly through a shared language;

- Responsiveness – an agent must have some kind of sensor device (physical or not) through which they perceive the surrounding environment and respond internally (by a change of state) and/or externally (through effectors) in a timely and according fashion to changes that occur in it.

[Jennings and Wooldridge, 1995] also reference the proposal of the following characteristics:

- Proactiveness – an agent's actions should not simply be on an action-reaction basis, it should display opportunistic, goal-directed behaviour, taking initiative when and where it is appropriate.

- Adaptability – the ability of an agent to modify its behaviour over time in response to changing environmental conditions or an increase in knowledge about its problem solving role;

- Mobility – the ability of an agent to change its physical location to enhance its problem solving;

- Veracity – the assumption that an agent will not knowingly communicate false information;

- Rationality – the assumption that an agent will act in order to achieve its goals and will not act in such a way as to prevent its goals being achieved without good cause.

Intelligent agents are expected to pursue their goals and execute theirs tasks in an optimized manner when compared to some given performance measures [Weiss G., 1999a]. This notion of intelligent agent is not related to omniscience, omnipotence, or failing proof. A single agent does not have all the knowledge of the system, nor are they required to – it would not make much sense to have more than on agent if that were the case, this also applies to the notion of being able to do all tasks. Finally, agents do fail; however, it is expectable that on such occasion an agent has the capability to learn from its own mistake. Agents should operate flexibly and rationally adapting themselves to environmental circumstances, given their own internal state and effectual capabilities.

Distributed Artificial Intelligence major focus is on processes such as problem solving, planning, searching, decision making, and learning [Weiss G., 1999a]; these processes are perfect for intelligent agents to show flexibility and rationality in their behaviour, and on the realization of such processes in multiagent scenarios.

2.2.3 Sensors and Effectors

An agent must be able to meet its design objectives. In order to do that, an agent must have at least some control over the environment where the agent is, this repertoire of actions available to an agents is the *effectoric capability* of the agent [Wooldridge M., 2009c]. This capability is subject to preconditioning defining which actions can be done and when.

In the majority of the systems, agents will have at most partial control over the environment being able to influence it. This means that, when embedded in a MAS, every action of an agent is non-deterministic in that, other agents' actions in the system might cause different results or even failure. Thus all agents must be prepared for the possibility of failure.

Sensors are the agent's input devices, like the effectors, they can be a piece of hardware or software. Sensors are the "eyes" of the agent to the environment where the agent belongs; a sensor might be a camera or infrared sensor, or a software input device. Each input is commonly known as a percept, an agent's percept sequence is the complete history of everything the agent has perceived. Figure 2 illustrates an agent and its interactions with the environment.

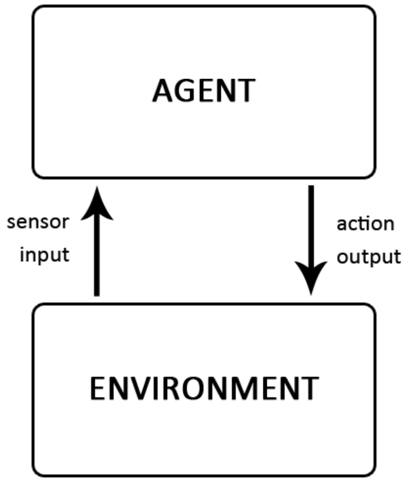


Figure 2 – An agent and its environment, adapted from [Wooldridge M., 2009c]

2.2.4 Multiagent Environment

A multiagent system is a system where several agents reside and may interact in a cooperative and/or competitive fashion. As such, some sort of coordination must be employed; this coordination is a form of interaction which is particularly important with respect to goal attainment and task completion [Weiss G., 1999a]. Coordination provides the means to achieve or avoid states of affairs that are considered as desirable or undesirable by one or several agents. [Russell and Norvig, 2009a] suggest the following classifications of environment properties.

2.2.4.1 Fully observable versus partially observable

If an agent's sensors give it access to the complete state of the environment at each point in time, then we say that the task environment is fully observable. From the agent's perspective, a partially observable environment, is one where it cannot obtain full information about the environment's state, for example, the agent's sensors may be physical and provide only local information about the environment. If the agent has no sensors at all then the environment is unobservable.

2.2.4.2 Competitive versus Cooperative

This classification is self-descriptive, although it must be pointed that, while designing a MAS, not every entity that at first appears to have be treated like an agent should be so. [Russell and Norvig, 2009a] give the example of a taxi driver A that can treat another vehicle B as an object and not an agent. They suggest that the key distinction between is “whether B’s behaviour is best described as maximizing a performance measure whose value depends on agent A’s behaviour”. In this taxi-driving environment, B and A agents can communicate to avoid collision and so it is a partially cooperative environment, of course, likewise A and B will compete for a parking spot, because the performance of each other will have a negative impact on the other, turning this environment to a partially competitive one.

Given this we can see that environments are not always exclusively cooperative nor are they exclusively competitive. In a cooperative environment, agents work together, gathering all their knowledge, effort, and capabilities to attain a common goal. On the other end, there is competition where agent’s goals are conflicting, leading to agents working against each other. While cooperative agents work as a team, failing or succeeding together, competitive agents work solely to maximize their own benefit thriving at the expense of others.

2.2.4.3 Deterministic versus stochastic

If the next state of the environment is completely determined by the current state and the action executed by the agent, then we say the environment is deterministic; otherwise, it is stochastic. Uncertainty may be ignored by an agent in a fully observable, deterministic environment (in this definition, uncertainty that comes from other agent’s actions is ignored).

Most real situations are so complex that makes it impossible to keep track of all the unobserved aspects, and for that reason they must be treated as stochastic.

2.2.4.4 Episodic versus sequential

In an episodic environment every experience is isolated from all others. In each episode the agent receives a percept and performs a single action – the next episode is not dependent on this one or any other previous episode. A good example of an episodic environment is one where an agent is responsible for the control of a robot arm in an assembly line; each part on the conveyor is treated individually.

On the opposite, in a sequential environment, every action may have long term consequences, and the agent is required not only to “think” ahead but also to keep track of every percept and the action taken.

2.2.4.5 Static versus Dynamic

A dynamic environment is one where changes on it can occur while an agent is deliberating; a static environment is the opposite. There is also the notion of semi dynamic where although the environment does not change, the performance score of the agent does (for example the time took to make a decision influences final score).

2.2.4.6 Discrete versus continuous

A discrete environment is one that is guaranteed to have a finite number of distinct states; a continuous environment on the other hand may be in uncountable many states. Chess is clearly discrete – the number of states is indeed very large but finite; while taxi driving is continuous.

2.2.4.7 Known versus unknown

[Russell and Norvig, 2009a] point out that, strictly speaking, this classification is not related to the environment but to the agent itself, and its knowledge about the environment. In a known environment, all the outcomes (or outcome probabilities in a stochastic environment) for any action are given. If and environment is unknown, the agent will have to learn how it works in order to make suitable decisions. [Russell and Norvig, 2009a] also remark that although it may look like they are the same, this distinction (known vs. unknown) is not the same as the distinction in 2.2.4.1 – an example of known partially observable environment is a solitaire card game where one knows the rules but has not seen cards that have not been turned over yet.

2.2.5 Agent Communication

Two of the most important paradigms in agent communication are [Botti *et al.*, 1995]:

- the actor paradigm, which is based on an object-oriented language where each agent has an independent life and communicates with others by sending messages;
- the blackboard paradigm, in which agents communicate by writing on a shared structure called a blackboard.

The first paradigm does not require a fully detailed explanation, although it must be emphasized that agents unlike objects, due to their independency can choose to not comply with those messages' orders. More detailed information about agent messaging, message formats and standards is beyond the reach of this work.

The blackboard is structured for organizing communications at various levels of abstraction, and an agent communicates with another one by writing on the blackboard. Those agents are activated (by the control component) when given patterns of information are present on the blackboard.

The blackboard model offers a powerful problem-solving architecture that is suitable in the following situations.

- Many diverse, specialized knowledge representations are needed
- An integration framework is needed that allows for heterogeneous problem-solving representations and expertise.

- Uncertain knowledge or limited data inhibits absolute determination of a solution. The incremental approach of the blackboard system will still allow progress to be made.

- Multilevel reasoning or flexible, dynamic control of problem-solving activities is required in an application.

The blackboard approach has been applied in numerous areas, including the following:

- process control
- planning and scheduling
- case-based reasoning
- knowledge-based simulation
- knowledge-based instruction
- symbolic learning

In each of these applications, the scope of the problem to be solved was the prime factor in selecting a blackboard approach. That is, deciding whether to use a blackboard approach should be based on the problem-solving requirements discussed above, rather than the specific application area [Corkill D., 1991].

2.3 MASCEM: Multiagent Simulator for Competitive Electricity Markets

The Multiagent Simulator for Competitive Electricity Markets – MASCEM [Praça *et al.*, 2003], [Pinto *et al.*, 2011c], [Vale *et al.*, 2011a] is a modelling and simulation tool that has been developed with the purpose of studying complex restructured electricity markets operation. MASCEM models the complex dynamic market players, including their interactions and medium/long-term gathering of data and experience, to support players' decisions according to their very own characteristics and objectives. MASCEM most important features are presented in Figure 3.

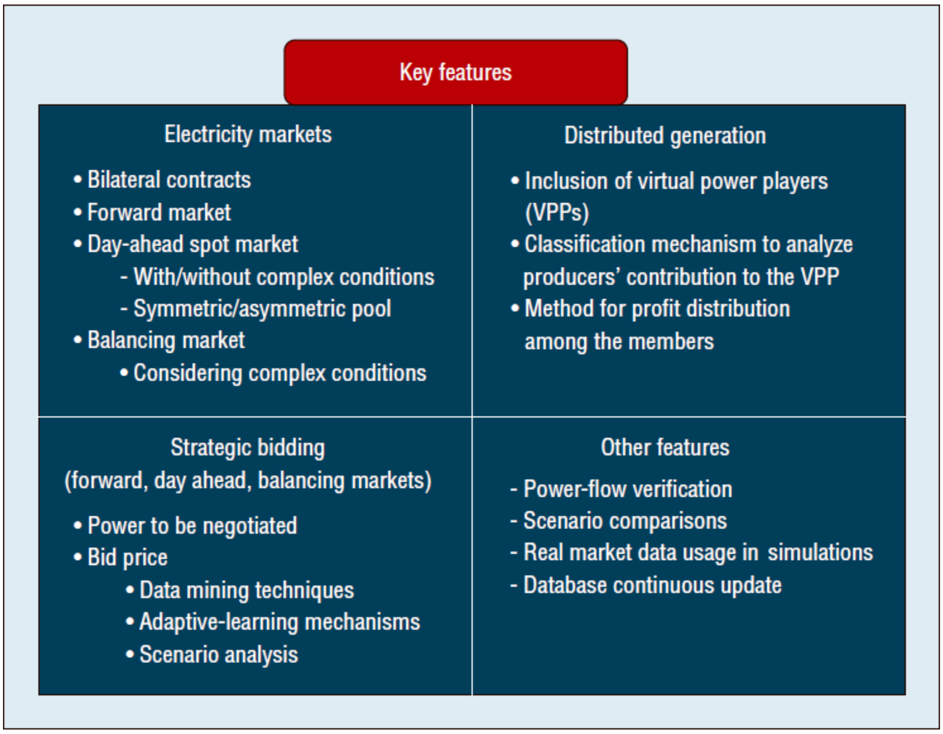


Figure 3 – MASCEM Key Features, from [Vale et al., 2011a]

MASCEM is implemented on the top of Open Agent Architecture (OAA) [OAA, 2007], using OAA AgentLib library, and Java Virtual Machine 1.6.0. The OAA's Interagent Communication Language (ICL) is the interface and communication language shared by all agents, no matter which machine they are running on or which programming language they are programmed in, allowing for integration of multiple software modules.

Communication and cooperation between agents are brokered by one or more facilitators, which are responsible for matching requests, from users and agents, with descriptions of the capabilities of other agents.

OAA is not a framework specifically devoted to develop simulations; some extensions were made to make it suitable to deal with the energy markets that MASCEM currently supports, namely to introduce the time evolution mechanism of the simulation.

MASCEM's goal is to be able to simulate as many market models and players types as possible so it can reproduce in a realistic way the operation of real electricity markets. This enables it to be used as a simulation and decision-support tool for short/medium term purposes but also as a tool to support long-term decisions, such as the ones taken by regulators.

Unlike traditional tools, MASCEM does not postulate a single decision maker with a single objective for the entire system. Rather, it allows agents representing the different independent entities in electricity markets to establish their own objectives and decision rules. Moreover, as the simulation progresses, agents can adapt their strategies based on the success or failure of previous efforts. In each situation, agents dynamically adapt their strategies according to the present context and using the dynamically updated detained knowledge.

MASCEM's key players reflect actual entities from real markets and provide a means for aggregating consumers and producers. Presently, there are agents representing market independent entities such as the system operator, which is another simulator that gets the economical dispatch and undertakes power-flow analysis to assure economical agreements can be implemented without disturbing power-grid stability and technical constraints.

The market operator agent regulates pool negotiations. This agent analyses bids presented to the pool and defines the market price and economical dispatch. It cooperates with the system operator by sending it the economical dispatch. The market operator agent uses different algorithms to account for complex conditions.

The need for understanding electricity markets' mechanisms and how the involved players' interactions affects the outcomes of the markets has contributed to the increased use of simulation tools in order to determine the best possible results in each market context for each participating entity. Multiagent-based software is particularly well fitted to analysing dynamic and adaptive systems with complex interactions among their constituents. Several of such modelling tools—designed to help researchers study restructured wholesale power markets—have emerged. In addition to MASCEM, other relevant tools in this domain are AMES, EMCAS, and MASI. Players in MASCEM are implemented as independent agents, with their own ability to perceive the states and changes in the world and to act accordingly. These agents are provided with bidding strategies, which must be adequate and refined to let them gain the highest possible advantage from each market context [Vale *et al.*, 2011a].

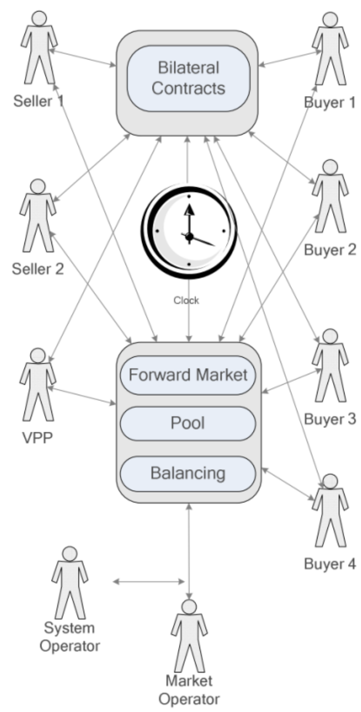


Figure 4 – MASCEM agents and negotiation framework, from [Praça *et al.*, 2003]

The seller and buyer agents are the two key players in the market. Sellers represent entities able to sell electricity in the market—for example, companies holding electricity production units. Buyers represent electricity consumers or even distribution companies. Sellers compete with each other because each seller is interested in maximizing its profits. They also cooperate with buyers while trying to establish a mutually profitable agreement.

2.3.1 Virtual Power Players in MASCEM

The aggregation of distributed generation plants gives place to the new concept of Virtual Power Player (VPP). VPPs integration into electricity markets is a very challenging domain that has been motivating MASCEM evolution; as it was referred before (see section 2.1), electricity production is a distributed problem, by nature, and VPPs give yet another dimension to this problem.

VPPs are responsible for managing the coalition of producers, which includes negotiating in the electricity market on behalf of the coalition and negotiating internally with their members, to guarantee that the terms of each member's contract are fair and suited to the VPPs' characteristics and objectives. For this process, MASCEM integrates a classification algorithm that analyses each producer's characteristics and tests their suitability to the VPPs' objectives. This provides the VPP with knowledge about which producers are most likely to favourably contribute to the VPPs' results, which lets it decide which producers to aggregate.

Coalition formation is the coming together of a number of distinct, autonomous agents that agree to coordinate and cooperate, acting as a coherent grouping, in the performance of a specific task. Such coalitions can improve the performance of the individual agents and/or the system as a whole. It is an important form of interaction in multiagent systems. The coalition formation process comprises several phases: coalition structure generation, optimization of the value of the coalition and payoff distribution [Pinto T., 2011].

Regarding the coalition formation process, for VPP modelling, the three main activities of coalition structure generation, optimization of the value of the coalition and payoff distribution should be considered under a scenario where agents operate in a dynamic and time dependent environment. This entails significant changes on MASCEM core model and communications infrastructure [Pinto T., 2011].

VPPs manage the information of their aggregates and are viewed from the market as seller agents. Each VPP is modelled as an independent Multiagent system, maintaining high performance and allowing agents to be installed on separate machines. To achieve this independence, individual VPP facilitators have been created to manage the communications between each VPP and its members independently from the rest of the simulation [Pinto *et al.*, 2009], [Oliveira *et al.*, 2009].

Figure 5 presents MASCEM's agent architecture – VPPs are visible to the Market Facilitator as a single agent (the VPP facilitator) hiding the coalition from the rest of the system.

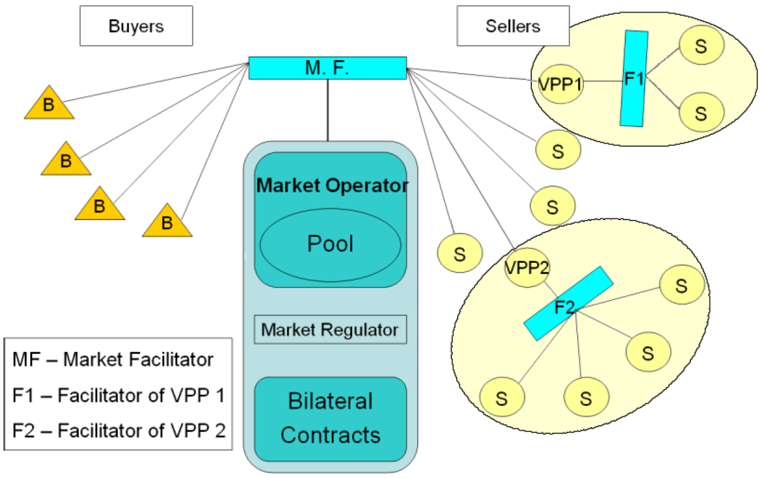


Figure 5 – MASCEM agent architecture, adapted from [Pinto *et al*., 2009]

To sell energy in the market VPP must forecast the generation of aggregated producers and "save" some power capacity to ensure a reserve to compensate a generation oscillation of producers with natural resources technologies dependent [Pinto T., 2011].

The VPP can use different market strategies, considering specific aspects such as producers established contracts and range of generation forecast. The prediction errors increase with the distance between the forecasting and the forecast times. The standard errors are given as a percentage of the installed capacity, since this is what the utilities are most interested in (installed capacity is easy to measure); sometimes they are given as the mean production or in absolute numbers [Pinto T., 2011].

MASCEM's modelling of VPPs enlarged the scope of negotiation procedures in this simulator, allowing the study of different types of negotiation outside the usual electricity markets' regulatory models [Pinto T., 2011].

2.3.2 Negotiation in MASCEM

MASCEM includes several negotiation mechanisms usually found in electricity markets, being able to simulate several types of markets, namely: Pool Markets, Bilateral Contracts, Balancing Markets and Forward Markets.

Figure 6 presents the negotiation sequence for one day simulation in MASCEM.

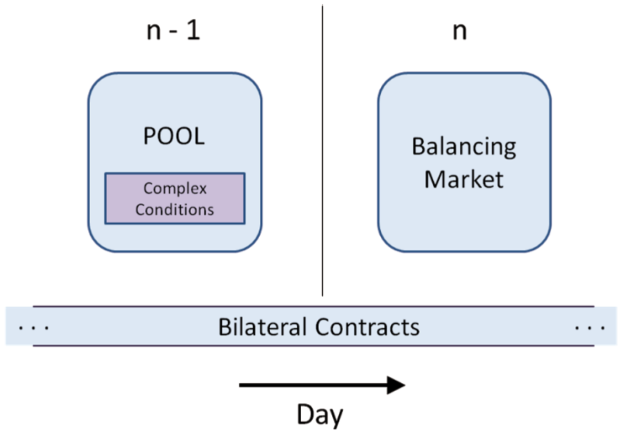


Figure 6 – MASCEM Negotiations timing for day n [Santos *et al.*, 2011]

Based on the previously obtained results, buyer and seller agents review their strategies for the future. The strategic behaviour of each agent defines its desired price and amount of power to be negotiated in each market.

Time-dependent strategies and behaviour-dependent strategies are part of each agent, and define the price to negotiate in the next day according to the results obtained previously. There are four types of time-dependent strategies [Praça *et al.*, 2003]:

- Determined – prices remain constant throughout the period of negotiation;
- Anxious – minor changes to the price are made after little trading time;
- Moderated – small changes to the price are made in an intermediate stage of negotiation period;
- Gluttonous – the price is significantly changed, but only in late trading.

On the other hand, the behaviour-dependent strategies are [Praça *et al.*, 2003]: Composed Goal Directed (when an agent has two consecutive goals, in which the definition of the second objective depends on the fulfilment of the first); Adapted Derivative Following (the results of price changes made in previous trading periods are analysed. If the agent finds that the change in the price of its proposals brought benefits, it maintains the same type of change for the next period. Otherwise, the change in price will go in the opposite direction); Market Price Following (this strategy bases the agent price fluctuations on the fluctuations of the market price).

Concerning the VPPs' operation, negotiations take place in some additional timings, namely in coalitions' formation and management, [Pinto *et al.*, 2011c], [Oliveira *et al.*, 2009]. This type of negotiation provides players with the capabilities of achieving the most advantageous coalition contracts, both for the aggregator (VPP) and for the members (sellers and buyers). These negotiations take into account the players' characteristics, objectives, and goals, and allows them to get alternative deals to those they could get by negotiating exclusively on the market.

The different types of negotiation approached in MASCEM, the different types of markets implemented, and the distinct interactions between the participating entities in different situations, create the fundamental need for the use of machine learning techniques in this simulator.

2.4 ALBidS – Adaptive Learning Strategic Bidding System

This section is partially abridged from [Pinto T., 2011] and [Pinto *et al.*, 2011a], its presence in this work is to provide a somewhat detailed context of what ALBidS is and what it does; for a deeper detail or experimental findings, please refer to [Pinto T., 2011]. Also, the structure of this section is different from Chapter 3 of [Pinto T., 2011]; given the scope of this work emphasis is given to the Reinforcement Learning Algorithms and Strategy Agents. Also because of this work's scope, Meta-learner Agents (although Strategy agents themselves) are presented in a different sub-section.

ALBidS [Pinto *et al.*, 2011a], [Pinto T., 2011] is a multiagent system directed to the definition of bidding strategies of market negotiating players. This system was developed in the Knowledge Engineering and Decision Support research Centre – GECAD – and is fully integrated with the MASCEM simulator.

In order to provide the negotiating players with competitive advantage in the electricity market it is essential to provide them strategies capable of dealing with the constant market changes, allowing adaptation to the competitors' actions and reactions. For that, it is necessary to have adequate forecast techniques to analyse the market data properly, namely the historic market prices. Prices prediction can be approached in several ways, namely through the use of statistical methods, data mining techniques, neural networks (NN), support vector machines (SVM), or several other methods [Pinto T., 2011].

This system was conceived to be capable of learning the best agent acting approaches, depending on each situation and context. This learning process takes into account the system's log, taking advantage of all the available information, including collected data during the use of the multiagent system itself. In order to achieve this purpose, several algorithms and learning methodologies were used, so that together they can contribute to the best decision making in each moment.

ALBidS algorithms and approaches encompass distinct areas, such as data mining techniques, data analysis and forecasting tools, pattern recognition, knowledge engineering approaches, artificial intelligence algorithms, and also the proposal of other algorithms directed to specific situations, such as the application of economic, mathematical, physics and psychology theories. The adequate combination of different types of algorithms is also considered relevant.

The ALBidS system includes context awareness which means the considered approaches are chosen in each moment according to the guarantees of success they offer in each context. The choosing process shall be performed through the application of algorithms driven by statistics and probabilities management. Through the consideration of several algorithms, based on completely different natures, a higher probability of at least one approach offering good results in each distinct situation and context is expected to be achieved.

ALBidS main objectives are described in [Pinto T., 2011] as listed:

- Development of a multiagent decision support tool directed to the strategic behaviour of market negotiating players, including:
    - Learning of the most adequate acting approaches, depending on each situation and the context it occurs, considering the system's log and the historic of past actions of the other agents considered in the system;
    - Proposal and testing of several learning and data analysis algorithms and methodologies, so that they can contribute together to the best decision making of the supported market players, including the development of algorithms directed to specific situations;
    - Implementation of a learning mechanism with the capability of choosing the most appropriate strategies at each moment, based on the methodologies performance statistics and probabilities; and the respective capability of calculating the disparity between each algorithm's proposals and the actual reality, so that it can properly update the confidence values of the algorithms' performances for each situation and context.
- Simulation of the intelligent action of players in electricity markets, concerning the following:
    - Integration of the developed system with the MASCEM electricity market simulator;
    - Utilization of the developed system to analyse the results of scenarios based on real electricity markets' data.

ALBidS' test scenarios uses real data extracted from the Iberian market – OMEL [OMEL, 2011].

ALBidS was developed regarding the accomplishment of these objectives, while providing the negotiating players with competitive advantage in electricity markets it is essential to endow them with strategies capable of dealing with the constant market changes, allowing adaptation to the competitors' actions and reactions.

2.4.1 Global Structure

ALBidS is composed by several agents and mechanisms; the central entity of the system is the Main Agent which is responsible for choosing the most appropriate answer among those received from every other agent performing each distinct algorithm, detaining the exclusive knowledge of its execution. This way the system can be executing all the algorithms in parallel, preventing the system's performance degradation, in the possible amount. As each strategy agent gets its answer, it sends it back to the Main Agent [Vale *et al.*, 2011a], [Pinto T., 2011].

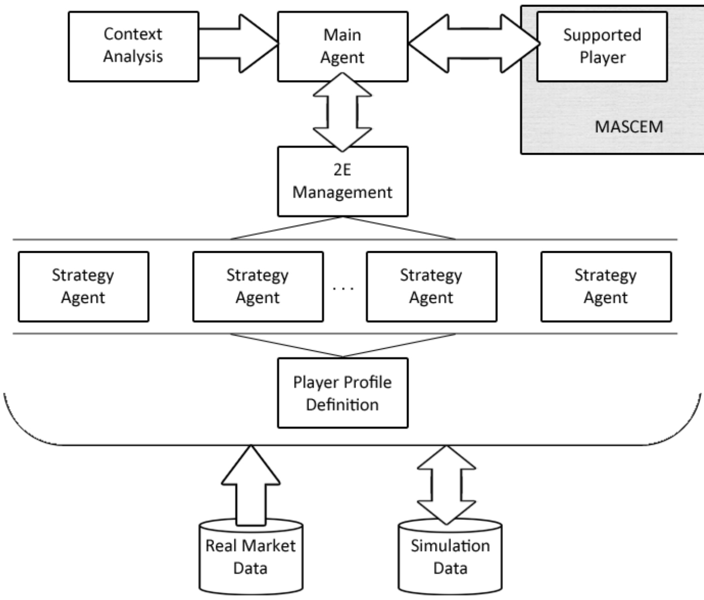


Figure 7 – ALBidS Global Structure [Pinto T., 2011]

Figure 7 presents the global structure of the ALBidS system, with the Main Agent as its central entity.

2.4.1.1 Main Agent

ALBidS is connected with the MASCEM simulator, providing a response to the negotiating players when they require intelligent support to act in the market. The connection between the two systems is managed by the Main Agent, using a Prolog Facilitator.

Being ALBidS a Multiagent system with an independent purpose from the MASCEM simulator, with its agents' interactions irrelevant to the functioning of the agents of MASCEM, and vice-versa, communication is managed independently from the rest of the simulation [Pinto *et al.*, 2011a]. This means having the communications between agents independent from those of MASCEM, to ensure the parallel processing from the two groups of agents. To achieve this, ALBidS' assigns communications management to an independent facilitator. Due to the OAA restrictions with the use of more than one facilitator simultaneously, ALBidS includes its own version of the OAA facilitator to manage the ALBidS agents' communications. ALBidS facilitators are implemented in LPA Prolog, as it guarantees a level of efficiency and speed of processing that Java cannot give [Pinto T., 2011].

This agent acts as an intermediary between the two systems. It receives requests from the negotiating players when they require decision support, and provides them the corresponding answers. These answers are provided after managing the ALBidS internal mechanism, including the interactions with the strategy agents (or bid proposal agents) – the agents responsible for executing the different strategies.

Besides dealing with the communications and containing the reinforcement learning algorithms, the Main Agent also integrates the Context Analysis module, and the Efficiency/Effectiveness Management module.

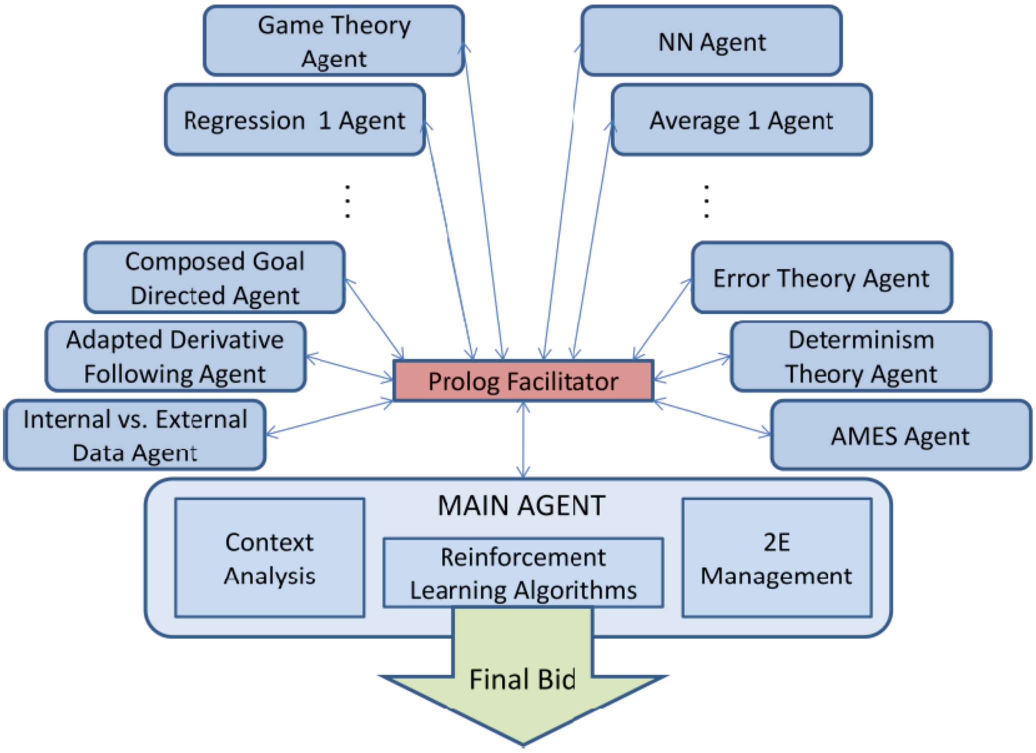


Figure 8 – Main Agent's role in ALBidS [Pinto T., 2011]

This agent's main responsibility is to execute the reinforcement learning algorithms [Vale *et al.*, 2011a]. In each moment and in each circumstance the technique that presents the best results for the actual scenario is chosen as the simulator's response. So, given as many answers to each problem as there are algorithms, the reinforcement learning algorithm will choose the one that is most likely to present the best answer according to the past experience of their responses and to the present characteristics of each situation, such as the considered day, the period, and the particular market context that the algorithms are being asked to forecast [Pinto T., 2011].

2.4.1.2 Context Analysis Mechanism

Contexts are an important factor in what concerns the adaptation of the approaches to be chosen as the final action to be performed in the market by the supported player. A mechanism to analyse and define different market negotiating contexts is present in ALBidS, hence providing the means for the chosen actions to be adapted and chosen depending of the different circumstances that are encountered at each moment.

The first step when analysing context in the electricity market environment is to consider its most basic conditionings, i.e. on what these negotiations depend: days and periods. These are the two main factors to consider when bidding in the market, since each bid much be submitted for each period of each day (see 2.1.2).

ALBidS context definition process takes into consideration the analysis of the situations concerning both perspectives, evolution throughout the days and throughout day periods. To perform this analysis, some influential conditionings that affect the prices in both cases were considered. The considered conditionings, or characteristics of a day and period are[Pinto T., 2011]:

- the market price for the period and day in matter;
- the amount of transacted power in the market;
- the wind intensity verified in that period of the day (this is important because it affects the production of wind plants, and therefore the total negotiated amount of power);
- the type of the day (whether it is a working day or weekend; if it is a holiday, or a special situation day, e.g. a day of an important event, such as an important game in a certain sport, which affects the energy consumption in that day, both because of the consumption in the stadium, and for increasing the number of people with the TV on to watch it).

The grouping of a day's periods depending on their context is performed through the application of a clustering mechanism. The clustering mechanism analyses the characteristics of each period throughout the days, and attributes each period to the cluster that presents the most similar characteristics. The clustering is performed using the K-Means clustering algorithm provided by MATLAB [Pinto T., 2011]. Further analysis on this mechanism is out of the scope of this work.

2.4.1.3 Player Profile Definition Mechanism

In order to build suitable profiles of competitor agents, it is essential to provide players with strategies capable of dealing with the possible changes in competitors' behaviour, allowing adaptation to their actions and reactions. For that, it is necessary to have adequate techniques to analyse the data properly, namely the historic of other agents past actions.

ALBidS player profile definition follows the same idea as the main system's methodology for the definition of market strategies.

The used reinforcement algorithm is the Roth-Erev algorithm [Roth and Erev, 1995] which will be presented in sub-section 2.4.2.2; it presents a distinct set of statistics for each acting agent, for their actions to be predicted independently from each other, and also for each period or market context. This means that an algorithm that may be presenting good results for a certain agent in a certain context, with its output chosen more often when bidding in this context, may possibly never be chosen as the output for another context. [Pinto T., 2011].

2.4.1.4 Efficiency/Effectiveness Management System

ALBidS also includes an Efficiency/Effectiveness (2E) management system so that the system can adapt to different simulation circumstances [Pinto T., 2011]. This mechanism provides the means for the system to adapt its execution time depending on the purpose of the simulation, i.e., if the expected results from ALBidS are as best as it is able to achieve, or, on the other hand, if the main requirement is for the system to be executed rapidly, since the purpose of the considered simulation is to analyse issues other than player's optimal performance in the electricity market [Pinto T., 2011]. The 2E Management mechanism manipulates the strategies both externally and internally. From the system's perspective this mechanism contributes by deciding which tools are used at each moment for each circumstance; depending on their observed performance in terms of efficiency and effectiveness. This way this mechanism can choose to exclude certain strategies when they are not fulfilling the ALBidS' requirements for the case in matter. The strategies chosen to be executed are also manipulated internally, so that they can adapt their individual results quality/execution time balance to the needs of the current simulation.

2.4.1.5 Reinforcement Learning Algorithms

This mechanism is used together with the context analysis mechanism. The main reinforcement algorithm presents a distinct set of statistics for each market context which means that an algorithm that may be presenting good results for a certain context, with its output chosen more often when bidding for this period, may possibly never be chosen to provide the final answer for another period that is defined as being in another context. The way the statistics are updated, and consequently the best answer chosen, can be defined by the user [Pinto T., 2011].

ALBidS learning mechanisms will be explored further in section 2.4.2.

2.4.1.6 Strategy Agents

Strategy agents are responsible for each distinct approach/algorithm in what concerns market negotiations and prices forecast. Currently, the following strategy agents are implemented in ALBidS:

- Composed Goal Directed Agent;
- Adaptive Derivative Following Agent;
- Market Price Following Agent;
- Average Agents;
- NN Agent;
- Regression Agents;
- AMES Agent;
- Game Theory Agent;
- Simulated Annealing Q-Learning Agent;
- Error Theory Agents;
- Economic Analysis Agent;
- Determinism Theory Agent; and
- Metalearner Agents.

More detail on each of these strategy agents will be presented on section 2.4.3.

2.4.1.7 Database Retrieval Agent

This agent has the access to the database containing the historic and current market data, which is constantly updated. It also accesses the database containing the agents' actions history, which is updated with every action that a player performs, such as bids sent to the market, or bilateral agreements that it binds [Pinto *et al.*, 2011a].

When asked, it gets the requested data from the required database and provides it to the requester, to use in its analysis. This agent was created to facilitate the access to the data, and prevent database access conflicts.

2.4.2 Reinforcement Learning Algorithms

MASCEM provides three reinforcement learning algorithms that can be chosen, all having in common the starting point. All the algorithms start with the same value of confidence, and then according to their particular performance that value will be updated. All algorithms also have the option of being attributed a weight value that defines their importance to the system. This means that a strategy that has a higher weight value will detach faster from the rest in case of either success or failure. There have been implemented three reinforcement learning algorithms, which differ in the way the stats for each strategy are updated. This way the system has the capability of choosing among the proposals using different perspectives, which improves the chances of one being more advantageous for each situation [Pinto T., 2011].

The implemented learning algorithms are described in the next sub-sections.

2.4.2.1 Simple Reinforcement Learning

Reinforcement learning is an area of machine learning in computer science, and therefore also a branch of Artificial Intelligence. It deals with the optimization of an agent's performance in a given environment, in order to maximize a reward it is awarded [Pinto T., 2011].

A reinforcement learning agent interacts with its environment in discrete time steps. At each time t, the agent receives an observation o_t, which typically includes the reward r_t. It then chooses an action a_t from the set of available actions. The environment moves to a new state s_{t+1} and the reward r_{t+1} associated with the transition (s_t, a_t, s_{t+1}) is determined. The goal of a reinforcement learning agent is to collect as much reward as possible. The agent can choose any action as a function of the history and it can even randomize its action selection [Pinto T., 2011].

The updating of the values is done through a direct decrement of the confidence value C in the time *t*, according to the absolute value of the difference between the prediction P and the real value R [Pinto *et al.*, 2011a]. The updating of the values is expressed by:

$$C_{t+1} = C_t - |(R - P)| \tag{2.1}$$

Reinforcement learning main and most distinguishing characteristics are trial-and-error and delayed reward [Rahimi-Kian *et al.*, 2005]. Optimal action requires reasoning about long term consequences; because the agent is not told what to do, it must explore which actions provide the most reward by trial-and-error. In the most interesting and challenging cases, actions may affect not only the immediate reward, but also the next situation, and through that all subsequent rewards.

This method of machine learning also presents another challenge: the trade-off between exploration and exploitation. While such an algorithm must prefer past effective actions that have proven themselves to be rewarding, it must also select actions not taken before to find if there are more profitable actions. The dilemma is that neither exploitation nor exploration can be pursued exclusively without failing at the task. Thus, reinforcement learning is particularly well suited to problems which include a long-term versus short-term reward trade-off [Pinto T., 2011].

2.4.2.2 Roth-Erev Reinforcement Learning

The main reasoning behind a reinforcement learning algorithm is that the tendency to perform an action should be strengthened, or reinforced, if it produces favourable results and weakened if it produces unfavourable results.

The main principle behind a reinforcement learning algorithm is that, not only are choices that were successful in the past more likely to be employed in the future, but similar choices will be employed more often as well. This is referred to as law of effect [Erev and Roth, 1998]. Another principle suggested by Roth and Erev is the power law of practice; this principle suggests that learning curves initially tend to be abrupt, and after some time they flatten out.

In order to experiment on each of these learning principles' responsibility over a person's learning process, Roth and Erev have developed a reinforcement learning algorithm, named as the Roth-Erev algorithm [Roth and Erev, 1995].

The differences from this methodology to the Simple Reinforcement Learning Algorithm concern the inclusion of a recency parameter, which defines the importance of the past experience to the evolution of the learning process. This way, taking advantage on this parameter, it is possible to define if, for each case, it will be attributed a higher importance for the recent events, adapting faster to the changing environment; or on the other hand, if the accumulated past experience, and the achieved results in the past, will be the most influent factor in defining the confidence in an action's performance [Vale *et al.*, 2011a], [Pinto T., 2011].

This revised Roth-Erev reinforcement learning algorithm that, besides the features of the previous algorithm, also includes a weight value W for the definition of the importance of past experience [Pinto *et al.*, 2011a]. This version is expressed by:

$$C_{t+1} = C_t * \mathrm{W} - |(R - P)| * (1 - W) \tag{2.2}$$

2.4.2.3 Bayes Theorem

This strategy implements an application of the Bayes Theorem, directing this theorem's principles to the development of a reinforcement learning algorithm based on the estimation of success probabilities [Vale *et al.*, 2011a].

The Bayes Theorem has been applied in several scopes throughout the times, taking advantage on this probability theory's capabilities of supporting applications directed to the most alternative contexts. One of this theorem's advantages that has been considered interesting was its applicability to be used as a reinforcement learning algorithm [Pinto *et al.*, 2011a]. This applicability is based on the use of a probability estimation to determine which of the different alternatives (strategies' suggestions) presents a higher probability of success in each context, therefore being considered as the most appropriate approach.

The Bayes Theorem reinforcement learning algorithm applies the Bayes Theorem through the implementation of a Bayesian network.

In this adaptation, each strategy (suggested action) is represented by a different knot. All strategy knots are connected to an output knot, which is responsible for the final decision on which is the most advantage action. This decision is based on the calculation of the success probability of each strategy, based on the observed events: if a strategy has accomplished to be the most successful amongst all in a certain negotiation period (event Yes), or not (event No) [Pinto T., 2011].

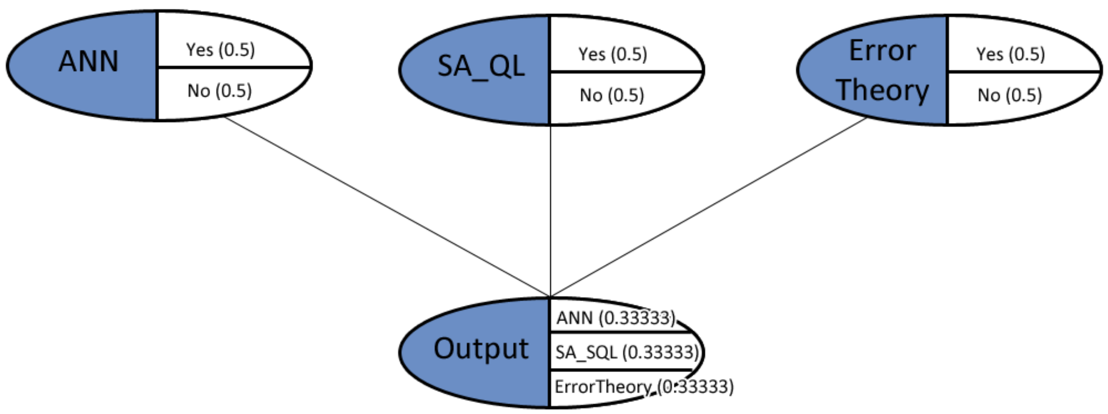


Figure 9 – Bayesian network's topology with three strategies adapted from [Pinto T., 2011]

Because a Bayesian Network does not represent a temporal probabilistic model, ALBidS implements this learning mechanism using a dynamic Bayesian Network; this dynamism has been achieved by including a Counting-Learning algorithm.

The Counting-Learning algorithm considers, for each case about to be learned, an experience value, which defines in what degree the considered case will affect the conditional probability of each knot. This algorithm guarantees that all knots' probabilities are updated.

The updating of the values is done through the propagation of the probability of each algorithm being successful given the facts of its past performance. The expected utility, or expected success of each algorithm is given by (2.3), being *E* the available evidences, *A* an action with possible outcomes O_i, $U(O_i|A)$ the utility of each of the outcome states given that action A is taken, $P(O_i|E,A)$ the conditional probability distribution over the possible outcome states, given that evidence *E* is observed and action *A* taken.

$$EU(A|E) = \sum_i P(O_i|E,A) * U(O_i|A) \tag{2.3}$$

2.4.3 Strategy Agents

The ALBidS system integrates several very distinct approaches in what concerns market negotiations, and prices forecast. It is possible to apply several different methodologies to achieve both these objectives. None of them presents an evident superiority relatively to the others in terms of results, especially when applied to different contexts and circumstances of simulation with distinct characteristics. For this reason, this system integrates a large number of distinct approaches: data mining techniques, forecasting methods, artificial intelligence methodologies, application of electricity markets directed strategies, mathematic approaches, economic theory based models, and the adaptation of physics theories. This way the system is able to take advantage of the best characteristics of each approach whenever they show to be advantageous [Pinto T., 2011].

The strategies differ from each other only from an internal point of view, i.e. their behaviour, while for the system they are all viewed in the same way, with the communications treated equally for all.

ALBidS includes the strategies initially developed for MASCEM [Praça *et al.*, 2003], these are Composed Goal Directed and Adaptive Derivative Following.

2.4.3.1 Composed Goal Directed

The composed goal-directed strategy is based on two consecutive objectives— for example, selling (or buying) all the available capacity (consumption needs) and then increasing the profit (or reducing the payoff). Following this strategy, sellers will lower their price if, in the previous period, they didn’t completely sell the available capacity. They’ll raise the price if they sold all the available capacity in the previous period. Similarly, demand agents will offer a higher price if they didn’t meet their consumption needs in the previous period and offer less if they succeeded in meeting their needs [Praça *et al.*, 2003].

2.4.3.2 Adaptive Derivative Following

This strategy is based on a derivative-following strategy proposed by Amy Greenwald, Jeffrey Kephart, and Gerald Tesauro. Adapted derivative following considers the revenue the seller earned last period as a result of the price change made between that period and the one before it. If that price change led to more revenue per unit, the seller makes a similar price change. If that price change produced less revenue per unit, the seller makes a different price change [Praça *et al.*, 2003].

For both these strategies, the next period’s offer price will be the previous period’s price adjusted by an amount that will be more or less than the previous price, depending on the strategy used. The price adjustment is determined by the same calculation:

$$price_{i+1} = price_i \pm ammount_{i+1} \tag{2.4}$$

$$ammount_{i+1} = price_i * \left(\beta + \frac{\Delta_i}{capacity\_available_i * \alpha}\right) \quad (2.5)$$

$$\Delta = capacity\_available - energy\_sold_i \quad (2.6)$$

Instead of adjusting the price each day by a fixed percentage, the formula scales the change by a ratio to sell all the available capacity. The amount of change increases with the difference between the amount of energy the seller wanted to sell and the amount it actually sold. *β* and *α* are scaling factors [Praça *et al.*, 2003].

2.4.3.3 Market Price Following

As the name suggests, this strategy follows the market price of the same period of the previous day. It is a very simple strategy, but it presents good results when prices show a tendency to stabilize in a certain period, for some consecutive days [Pinto *et al*., 2011a], [Vale et al., 2011a].

2.4.3.4 Average 1 Agent

This is the agent that performs the first average of prices from the market historic database. It uses the data from the 30 days prior to the current simulation day, considering only the same period as the current case, of the same week day. This allows to have a strategy based on the tendencies per week day and per period.

2.4.3.5 Average 2 Agent

This agent performs an average of the market prices considering the data from one week prior to the current simulation day, considering only business days, and only the same period as the current case. This strategy is only performed when the simulation is at a business day. This approach, considering only the most recent days and ignoring the distant past, gives us a proposal that can very quickly adapt to the most recent changes in the market.

2.4.3.6 Average 3 Agent

This agent uses an average of the data from the four months prior to the current simulation day, considering only the same period as the current case. This offers an approach based on a longer term analysis. Even though this type of strategies, based on averages, may seem too simple, they present good results when forecasting the market prices, taking only a small amount of time for their execution.

2.4.3.7 Regression 1 Agent

This agent performs a regression on the data from the four months prior to the current simulation day, considering only the same period of the day, similarly to the method used by Average 3 Agent.

2.4.3.8 Regression 2 Agent

This agent performs a regression on the data of the last week, considering only business days. This strategy is only performed when the simulation is at a business day.

2.4.3.9 Dynamic Artificial Neural Network Agent

This agent uses a feed-forward neural network (NN) trained with the historic market prices, with an input layer of eight units, regarding the prices and powers of the same period of the previous day, and the same week days of the previous three weeks. The intermediate hidden layer has four units and the output has one unit – the predicted market price for the period in question. The neural network topology is presented in Figure 10.

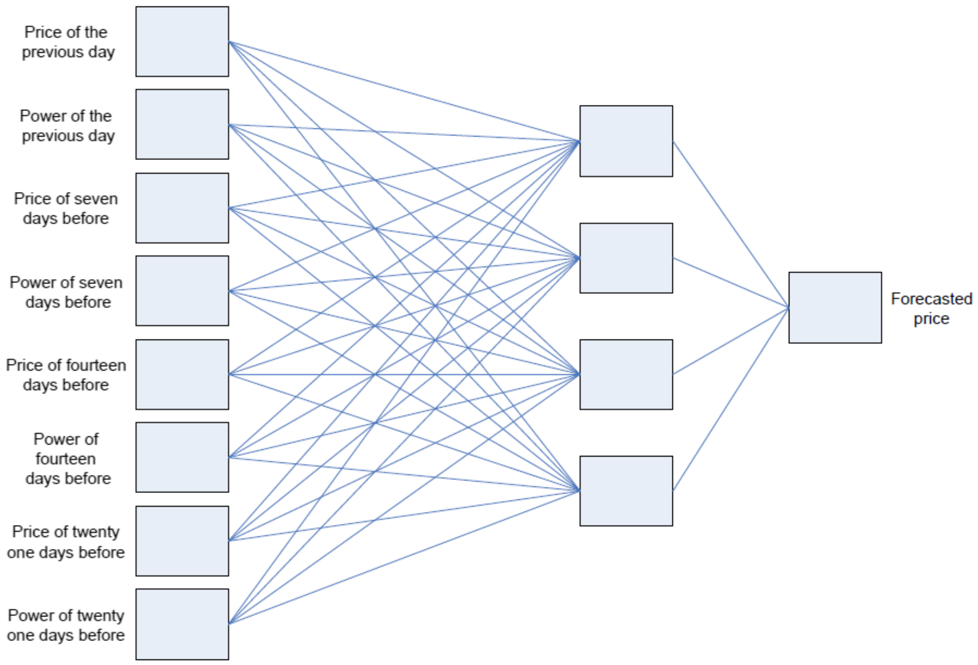


Figure 10 – Neural Network topology [Pinto *et al.*, 2011a]

The considered NN is characterized as a feedforward neural network, receiving as inputs the market prices and total amount of negotiated energy in the market, referring to: the day before to the desired forecasted day, one week before, two weeks before, and three weeks before. The NN considers four nodes in the intermediate layer, and one output – the forecasted market price. This topology was defined in [Vale et al., 2011b] and [Pinto *et al.*, 2011c]

The dynamism of the NN is achieved by a retraining of the network in each iteration, in order to always consider the most recent information, instead of the usual approach when managing NNs, which is training it once, and then use that trained NN to perform the forecast from that point forward. With the dynamism of the NN, the constant adaptation and adequacy of the forecast taking into account the most recent events is the main goal [Pinto T., 2011].

Experimentation on the training of this NN led to the conclusion that the training limit for the predictions should not be above 730 days (two years) [Pinto T., 2011], with limits above this one predictions tend to get worst, probably due to the inclusion of data with no longer up-to-date characteristics.

2.4.3.10 AMES Agent

This agent performs an adaptation of the AMES bidding strategy [Li and Tesfatsion, 2009], [Sun and Tesfatsion, 2007].

This strategy is based on a study of the efficiency and reliability of the Wholesale Power Market Platform (WPMP), a market design proposed by the U.S. Federal Energy Regulatory Commission for common adoption by all U.S. wholesale power markets. MASCEM is a simulator directed to markets with different characteristics, including asymmetrical and symmetrical pool markets (see 2.1.2), nevertheless the study of the optimal cost coefficients for WPMP can be used as a basis for the calculation of the production cost function for other type of markets' generators, and consequently define the minimum price that must be bided in order to cover such costs, and provide profits [Pinto *et al.*, 2011b], [Pinto T. 2011].

The strategy uses the Roth-Erev reinforcement learning algorithm to choose the best among a set of possible bids that are calculated based on the relation cost/profit that the player presents when producing electricity. The various possible bids differ from each other due to the distinct combination of the input parameters. The most combinations we set, the best chances there are of getting a good result. However, the number of combinations affects the processing time and the number of runs required for a satisfactory convergence.

For this reason, ALBidS implements AMES strategy using Simulated Annealing heuristic in order to determine their action-selection strategy and to balance exploration and exploitation, which can accelerate convergence to the optimum by avoiding the excessive exploration during the learning process [Pinto T., 2011].

2.4.3.11 Simulated Annealing Q-Learning Agent

This strategy uses the Simulated Annealing heuristic to accelerate the process of convergence of the Q-Learning algorithm in choosing the most appropriate from a set of different possible bids to be used by the market negotiating agent whose behaviour is being supported by ALBidS.

The Q-Learning algorithm [Rahimi-Kian *et al.*, 2005] is a popular reinforcement learning method. It is an algorithm that allows the autonomous establishment of an interactive action policy. It is demonstrated that the Q-learning algorithm converges to the optimal proceeding when the learning state-action pairs Q is represented in a table containing the full information of each pair-value [Juang and Lu, 2009].

In Q-learning, a Q-function is used as a prediction function that estimates the expected return (optimal evaluation) from the current state and action pair; this Q-function mapping is represented by (2.7):

$$Q : s \times a \rightarrow U \tag{2.7}$$

where U is the expected utility value when executing an action a in the state s. As long as the state and action states do not omit relevant information, nor introduce new information, once the optimal function Q is learned the agent will know precisely which action results on the higher future reward, in a particular situation s [Juang and Lu, 2009], [Pinto T., 2011].

The $Q(s,a)$ function, regarding the future expected reward when action a is chosen in the state s, is learned through try and error, following the equation (2.8):

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha[r_t + \gamma V_t(s_{t+1}) - Q_t(s_t, a_t)] \tag{2.8}$$

Where α is the learning rate; r is the reward or cost resulting from performing the action a in the state s; γ is the discount factor; and U (2.9) is the state s utility resulting from action a, obtained using the Q function learned so far.

$$U_t(S_{t+1}) = \max_{a} Q(s_{t+1}, a) \tag{2.9}$$

The Q-Learning algorithm is executed as follows [Pinto T., 2011]:

- For each s and a, initialize $Q(s,a) = 0$;
- Observe s;
- Repeat until the stopping criterion is satisfied:
  - Select action a, using the current action policy;
  - Execute action a;
  - Receive immediate reward $r(s,a)$;
  - Observe new state s';
  - Update $Q(s,a)$;
  - $s \leftarrow s'$.

As the visiting of all state-action pairs tends to infinite, the method guarantees a generation of an estimative of Q_t which converges to the value of Q. In fact, the actions policy converges to the optimal policy in a finite time, however slowly.

ALBidS implementation of Q-learning accelerates this process through the application of the Simulated Annealing heuristic.

2.4.3.12 Game Theory Agent

This agent uses a scenario analysis algorithm based on the application of the Game Theory.

Game theory deals with circumstances where a person's success is based upon the choices of others. This theory has been applied in many areas, namely in mathematics, economics, political science, psychology, and computer science.

An important issue in game theory is the concept of perfect information. A game is regarded to as being of perfect information if all players know the moves previously made by all other players. Thus, only sequential games can be games of perfect information, since in simultaneous games not every player knows the actions of the others. Most games studied in game theory are imperfect information games, such as poker and contract bridge. Although there are some interesting examples of perfect information games, including chess, go, and mancala. Perfect information is often confused with complete information, which is a similar concept. Complete information requires that every player know the strategies and payoffs available to the other players but not necessarily the actions taken.

In ALBidS the algorithm is based on the analysis of several bids under different scenarios. The analysis results are used to build a matrix which supports the application of a decision method to select the bid to propose. The agent uses the historical information about market behaviour and about other agents' characteristics and past actions. This algorithm's organization is presented in Figure 11.

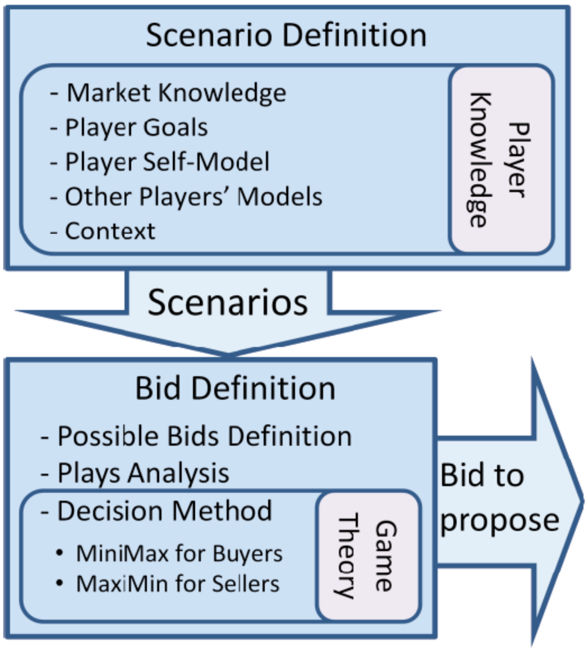


Figure 11 – Scenario Analysis Algorithm [Pinto *et al.*, 2011a]

To get warrantable data, the agent performs a statistical analysis of the historic data. With the gathered information, it builds a profile of other agents, including information about their expected proposed prices, limit prices, and capacities. With these profiles, and based on the agent own objectives, several scenarios, and the possible advantageous bids for each one, are defined.

As Seller and Buyer agents interact with each other in MASCEM environment, taking into account that their results are influenced by competitors' decisions, Game Theory is well suited for analysing these kinds of situations, assuming that each player seeks to:

- Maximize the minimum possible gain by using the MaxiMin decision method, when this method is being used by a Seller agent;

- Minimize the maximum possible loss, selecting the strategy with the smallest maximum payoff by using the MiniMax decision method, when this method is being used by a Buyer agent.

After each negotiation period, an agent may increase, decrease or maintain its bid, increasing the number of scenarios to analyse.

2.4.3.13 Error Theory Agents

Given that forecasts and predictions are always subject to some error, it is important to analyse that error properly, in order to try to overcome that. When analysing the forecasting errors' distribution over time, it becomes visible that many times those errors show patterns in their occurrence. In some cases, forecasts fail by predicting higher or lower values than the reality, in recurrent events that may have something in common (e.g. their context).

This strategy's goal is to analyse the forecasting errors' evolution of a certain forecasting method, to try finding patterns in that error sequence and provide a prediction on the next error, which will be used to adequate the initial forecast.

When a prediction is made, there will always be some error or uncertainty. For any measurement, there are a set of factors that may cause deviation from the actual (theoretical) value. Most of these factors have a negligible effect on the outcome and usually can be ignored. However, some effects can cause a significant change, or error, in the final result. In order to achieve a useful prediction, it is necessary to have an idea of the amount of the errors [Príncipe, 2010].

ALBidS implements this strategy with a dynamic neural network (NN) used to forecast the errors. Each iteration the NN is re-trained, so that the forecast values are always updated according to the most recent observations. The NN receives as input the prediction error history data of the market's prices, and is trained to generate an output value, which will be the expected error. Then, this error is used to adjust the value of a prediction made by other forecasting strategy. Errors are stored in a market's history, registered in a database.

When defining the architecture of the NN, a matter of high importance had to be analysed – the way the errors' sequence is "looked at" when trying to forecast it. This is a very important issue, for it does not matter how much data one has access to if that data is not properly used. For this reason, three different approaches are considered in ALBidS.

These three approaches use a NN with one value in the output layer - the value of the expected error, two intermediate nodes, and an input layer of four units. The input layer considers different values in the different approaches. These values depend on how the history of the error is considered:

- Error Theory A – this strategy makes a prediction along the 24 periods per day ,using for the training of each period the error of the same period for:
    - the previous day;
    - the previous 7 days;
    - the previous 14 days;
    - the previous 21 days.

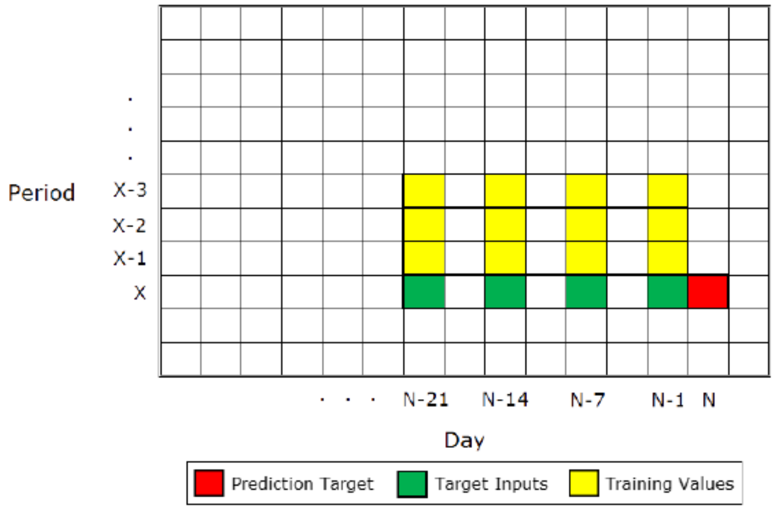


Figure 12 – Error Theory A training data structure [Pinto T., 2011]

Figure 12 helps understanding the concept: if an error prediction is required for day N and period X, the input values of the NN are N-1, N-7, N-14 and N-21, all for period X. The data of the previous periods is used to train the NN, considering the same days.

- Error Theory B - This strategy makes a prediction along the days, using the error of the following periods:
    - prior period;
    - 2 previous periods;
    - 3 previous periods;
    - 4 previous periods.
- Error Theory C - this strategy makes a prediction considering always the errors of the same period (the time period in question), using the error for:
    - the previous day;
    - the previous 7 days;
    - the previous 14 days;
    - the previous 21 days.

After experimentation it was concluded that the Error Theory C strategy, concerning the tested start day and time, has a better performance than the Error Theory strategy B, regarding the level of proximity of the forecast with the actual values of the market, but worse than Error Theory strategy A [Pinto T., 2011].

2.4.3.14 Economic Analysis Agent

The Economic Analysis Agent implements a strategy based on the two most commonly used approaches of forecasting in a company's scope. These approaches are the internal data analysis of the company, and the external, or sectorial, data analysis.

The most important and widely used strategic planning methods for a companies' business, base their analysis in the identification and grouping of the key pieces of information into two main categories: Internal and External data.

The Economic Analysis strategy's main principle is to decide when are the most appropriate moments to opt by a riskier or a safer approach in negotiating in the market. This decision is based on the connection between the internal and external data analysis of a company.

The internal analysis can be viewed as the company's economic development, i.e., the increasing or decreasing of the achieved profits. The profits take into consideration the company's fixed costs FC, such as the personnel expenses, the infrastructures' costs, the overheads, continuous maintenance, etc. Additionally, it also considers the variable costs, which are dependent on the production P, and are usually represented as two factors: a and b. The profits can be defined as in (2.10).

$$Profits = Total\ Income - (FC + P * a + P * b^2) \tag{2.10}$$

The analysis on the profits evolution is performed through a comparison between the most recent achieved profits, with the immediately previous ones. If the evolution is crescent, i.e., the recent profits are increasing, it is considered that the company is consolidating its position on the market, and therefore it is in a position where it can afford to take risks, in order to try obtaining higher profits. On the other hand, if the recent profits tendency is decreasing, the company must play safe, acting towards equilibrium, in a way to assure the continuous achievement of incomes, even if they are not as high as they could be.

When the decision goes for risking, and trying to achieve the higher possible profits, the Economic Analysis strategy uses as reference for the bid price, the forecasted market price, as it is the threshold of where a bid should be located, in order to obtain profits. The market price forecast used by this strategy is provided by the Dynamic Artificial Neural Network strategy, presented in sub-section 2.4.3.9.

When the decision is to play safe, ALBidS' Economic Analysis Agent uses K-Means clustering mechanism to group the companies acting in the electricity market, in different groups, according to their characteristics. Companies are grouped according to their similarity in dimension (amount of produced energy), most recent bid prices, and average price for the last month and year [Pinto T., 2011].

This is an application of External/Sectorial analysis as it groups the companies into three different clusters: one representing the most influential companies, one representing the most similar companies to the one ALBidS is supporting, and one representing the less influent companies over the market.

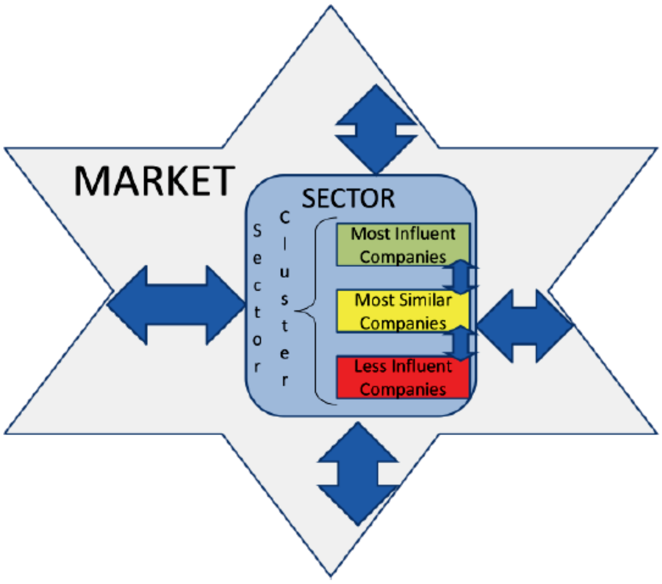


Figure 13 – External/Sectorial Analysis [Pinto T., 2011]

Once the clustering is finished, the average bid price of the companies grouped in the same cluster as the supported one, is determined as the sector reference price, as it is used as reference for the situations in which the decision was to act towards equilibrium. The only situation in which this may not apply is when the supported company is placed in a highly competitive cluster, with high risking prices. In this case it is used the lower from the cluster reference or market price forecast as reference value.

Concerning both reference values (sector reference price in case of equilibrium, or market price in case of risking), they are subject to a decrement before being defined as the bid price for the supported player. This decrement is based on a risk factor, chosen by the user, with the purpose of guaranteeing that the final bid price is located below the reference, regardless of which. The risk factor is a value between 0 and 1, and the higher the value is, the smaller is the decrement, meaning a higher proximity to the used reference values, and therefore increasing the risk to which it is subject. The initial decrement is calculated as in (2.11).

$$Decrement = 1 - Risk\ factor \tag{2.11}$$

The decrement will stay fixed if risks aren't to be taken. Otherwise, if risking for continuous number of periods, decrement lowers according to (2.12), slowly increasing the risk, until the bid price is equal to the reference value. If the sequence of risking periods is interrupted, the decrement returns to its initial value.

$$New\ Decrement = Decrement * (1 - Risk\ factor) \tag{2.12}$$

An adequate balance between the decision of taking higher risks, and acting safe, towards equilibrium, is the main goal of the Economic Analysis strategy. The decision making in what concerns the adequate times to risk is the essential concern of this strategy. For that, the internal vs. external data analysis gives its contribution [Pinto T., 2011].

Experimental studies have shown that following a day of bad results – not selling, or selling a low amount of power – the agent's bid price decreases, lowering the risk, and acting towards equilibrium. When the incomes are higher, the bid price for the following day is much closer to the market price, meaning a higher risk and the possibility to achieve higher profits.

2.4.3.15 Determinism Theory Agent

Determinism is the theory that all events, all happenings, are no more than consequences of a certain set of causes [Winnie J., 1996], and therefore all events are determined, because the event that was caused by other, is itself one of the causes for the determination of various other happenings [Einstein A., 1905]. This theory is based on the concept of cause and effect [Bunge M., 1959], which states that things do not happen by chance. All things result from the causes that impelled that happening to occur.

According to this theory, although all events are determined, and there is no chance of things happening any other way, it is impossible to predict a complex event based on this paradigm, because that event is caused by infinite other events, and, there does not exist the capability of considering, analysing, and combining infinite variables, to perform that prediction [Green C., 2003].

The main conclusion to take from the analysis of the Determinism Theory is that it is considered impossible to predict a future event, or effect, through the analysis of its causes, although such event is predetermined, because of the impossibility of considering all the causes, or variables, which affect the event, for they are infinite.

However, in a controlled environment, such as in simulation, which represents the global reality in a condensed and simplified environment, the consideration of all variables becomes much closer to being possible. Although a simulation environment and its results cannot fully reflect the reality; as they represent the essential aspects of a certain part of the reality, they can be (and are widely) used to provide important pointers and possibilities about how this reality shall react to the events tested in the simulation environment. Therefore, the application of such theory can prove both, to be possible in this environment, and also to produce results which can reflect the reality.

ALBidS approach to the implementation of this strategy started by identifying the variables that are required to be predicted: the market price for a certain period of a certain day, and a total amount of power traded by each market agent - these are the main variables as they influence directly the amount of income each agent will achieve, and the main goal of ALBidS' supported agent is the achievement of the higher possible profits. As causes for the effect that is going to be predicted, this strategy uses market negotiating agents' behaviours, namely their proposed bids - these are the factors that will determine the market price and amounts of traded power, through the application of the spot market symmetric algorithm (see section 2.1.2).

Because the player being supported has the equivalent influence to the future state of the world as other competing players do, ALBidS' approach is that after predicting (and henceforth considering them static) the other players' action, the only variation that can affect the future state is the supported player's action. As such, optimizing this action will grant the best possible results. Figure 14 presents a global overview over the Determinism Theory strategy approach.

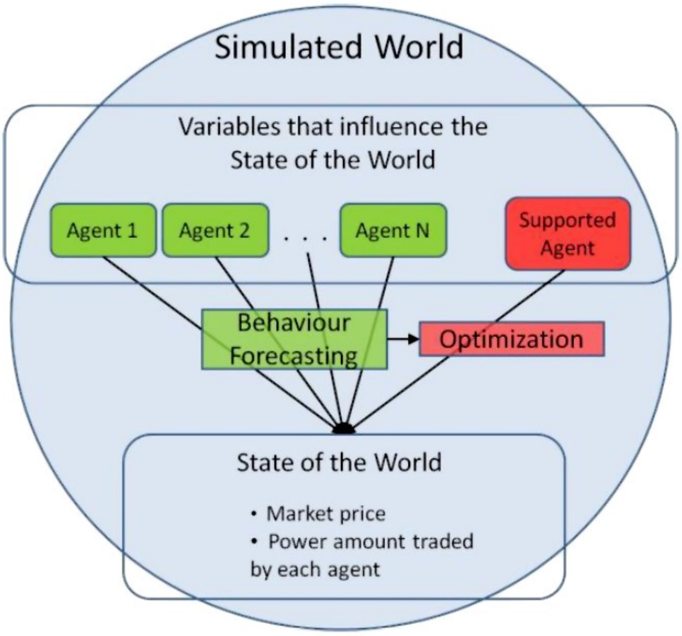


Figure 14 – Determinism theory approach [Pinto *et al.*, 2011a]

Forecasting of other players' actions is done using the Player Profile Definition Mechanism (see 2.4.1.3).

The input variables are the predicted prices and the bid power of the supported player; the decision variable is the supported agent's bid price, which is optimized according to the objective function: the market clearing algorithm, of the symmetric pool (see 2.1.2). This algorithm is what defines the market clearing price and the power amount traded by each agent.

The optimization's goal is to find the bid price that provides the maximum profit for the supported agent. The way the optimization is performed depends on the efficiency/effectiveness preference. So, for a high preference on the effectiveness of the method, the optimization is performed using an explicit enumeration of all the admissible solutions, i.e., the experimentation of all the possible bid prices for this agent. Clearly, the required execution time for processing such an algorithm is high – when the user requested for high effectiveness of the result rather than efficiency, this is not a problem, however when working dealing with tighter schedules for a response, heuristics are used. The implemented heuristics are: the *Simulated Annealing (SA)*, the *Tabu Search (TS)*, and the *Particle Swarm Optimization (PSO)*.

2.4.4 Metalearner Agents

ALBidS has two metalearner which are explored on the next sub-sections.

2.4.4.1 Metalearning Overview

Metalearning studies how learning systems can increase in efficiency through experience; the goal is to understand how learning itself can become flexible according to the domain or task under study [Vilalta and Drissi, 2002].

Metalearning is one-step higher than ordinary learning and means Learning-About-Learning. It is used to improve the overall classification (prediction) accuracy by exploring the multiple learning. Vaguely speaking, the meta-learning can be viewed as the learning from information generated by a set of (base) learners, or, using other words, as the learning of meta-knowledge on the learned information. [Bruha I., 2004]. The main goal is to use such meta-data to understand how automatic learning can become flexible in solving different kinds of learning problems, hence to improve the performance of existing learning algorithms.

Metalearning aims at discovering ways to dynamically search for the best learning strategy as the number of tasks increases [Vilalta and Drissi, 2002]. A computer program qualifies as a learning machine if its performance improves with experience. Experience is best understood as the knowledge gained from the analysis of several tasks; the definition is not limited to the ability to refine a hypothesis after presenting examples that belong to one task. Hence, meta-learning advocates the need for continuous adaptation of the learner at different levels of abstraction. If a base-learner fails to perform efficiently, one would expect the learning mechanism itself to adapt in case the same task is presented again. Thus, learning can take place not only at the example (i.e. base) level, but also at the across-task (i.e. meta) level [Vilalta and Drissi, 2002].

All learning systems work by adapting to a specific environment, which reduces to imposing a partial ordering or bias on the set of possible hypotheses explaining a concept. Meta-learning differs from base-learning in the scope of the level of adaptation: meta-learning studies how to choose the right bias dynamically, as opposed to base-learning where the bias is fixed *a priori*, or user parameterized [Vilalta and Drissi, 2002].

Flexibility is very important because each learning algorithm is based on a set of assumptions about the data, its bias. This means that it will only learn well if the bias matches the data in the learning problem. A learning algorithm may perform very well on one learning problem, but very badly on the next. From a non-expert point of view, this poses strong restrictions on the use of machine learning or data mining techniques, since the relationship between the learning problem and the effectiveness of different learning algorithms is not yet understood [Pinto T., 2011].

By using different kinds of meta-data, like properties of the learning problem, algorithm, or patterns previously derived from the data, it is possible to select, alter or combine different learning algorithms to effectively solve a given learning problem [Pinto T., 2011].

One usual view of meta-learning is stacked generalisation which works by combining a number of (different) learning algorithms. The meta-data is formed by the predictions of those different algorithms. Then another learning algorithm learns from this meta-data to predict which combinations of algorithms give generally good results. Given a new learning problem, the predictions of the selected set of algorithms are combined (e.g. by weighted voting) to provide the final prediction. Since each algorithm is deemed to work on a subset of problems, a combination is hoped to be more flexible and still able to make good predictions.

Originally proposed by Wolpert [Wolpert D., 1992], in stacked generalisation a set of q base-learners are applied to a training set T_{train}: $\{(\widetilde{X}_\iota, c_i)\}_{i-1}^{m}$ to produce q hypotheses, $\{h_j\}_{j=1}^{q}$, also called level-0 generalisers. Meta-learning takes place when training set T_{train} is redefined into a new set T'_{train}. The redefinition replaces each vector $\widetilde{X}_\iota$ with the class predicted by each of the q hypothesis on $\widetilde{X}_\iota$ [Vilalta and Drissi, 2002], according to (2.13).

$$T'_{train} = \{(\widetilde{X}_\iota, c_i)\} = \{((h_1(\widetilde{X}_\iota), h_2(\widetilde{X}_\iota), \ldots, h_q(\widetilde{X}_\iota)), c_i)\} \tag{2.13}$$

The new training set T'_{train} serves as input to a set of meta-learners, which produce a new set of hypotheses or level-1 generalisers. The redefinition of T_{train} into T'_{train} is done by k-fold cross validation [Vilalta and Drissi, 2002].

Stacked generalisation is considered a form of meta-learning because the transformation of the training set conveys information about the predictions of the base-learners. Stacked generalisation has a severe limitation in that both base-learners and meta-learners have a fixed form of bias (i.e. no dynamic selection of bias takes place) [Vilalta and Drissi, 2002].

Research in the stacked-generalization paradigm investigates what base-learners and meta-learners produce best empirical results. After transforming the original training set, each example contains the predictions of the base-learners, but it may also contain the original features. Results show how certain combinations of learners and meta-learners can yield significant improvements in accuracy [Vilalta and Drissi, 2002].

As of [Pinto T., 2011], ALBidS had two strategy agents with metalearning capability, which will be explored next.

2.4.4.2 Simple Metalearner

ALBidS implementation of a simple metalearner follows the works of Georg Zimmermann, where simple averages of the outputs of a NN are used to originate a final output, in order to overcome the uncertainty that affects the NN forecasts [Pinto T., 2011].

ALBidS' Simple Metalearner processing is a simple averaging between all the outputs of the strategies that are used in a certain situation. Note that the used strategies for this average depend both on the user initial preferences, and also on the requirements demanded by the Efficiency/Effectiveness Management mechanism (see section 2.4.1.4) [Pinto T., 2011].

2.4.4.3 Weighted Metalearner

ALBidS' Weighted Metalearner extends the Simple Metalearner by creating a tool that can, in fact, be called a metalearner using the concept of stacked generalisation. This metalearner's inputs are the outputs of the various approaches, adding the possibility of attributing importance weights to each of these inputs. These weights offer the chance for the metalearner to adapt its output, giving higher focus to the results of the strategies that are proving to be more adequate, while partially or completely ignoring the contribution of the strategies which are presenting worst results [Pinto T., 2011].

This procedure allows the Weighted Metalearner to adapt its output according to the observed results of each of its inputs. The weights used for defining the importance that each input has for the Metalearner, are based on the confidence values of the main reinforcement learning algorithm used by ALBidS (see sections 2.4.1.1 and 2.4.1.5). The reinforcement learning algorithm's confidence values are adapted and updated according to the results each strategy is presented, hence being exactly what this metalearner requires for understanding which strategies' outputs it ought to consider as most influent to the final output [Pinto T., 2011].

The generation of this output $\bar{x}_p$ is performed through a weighted average, using the reinforcement learning algorithm's confidence values as weights $p_1, p_2, p_3, \dots p_n$ for each strategy's output's $x_1, x_2, x_3, \dots, x_n$ as contribution to the final metalearner's solution. The procedure is expressed in (2.14).

$$\bar{x}_p = \frac{p_1 * x_1 + p_2 * x_2 + \cdots + p_n * x_n}{p_1 + p_2 + \cdots + p_n} = \frac{\sum_{i=1}^{n} p_i * x_i}{\sum_{i=1}^{n} p_i} \tag{2.14}$$

The adapted output of the Weighted Metalearner is expected to be able to generate better results than the Simple Metalearner, since it takes higher regard for the inputs that are expected to point the final result towards a better solution.

Experimentation in [Pinto T., 2011] was done using Bayes Theorem algorithm as main reinforcement algorithm for the confidence values for the Weighted Metalearner algorithm. Additionally, in these experiments, as supporting strategies, nine random strategies of the already presented, were selected. These strategies were the same for both simulations.

Comparing the Weighted Metalearner's performance with the Simple Metalearner, a much more constant behaviour is found. The adjustment of the final bid, taking into account the strategies' results has proven to be an added value in suiting the Metalearner's results.

3 Six Thinking Hats

The Six Thinking Hats is a parallel thinking method built to change the way meetings are run, and stakeholders work and interact [de Bono E., 1985]. This method proposes to, among other results, increase the speed at which decisions are made without hastening the process.

The method also “promises” to harness and take full advantage of the intelligence, information and experience of each and every party present in the meeting. By using this method, participants are also invited to discard, almost entirely, any conflict that could emerge in the meeting. Taking full advantage of stakeholder’s capabilities should also grant better decisions.

The purview of this thesis is neither to teach nor train readers in the usage of this method; however, a proper introduction of its main concepts should be given, this is given in this chapter.

3.1 The method and its advantages

The Six Thinking Hats method offers itself as the “exact opposite of argument, adversarial, confrontational thinking”. Using this method every participant is asked to take one direction of thinking at a time, always avoiding confrontation, or non-constructive criticism, or argumentation where one party tries to prove the other wrong is undesirable; this allows full exploration of the subject being discussed [de Bono E., 1985].

Should it be come to choosing between two opposite goals or options, this step must be taken down the road – one of the key concepts is that a meeting is a way forward while fully exploring the subject [de Bono E., 1985].

For each direction of thinking, this method associates a hat with a distinctive colour. On the following chapters, a short summary of the role of each hat is presented. This method also assumes the existence of a moderator, responsible of picking the direction which to take next and enforcing rules

3.1.1 White Hat

White is neutral and objective. The white hat is concerned with objective facts and figures [de Bono E., 1985]. Using this hat, the participants are asked to supply any information on the subject they have or are aware of; no judging or opinions are allowed. Each detail, each piece of information given in this step is like the piece of a puzzle, or a map, each one enriching and completing our map. Once the map is finished, the route becomes more or less obvious to everyone.

It is the moderator's job to not allow any ready-made idea, feeling, opinion, or anything that is not neural information, to creep in the discussion. In fact, in practice of this hat, the moderator should establish a two-tier system: *believed facts* and *checked facts*, i.e. it is important to be able to classify all facts given during this stage of the method.

1. Checked fact – all facts that are measurable, confirmed;

2. Beliefs or hearsays – some information put out in good faith or personal belief.

This method acknowledges that it is not possible, or desirable to have each and every fact scrutinized with the rigor of scientific experimentation, however as stated before, opinions and feelings must be trimmed out of this map we are now building.

3.1.2 Red Hat

In a normal business discussion you are not supposed to allow your emotions to enter in. They enter anyway – you merely disguise them as logic. The red hat provides a unique and special opportunity for feelings, emotions and intuition to be put forward as such. [de Bono E., 1985]

This step of the method gives the thinker a channel to express any simple feeling (fear, like/dislike, suspicion, mistrust), or the more complex like hunch and intuition on the subject under debate, or even at the conduct of the meeting itself. No justification or details about the feeling are required; emotions do not have to be logical or consistent.

The red hat makes feelings visible so that they can become part of the thinking map and also part of the value system that chooses the route on the map.

3.1.3 Black Hat

The most used and perhaps the most important of all hats, the black hat is the hat of caution. The black hat is all about carefulness, awareness and survival. The black hat points out how something does not fit our resources, our policy, our strategy, our ethics, our values, and so forth [de Bono E., 1985].

Black hat thinking brings experience into the playing board, under this hat we look for patterns that do not match our own experience.

Although caution is a good thing, preventing us from being wiped out of existence, there are some people o tend to overuse this black always probing for the error, for the fault, especially on someone else's ideas or thoughts (which is typical with the Western argumentative habits). Constant destructive criticism is very bad for the discussion, reason why time framing of this step is essential.

A warning must be issued, though: black hat thinking is not a permit to go back to argumentative discussion. Procedural errors can be pointed out; parallel statements that express a different point of view can be laid down. In the end there should be a clear map of possible problems or obstacles, these ones need to be clarified and elaborated.

3.1.4 Yellow Hat

The exact opposite of the black, the yellow hat thinking has the role of bringing positive thoughts to our thinking map. At this stage of the discussion, one is expected to bring upfront only constructive and positive thoughts.

The nature of optimistic thinking allows it to cover a very broad spectrum of ideas, ranging from the logical and practical at one end, to dreams, visions and hopes at the other. This also includes foolish thoughts, thoughts that are too impractical or truly over-optimistic, therefore an effort should be made to stand somewhere on the spectrum, keeping away from the edges – while the edge where all is sound and safe will hardly grant any progress, over-optimistic ideas may be hazardous to our decision.

A thinker wearing this hat is expected to make an active effort of seeking the value of the ideas in our map; however, it is very important that once this value is found it is logically based.

3.1.5 Green Hat

The green hat is the energy hat. Think of vegetation. Think of growth. Think of new leaves and branches. The green hat is the creative hat [de Bono E., 1985].

Under the green hat we put forward new ideas. Under the green hat we lay out options and alternatives. These include both the obvious alternatives and fresh ones. Under the green hat we seek to modify and improve suggested ideas. Under the green hat you are permitted to put forward "possibilities". Possibilities play a much bigger role in thinking than most people believe. Without possibilities you cannot make progress. Two thousand years ago, Chinese technology was way ahead of Western technology. Then progress seemed to come to an end. The explanation often given is that the Chinese did not develop the hypothesis. Without this key of mental software it was impossible to make progress [de Bono E., 1985].

Creativity involves provocation, exploration and risk taking. Creativity involves "thought experiments" [de Bono E., 1985].

In Six Thinking Hats [de Bono E., 1985], there is mention to a term not introduced in this book: "lateral thinking". This mention is related to the fact that *creative* can, according to de Bono, be vague – creativity seems to cover everything from creating confusion to creating a symphony. Lateral thinking is very precisely concerned with changing concepts and perceptions [de Bono E., 1985].

Lateral thinking, which will not be explored in full detail here, is about cutting across pattern instead of just following along them. When cutting across to a new pattern is seen to make sense, we have the “eureka” effect [de Bono E., 1985]. “Normal” thinking uses *judgment* – “how this idea compares to what I know?” – this relates very closely to black hat thinking. What is asked in green hat thinking is that we use an idiom de Bono coined: *movement*. Movement stands for using an idea for its forward effect, we use an idea to see where it will lead us to.

Edward de Bono also emphasizes the need for provocation, the unplanned mistake or accident that sets off a new idea. Lateral thinking is an active process, and so there is the need to intentionally set off these provocations – in other words, the green hat thinker is asked to put some “crazy” ideas into discussion. Crazy, however, is not a synonym to absurd or illogical, we are asking for some non-ordinary ideas or associations. De Bono even suggests the usage of a random word at meetings, drawn right out of a dictionary to try and sprout for some new paths of thinking, cutting across patterns.

3.1.6 Blue Hat

The blue hat is for thinking about thinking, as a maestro is about getting the best out of the orchestra by seeing that what should be done is done at the right time. The blue hat is for process control.

It is under the initial blue hat that the agenda or sequence of use of the other hats is laid out. The blue hat sets the thinking “strategy”. During the session the blue hat keeps the discipline and ensures that people keep to the relevant hat. The blue hat also announces a change of hats.

Typically the blue hat is worn by the facilitator, chairperson or leader of the session. This is a permanent role. Any participant can be asked to, or voluntarily use the blue hat to, for example, examine if the building of our “thinking map” is going the right way.

At the end of a session the blue hat asks for the outcome. This may be in the form of a summary, a conclusion, a decision, a solution and so on. The blue hat may even acknowledge that little progress has been made. Under the final blue hat, the next steps can be laid out. These might be action steps, or further thinking on some points.

3.1.7 Summary

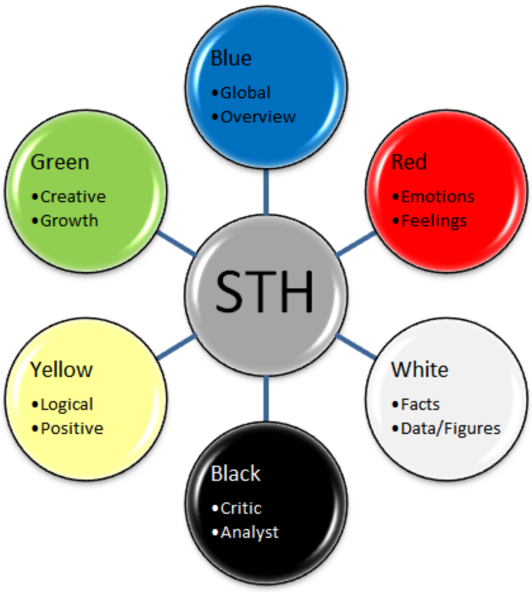


Figure 15 – STH overview

Figure 15 illustrates a summary of STH methodology, it is visible that the Blue Hat is not in the centre of the figure, although it plays a crucial role in the meeting, because any participant besides the moderator can intervene in a blue-hat way.

The method of the six hats to think was designed to remove thought from habitual argumentative style and to take it to a cartographic style. This is done by process of two stages: first it is to develop the map; second it is to choose the route. If the map is sufficiently good, a better route usually is obvious.

The greater value of the hats is its own artificiality. It offers a formality and one convention to, as much as possible, require a certain type of thought from every participant. STH lays down the rules of the game of the thought. Whoever plays it is going to know these rules – and people are generally good at following rules and playing games.

3.2 ALBidS Methodology based on STH

This work's focus is creating a decision support tool that acts based on STH method. For this development, it was decided to use ALBidS architecture while adapting agent strategies to each different "hat thinking way". This means that the scope of this support will be solely for bidding in the Spot Market.

It was also necessary to relate MASCEM and ALBidS' internal working to that of a common STH method-driven meeting, mainly these characteristics:

- A moderator entity, leading different interveners in the decision – the blue hat or the moderator sets the agenda for the meeting, determines its closing and final decision.

- Different ways of thinking about the problem to solve
- No arguing or debating, no confrontation

Figure 16 depicts the proposed architecture for this system. The approach that was taken was to build STH as an ALBidS' strategy agent. Enacting the STH's Blue Hat role as moderator, this agent will be responsible for controlling the process and ruling the final decision that will be communicated to ALBidS.

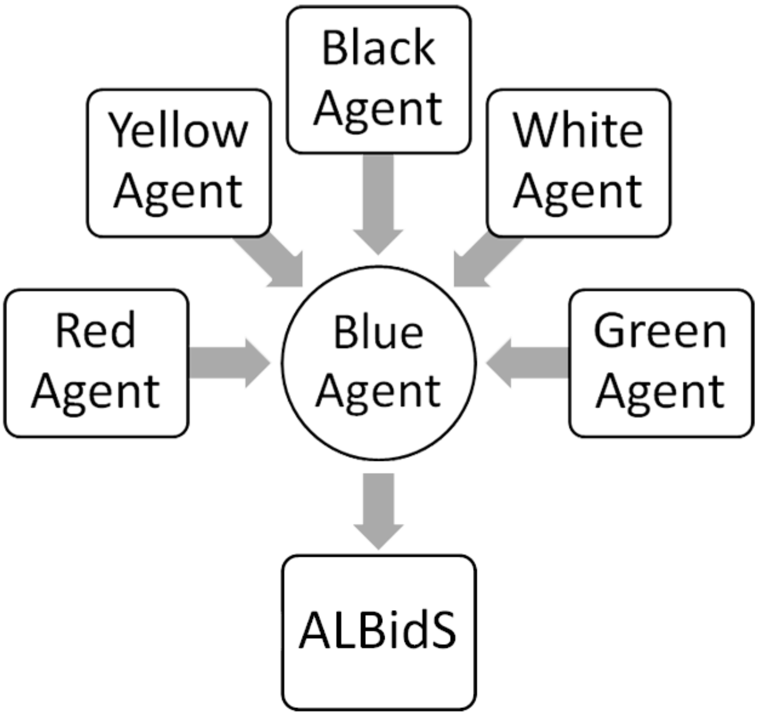


Figure 16 – STH Architecture

As denoted before, the STH method proposes to end or mitigate any conflict or confrontation that is natural to occur in a meeting (at least in Western Civilizations [De Bono E., 1985]); however unlike a human meeting, in this system it won't be necessary to avoid conflict or arguing because there is no communication or negotiation between.

Unlike STH where each intervenient is asked to wear every hat, in our approach, each intervenient will only act out a role and this will be played by an ALBidS' strategy agent. For this purpose we mapped a relation between each Hat's behaviour and the existing ALBidS' strategies, agents or mechanisms.

Also, in order to take advantage of what was already done, we chose not to try to assemble a exactly STH-like "thinking map" to where all interveners would contribute with their part (feasible with a blackboard approach, for example); [De Bono E., 1985] states that after our "thinking map" is done, the solution will be pretty obvious to every participant in the meeting.

Because we are dealing with multiagent systems and concepts such as "pretty obvious" and "emotions" are unknown (or hard to develop) to these artificial entities, we had to make some compromises and decisions on what would represent these concepts. The reasons that led to the Hat-Agent relation were:

- White Agent – this hat is responsible for bringing data, figures, and numbers into discussion, as such little or no treatment should be done in the data; it was clear that we could use an Average approach for this role.

- Red Agent – we decided to examine the emotions as something related to the recent past. Though not a true emotional decision or fulfilling the role De Bono planned for this hat, it is reasonable to assume a human intervenient could say something like "Due to our recent past, I've got a hunch that we should not raise prices"; a regression approach was chosen for this role.

- Black Agent – this role is for caution, to avoid danger. It is quite fair to think of it as knowing our position in the market, and knowing what we can or can't do; this is why we chose to fill this role with an Economic Analysis with little or no propensity for risk.

- Yellow Agent – Optimism without foolishness, this is the best way to describe the yellow hat thinker; also as the opposite of the black view, we saw this agent as a logical analyser with some appetite for risk – hence, once again the Economic Analysis this time with medium to high risk.

- Green Agent – the evolutionary traits of Particle Swarm Optimization used in Determinism Theory Agent made this strategy a suitable candidate for the creative thinking part. We are looking for logical and adjusted decisions while somehow different from the usual.

- Blue Agent – the one responsible for the development of the process and the final decision. This is the STH agent, the one that will gather every other agent decisions, summarize, and deliver a final answer to ALBidS. This agent's capabilities and structure is fully analysed in the section 3.3.

Table 1 shows the final version of this relation.

Table 1 – Relation between STH's roles and existing ALBidS' entities

STH's Role	ALBidS' Existing Agent
White	Average 1 Agent
Red	Regression 2 Agent
Black	Economic Analysis Agent – with low risk
Yellow	Economic Analysis Agent – with medium-high risk
Green	Determinism Theory Agent – using PSO
Blue	Not Available

Each of these Strategy Agents will be, henceforth and on the context of STH, referred to by the colour of its respective hat – in other words Red Hat Agent will refer to Average 2 Agent – or collectively as STH Agents.

Another compromise, that was already referred, was the rejection of using a collaborative blackboard system to build our “thinking map”. Though this would be the natural approach, it would have forced us to develop an independent MAS, and with it the development of another facilitator beyond those already present in MASCEM (OAA) and the Prolog Facilitator of ALBidS.

3.3 STH Agent – A new strategy/metalearner agent

In STH method, all decisions pass, ultimately, through the moderator. For this reason our Blue Agent will be all similar to the already existent ALBidS’ Metalearners, with different inputs.

Our goal for this work is to provide a decision that would reflect the combined efforts of several different views working together. We rejected the collaborative blackboard approach because, since there is no need for confrontation or arguing, the same effect can be achieved by having White, Black, Red, Yellow, and Green agents delivering their chosen bids to the Blue agent.

Unlike all other agents in ALBidS, Blue Agent’s answer is a set of ordered bids corresponding to each of the 24[1] periods of the day; this choice is related to our development decisions, which will be explained next.

In our first tests for a single period, the STH Agent’s answers varied between 3.68 and 8.65; for such precision we would have $(8.65 - 3.68) * 100 + 1 = 498$ possible prices, which would mean that cardinality of our solution space would be:

$$Pr_n^d = n^d = 498^{24} \tag{3.1}$$

The result, in decimal notation, is a 65 digits number – finding the optimal solution in such a large space would be computationally impractical, we had to turn on to heuristics; our choice was to use Genetic Algorithm heuristic. It was the choice for a set of ordered bids as answers for all STH agents that enabled us to use this heuristic.

3.3.1 Genetic Algorithm Heuristic

Genetic algorithms represent a class of algorithms based on a simplified computational model of the biological evolution process. They represent a class of general purpose adaptive search techniques that have the properties of parallel search and an enhanced ability to avoid local optima. [Lee I. *et al.*, 1997].

In 1859, Darwin’s Theory of Evolution by Natural Selection [Darwin C., 1859][2], introduced the term Natural Selection, analogous to selective breeding, a process by which animals and plants with traits considered desirable by human breeders are systematically favoured for reproduction. The concept of natural selection was originally developed in the absence of a valid theory of heredity; at the time of Darwin's writing, nothing was known of modern

[1] MASCEM’s database has only data regarding 24 periods for each day
[2] Independently developed by Alfred Russel Wallace (1958)

genetics[3,4]. The union of traditional Darwinian evolution with subsequent discoveries in classical and molecular genetics is termed the modern evolutionary synthesis. Natural selection remains the primary explanation for adaptive evolution.

The central idea is simple: variations occur in reproduction and will be preserved in successive generations approximately in proportion to their effect on reproductive fitness – the survival of the fittest.

This concept is applied in genetic algorithms, where each solution or individual in the population is described by a string of variables, and the search performs operations on the population of solutions. Individuals' potential to reproduce and pass their genes on to the next generation is determined by their fitness functions, which are evaluated with respect to the objective function of the problem at hand.

The purpose of the use of a genetic algorithm is to find the individual from the search space with the best "genetic material". The quality of an individual is measured with an evaluation function. The part of the search space to be examined is called the population [Larrañaga *et al.*, 1996].

Once the initial population is generated (randomly) or arbitrarily chosen, three common operations are used to generate offspring or children to form the next generation [Lee C. *et al.*, 1997]:

1. Selection – this is the stage of a genetic algorithm in which individual genomes are chosen from a population for later breeding.

A generic selection procedure may be implemented as follows:

   a. The fitness function is evaluated for each individual, providing fitness values, which are then normalized. Normalization means dividing the fitness value of each individual by the sum of all fitness values, so that the sum of all resulting fitness values equals 1;

   b. The population is sorted by descending fitness values;

   c. Accumulated normalized fitness values are computed (the accumulated fitness value of an individual is the sum of its own fitness value plus the fitness values of all the previous individuals). The accumulated fitness of the last individual should be 1 (otherwise something went wrong in the normalization step);

   d. A random number R between 0 and 1 is chosen;

   e. The selected individual is the first one whose accumulated normalized value is greater than R.

[3] Only seven years later, in 1866, would modern genetics took its first steps, with the publication of Gregor Mendel's Experiments on Plant Hybridization [Mendel G., 1866]

[4] And only near a century later, would Watson and Crick (1953) identify the structure of the DNA molecule and its alphabet

2. Crossover – an exchange of a portion of each individual's (involved in the reproduction) genetic material;

Crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next. It is analogous to reproduction and biological crossover, upon which genetic algorithms are based. Crossover is a process of taking more than one parent solutions and producing a child solution from them. There are several methods for selection of the chromosomes which will be not explored here.

There are many crossover techniques, the most simple of them being:

a. One-point crossover:

A single crossover point on both parents' organism strings is selected. All data beyond that point in either organism string is swapped between the two parent organisms. Figure 17 illustrates this operation.

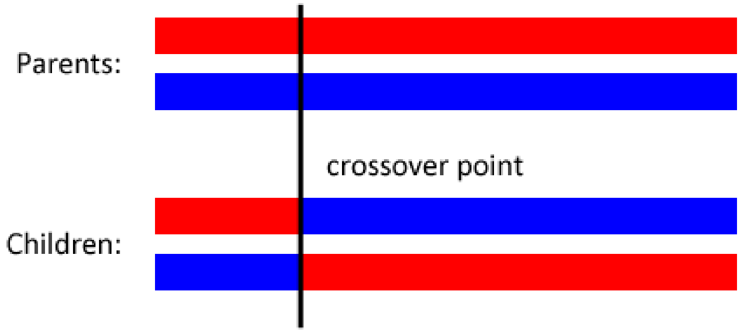


Figure 17 – GA's One Point Crossover

b. Two-point crossover:

Two points are selected on the parent organism strings. Everything between the two points is swapped between the parent organisms, rendering two child organisms depicted in Figure 18.

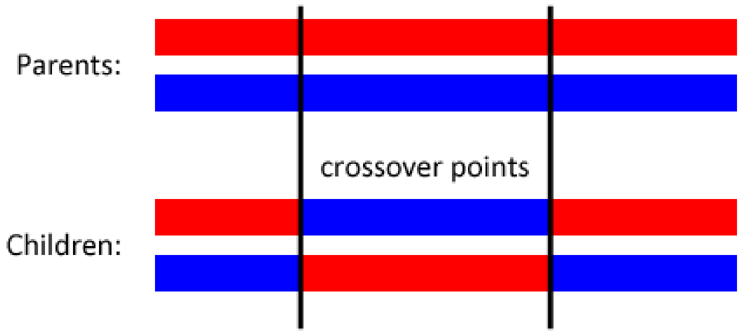


Figure 18 – GA's Two Point Crossover

Other techniques are available but they are not relevant for this work.

3. Mutation, a random modification of each individual

Mutation is used to maintain genetic diversity from one generation of a population of algorithm chromosomes to the next. It is analogous to biological mutation that may occur naturally during meiosis [Cederberg and Rannug, 2006] or DNA replication [Bertram J., 2000]. Mutation alters one or more gene values in a chromosome from its initial state. In mutation, the solution may change entirely from the previous solution. Hence GA can come to better solution by using mutation. Mutation occurs during evolution according to a user-definable mutation probability. This probability is normally set low. If it is set too high, the search will turn into a primitive random search.

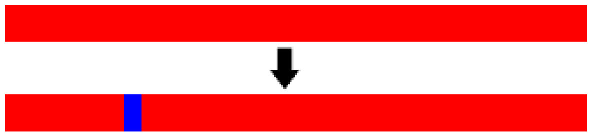


Figure 19 – GA's Mutation

After a period of time, the most suitable individuals will dominate the population, providing an optimal (or near-optimal) solution.

3.3.2 GA in STH for electricity markets

Generally, the first step of GA procedures is to initialize the population either randomly or by using seeds. In this work, our population will not be random but a result of executing the other strategies.

The individuals composing the population for our study are the ordered sets of bid values that each agent submitted to the Blue Agent, in other words every individual is the set of 24 bids for each period of the following day. Figure 20 illustrates a sample individual and the detail of its chromosomes; the crossover points are also depicted.

Period	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
W.A. Bids	1.67	2.93	4.94	1.38	1.59	2.8	4.03	4.1	4.57	3.85	3.74	1.16	0.98	3.99	3.07	4.01	2.45	4.33	1.29	2.29	2.15	2.9	1.71	3.22

Crossover point 1 Crossover point 2

Figure 20 – White Agent Bids sample

In order to introduce some variation to our genetic pool we decided to add the following individuals:

- GAI_{minima} – an individual with the minima of bids for each period of STH agents' results;
- GAI_{maxima} – an individual with the maxima of bids for each period of STH agents' results;
- $GAI_{average}$ – an individual with the averages of STH agents' results;
- GAI_{D1} and GAI_{D2} – two individuals with R random values in the interval expressed by (3.2)

$$\begin{cases} \vartheta = |S_{max} - S_{min}| * p, S_{max} \geq S_{min}, 0 \leq p < 1 \\ R \in [S_{min} - \vartheta, S_{max} + \vartheta] \end{cases} \tag{3.2}$$

where S_{min} and S_{max} are, respectively, the minimum and the maximum of all values provided by STH agents, and p is user defined;

- GAI_{R1} and GAI_{R2} – two individuals with random values between 0 and 10 – we chose the top value after analysis of historic data in the database – bids don't usually go above 10 cents of Euro; it also doesn't make sense to bid a negative value[5].

Although it was possible tackle this problem by dividing the population into several smaller populations, applying different GAs in each [Huang *et al.*, 2010] and then breeding across populations; we chose to use a single population in this study.

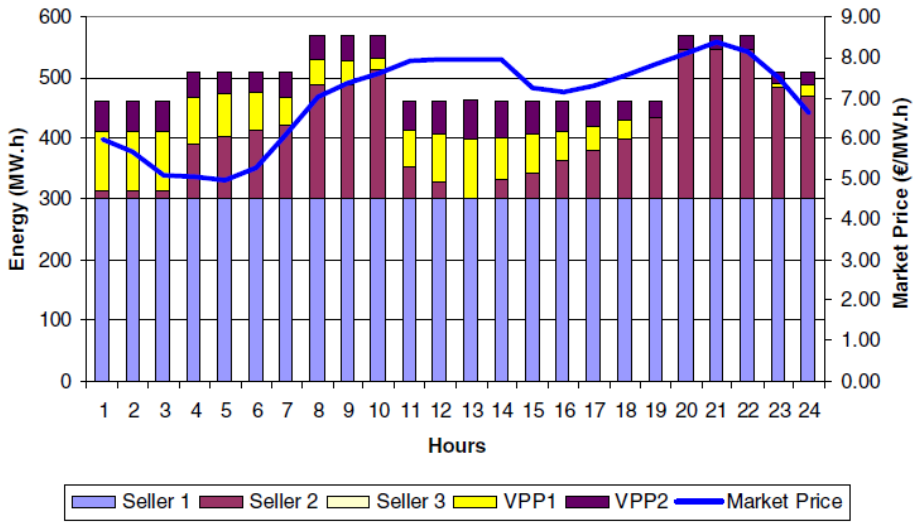


Figure 21 – Energy market transaction by seller [Pinto *et al.*, 2011a].

Figure 21 illustrates market transaction by seller, looking only at the total energy sold it is clear that there is some separation between some periods of the day, therefore we decided to use two crossover points, one for the sunrise and the other for the sunset, thus dividing the day and night periods. The reasons behind this decision are quite clear – electricity consumption changes during night time. ALBidS' context analysis mechanism (see section 2.4.1.2) provides support for this statement as one of the analysis on [Pinto T., 2011] shows a "clear separation between periods from ten to twenty three from the rest". The values 10 and 23 were chosen as our standard crossover values instead of the sunrise and sunset for the simulation days.

For our fitness function, we run a lesser featured simulation to predict the bidding values of all other competing agents; this way we have an approximation to what the market price will be,

[5] Although it may be possible in the future – a producer may have to pay consumers to "buy" their electricity because otherwise it would be much more expensive to shut down the energy generators.

providing us with an quite accurate evaluation of the potential of our solution. Because this simulation is implemented in Prolog – for performance reasons – we were unable (without further development) to use MATLAB [MATLAB, 2012] GA solving, hence we chose to develop this algorithm in Prolog.

3.3.3 Integration with ALBidS

STH agent was built to be seen by ALBidS like any other strategy agent, using the already existing structure. During operation, STH agent creates instances of the other strategy agents using a factory method pattern, and then invokes the appropriate methods to get each agent's values.

All ALBidS Strategy agents operate at the period level – agents are prepared to receive as input a day and a period to give their bid answer, this means that to bid on the spot market, ALBidS must ask for an agent's answer 24 times, one for each period that must be then organized and sent to the market operator.

The STH method as it is implemented operates at the day level, meaning that it always answers with a set of bids for all periods of the following day.

4 Case Studying

4.1 Introduction

The case studies shown in this section are the result of tests trying to determine if there are any advantages in using this new strategy to complement the existent ones.

Given that this strategy uses several others, each one with its own parameterization, these studies will use always the same values for each of them. These default values are not incidentally the same used in [Pinto T., 2011], these are presented next.

4.1.1 Test Scenario

For proper comparison purposes the test scenario will be identical to the ones used in [Pinto T., 2011]. The test scenario involves 7 buyers and 5 sellers (3 regular sellers and 2 VPPs). This group of agents has been created with the intention of representing the Spanish reality, reduced to a smaller group, containing the essential aspects of different parts of the market, allowing a better individual analysis and study of the interactions and potentiality of each of those actors [Vale et al., 2011a]. Figure 22 presents the test scenario structure.

The simulations consider different biddings for each agent. Seller 2, which is used as the test reference, will use the STH strategy.

The other players' bids are defined as follows:

- Buyer 1 – This buyer buys power independently of the market price. The offer price is 18.30 c€/kWh (this value is much higher than average market price);
- Buyer 2 – This buyer bid price varies between two fixed prices, depending on the periods when it really needs to buy, and the ones in which the need is lower. The two variations are 10.00 and 8.00 c€/kWh;
- Buyer 3 – This buyer bid price is fixed at 4.90 c€/kWh;

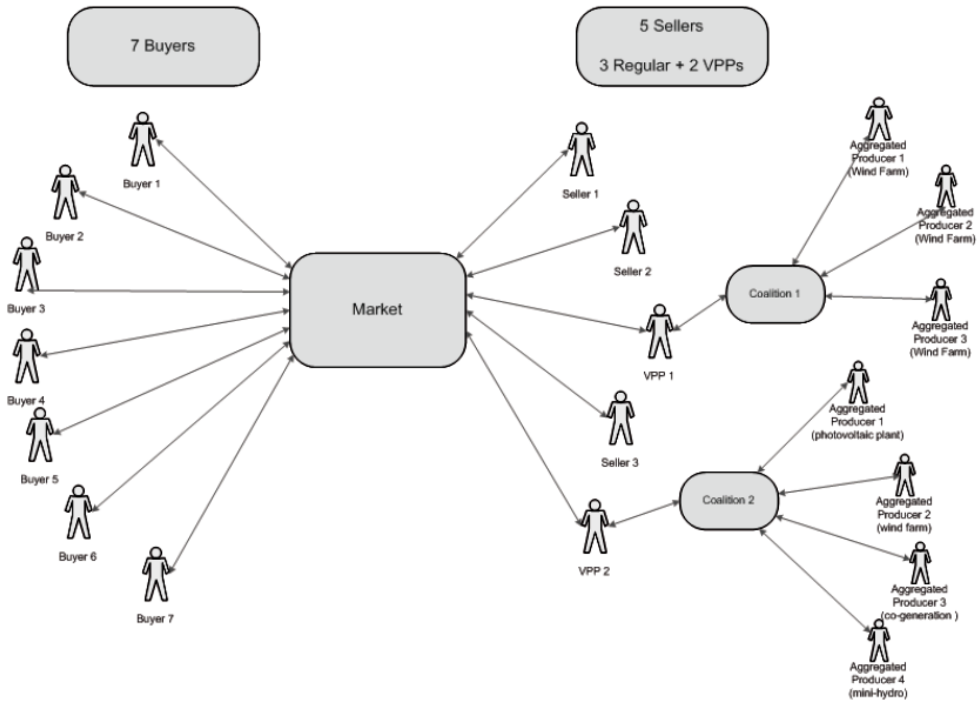


Figure 22 – Test Scenario

- Buyer 4 – This buyer bid considers the average prices of the last four Wednesdays;
- Buyer 5 – This buyer bid considers the average prices of the last four months;
- Buyer 6 – This buyer bid considers the average prices of the last week (considering only business days);
- Buyer 7 – This buyer only buys power if market prices are lower than the usually verified market price (around 4.0 to 8.0 c€/kWh), by bidding a much lower value: 2.0 or 3.0 c€/kWh, depending on whether the current negotiation period is at a peak time of the day;
- Seller 1 – This seller needs to sell all the power that he produces. The offer price is 0.00 c€/kWh;
- Seller 3 – This seller bid considers the average prices of the last four months with an increment of 0.5 c€/kWh;
- VPP 1 – Includes four wind farms and offers a fixed value along the day. The offer price is 3.50 c€/kWh;
- VPP 2 – Includes one photovoltaic, one co-generation and one mini-hydro plants; the offer price is based on the costs of co-generation and the amount to sell is based on the total forecasted production.

This test scenario is used in all of the simulations in this chapter.

4.1.2 Strategy Parameterization

All the strategies in ALBidS receive as parameters the current date and the period for the simulation.

- The Average 1 (White) and Regression 2 (Red) strategies have no parameters that affect their answers

- Economic Analysis (Black with low risk, Yellow with medium-high risk) strategy's parameter is the risk factor that the agent should take in the decision. We have set these values respectively 0.2 for Black and 0.8 for Yellow. For both of these agents, the Own agent ID is that of the Seller 2.

- Determinism Theory Agent using PSO (Green), we set the Efficiency/Effectiveness to 50 so that it would use a heuristic (100% Effectiveness would make it use Explicit Enumeration), and an hourly production of 50MW/h.

4.2 Case Study 1 – STH performance in the market

In this case study we evaluate STH's performance in the market. Six simulation results are presented, each for September the 1st, 2008. Table 2 presents the different parameters used for GAs sensibility test.

Table 2 – GAs sensibility test parameterizations

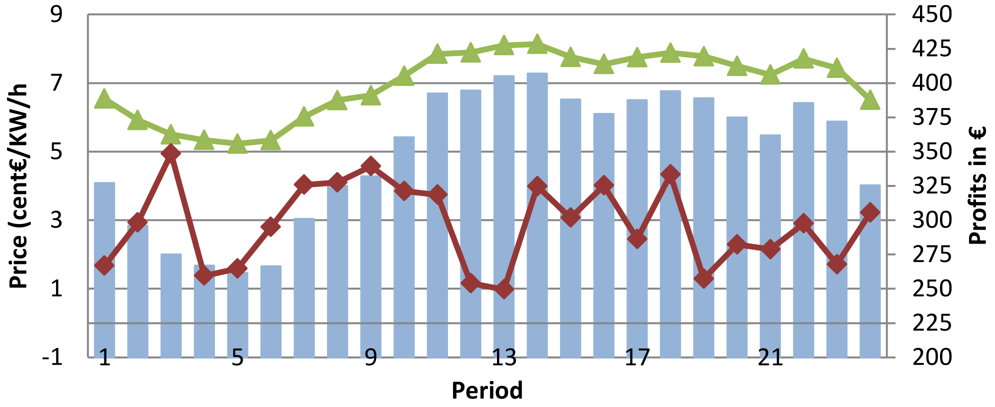
#	Strategies Weights	Number of Generations	Deviation	Crossover Points	Mutation Prob.
1	Same	100	0.10	10 and 23	0.025
2	Given RLA	100	0.10	10 and 23	0.025
3	Same	200	0.10	10 and 23	0.025
4	Same	100	0.20	10 and 23	0.025
5	Same	100	0.10	8 and 21	0.025
6	Same	100	0.10	10 and 23	0.5

The reason the default crossover points are 10 and 23 was explained earlier, it is based on analysis made by K-Means clustering in [Pinto T., 2011]. The values 8 and 21 are based on sunrise and sunset information for Madrid at September, the 1st, 2008 taken from TimeAndDate.com [timeandate.com, 2008]. The graphs in

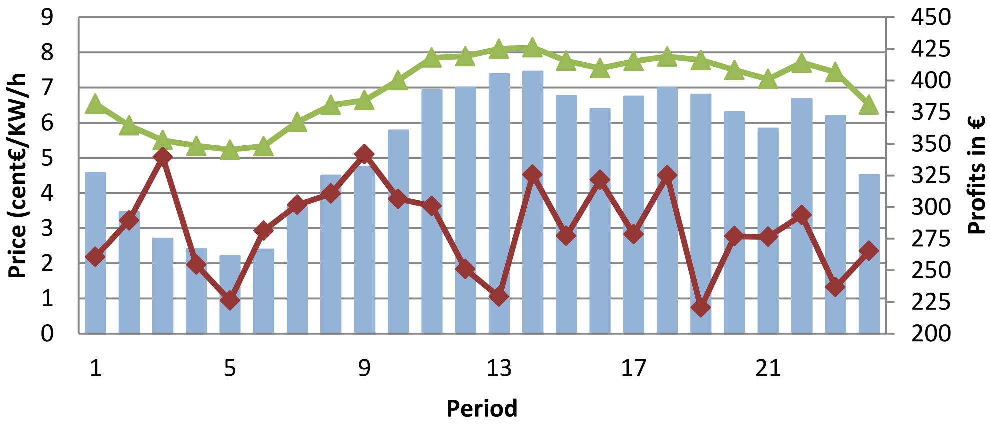
Figure 23 show each of these different parameterizations.

For the default number of generations we opt for a number high enough so that would be chance of mutation actually happening. We found 10% to be a reasonable value for the deviation – we didn't want the values to be outside the boundaries 0 and 10 we set earlier. The charts in Figure 21 show the sensibility test results.

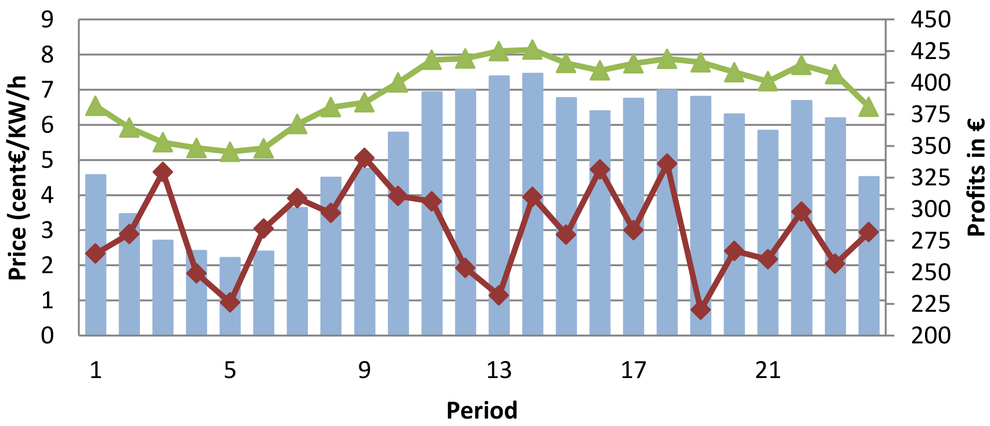
a)



b)



c)



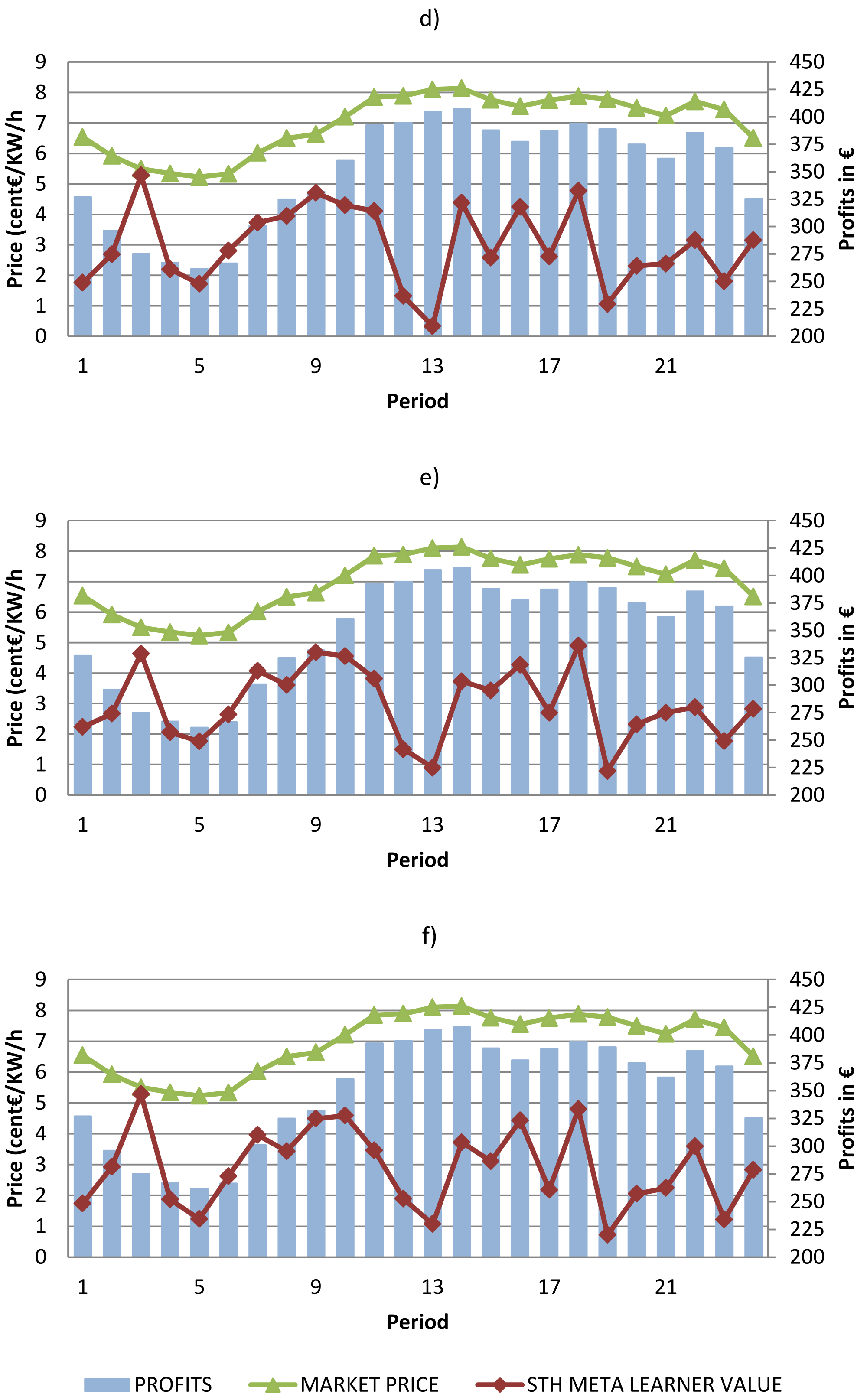


Figure 23 – GA sensibility test results: a) Control settings, b) RLA Weights, c) 200 generations, d) 20% deviation, e) 8 and 21 crossover points, f) 0.5 Probability of mutation

None of these parameterizations detaches from the others, all get good results being able to sell all energy and in all periods of the day as shown on Table 3.

Table 3 – Parameterizations' Profits and Execution time

Parameterization	1	2	3	4	5	6
Profit (€)	8,366.50	8,366.50	8,366.50	8,366.50	8,366.50	8,366.50
Execution Time (ms)	59,837	63,176	115,480	60,610	60,804	60,959

In terms of execution times, no single strategy detaches itself, the larger running time value for parameterization 3 is consistent with a bigger workload on GA. The good performance values in the market are explained by STH's permanent choice of small values that are likely to be below market value.

Beyond those shown here we decided to do some additional testing to find if even with some not so reasonable parameters, would STH behave differently – we found no differences for ten thousand iterations, or a mutation probability of 0.9. The main reason for this behaviour is that generally, as shown in Figure 24, STH's answers are very influenced by the answers given by Green Agent, even with the same weights for every strategy.

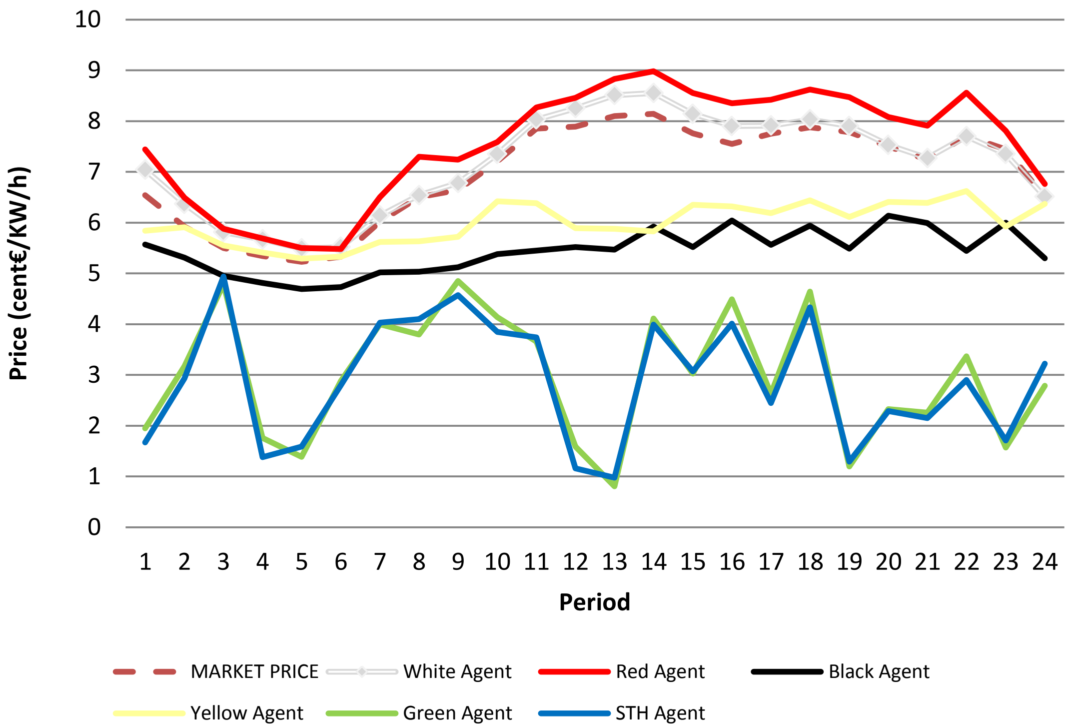


Figure 24 – Individual Hat Agent's answers and STH final answer for default settings

Tests show that the fitness function will, more often than not, opt for the lowest values in the genetic pool – allowing their survival and reproduction – because of the way the market price is chosen (see section 2.1.2), with the current mechanism bids lower than the established market price will be accepted.

Some remarks on the other agents' answers are due and listed here:

- Not surprisingly both White and Red Agents' answers are very close to the real market price. Although bidding those values were not chosen and would, in most periods, cause no energy to be sold, we believe that their presence in the pool can be of great value.

- Careful thinking usually pays off, and the Black Agent's chosen bids are an example of that. Despite that only in period 3 it contributed directly for the final answer, in our opinion, this agent's answers can grant some safety.

- Yellow Agent's answers were astoundingly close to Black Agent's ones, even with a fourfold risk. However, we do not believe that these values should be discarded, with proper weight adjustments they can prove very valuable providing answers very close but seldom higher to the market price.

It is clear, after this analysis that the fitness function greatly influences STH's general behaviour, to test this hypothesis we chose to do some additional tests to STH's behaviour with arbitrarily chosen weights, in Table 4 are the weights chosen for this test.

Table 4 – Fitness function's behaviour testing

	White Agent	Red Agent	Black Agent	Yellow Agent	Green Agent
Answer Weight	32	16	8	4	2

We intended to have Green Agent's answers to have, by comparison, an insignificant weight.

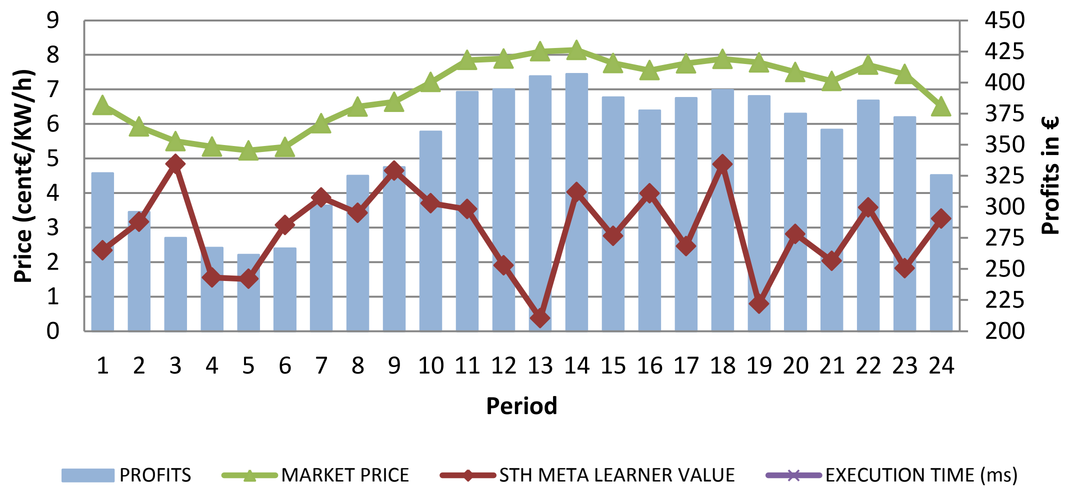


Figure 25 – Fitness function stress test results

This shows that the fitness function hardly allows individuals other than Green Agent's answers to reproduce. We recall that the Green Agent is, in fact, the Determinism Theory

Strategy and that this strategy takes advantage of the Player Profile Definition, trying to predict every other agent's bids, whilst our fitness function simulates the market to predict the same thing.

Unless another agent provides better answers for all periods of a crossover section, STH will almost certainly follow Green Agent's answers. Notice that this does not tell us that we should discard STH and keep with the Determinism Theory Agent, it does, however, indicate that further testing is needed (see section 5 below).

4.3 Case Study 2 – STH versus Simple Metalearner

For a description of the Simple Metalearner, please refer to section 2.4.4.2.

To compare the performance of these two strategy agents, we ran a market simulation for an arbitrary period along 61 days starting September, 1st 2008 for each. The chart in Figure 26 presents the incomes obtained by Seller 2 in both simulations.

We chose to use default parameterization set for running STH, because as seen earlier none of the tested sets outperforms the others.

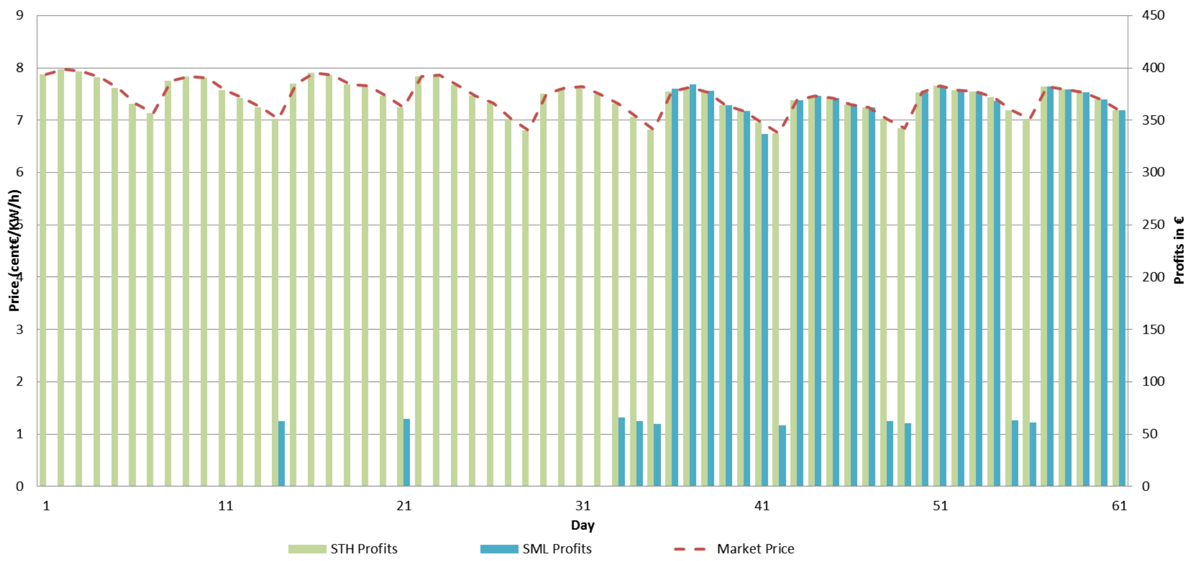


Figure 26 – Simple Metalearner results versus STH's results

It is visible that Simple Metalearner (SML) strategy begins with bad results, which start to improve after some days. This is due to some supporting strategies' worst suggestions at the start, while they do not have the experience to learn adequately yet. As this metalearner considers all suggestions in a similar way, the good outputs that some strategies may be presenting are muffled by the bad ones. The consideration of all suggestions in an equal way results in a bad performance of the Simple Metalearner especially in the first days.

STH keeps an excellent performance through all days of the simulation; due to the selection of the values being performed by genetic algorithms, it does not suffer from inadequate suggestions from the other strategies, even though they do exist like seen in section 4.2.

Table 5 presents profits obtained by each of the strategies; STH presents an amazing 169.88% improvement.

Table 5 – SML and STH Profits

	SML	STH
Profit (€)	8,414.77	22,710.00

In the previous section we stated that the STH meta-learner is following closely the values of the Green Agent, as such we wanted to test its influence in this simulation. Figure 27 illustrates this influence.

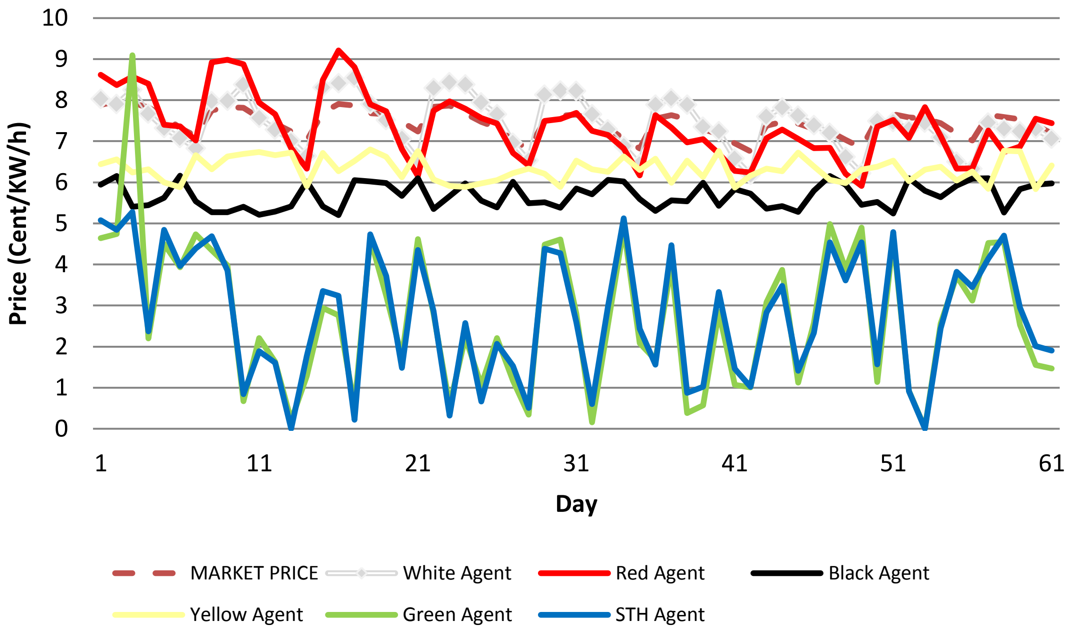


Figure 27 – Individual agent answers and STH final answer along period 18 of 61 days starting 1st September 2008

It becomes clear that Green Agent's answers for being so low have almost fourfold weight in the algorithm as they contribute solely for the genes of the individual GAI_{minima}, and strongly contribute for the genes of the individuals GAI_{D1} and GAI_{D2}, hence the small deviation often visible in the chart. This also helps to explain why the behaviour remained the same even with a very small weight value for the Green Agent, the random values individuals remained with a weight of one. Nevertheless it is worthy to focus that on periods 1 to 3, STH chose answers different from that of the Green Agent – as these were quite bad.

4.4 Case Study 3 – STH versus Weighted Metalearner

For a description of the Weighted Metalearner, please refer to section 2.4.4.3.

Like the last simulation, we compared STH's strategy performance with the Weighted Metalearner (WML) strategy using the bids of a market simulation of 61 days starting on September, 1st 2008.

For these simulations, the main reinforcement learning algorithm, whose confidence values are used by the Weighted Metalearner, is the Bayes Theorem algorithm.

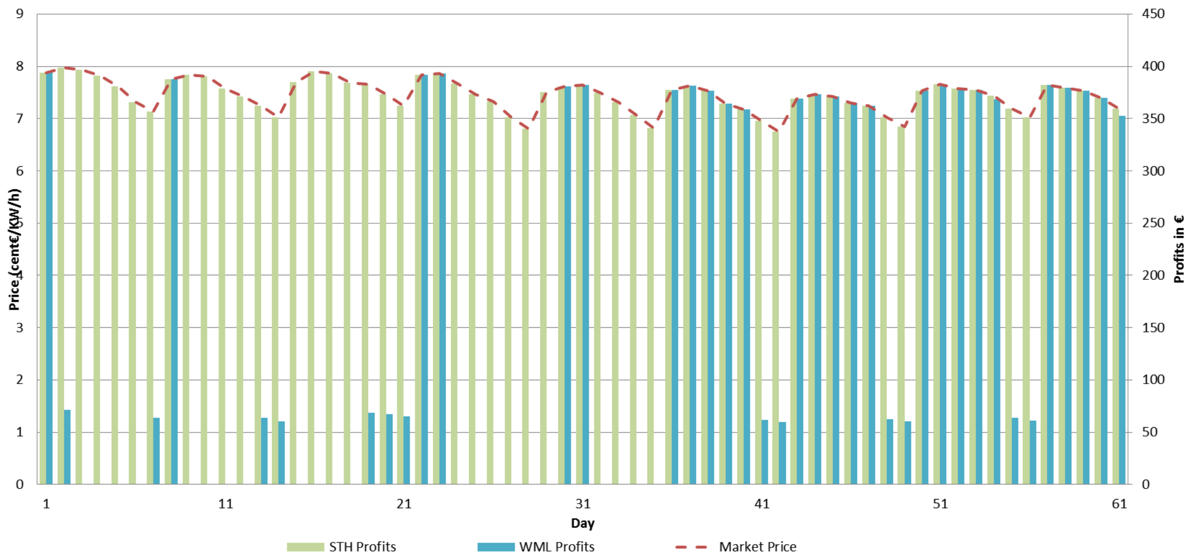


Figure 28 – Weighted Metalearner results versus STH's results

Figure 28 shows why it is important to use and learn which different weights should be attributed to each strategy's outputs. This method considers adequate weights to instigate the attribution of higher importance to the strategies that are presenting best results at each time and therefor got much better results than SML.

Analogously to SML, WML takes some time to learn about which strategy is outputting the best results. STH still outperforms WML but this time, however, by a lesser margin, as seen in Table 6.

Table 6 – WML and STH Profits

	SML	STH
Profit (€)	10,604.65	22,710.00

Both SML and WML simulation values are the same used in [Pinto T., 2011], more details about these simulations including parameters, experimental findings and internal behaviour of the mechanisms used are to be found for SML and WML in section 3.7.9.3 of [Pinto T., 2011].

4.5 Case Study 4 – STH versus AMES Agent

The input parameters of AMES strategy were the same used in [Pinto T., 2011], which will not be detailed here (see sub-section 3.7.3 of that document).

Comparing STH strategy performance to AMES agent's strategy, we can see that similar behaviour to SML and WML occurs – we recall that the AMES strategy uses the Roth-Erev reinforcement learning algorithm to choose the best among a set of possible bids that are calculated based on the relation cost/profit that the player presents when producing electricity (see sub-section 2.4.3.10). Due to this, even with SA, it takes some time for AMES strategy to learn about what values to bid.

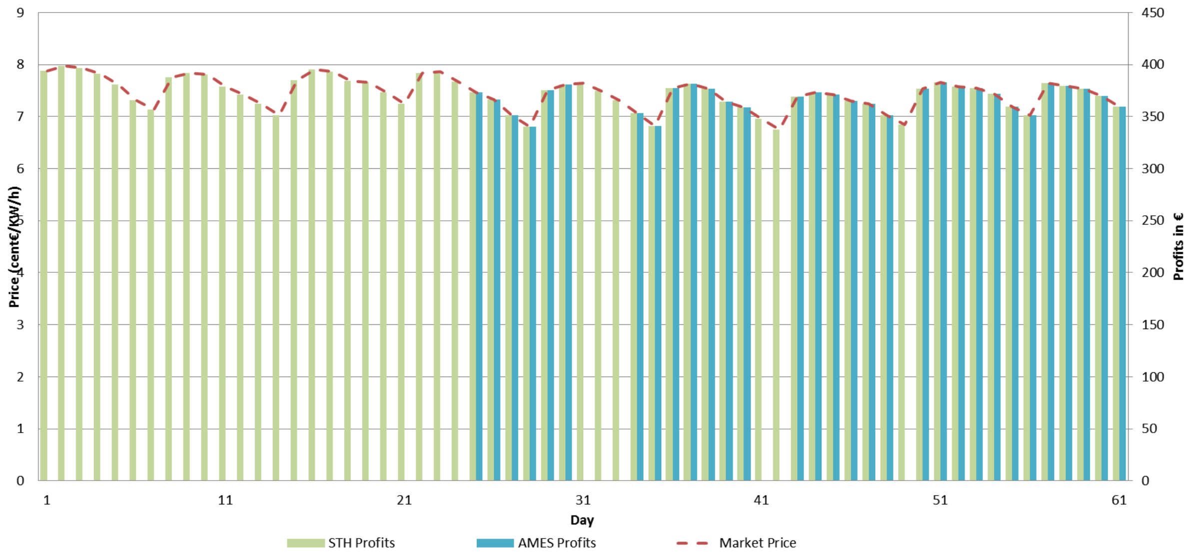


Figure 29 – AMES Strategy results versus STH's results

We can see both strategies profits in Table 7.

Table 7 – AMES and STH Profits

	AMES	STH
Profit (€)	11,398.50	22,710.00

Note that a certain offer that performs well in a certain day does not guarantee success in all days. There are safer offers, which guarantee incomes in almost all cases, but being incapable of achieving the highest possible gain in each day; and riskier offers, which are able to obtain the higher possible gains, but that risk translates into failure in some other days.

5 Conclusion and Future Work

"*The biggest enemy of thinking is complexity, for that leads to confusion. When thinking is clear and simple, it becomes more enjoyable and more effective.*" [De Bono E., 1985]

This dissertation presented a new path for research in agent-based decision, namely on electricity markets using MASCEM and ALBidS.

ALBidS combines several different strategic approaches, seen as tools by the Main Agent; this works extended ALBidS by presenting yet another strategy combining previously existing strategies in a different way.

The Six Thinking Hats method was stepping stone for all the work presented here. It even revealed itself useful to take some decisions about the routes this dissertation should take. One of the biggest challenges was how to "de-humanize" concepts such as emotions, obvious and even thinking.

The STH ALBidS' meta-learner can be seen as a different way to assemble information about the decisions at hand. It treats several agents as if they were different persons in a meeting addressing a single decision; it then applies the concept natural of the survival of the fittest over the aggregate of all the ideas and routes that spanned at this same meeting to pick one final decision. Unlike ALBidS main concept where all strategies' answers are independent and concurrent and mutually exclusive, STH tries to take the best of each "idea" in the table, combining the best of them all.

After experimentation, we believe that the fitness function as it is, is trimming away effectively any unsafe answer always choosing low bids; although this is most wanted and has proven to give a steady income, it is not safe to believe that this method is infallible. Our function is taking advantage of a mechanism similar to that of the Determinism Theory Strategy (see subsection 2.4.3.15), trying to predict every other agent's bids, as such, STH is now following the trends Green Agent gives.

During development lots of ideas of what add to STH and ALBidS and now we find that there are many paths to tread, these ideas are summed on the next subsections.

5.1 What can be changed about STH

In this section we describe what could have been done differently or what went not that good during the elaboration of this work.

We have seen that STH's answers suffered from deep bias to lower values that guaranteed selling all the available energy, this behaviour was not expected, and it may be that the initial gene pool configuration should be revised. Also, for a suitable analysis of the GA behaviour, especially about what concerns crossover zones, some tests without the Green Agent should be done.

5.2 Future development

Here are some suggestions for future work on STH's working in energy markets.

1. Having Hat Agents providing more than one individual to the initial gene pool used by GA's, for example:

    a. White Agent fetching other historic data like from last year, or the year before (properly adjusted to fit on the type of day/weekday)

    b. Black and yellow supplying one individual with fixed risk, and another with random risk in a range.

2. Green Agent using evolutionary algorithms other than PSO, for example Genetic Algorithms over data from some arbitrary periods (see 1.a)

3. Analyse the genetic algorithm behaviour with different crossover points or a different number crossover points, namely:

    a. Using the sunrise-sunset data from a service like timeandate.com [timeandate.com, 2008] to take full advantage of the difference between these events each day of the year;

    b. Test for a bigger number of crossover points separating, for example, early morning from mid-morning and lunch, and late afternoon.

4. Test for different crossover techniques like Uniform Crossover or Half Uniform Crossover

5. Change the number of descendants in each generation

6. Change STH to use a blackboard system

7. Change STH to have one agents outputs being the other inputs, and perform in a linear fashion

8. Adapt ALBidS and STH to be used for bilateral contracts negotiation

9. Test STH with different market-clearing tools, namely one that favours bids lower than but closer to the market price; the study of this behaviour could also be interesting in bilateral contracts negotiation.

As a final remark about this work, in our understanding, using a strategy that effectively combines the best of others can prove to be a great asset to the decisions made by ALBidS. This work is, hopefully, just the first of many exploring this path.

References

[Bertram J., 2000] John S Bertram, The molecular biology of cancer, Molecular Aspects of Medicine, Volume 21, Issue 6, December 2000, Pages 167-223, ISSN 0098-2997, 10.1016/S0098-2997(00)00007-8. (http://www.sciencedirect.com/science/article/pii/S0098299700000078)

[Botti *et al.*, 1995] V. Botti, F. Barber, A. Crespo, E. Onaindia, A. Garcia-Fornes, I. Ripoll, D. Gallardo, L. Hernández, A temporal blackboard for a Multiagent environment, Data & Knowledge Engineering, Volume 15, Issue 3, June 1995, Pages 189-211, URL: http://www.sciencedirect.com/science/article/pii/0169023X9500007F

[Bruha I., 2004] Bruha, I., "Meta-Learner for Unknown Attribute Values Processing: Dealing with Inconsistency of Meta-Databases", Journal of Intelligent Information Systems, vol. 22, no. 1, 2004

[Bunge M., 1959] Bunge, M., "Causality: The Place of the Causal Principle in Modern Science", Harvard University Press, 1959

[Cederberg and Rannug, 2006] Håkan Cederberg, Ulf Rannug, Mechanisms of human minisatellite mutation in yeast, Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis, Volume 598, Issues 1–2, 25 June 2006, Pages 132-143, ISSN 0027-5107, 10.1016/j.mrfmmm.2006.01.010. (http://www.sciencedirect.com/science/article/pii/S0027510706000534)

[Corkill D., 1991] Corkill D. (1991). Blackboard systems. AI Experts. 6 (1), 40-47 URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.137.7801

[Darwin C., 1859] Darwin, C. R. 1859. On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life. London: John Murray. Tenth thousand, p. 53 (http://darwin-online.org.uk/content/frameset?itemID=F387&viewtype=text&pageseq=82) RN1

[de Bono E., 1985] Edward de Bono. *Six Thinking Hats*. England: Penguin Books, 1985

[Einstein A., 1905] Einstein, A., "Zur Elektrodynamik bewegter Koerper", Annalen der Physik, vol.17, pp. 891-921, 1905

[Erev and Roth, 1998] Erev, I, & Roth, A 1998, 'Predicting How People Play Games: Reinforcement Learning in Experimental Games with Unique, Mixed.', American Economic Review, 88, 4, p. 848, Business Source Complete, EBSCOhost, viewed 21 September 2012.

[Green C., 2003] Green, C., "The Lost Cause: Causation and the Mind-Body Problem", Oxford Forum Publisher: Oxford Forum, vol. 1, 2003

[Huang *et al.*, 2010] Ming Huang; Pengfei Liu; Xu Liang; , "An improved multi-population genetic algorithm for job shop scheduling problem," Progress in Informatics and Computing (PIC), 2010 IEEE International Conference on , vol.1, no., pp.272-275, 10-12 Dec. 2010

doi: 10.1109/PIC.2010.5687449
URL:
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5687449&isnumber=5687395

[Jennings and Wooldridge, 1995] Jennings, N. R. and Wooldridge, M. J. *Applying Agent Technology*. Int. Journal of Applied Artificial Intelligence, 9, (4), 351-369, 1995

[Jennings N., 1994] Jennings, N. R. (1994) *The ARCHON System and its Applications*. In, 2nd Int. Conf. on Cooperating Knowledge Based Systems (CKBS-94), Keele, UK, 13-29.

[Juang and Lu, 2009] Chia-Feng Juang; Chun-Ming Lu; , "Ant Colony Optimization Incorporated With Fuzzy Q-Learning for Reinforcement Fuzzy Control," Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on , vol.39, no.3, pp.597-608, May 2009
doi: 10.1109/TSMCA.2009.2014539
URL:
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4796256&isnumber=4815381

[Koritarov, 2004] Koritarov, V., "Real-World Market Representation with Agents: Modeling the Electricity Market as a Complex Adaptive System with an Agent-Based Approach", IEEE Power & Energy magazine, pp. 39-46, 2004

[Larrañaga *et al.*, 1996] Larrañaga, P.; Kuijpers, C.M.H.; Murga, R.H.; Yurramendi, Y.; , "Learning Bayesian network structures by searching for the best ordering with genetic algorithms," Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on , vol.26, no.4, pp.487-493, Jul 1996
doi: 10.1109/3468.508827
URL:
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=508827&isnumber=10954

[Lee C. *et al.*, 1997] C.-Y. Lee, S. Piramuthu & Y.-K. Tsai (1997): Job shop scheduling with a genetic algorithm and machine learning, International
Journal of Production Research, 35:4, 1171-1191

[Lee I. *et al.*, 1997] Lee, I.; Sikora, R.; Shaw, M.J.; , "A genetic algorithm-based approach to flexible flow-line scheduling with variable lot sizes," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on , vol.27, no.1, pp.36-54, Feb 1997
doi: 10.1109/3477.552184
URL:
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=552184&isnumber=11996

[Li and Tesfatsion, 2009] Li, H. and Tesfatsion, L., "Development of Open Source Software for Power Market Research: The AMES Test Bed", Journal of Energy Markets, vol. 2, no. 2, pp. 111-128, 2009

[Mendel G., 1866] Mendel, G., 1866. Versuche über Pflanzen-Hybriden. Verh. Naturforsch. Ver. Brünn 4: 3–47 (in English in 1901, J. R. Hortic. Soc. 26: 1–32)

[Morais *et al.*, 2008] Morais, H.; Cardoso, M.; Khodr, H.; Praça, I.; Vale, Z.; , "Virtual power producers market strategies," Electricity Market, 2008. EEM 2008. 5th International Conference on European , vol., no., pp.1-6, 28-30 May 2008

doi: 10.1109/EEM.2008.4579099
URL:
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4579099&isnumber=4578994

[Oliveira *et al.*, 2009] Oliveira, P.; Pinto, T.; Morais, H.; Vale, Z.A.; Praça, I.; , "MASCEM – An Electricity Market Simulator providing Coalition Support for Virtual Power Players," Intelligent System Applications to Power Systems, 2009. ISAP '09. 15th International Conference on , vol., no., pp.1-6, 8-12 Nov. 2009
doi: 10.1109/ISAP.2009.5352933
URL:
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5352933&isnumber=5352811

[Pinto *et al.*, 2009a] Pinto, T.; Vale, Z.A.; Morais, H.; Praça, I.; Ramos, C.; , "Multiagent based electricity market simulator with VPP: Conceptual and implementation issues," Power & Energy Society General Meeting, 2009. PES '09. IEEE , vol., no., pp.1-9, 26-30 July 2009
doi: 10.1109/PES.2009.5275565
URL:
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5275565&isnumber=5260217

[Pinto *et al.*, 2011a] Pinto, T.; Vale, Z.; Rodrigues, F.; Praça, I.; Morais, H.; , "Multiagent system for adaptive strategy formulation in electricity markets," Intelligent Agent (IA), 2011 IEEE Symposium on , vol., no., pp.1-8, 11-15 April 2011
doi: 10.1109/IA.2011.5953609
URL:
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5953609&isnumber=5953603

[Pinto *et al.*, 2011b] Pinto, T.; Vale, Z.; Rodrigues, F.; Praça, I.; Morais, H.; , "Cost dependent strategy for electricity markets bidding based on adaptive reinforcement learning," Intelligent System Application to Power Systems (ISAP), 2011 16th International Conference on , vol., no., pp.1-6, 25-28 Sept. 2011
doi: 10.1109/ISAP.2011.6082167
URL:
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6082167&isnumber=6082157

[Pinto *et al.*, 2011c] T. Pinto, H. Morais, P. Oliveira, Z. Vale, I. Praça, C. Ramos, A new approach for Multiagent coalition formation and management in the scope of electricity markets, Energy, Volume 36, Issue 8, August 2011, Pages 5004-5015, ISSN 0360-5442, 10.1016/j.energy.2011.05.045.
URL:
http://www.sciencedirect.com/science/article/pii/S0360544211003707

[Pinto T., 2011] Pinto, T.; Adaptive Learning In Agents Behaviour: A Framework for Electricity Markets Simulation. Dissertation to obtain the Master of Science Degree in Computer Science – Specialization in Knowledge-Based and Decision Support Technologies, ISEP-IPP, 2011.

[Praça *et al.*, 2003] Isabel Praça, Carlos Ramos, Zita Vale, Manuel Cordeiro. *MASCEM: A Multiagent System That Simulates Competitive Electricity Markets*. Intelligent Systems, IEEE. 18 (6), 54-60, 2003

[Praça *et al.*, 2008] Praça, I.; Morais, H.; Ramos, C.; Vale, Z.; Khodr, H.; , “Multiagent electricity market simulation with dynamic strategies & virtual power producers,” Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE , vol., no., pp.1-8, 20-24 July 2008
doi: 10.1109/PES.2008.4596710
URL:
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4596710&isnumber=4595968

[Príncipe, 2010] Principe, J., “Information Theoretic Learning”, Springer, Information Science and Statistics series, 2010

[Rahimi-Kian *et al.*, 2005] Rahimi-Kian, A.; Sadeghi, B.; Thomas, R.J.; , “Q-learning based supplier-agents for electricity markets,” Power Engineering Society General Meeting, 2005. IEEE , vol., no., pp. 420- 427 Vol. 1, 12-16 June 2005
doi: 10.1109/PES.2005.1489514
URL:
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1489514&isnumber=32012

[Roth and Erev, 1995] Alvin E. Roth, Ido Erev, Learning in extensive-form games: Experimental data and simple dynamic models in the intermediate term, Games and Economic Behaviour, Volume 8, Issue 1, 1995, Pages 164-212, ISSN 0899-8256, 10.1016/S0899-8256(05)80020-X.
URL:
http://www.sciencedirect.com/science/article/pii/S089982560580020X

[Russell and Norvig, 2009a] Russell, S., Norvig, P. (2009). Intelligent Agents. In: *Artificial Intelligence: A Modern Approach*. 3rd ed. Stanford: Prentice Hall. 42-46

[Santos *et al.*, 2011] Santos, G.; Pinto, T.; Morais, H.; Praça, I.; Vale, Z.; , “Complex market integration in MASCEM electricity market simulator,” Energy Market (EEM), 2011 8th International Conference on the European , vol., no., pp.256-261, 25-27 May 2011
doi: 10.1109/EEM.2011.5953019
URL:
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5953019&isnumber=5952524

[Sun and Tesfatsion, 2007] Sun, J. and Tesfatsion, L., “Dynamic Testing of Wholesale Power Market
Designs: An Open-Source Agent-Based Framework”, Computational Economics,
vol. 30, no. 3, pp. 291-327, 2007
URL: http://dx.doi.org/10.1007/s10614-007-9095-1

[Vale *et al.*, 2011a] Vale, Z.; Pinto, T.; Praça, I.; Morais, H.; , “MASCEM: Electricity Markets Simulation with Strategic Agents,” Intelligent Systems, IEEE , vol.26, no.2, pp.9-17, March-April 2011
doi: 10.1109/MIS.2011.3
URL:
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5696716&isnumber=5751209

[Vale et al., 2011b] Vale, Z. *et al.*, “Electricity Markets Simulation: MASCEM contributions to the challenging reality”, Handbook of Networks in Power Systems, Springer-Verlag, 2011

	http://www.springerlink.com/content/jml6t2n863l1q404/
[Vilalta and Drissi, 2002]	Vilalta, R, Drissi, Y. (2002). A Perspective View and Survey of Meta-Learning. Artificial Intelligence Review. 18 (2), p77-95. doi: 10.1023/A:1019956318069 URL: http://dx.doi.org/10.1023/A:1019956318069
[Weiss G., 1999a]	Gerhard Weiss *et al.* Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, pp. 1-27, 1999
[Weiss G., 1999b]	Gerhard Weiss *et al.* Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, pp. 27-29, 1999
[Winnie J., 1996]	Winnie, J., "Deterministic Chaos and the Nature of Chance", The Cosmos of Science - Essays of Exploration, University of Pitsburgh Press, pp. 299–324, 1996
[Wolpert D., 1992]	Wolpert D. H. (1992). Stacked Generalization. Neural Networks. 5, p241-259. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.56.1533
[Wooldridge M., 2009a]	Michael Wooldridge. (2009). Preface. In: *An Introduction to Multiagent Systems*. Oxford: Wiley. pp. xi-xviii.
[Wooldridge M., 2009b]	Michael Wooldridge. (2009). Introduction. In: *An Introduction to Multiagent Systems*. Oxford: Wiley. pp. 1-12.
[Wooldridge M., 2009c]	Michael Wooldridge. (2009). Intelligent Agents. In: *An Introduction to Multiagent Systems*. Oxford: Wiley. pp. 15-45.

URL References

[MATLAB, 2012]	The MathWorks, Inc. (2012). Genetic Algorithm Solver - Global Optimization Toolbox for MATLAB & Simulink. Available: http://www.mathworks.com/products/global-optimization/description4.html. Last accessed 04th Oct 2012.
[OAA, 2007]	SRI International. (2007). The Open Agent Architecture. Available: http://www.ai.sri.com/oaa/. Last accessed 11th Oct 2012.
[OMEL, 2011]	Operador del Mercado Ibérico de Energia. (2011). homepage. Available: http://www.omel.es/. Last accessed August 2011.
[timeandate.com, 2008]	timeanddate.com. (2008). Sunrise and sunset in Madrid. Available: http://www.timeanddate.com/worldclock/astronomy.html?n=141&month=9&year=2008&obj=sun&afl=-11&day=1. Last accessed 08th Sept 2012.