FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Crew and Aircraft Electronic Market

Luís Brochado Pinto dos Reis

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Ana Paula Rocha

Co-supervisor: António Castro

June 27, 2017

Crew and Aircraft Electronic Market

Luís Brochado Pinto dos Reis

Mestrado Integrado em Engenharia Informática e Computação

June 27, 2017

Abstract

In the Air Transportation domain, a disruption in an operational plan means that a flight, for some reason (due to an irregular event) can not meet the planned schedule, causing delays or canceled flights. Disruptions can be classified in two types, a massive one, which makes impossible to fly safely in the affected area and leads to canceled flights, or a smaller and most frequent one, which cause delays. Studies in this area show that airline companies lose between 2% to 3% of their annual revenue, as consequence of these disruptions. The impact caused by small disruptions in companies' profits can be reduced by at least 20%, through a better Recovery Process.

Resulting from the lack of collaboration between airline companies, operation recovery from a disruption works in a restricted solution space. To wider this space, this dissertation proposes the usage of an Electronic Market modeled as a Multi-Agent System. Airline companies will use the electronic market to negotiate their needs in order to apply their optimal recovery plan. In the specific case of this dissertation, the Airline Operations Control Center (AOCC), which is the entity responsible for dealing with irregular operations, will play the role of the buyer agent and the other airline companies the role of seller agents. The negotiation object is an abstraction named "*Flight*", which includes aircraft and crew members, based on the needs of the airline company. The flight's characteristics to be negotiated are cost and availability and if two flights have the same values for those characteristics, they will be considered the same solution even if the resources composing it are distinct.

The proposed negotiation occurs in several rounds. The AOCC (buyer agent) gives feedback over the proposals committed on each round by the sellers interested in leasing the asked resource(s), leading to new proposals until all sellers refuse to negotiate any further. The seller agent uses learning to calculate a new proposal in each round, through reuse similar cases of past experiences (using Case-Based Reasoning - CBR). At the end of the negotiation, the buyer agent selects the seller who proposed the most advantageous solution as the winner.

The main goal of the AOCC is to minimize the costs caused by a disruption, what can be better achieved by leasing resources from other companies. In order to validate this concept and to understand whether it is advantageous or not, an evaluation is performed to show that solutions obtained with recourse to the electronic market are more cost-effective than solutions obtained with the company's own resources.

Resumo

No domínio do transporte aéreo, uma interrupção de um plano operacional significa um voo que por algum motivo (devido a um evento irregular) não consegue cumprir o seu plano, provocando atrasos ou até o cancelamento de voos. As interrupções de voos podem ser de dois tipos, de grande escala, que impossibilita a realização de voos em segurança na área afetada, ou então, interrupções de menor dimensão e mais frequentes, que causam atrasos. As companhias aéreas perdem entre 2% a 3% da sua receita anual, como consequência de interrupções. O impacto provocado por interrupções de menor dimensão nos lucros das companhias aéreas pode ser reduzido em pelos menos 20%, com um melhor Processo de Recuperação.

Como resultado da falta de colaboração entre companhias aéras, a recuperação de operações atua num espaço de soluções reduzido, assim, para expandir esse espaço, esta dissertação propõe a utilização de um Mercado Eletrónico baseado num Sistema Multi-Agente. As companhias aéreas utilizarão o mercado eletrónico para negociarem as suas necessidades, de modo a aplicarem o seu plano de recuperação ótimo. No caso específico desta dissertação, o Centro de Controlo de Operações (CCO), que é a entidade responsável por gerir as operações irregulares, será considerado o agente comprador e as outras companhias aéreas, os agentes vendedores. O objeto da negociação é uma abstração nomeada de "*Voo*", que inclui aeronaves e tripulantes, baseado nos recursos necessários para minimizar o atraso do voo interrompido. As características em negociação de um voo são o seu custo e disponibilidade, e se essas características tiverem o mesmo valor para dois voos, significa que estes voos são considerados a mesma solução, ainda que os recursos de cada um sejam diferentes.

O mercado funciona com base no *feedback* do CCO sobre as propostas enviadas em cada ronda, pelos vendedores interessados em alugar o(s) recurso(s) pedido(s), conduzindo a novas propostas (renegociando o mesmo voo ou um diferente) até que todos os vendedores recusem continuar a negociar. É então que o comprador informa o vendedor com a solução mais vantajosa que a sua proposta foi aceite e informa todos os outros vendedores que as suas propostas foram rejeitadas.

O objectivo principal do CCO passa por minimizar os custos derivados de qualquer interrupção, o que consideramos que pode ser melhor atingido através do aluguer de recursos de outras companhias. De modo a validar este conceito e para compreender se é vantajoso ou não, será necessário avaliar se os resultados obtidos com recurso ao mercado eletrónico têm uma relação custo/benefício maior do que as soluções obtidas com os recursos próprios de cada companhia.

Acknowledgements

I would like to start by thanking to Alan Turing, the father of Artificial Intelligence, my major area of interest and whose work kept me motivated during this masters degree.

Secondly, I would like to thank my supervisors, Professor Ana Paula Rocha and Professor António Castro, for trusting me to develop this project and also for all the support given whenever requested.

Thirdly, I would like to thank all my friends that were present in this journey, specially Diogo Pinto, Fábio Silva, João Cardoso, João Carlos Santos, Sara Paiva, Tiago Silva and Wilson Oliveira, who have stood by my side me in the most somber period of my life.

I could never forget my fellows, $\rho 7$ Boyz, for all the amusing moments and all the cooperation during the hard task of writing up the dissertation.

Additionally, I would like to have a special gratitude word towards Professor António Augusto de Sousa and Doctor Helena Lopes, who were present in the worst year of my life and helped me overcome my problem.

I also want to thank Cátia Nunes, a person that was most important to me, who always supported and guided me in all the bad moments I found myself into, who always had a smile that could enlighten the darkest of the skies.

Lastly, I would like to thank to my grandfather, whom I miss everyday, the man responsible for making me believe that I could chase my dreams.

Finally, I would like to express the deepest gratitude towards my mother, Branca Teresa, who always supported me and kept me focused, who made enormous sacrifices to make my graduation possible.

Luís Brochado Pinto dos Reis

"I have lived my life as best I could, not knowing its purpose, but drawn forward like a moth to a distant moon. And here, at last, I discover a strange truth. That I am only a conduit for a message that eludes my understanding. Who are we, who have been so blessed to share our stories like this? To speak across centuries? Maybe you will answer all the questions I have asked. Maybe you will be the one to make all this suffering worth something in the end."

Ezio Auditore da Firenze, Assassin's Creed: Revelations

Contents

List of Figures

LIST OF FIGURES

List of Tables

Abbreviations

AI	Artificial Intelligence
AOCC	Airline Operations Control Centre
CBR	Case-based Reasoning
DF	Directory Facilitator
EM	Electronic Market
JADE	Java Agent Development Framework
MAS	Multi-Agent System
MASDIMA	Multi-Agent System for Disruption Management

Chapter 1

Introduction

Even if we don't realize its presence, Artificial Intelligence (AI) has grown in the last decades. It has become of great use in people's day-to-day lives, from achieving simple tasks, like getting a recommendation while shopping at Amazon, to more complex ones, like automating your home [sma]. In the Air Transportation domain, more specifically in the operations recovery problem (or disruption management, which is the process that tries to minimize irregular operations), AI, and specially Distributed AI (DAI) can be a useful resource due to its ability to deal with problem's high complexity, distribution and dynamism.

This chapter will present the context, motivation and goals of this dissertation along with its structure.

1.1 Context

One of the most important roles in an airline company is the Airline Operations Control Center (AOCC), the entity responsible for monitoring flights, detecting irregular events and managing consequent irregular operations (disrupted flight plans) trying to minimize the costs caused by these irregular events. An irregular event, also called a flight disruption, is an event that causes the flight to fail its planned schedule. As a main function, the AOCC must ensure that the operations plan is followed or, when impossible to do so, find alternative plans to reduce delays and consequently the costs caused by delayed flights. A disruption can be a massive one, commonly caused by natural disasters, like hurricanes or volcanic eruptions, making impossible to fly in the affected area and leading to canceled flights or it can be a more frequent and small type of disruption, usually caused by aircraft malfunctions, crew absenteeism or bad weather.

1.2 Motivation and Objectives

The estimated costs caused by irregular operations to airline companies, is between 2% to 3% [CRO12],[CCZ10] of its annual revenue, which even for small companies, can be a loss of million euros, i.e. according to TAP's 2010 annual report [TAP17], the cost of all irregular operations during that year was comprehended between 39.7M€ and 59.5M€, of a total revenue of 1,986.3M€.

Due to the usage of their own resources in the recovery process, airline companies can not always ensure the solution reliability in terms of cost reduction. As there are studies that point to a cost reduction of at least 20% [Irr96] if there was a better recovery process, this dissertation's main goal is to evaluate how the promotion of collaboration between airline companies when dealing with problems that arise during their own operational plan, changes each company's solution space, which is narrowed as consequence of the lack of collaboration between airline companies.

In order to do this, a multi-agent system based electronic market is introduced to help expand airline companies' solutions space, by leasing each other the required resources (aircraft and crew members). The agreement is reach through an automatic negotiation process, where participants change proposals and counter-proposals based in their own preferences and availability, as well as experience, using a case-based reasoning approach.

1.3 Dissertation Structure

Besides the introduction, this dissertation includes five more chapters.

Chapter 2 includes a literature review on the topics of Disruption Management, Multi-Agent Systems and Electronic Markets.

Chapter 3 contains a more thorough description of the problem and an overview over the approach chosen to solve it.

The details of the implemented solution are described in chapter 4.

Chapter 5 presents the experiments and discusses the results obtained.

Finally, chapter 6 concludes this dissertation by presenting its main contributions and suggestions for future work.

Chapter 2

Literature Review

This chapter presents a summary over the literature reviewed on the problem and the proposed solution in order to understand the current state of the art. It focuses on the topics of Disruption Management, Multi-Agent Systems and Electronic Markets.

2.1 Arline Disruption Management

On the AOCC (Airline Operations Control Center) relies the responsibility to ensure that flights meet their planned schedule or, if any problem arises, to find a viable solution that minimizes both the impact in the operational plan and the cost of it. Operations management is essentially a manual process that among other functions includes monitoring, event detection and problems resolution and, strongly depends on the tactical knowledge of the AOCC's members [Cas08]. The AOCC acting cycle is illustrated in figure 2.1 .

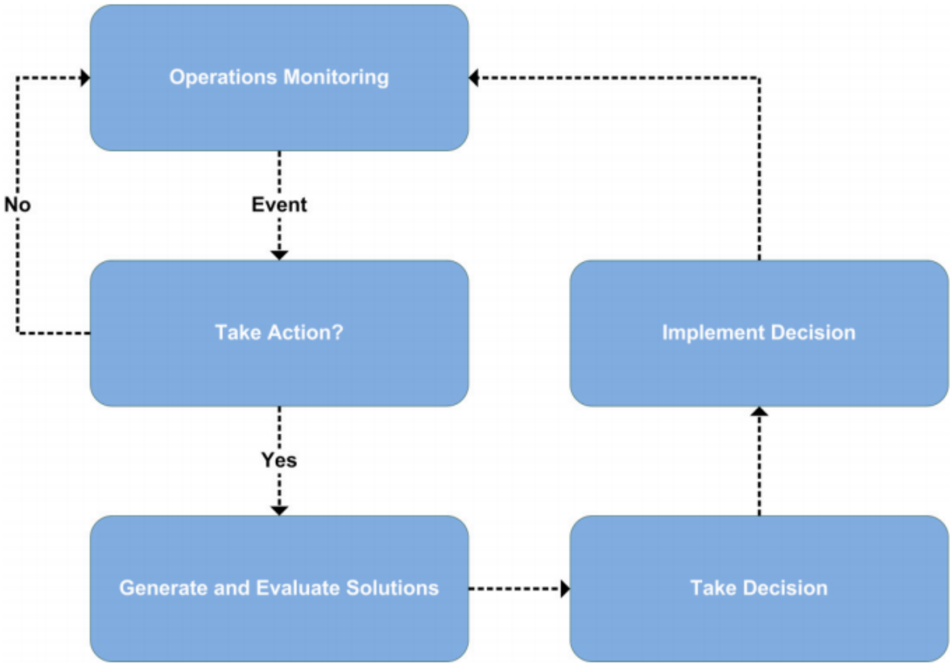


Figure 2.1: AOCC Acting Cycle. Source: [Cas08]

Every time an irregular event that has an impact on the scheduled plan is detected, the AOCC's team has to plan carefully an alternative schedule, to ensure that it minimizes at most the disruption cost. A disruption can be view as composed by four dimensions [dC13]:

- Disrupted Aircraft

  Aircraft that can not meet its schedule plan.

- Disrupted Crew Member

  Crew member that can not meet its schedule plan.

- Disrupted Passenger

  I: Passenger that lost one or more flight connections due to disrupted flights.

  II: Passenger whose itinerary contains a disrupted flight.

- Disrupted Flight

  Flight that can not meet its schedule plan.

In disruption management, the AOCC commits to recover all the dimensions affected, so the recovery process has four possible action areas [dC13]:

- Aircraft Recovery

  The process of assigning individual aircraft to a disrupted flight minimizing a specific objective (usually the cost and flight delay) while complying with the required rules.

- Crew Recovery

  The process of assigning individual crew members to a disrupted flight minimizing a specific objective (usually the cost and delay) while complying with the required rules.

- Flight Recovery

  The process of repairing a flight schedule after a disruption, through specific actions like delay, cancel or divert flights from their original schedule, so that the flight delay is minimized. It is closely related to the aircraft and crew recovery process, since it depends on the availability of these resources to be successfully solved.

- Passenger Recovery

  The process of finding alternate itineraries, commencing at the disrupted passenger location and terminating at their destination or a location nearby, while minimizing a specific objective (usually the passenger trip time and the airline costs).

Currently, we can find in the literature three different approaches to the airline disruption management problem, as follows [CRO14]:

- Sequential approach

  A process that is able to recover all problem dimensions separately but not simultaneously. (Usually, the dimensions are solved sequentially, which imposes an importance factor on them)

- Integrated approach

  A process that is able to recover all problem dimensions simultaneously. (Does not impose an importance factor to any of the dimensions)

- Partial-Integrated approach

  A process that is able to recover at least two, but not all, of the problem dimensions, simultaneously or not.

MASDIMA (described later) addresses the Aircraft, Crew and Passenger recovery problem using an integrated approach [dC13]. The Passenger recovery as also being approached in [Lim16], and it presents a Multi-Agent System, where passengers, through argumentation, negotiate with airline companies, to solve their problem in a personalized way.

2.2 Agents and Multi-Agent Systems

As Wooldridge [Woo09] defined, "An agent is a computer system that is situated in some environment,and that is capable of autonomous action in this environment in order o meet its design objectives." To support this idea, Russel and Norvig [Woo09] consider that an intelligent agent is characterized by the following capabilities:

- Reactivity

  Intelligent agents' ability to perceive their environment, and respond to changes that occur in order to meet their objectives.

- Proactiveness

  Intelligent agents' ability to exhibit goal-directed behaviour by taking the initiative in order to meet their objectives.

- Social ability

  Intelligent agents' capability of interacting with other agents in order to meet their objective.

According to its architecture and the environment it is involved in, the behaviour of an intelligent agent may fluctuate. To explain the environment properties, Russel and Norvig suggest the following [Nil96]:

I: Accessible or Inaccessible

II: Deterministic or Nondeterministic

III: Episodic or NonEpisodic

IV: Static or Dynamic

V: Discrete or Continuous

An Accessible or Inaccessible environment depends on if an agent has access to the complete state of the environment (Accessible) or not (Inaccessible). The second property refers if an agent next state can be determined by its current state and the actions selected (Deterministic). An Episodic environment is an environment where the agents do not need to think ahead, once the agents' action depends on the episode itself. If the environment can change while an agent is deliberating it is considered a dynamic environment, otherwise it is static. If the environment does not change with time but the agent's performance score does, then the environment is considered semidynamic. An environment with limited number of distinct percepts and actions on it.

Although nowadays the term "Multi-Agent System" (MAS) is used to refer all types of systems composed of agents, it once was defined as agents that work together to solve problems that are beyond the individual capabilities or knowledge of each one of them. Those agents are autonomous and may be heterogeneous in nature. The characteristics of MAS can then be summarized in the following points [JSW98]:

- Each agent has a limited viewpoint
- There is no global system control
- Data is decentralized
- Computation is asynchronous

2.2.1 Agents Utilities and Preferences

Assuming that agents in an multi-agent system are *self-interested* agents, i.e., each agent has its own preferences and desires about how the world should be. Those preferences are formally captured by means of *utility functions*, unique for each agent, which assigns to every outcome or state on an agent's preference set, a real number indicating how good the outcome is for that agent [Woo09]. The larger the number, the better from the point of view of the agent with the utility function. So, according to the same author, utility functions are just a representation of agent's preferences.

2.2.2 JADE

"The first software developments, that eventually became the JADE platform, were started by Telecom Italia (formely CSELT) in late 1998, motivated by the need to validate the early FIPA[1] specifications." [BCG07]. So, according to Bellifemine [BPR99], JADE (Java Agent DEvelopment Framework) a software framework to develop agent applications in compliance with FIPA specifications for inter-operable multi-agent systems. While being implemented in the Java language. it simplifies the implementation of multi-agent systems through a middle-ware and through a set of graphical tools that support the debugging and deployment phases [JAD17]. In order to make the implementation easier, JADE provides to programmers the following list of core functionalities [BCG07]:

- A fully distributed system inhabited by agents;
- Full compliance with the FIPA specifications;
- Efficient transport of asynchronous messages;
- Implementations of both white pages and yellow pages;
- An effective agent life-cycle management;
- Support for agent mobility;
- A subscription mechanism for agents and even external applications;
- A set of graphical tools to support programmers when debugging and monitoring;
- Support for ontologies and content languages;
- A library of interaction protocols;
- Integration with various Web-based technologies;

[1] The Foundation of Intelligent Physical Agents (FIPA) is an IEEE Computer Society standards organization that promotes agent-based technology and the interoperability of its standards with other technologies [FIP17].

- Support for J2ME[2] platform and the wireless environment;
- An in-process interface for launching/controlling a platform and its distributed components from an external application;
- An extensible kernel designed to allow programmers to extend platform functionality through the addition of kernel-level distributed services.

Still according to Bellifemine [BPR99], to simplify multi-agent systems development is JADE's main goal and, in order to do that, it offers the following list of features to an agent programmer:

- FIPA-compliant Agent Platform, which includes the AMS (Agent Management System), the DF (Directory Facilitator), and the ACC (Agent Communication Channel). All these three agents are automatically activated at the agent platform start-up;
- Distributed agent platform. The agent platform can be split on several hosts (provided that there is no firewall between them). Only one Java application, and therefore only one Java Virtual Machine, is executed on each host. Agents are implemented as one Java thread and Java events are used for effective and light-weight communication between agents on the same host. Parallel tasks can be still executed by one agent, and JADE schedules these tasks in a more efficient (and even simpler for the skilled programmer) way than the Java Virtual Machine does for threads;
- A number of FIPA-compliant DFs (Directory Facilitator) can be started at run time in order to implement multi-domain applications, where the notion of domain is a logical one as described in FIPA97 Part 1;
- Programming interface to simplify registration of agent services with one, or more, domains (i.e. DF);
- Transport mechanism and interface to send/receive messages to/from other agents;
- FIPA97-compliant IIOP protocol to connect different agent platforms;
- Light-weight transport of ACL messages inside the same agent platform, as messages are transferred encoded as Java objects, rather than strings, in order to avoid marshalling and unmarshalling procedures. When sender or receiver do not belong to the same platform, the message is automatically converted to /from the FIPA compliant string format. In this way, this conversion is hidden to the agent implementers that only need to deal with the same class of Java object;
- Library of FIPA interaction protocols ready to be used;
- Automatic registration of agents with the AMS;

[2] J2ME allows developers to use Java and the J2ME wireless toolkit to create applications and programs for wireless and mobile devices [J2M17].

- FIPA-compliant naming service: at start-up agents obtain their GUID (Globally Unique Identifier) from the platform;
- Graphical user interface to manage several agents and agent platforms from the same agent. The activity of each platform can be monitored and logged.

The communication among agents in a JADE-based application is made using the FIPA-ACL message specification. The FIPA ACL specifies a standard message language by setting out the encoding, semantics and pragmatics of the messages [BPR99]. According to the same author, Bellifemine et al. [BPR99], JADE uses the behaviour abstraction to model the tasks that an agent is able to perform and agents instantiate their behaviours according to the needs and capabilities. The framework provides three types of atomic behaviours:

- SimpleBehaviour: An atomic behaviour that must be used by the agent developer to implement atomic actions of the agent work;
- CyclicBehaviour: Atomic behaviour that must be executed forever;
- OneShotBehaviour: Atomic behaviour that executes just once.

It also provides more complex behaviours that can be implemented by extending the atomic ones.

2.2.3 MASDIMA

MASDIMA, Multi-Agent System for Disruption Management, is a Multi-Agent System responsible for solving disruptions in airline operational plans [CRO12]. Its architecture is illustrated in figure 2.2 and comprises three main decision levels: *bottom level*, composed of multiple *specialists* for each of the three dimensions (aircraft, crew and passenger); *middle level*, composed of three *managers*, one for each dimension, that selects the best solution proposal of its own dimension and cooperates with others to complete the global solution; *top level*, includes the *supervisor* responsible for presenting the final solution to the user. Each manager uses an utility function to evaluate a specific solution proposal, that takes into account two variables: cost and delay (for passengers manager, cost comprises direct and quality costs). Due to its architecture, MASDIMA manages the disruption problem by achieving the best integrated solution.

2.2.4 Case-based Reasoning

According to Riesbeck and Schank [RS13]: "A case-based reasoner solves problems by using or adapting solutions to old problems.", i.e. case-based reasoning (CBR) focuses on the reuse of knowledge acquired from previous experiences in order to solve new problems. So, CBR is a problem solving paradigm different from other major AI approaches because its approach is an incremental and sustained learning, since new experiences are retained each time a problem is solved making those available for future problems [AP94]. Being a multi-disciplinary subject, it can be interpreted by three distinct groups [Aha98]:

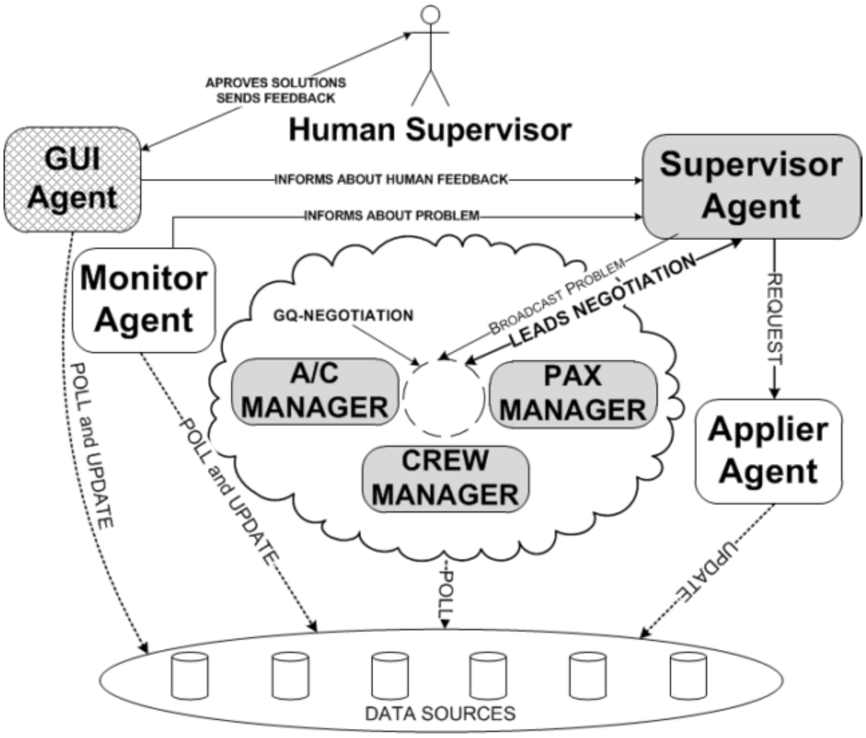


Figure 2.2: MASDIMA MAS Architecture. Source: [CRO12]

- Cognitive scientists: CBR is a plausible high-level model for cognitive processing [Kol14];
- Artificial intelligence researchers: CBR is a computational paradigm for problem solving [AP94];
- Expert system practitioners: CBR is a design model for expert systems that can be used in either stand-alone or embedded architectures [Wat98].

Its cycle is composed by four activities, Retrieve, Reuse, Revise and Retain, also called the four-REs [Wat99]:

- Retrieve similar cases to the problem description;
- Reuse a solution suggested by a similar case;
- Revise or adapt that solution to better fit the new problem if necessary;
- Retain the new solution once it has been confirmed or validated.

2.2.5 Softmax

Softmax action selection is the method that varies the action probabilities as a graded function of estimated value, i.e. the best action is given the highest selection probability, but all the others are ranked and weighted according to their value estimates, imposing a differentiation factor between better and worst actions, unlike ε-greedy action selection that chooses equally among all actions. This method commonly uses a Gibbs or Boltzmann distribution [SB98].

2.3 Electronic Markets

According to Troy et al. [SS97], EMs are the foundation of electronic commerce (e-commerce from now on). They define an EM (also known as electronic marketplace) as an inter-organizational information system that allows the participants (buyers and sellers) to exchange information about prices and information, products and services.

2.3.1 Electronic Commerce

With the Internet explosion in the late 1990s, the e-commerce became an area of interest. Nowadays it is commonly associated with online information, products and services transactions. E-commerce involves several layers of infrastructure (among many others it includes the messaging and information distribution infrastructure and common business services infrastructure). It also encompasses a wide range of applications (among others, marketing, payment systems and electronic markets) [SS97]. In this context, Wooldridge [Woo09] suggests that the use of the Web to commercial purposes (e-commerce) is limited due to:

- Trust

  Due to the lack of previous contact with some brands, it is difficult to consumers to know which vendors are reliable or not, in an online global market.

- Privacy and security

  The major issue is the concern that consumers have about the security of their personal information during the usage of e-commerce systems.

- Billing/Revenue

  As the Web was not designed with revenue models in mind, the billing mechanisms must be implemented over the Web's basic structure.

- Reliability

  Internet's unreliability (connections and data are frequently lost, and its performance is unpredictable) represents a massive obstacle to the usage of e-commerce.

2.4 Automated Negotiation

In Multi-Agent Systems it is required for an agent to interact with other agents whom may not share common goals. This leads to the need to reach agreements [Woo09] trough an automated negotiation process. The existent automated negotiation systems are composed by three major groups [OR01]:

- Auctions;
- Game Theory;
- *Negotiation*.

Although it only considers a single attribute, due to its simplicity and well predefined rules, auctions [VJ00] are a very popular negotiation mechanism.

Game Theory [RZ94] is a mechanism that can only be applied to perfect information and rationality contexts.

Negotiation is the generic name given to other techniques where agents must reach agreements on matters of mutual interest [Woo09]. These techniques are more flexible than auctions and game theory in terms of preexistent protocols and rules, thus more suitable for open and dynamic environments [OR01]. An overview over each one of these three groups is presented in the following subsections.

2.4.1 Auctions

Online auctions have become so popular because its interaction scenarios are extremely simple and that makes it easier to automate them. Although their simplicity seems attractive when comes to chose a way to reach agreements, auctions present both a rich collection of problems, concerning collusion and lies (to lower or inflate the price of the good under negotiation, respectively), and a powerful tool that automated agents can use for allocation goods, tasks and resources [Woo09]. Hereupon, an auction takes place between an agent known as auctioneer, and a group of agents, known as bidders. The more usual auction configurations, the auctioneer wants to maximize the price of the good under negotiation and the bidders want to minimize it. There are three dimensions in what an auction protocol may vary [Woo09]:

- Winner determination: who gets the good that bidders are bidding for;
- *Open cry* or *sealed-bid*: whether or not the bids made by the agents are known to each other;
- Bidding process: the mechanism by which bidding proceeds. It can be single rounded, ascending or descending auctions.

So, even considering the problems that may arise from this type of negotiation process, auctions are considered a powerful technique for allocating goods to agents [Woo09].

2.4.2 Game Theory

The negotiation process present in Game Theory is an iterative one, where agents exchange proposals, concerning global agreements, to fulfill its objectives [RZ94]. In every iteration an agent's proposal utility can not be higher than the previous iteration i.e., an agent has to cede over the negotiation process.

As this mechanism assumes two very limiting prerequisites, perfect information and rationality, it is not appropriated for all the scenarios.

For this mechanism to be used, it would required a context where the information would be equally understood by all participating agents and, where the participating agents are computationally unlimited and have complete information either about its own negotiation options or about other agents negotiation options (rationality).

2.4.3 Other Negotiation Techniques

Agent-based technology allows the implementation of more flexible mechanisms such as:

- Argumentation;
- Planning by Contracting;
- Heuristic Approximation;
- Multi-Attribute Negotiation;

An overview over each one of these mechanisms is presented in the following subsubsections.

2.4.3.1 Argumentation

To negotiate an agent needs to be capable of generate proposals and answer to the ones received. Proposals are possible solution to the current problem and can vary from a complete solution to a partial one or even to a set of solutions, either complete or partial ones. A proposal is generated through arguments based on agents intentions. Arguments are expressions with the purpose to change opponents (other agents) intentions, and consequently actions. If one agent is only capable to accept or refuse proposals, the negotiation may become extremely long (regarding a scenario with multiple rounds) or inefficient (regarding a scenario with only one round) because the proposer can not understand why its proposal has been rejected, if it is close to reach an agreement or not, or even what course it should follow in its solution space. To improve the efficiency of the negotiation process, the recipient needs to be able to provide more useful feedback on the proposals it receives, which can be [JFL+01]:

- Critique: comments on which parts of the proposal the agent agrees or not
  - Indication of restrictions to certain attributes;
  - Indication of acceptance or rejection of certain parts of the proposal.
- Counter-Proposal: an alternative proposal generated in response to the one received. The feedback is given in a less explicit way but usually with more detail that in critique case.

The feedback is useful for the proposer to generate a new proposal (if it chooses to do so) in the next round, moving towards a possible agreement.

As these two mechanisms, which form the basis for argumentation-based negotiation, are unfounded statements of what the agent intends to obtain, other builders have been developed:

- Agents provide meta-level information in the form of arguments to support its position. These arguments can be classified in three types [JFL$^+$01]:
  - Threat: failure to accept this proposal, by the opponent, means that something negative will happen to it;
  - Reward: acceptance of this proposal, by the opponent, means that something positive will happen to it;
  - Appeal: The opponent should prefer this proposal over that alternative or some reason.
- If an explanation [PSJ98] is attached to a proposal, counter-proposal or critique it is more probable that an agreement may be reached quickly. An explanation is additional information explaining why a proposal, counter-proposal or critique was made by an agent, i.e, an explanation is a form of justification that the agent supplies for its position. Including the feedback (explanation) on why certain proposal has been rejected, the agent can be helping the other part of the negotiation to focus it search to a more adequate solution.

2.4.3.2 Planning by Contracting

In the Planning by Contracting mechanism, the negotiation protocol may affect the strategic behaviour of the participants, particularly the plan formulation of the agent that started the negotiation [CJGM97]. In this mechanism the negotiation is an interaction process that does not support counter-proposals and is composed by three phases:

- Call-for-bids;
- Bidding:
- Bid Acceptance.

A customer agent issues an announcement, suppliers reply with bids and the customer ends by accepting the best proposal. After this process both parties are committed to the agreement and subject to the decommitement penalties. Penalties are specified by the customer and are design to be functions of time, the higher the later the contracted supplier decides not to comply with the agreement. The customer creates a plan to satisfy its goal and announces to the market the required subtasks. This announcement includes a task description, a bid deadline or the time by which the suppliers must respond, the time at which the customer will begin considering the bids, the earliest time at which bid acceptances will be sent and the penalty functions for each subtask. The customer collects the proposals issued by the suppliers and evaluate them. The goal is to find a task combination that minimizes the combination between cost and risk (risk factors include the expected cost of penalties associated with decommitement, the cost of plan failure and other factors) and allows the coverage of all tasks. If it is not possible to satisfy the initial plan or if the customer is not satisfied with the proposals received, it can change its plan and announce new tasks to be satisfied. In the end, it sends the correspondent contracts to the selected agents. During the negotiation process, interactions can become quite complex due to the fact that the hired suppliers can decommit (paying the respective penalty) and, in this case, customer needs to remake its plan.

2.4.3.3 Heuristic Approximation

This negotiation mechanism [JFL+01] assumes that agents have a set of negotiation tactics, being the proposals generated by a linear combination of these tactics. They define an agent negotiation behaviour in an heuristic form and are classified in three different classes [FSJB99]

- Time-dependent tactics: agents make its proposals according to the available time to negotiate;
- Resource-dependent tactics: agents make its proposals based on a certain resource availability;
- Behaviour-dependent tactics: agents try to duplicate its opponents behaviour.

2.4.3.4 Multi-Attribute Negotiation

It is common to be in the situation where the negotiation decision does not considers only one attribute but multiple attributes instead. For instance, when buying any product, the buyer considers the price as an important attribute in its decision but the delivery time or the product quality may also be (usually are) factors to be considered in the decision of buying or not a certain product.

Giving different utility values to the different attributes under negotiation solves the problem of multi-attribute evaluation. The most common proposal evaluation formula is a linear combination of the attribute correspondent values, weighted by the respective utilities. Therefore, a multi-attribute negotiation is converted to a single-attribute one, to be made over the evaluation value. The following examples use this conversion method of multi-attribute negotiation to single-attribute negotiation;

- [dOFSG99]: Vickrey auction is used in an automobile negotiation, with one buyer and multiple sellers. The attributes under negotiation are not only the price but also the maximum speed, number of doors, brand, etc.;
- [VJ00]: English auction is used in the negotiation of services characterized by multiple attributes;
- [MSJ98],[CO00]: An heuristic negotiation mechanism is used, where proposals and counter-proposals are the result of a weighted combination of different tactics.

All the referred works compels to the attribution of a numeric value to each attribute utility. However, in some cases it can be difficult to give an exact numeric value to an attribute utility which leads to a more intuitive situation that is just to impose a preferential order over the domain values for the different attributes or on the attributes itself.

2.5 Negotiation Analysis

In negotiation analysis there are two cases to consider [RRM02], Two-Party Distributive Negotiations and Two-Party Integrative Negotiations. A distributive negotiation means a negotiation (or a part of a larger one) concerned with the division of a single good. Distributive and Integrated negotiations are opposites, where distributive negotiation is about getting a bigger piece for oneself while integrative is about making the pie bigger. This means that a distributive negotiation is a Win-Lose negotiation while the integrative negotiation is a negotiation where joint gains are a potential result, which means that is a win-win negotiation. Other names for the same concepts are claiming and creating, for distributive and integrative respectively.

2.6 Summary

This chapter presented a literature review over the topics Airline Disruption Management, Agents and Multi-Agent Systems, Electronic Markets and Automated Negotiation. The review focused specially on describing all the constituents of the adopted methodologies used in this dissertation, trying to relate each topic with the others. It is possible to conclude that all the topics mentioned in this chapter fit together to develop the proposed solution as they are complementary to each other.

Chapter 3

Detailed Problem Description and Proposed Solution

This chapter presents an overview over the problem and some high level considerations over the proposed solution, which will be detailed in the next chapter.

3.1 Problem Description

In the Air Transportation domain, when a disruption occurs, it is necessary to take action. Depending on the type of disruption, it may be necessary to cancel the flight or just to find an alternative solution to reduce that flight's delay and the resultant cost. To find the optimal recovery plan, a reallocation of resources is needed and in some scenarios it is not possible due to the limited resources. As previously stated, each disruption is composed by four dimensions[dC13]: aircraft, crew member, passenger and flight. The AOCC tries to recover all dimensions affected through three different approaches[CRO14]: sequential, integrated and partial-integrated, but even with different approaches, whether imposing or not an importance factor, the major problem is the lack of available resources. For this reason, the solutions space is narrowed when compared to a scenario where companies have access to all other companies' resources, present in the same airport.

3.2 Proposed Solution

In this section an unique approach to help improving the disruption management problem is briefly described. The main goal is to expand the resources available for airline companies, i.e. to expand the solutions space which is narrowed because the resource considered for disruption management are the injured company own resources. The solution proposed is a Multi-Agent System based Electronic Market which, making use of the technologies mentioned, in chapter 2, will be defined to help companies searching for better solutions when a disruption occurs. Two types of agents will be developed to represent each type of company, agents to represent injured companies (buyer) and agents to represent resources provider companies (seller). The EM will allow to the companies to negotiate between themselves the resources to be leased so that the injured company applies a better solution, if one is found in the EM, and the solution provider company to get some profit with unused resources.

3.2.1 Negotiation Protocol

The negotiation protocol used in the EM, FIPA Iterated Contract Net, was chosen because it allows multi-round iterative bidding. This way, it is ensured that there is a wide solutions space where the best is selected. This protocol works with an initiator (also known as buyer in this dissertation's specific case) and multiple responders (sellers in the same context), where the initiator asks for a flight with specific attributes as scheduled departure and trip time (aircraft capacity also, if an aircraft is needed) and the responders answer with a pair of attributes, a flight's price and availability.

3.2.2 Electronic Market Working Flow

In the EM there are two different entities, buyer that represents an injured company or a service consumer company, and seller that represents a service provider company. Here, buyers and sellers will negotiate flights, which are composed by one crew and one aircraft, when needed. The negotiation is a process where proposals are exchanged and each one contains a flight's price and availability. Proposals were created to ensure that all private data is kept private, for instance, if sellers would know the buyer's disruption cost, their strategy would be to ask for a price slightly lower than that cost, making the market an unpractical alternative for buyer. The only restriction present in the market is that when the flight asked needs an aircraft, the seller must also provide a crew to handle it.

After explaining the market operating flow, it is time to introduce the negotiation process. For it to be possible sellers must register first, otherwise will be no one in the market to be asked for some resource(s). So, the first step for the market to work is to have multiple sellers registered and waiting for some buyer to register too. When a buyer registers in the market, it retrieves from the DFS a list with all other companies registered. Everything is now ready to start negotiating, so, buyer sends to all sellers a message containing the disrupted flight, giving crucial information as the scheduled departure time, triptime and delay, so sellers have a filter to exclude unavailable or unwanted resource(s). That first message also contains the flight data, whether is needed an aircraft or not, how many crew members from each rank, and its price and availability, that are set to 0. After sent the message buyer gives a timeout, for sellers to respond. After that timeout if a seller did not responded, it is removed from the negotiation. When a seller receives the first message, it processes the message and creates a proposal, with price and availability, and replies to buyer with it.

First round is concluded and until the end of the negotiation, all rounds are processed in the same method, explained as follows. Buyer receives one proposal for each seller that responded to the initiation message, evaluates them and creates a reply containing some feedback over the price and availability proposed. Then buyer selects the best proposal, if the current round is the second round, buyer just selects the best proposal but, if it is not, buyer compares all proposals with the best proposal received so far and if there is one that is better, that one is selected as the new best proposal, else, the best proposal remains unchanged, as explained in 4.2.5. So, when a seller receives the feedback over the last proposal made, it updates its experience record by updating the concerned proposal's evaluation. Then, it searches in the record for similar feedback and if there are any similar feedback present, seller selects one among all and replicates the action that was previously taken. If none experience was found, a new entry is added to the record, and seller processes the feedback, if possible adapting the proposal according to the feedback received, and sends a new proposal. If it is not possible to adapt the proposal to the feedback received, seller proposes another flight. When a seller does not have any more flights to propose, it sends a refusal message.

The negotiation is over when all sellers have sent a refusal message and then the market enters in the last round. In the last round, buyer sends an accept message to the best proposal's owner, with the accepted proposal data and a reject message to all others. As soon as a seller receives a reject message, it resets its structure and waits for a new buyer to register in the market. The seller that received the accept message searches for the flight corresponding to the one proposed and sends back to seller a termination message with all relevant data about the leased resource(s). Then, it updates its data set and resets its structure to wait for a new buyer to register in the market. Upon receiving the termination message, buyer unregisters himself from the market as the negotiation is over. The communication diagram on figure 3.1 presents a better comprehension about what has been explained.

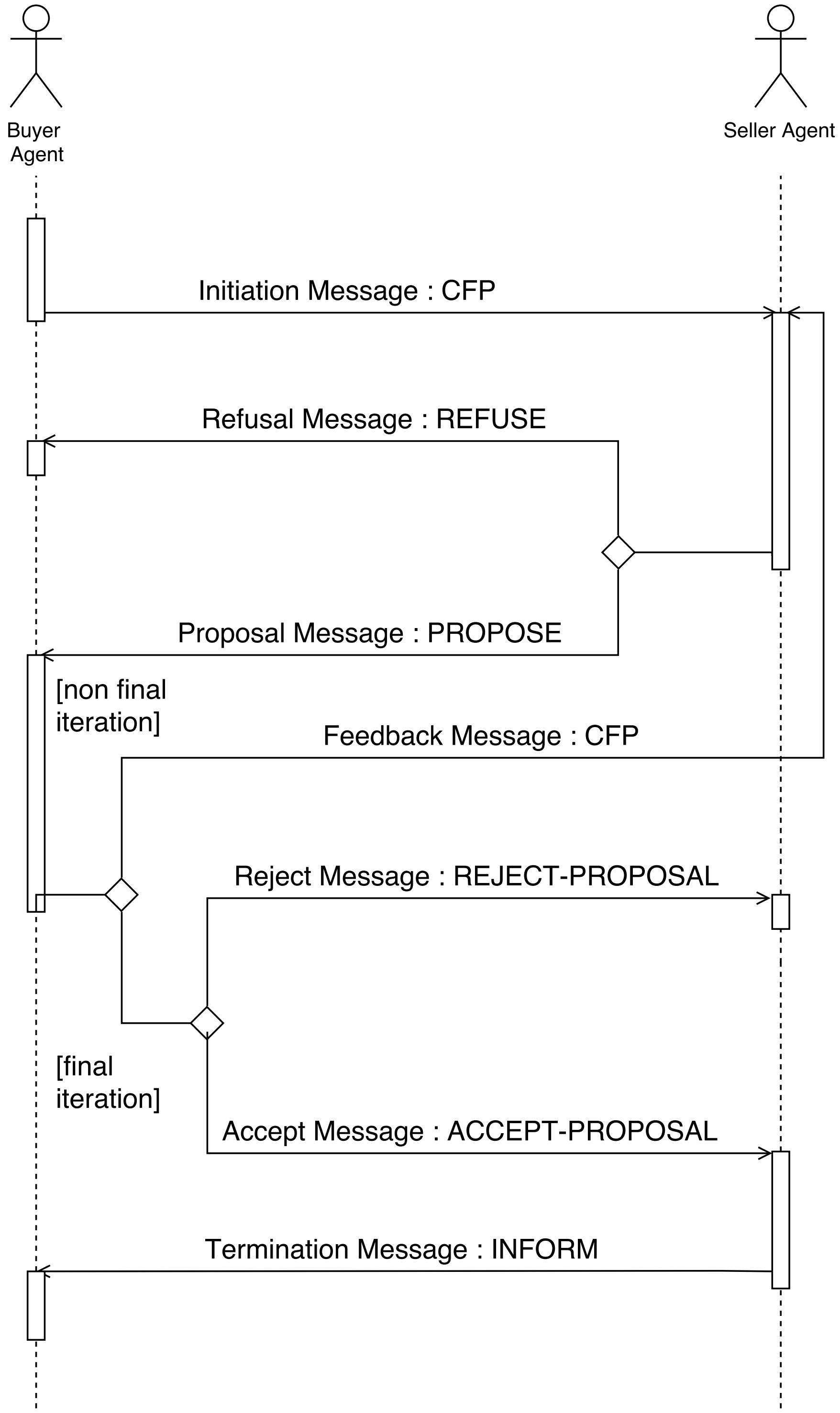


Figure 3.1: Electronic Market Communication Diagram

3.2.3 Directory Facilitator Service (DFS)

As previously stated, the EM will be implemented using JADE, which provides a Directory Facilitator Service or DFService or even DFS is a yellow pages service provider agent. It is used in this dissertation's context because it implements a set of static methods to communicate with a standard FIPA DF service. Among the available methods are used register, deregister and search actions, for agents to register and deregister on the EM and search to find service providers (sellers). These methods block every agent activity until the action is successfully executed or an exception (jade.domain.FIPAException) is thrown. There are cases where it is more convenient to execute these tasks in a non-blocking way, which JADE and DFS have the appropriate classes to handle these situations.

3.3 Summary

This chapter presented a detailed problem description and the solution's operating and workflow. The EM data structure and implementation will be detailed in the next chapter.

Chapter 4

Multi-Agent System based Electronic Market

4.1 Electronic Market

4.1.1 Flight: the negotiation object

Being the object under negotiation, a flight is composed by a list of crew members and an aircraft (resources). Aside from these two components, a flight is characterized by both availability and cost, which varies from flight to flight. The remaining flight's attributes are dates, origin, destination and fleet. The flight data structure is represented in table 4.1.

Table 4.1: Flight Data Structure

Attribute	Data Type	Unit	Explanation
Availability	Long	Millisecond	Worst availability between a flight's resources
Cost	Double	Euro	Leasing associated cost
Scheduled Departure	Long	Millisecond	Date planned for the disrupted flight
Triptime	Long	Millisecond	Flight duration
Delay	Long	Millisecond	Disrupted flight delay
Fleet	String	-	Disrupted flight fleet
Origin	String	-	Origin city
Destination	String	-	Destination city

4.1.1.1 Crew Member

Crew Member represents one of the resources available in the market and are identified by their rank and availability, being the first identification criterion the tank, and within the same rank, the criterion is the availability. Alongside with the already mentioned attributes, a crew member also has a number, status, seniority, identification and hourly salary. The crew member data structure is represented in table 4.2.

Table 4.2: Crew Member Data Structure

Attribute	Data Type	Unit	Explanation
Rank	String	-	Crew member rank
Availability	Long	Millisecond	Crew member availability
Crew member id	Double	-	Crew member identification
Seniority	Integer	-	Crew member seniority within its rank
Status	String	-	Crew member status
Number	Integer	-	# crew members needed from certain rank
Hourly salary	Integer	Euro	Crew member hourly salary

4.1.1.2 Aircraft

Aircraft represents the other type of resources available in the market. As in both flights and crew members, aircraft is also identified by its availability and cost. It also has the rank and number of crew members needed to handle the aircraft, the aircraft identification and capacity. The aircraft data structure is represented in table 4.3.

Table 4.3: Aircraft Data Structure

Attribute	Data Type	Unit	Explanation
Availability	Long	Millisecond	Aircraft availability
Capacity	Integer	-	Aircraft capacity (number of seats)
Tail Number	String	-	Aircraft identification
Nautical mile cost	Double	Euro	Cost per nautical mile traveled
Maintenance cost	Double	Euro	Maintenance average cost per flight minute
Fuel Cost	Double	Euro	Fuel average cost per flight minute
Crew members needed from rank	Integer	-	# crew members needed from certain rank

Aircraft has five different crew member ranks, but only one is illustrated in table 4.3 to keep it simpler.

4.1.1.3 Availability

Each resource's main attributes are price and availability, which identify them as alternatives to the disrupted flight. A resource's availability is comprehend between 0 and a maximum value, delay. An availability expressed in 0 ms, stands for a resource that make up for the delay that buyer wants to minimize, while an availability different from 0 ms, stands for a delayed resource when compared to the planned scheduled departure, but still not as delayed as the original delay. So a resource's availability is comprehended in a interval as represented in the equation (4.1).

$$\forall \alpha : \ \alpha \in [0 \ , \ \lambda[\tag{4.1}$$

where:

- α is the resource's availability;
- λ is the delay to be minimized.

As availability is represented as a value instead of a range of values, it is immutable, this is, it can not be changed. So, a resource is only available from its availability value. It is represented in milliseconds, counting from 1970-01-01 at 00:00:00, to make easier to make operations with dates. So, the day 1992-04-15 at 11:30:00 is represented by 703337400000 ms.

4.1.2 Proposal: The negotiation data

The proposal object is the courier between buyer and sellers, containing only the current price and availability. A proposal also has the feedback over the attributes being negotiated (availability and price). The proposal data structure is represented in table 4.4.

Table 4.4: Proposal Data Structure

Attribute	Data Type	Unit	Explanation
Availability	Long	Millisecond	Flight under negotiation availability
Price	Double	Euro	Flight under negotiation price
Availability comment	String	-	Buyer feedback over last proposal's availability
Price comment	String	-	Buyer feedback over last proposal's price
Flight under negotiation	Flight	-	Flight under negotiation at the current round
Sender	AID	-	Sender identification

The main purpose of a proposal is to ensure that all private data is kept that way, so, the only scenarios where a proposal contains a flight is when buyer agent asks to all other agents in the EM for similar resources, and when seller informs buyer of the agreed resources to be leased.

4.1.3 Communication

During each negotiation, agents exchange messages with various purposes, represented in table 4.5.

Excluding the refusal and reject cases, all messages contain a proposal. In these two specific cases, there is no need to attach a proposal to the message, as they are merely informative messages. That said, the initiation message, contains a proposal with the resource or resources missing, setting a default value for price and availability (0 for each one). The proposal message

Table 4.5: Messages, their functions and their contents

Message Function	ACL Message		
	Performative	**Content**	**Sender**
Initiation	CFP	Resource(s) Missing	Buyer
Proposal	PROPOSE	Proposal	Seller
Feedback	CFP	Feedback over last proposal	Buyer
Refusal	REFUSE	None	Seller
Reject	REJECT-PROPOSAL	None	Buyer
Accept	ACCEPT-PROPOSAL	Proposal accepted	Buyer
Termination	INFORM	Resource(s) details	Seller

contains a resource or resources' price and availability. The feedback message contains the proposal received with some feedback over price and availability. As previously stated, both refusal and reject messages have no content, the first to inform buyer that the seller is not available to negotiate any further and the second to inform the seller that all of his proposals have been rejected. Finally, the accept message contains the accepted proposal, so the seller learns what resource(s) he must lease. The termination message contains all data regarding the resource(s) to be leased, so buyer knows what specific resources it is going to receive.

4.1.4 Technical Issues

Java Database Connectivity (JDBC) is an API (application programming interface) that defines how a client may access a database. It is an API for Java programming language and in the specific case of this dissertation, it is used for each agent to access to its database, whether to load or save data. A local server was created, for testing purposes, using MySQL Workbench.

Date4j is an alternative library for JDK (Java Development Kit) date classes that makes dates easier to be manipulated among other features. It is used in this dissertation's context due to previous experience with the library and because it makes simpler to convert dates to milliseconds, the availability's unit measure.

4.2 Buyer Agent

4.2.1 Disruption

A disruption can be viewed as a flight that for some reason cannot meet its planned schedule, and needs to be replaced or compensated. In order to do so, the injured company (hereinafter buyer) needs to collect some data from current disruption to help the other companies registered in the market (hereinafter sellers) reducing unfeasible or unavailable resources. Each disruption is identified by a code (hereinafter resource affected), which is unique, and the most important attributes, scheduled time of departure, triptime, departure delay and resource needed, which are used to detail the disruption. The resource needed attribute can be one of three, adding a few more attributes to be considered:

- Aircraft: in this case, a crew must be leased as well, from the same company that leased the aircraft.

- Complete crew;

- Crew member;

The extra attributes to be considered in each scenario are aircraft capacity in the first, crew members number from each rank in the second, crew members number from the rank asked in the third. The costs of each resource (aircraft or crew) disruption are also known and they are used to set the maximum limit that buyer is willing to spend. When the required resource is a crew member, the buyer considers the maximum cost to be 1.5 of its own crew member salary, where the selected crew member is the buyer's crew member from that rank with the highest salary.

4.2.2 Buyer Data Structure

The buyer data structure is represented in table 4.6.

Table 4.6: Buyer Data Structure

Attribute	Data Type	Unit	Explanation
Best Proposal	Proposal	-	Best proposal received at the moment
Disrupted Flight	Flight	-	Flight under negotiation price
Maximum Disruption Cost	Double	Euro	Disruption total cost
Aircraft Disruption Cost	Double	Euro	Disrupted aircraft cost
Crew Disruption Cost	Double	Euro	Disrupted crew/crew member cost
Round	Integer	-	Round number
Sellers	ArrayList	-	List of all sellers registered in the EM
Resource Affected	String	-	Disrupted resource type

4.2.3 Buyer Life Cycle

As the buyer is launched, it retrieves the disruption from the data set, identified by the affected resource, and then registers itself in the DFS and creates a list of potential providers, with all agents registered in the service (DFS). Then, it prepares the initiation message, detailed in 4.1.3, and sends it to all sellers in the potential providers list. The negotiation ends when there are no more sellers in the potential providers list, and the buyer notifies the seller which is the best proposal's owner that its proposal is the one chosen and that an agreement has been reached, rejecting all other proposals. Buyer life cycle is illustrated in algorithm 1.

Each round consists in processing all the messages received, which implies to compare all the proposals received with the best proposal until the moment, updates it if necessary, and gives some feedback over each proposal received to enable sellers to improve it. After each round (after buyer sends a message) it waits a short timeout and then processes all the responses received and if the

Algorithm 1: Buyer Agent life cycle

Input: Disruption
Output: Solution
missingResources := findMissingResources(disruption);
bestProposal := null;
firstProposal := createProposal(missingResources);
sendFirstCFP(firstProposal);
while *number active sellers > 0* **do**
    feedbackMessages := handleAllResponses(messagesReceived);
    newRound(feedbackMessages);
end
lastRound(feedbackMessages);
Solution := bestProposal;

number of responses is less than the number of sellers in the potential providers list, it removes them from the list proceeds. The message processing algorithm is presented in algorithm 2.

After receiving all messages, buyer iterates over the responses array. For each proposal received, buyer evaluates it and creates a new proposal with some feedback. The feedback algorithm and its explanation are detailed later. Although the feedback given to each proposal is not a mere comparison with the best proposal received, all proposal's are still compared to the best proposal through proposal's utility and, if there is one with better utility, it is set as new best proposal. If a proposal's utility is equal to best proposal's utility, the tie break criterion is the price, the lower the price, the best so, in this scenario, the proposal with lower price is the best proposal (leave it unchanged if it already is the best proposal or update it if it is not). When there are no more active sellers, buyer sets all messages to reject except the one that was sent by best proposal's owner, that is set to accept. After receiving an inform message, containing all data over the resources leased, buyer unregisters himself from the DFS.

4.2.4 Buyer Proposal Evaluation

As previously mentioned in 2.2.1, concerning an agent's preferences set, that agent's utility is a way of representing its preferences, i.e., the higher the utility, more preferred is the outcome or state. Considering this, in this dissertation specific case, a proposal is evaluated through its utility, which is represented in equation (4.2).

$$\forall \mu : \ \mu \in [0, 1] \tag{4.2}$$

Where:

- μ is the proposal's utility;

One proposal's utility is measured in order to its availability and price, where buyer tries to minimize both (availability's variation has been explained before in 4.1.1.3). The utility of each proposal is given by equation (4.3).

Algorithm 2: Handle All Responses Algorithm

```
Input: Messages Received
Output: Feedback
if messagesReceived.size() < number of active sellers then
    number of active sellers := messagesReceived.size();
end
index := 0;
accept := null;
while index < messagesReceived.size() do
    message := getMessage(messagesReceived,index);
    if number of active sellers = 0 then
        accept := message.createReply();
        accept.setContent(bestProposal);
        Feedback.add(accept);
        setRejects(Feedback);
    end
    if message.getPerformative() = PROPOSE then
        reply := message.createReply();
        Feply.setPerformative(CFP);
        feedback.add(reply);
        proposal := convertMsgToProposal(message);
        compareWithBestProposal(proposal);
    end
    if message.getPerformative() = REFUSE then
        number of active sellers := number of active sellers - 1 ;
        removeSeller(index, messagesReceived);
    end
    index++;
end
if accept = null then
    Feedback := setFeedback(messagesReceived);
end
```

$$\mu = \mu_\alpha \times \beta + \mu_\rho \times (1 - \beta) \tag{4.3}$$

where:

- μ is the proposal's utility;
- μ_α is the proposal availability's utility;
- μ_ρ is the proposal price's utility;
- β is availability's importance factor;

With:

$$\forall \beta : \ \beta \in [0,1] \tag{4.4}$$

where:

- β is availability's importance factor.

By default, both parameters (availability and price) are equally valued, so the default formula is:

$$\mu = \mu_\alpha \times 0.5 + \mu_\rho \times 0.5 \tag{4.5}$$

where:

- μ is the proposal's utility;
- μ_α is the proposal availability's utility;
- μ_ρ is the proposal price's utility;

As shown above in equations (4.3) and (4.5), a proposal's utility is composed by availability's utility (μ_α) and price's utility (μ_p). As previously stated in equation (4.1), availability may vary from 0 ms to delay, which implies that when availability is equal to delay, the availability's utility should be 0, because that is the worst scenario that can be proposed while the best possible scenario is when the proposed availability is equal to 0, and then, the availability's utility should be 1. The main purpose of availability's utility is to relate the proposed availability with the delay but as previously stated, when availability and delay are equals, the utility should be 0, but the quotient between both is 1. This is ensured as demonstrated in equations (4.6), (4.7) and (4.8).

$$\lim_{\alpha \to \lambda} \frac{\alpha}{\lambda} = 1 \implies 1 - \frac{\alpha}{\lambda} = 0 \tag{4.6}$$

And so, the availability's utility is expressed by:

$$\mu_\alpha = 1 - \frac{\alpha}{\lambda} \tag{4.7}$$

where:

- μ_α is the proposal availability's utility;
- α is the proposed availability;
- λ is the delay to be minimized.

As:

$$\forall \alpha, \lambda : \frac{\alpha}{\lambda} \in [0,1] \implies (1 - \frac{\alpha}{\lambda}) \in [0,1] \implies \mu_\alpha \in [0,1] \tag{4.8}$$

The same principle is applied to price's utility, which means, the lower the price, the best the utility, as equations (4.9), (4.10) and (4.11) show.

$$\lim_{\rho \to \tau} \frac{\rho}{\tau} = 1 \implies 1 - \frac{\rho}{\tau} = 0 \tag{4.9}$$

And so, the price's utility is expressed by:

$$\mu_\rho = 1 - \frac{\rho}{\tau} \tag{4.10}$$

where:

- μ_ρ is the proposal price's utility;
- ρ is the proposed price;
- τ is the disruption cost.

As:

$$\forall \rho, \tau : \frac{\rho}{\tau} \in [0,1] \implies (1 - \frac{\rho}{\tau}) \in [0,1] \implies \mu_\rho \in [0,1] \tag{4.11}$$

So, a proposal's utility is better as availability and price proposed are lower. Each attribute contribution to a general utility is calculated and then each attribute contribution is multiplied by its importance factor. A proposal's utility is the sum of both attributes contribution.

4.2.5 Proposal's Feedback

A proposal's feedback is composed by two comments over the attributes being negotiated (price and availability) and has three qualitative options for each attribute:

- OK: which means that there is no need to improve the attribute that received this feedback;
- LOWER: which means that the attribute that received this feedback needs a little improvement;
- MUCH LOWER: which means that the attribute that received this feedback needs a lot of improvement.

Availability is the first attribute to get its feedback, due to the fact that price is written in order to it. Availability is directly related to the delay reduction, as shown in table 4.7.

Table 4.7: Availability Feedback

Delay Reduction Percentage	Feedback
81% - 100%	OK
61% - 80%	LOWER
1% - 60%	MUCH LOWER

The criterion to decide what feedback to apply to a proposal's availability depends on the delay reduction percentage, i.e., calculate the delay reduction percentage using the proposal's availability in order to understand how much the delay would be reduced. The delay reduction percentage is presented in equation (4.12) and its range in equation (4.13).

$$\theta = \frac{\alpha}{\lambda} \tag{4.12}$$

and:

$$\forall \theta \in [0,1] \tag{4.13}$$

where:

- $\theta$is the delay reduction percentage;
- α is the availability proposed;
- λ is the delay to be minimized.

As illustrated in table 4.7, if the delay would have a reduction between 81% and 100% (a 100% delay reduction is represented by an availability of 0 ms and that the delay is completely reduced) the feedback on that proposal's availability would be OK. For instance, if one resource had a delay of 40 minutes and one of the proposals received would reduce the delay to 5 minutes (a delay reduction of 87.5%), the feedback on that proposal's availability would be OK, which means that there is no need to improve/change that proposal's availability. If the delay would have a reduction between 61% and 80% the feedback on that proposal's availability would be LOWER, and the feedback would be MUCH LOWER if the delay would have a reduction between 1% and 60%.

The feedback on a proposal's price is more complex than the availability's feedback. The price's feedback is dependent on availability's feedback because in the presence of a better solution (better availability) buyer is willing to pay more than for a worse solution (worse availability). So, table 4.8 shows what percentage of buyer's disruption cost, it is willing to pay for an EM solution based on proposal's availability feedback. This way, the solution obtained through the EM is not as expensive as the disruption itself and it is ensured that it only accepts proposals that are more cost-effective in his perspective or else the EM would not be an advantageous alternative. For example, the proposal's availability is LOWER. If the proposal's price is between 49% and 70% of buyer's disruption cost, the price's feedback will be LOWER as well.

Table 4.8: Price Feedback

Availability Feedback	**EM Tolerable Cost**	
	Disruption Cost Percentage	**Feedback**
OK	1% - 69%	OK
OK	70% - 100%	LOWER
LOWER	1% - 44%	OK
LOWER	45% - 79%	LOWER
LOWER	80% - 100%	MUCH LOWER
MUCH LOWER	1% - 14%	OK
MUCH LOWER	15% - 44%	LOWER
MUCH LOWER	45% - 100%	MUCH LOWER

As happened with availability attribute, the disruption cost percentage that buyer is willing to pay for an EM solution is calculated so buyer can understand where the price is situated and to know how to evaluate it in order to resource's availability. The price reduction percentage is presented in equation (4.14) and its range in equation (4.15).

$$\varphi = \frac{\rho}{\tau} \tag{4.14}$$

and:

$$\forall \varphi \in [0, 1] \tag{4.15}$$

where:

- $\phi$is the disruption cost percentage;
- ρ is the price proposed;
- τ is the disruption cost.

To a better comprehension on how the feedback on availability and price works, the proposal feedback algorithm is presented in algorithm 3.

After evaluating all proposals in responses array, buyer initiates a new round, prepares a feedback message for each seller, containing their last proposal and its feedback, sending it to respective sellers. In order to understand how a proposal is evaluated, the proposal evaluation algorithm is presented in algorithm 4.

4.3 Seller Agent

4.3.1 Seller Data Structure

The Seller data structure is represented in table 4.9.

Algorithm 3: Set Proposals Feedback Algorithm

```
Input: Messages Received, Feedback to Send
foreach message in Messages Received do
    proposal := convertMsgToProposal(message);
    feedbackMessage := feedback.getMsgBySender(message.getSender());
    proposalWithFeedback := proposal;
    pricePercent := proposedPrice/maximumCost;
    availabilityPercent := proposedAvailability/delay;
    if 0.6 <= availabilityPercent < 1.1 then
        proposalWithFeedback.setAvailabilityFeedback(MUCH LOWER);
        if 0 <= pricePercent < 0.15 then
            proposalWithFeedback.setPriceFeedback(OK);
        else if 0.15 <= pricePercent < 0.45 then
            proposalWithFeedback.setPriceFeedback(LOWER);
        else if 0.45 <= pricePercent < 1.1 then
            proposalWithFeedback.setPriceFeedback(MUCH LOWER);
    else if 0.2 <= availabilityPercent < 0.60 then
        proposalWithFeedback.setAvailabilityFeedback(LOWER);
        if 0 <= pricePercent < 0.45 then
            proposalWithFeedback.setPriceFeedback(OK);
        else if 0.45 <= pricePercent < 0.80 then
            proposalWithFeedback.setPriceFeedback(LOWER);
        else if 0.8 <= pricePercent < 1.1 then
            proposalWithFeedback.setPriceFeedback(MUCH LOWER);
    else if 0 <= availabilityPercent < 0.20 then
        proposalWithFeedback.setAvailabilityFeedback(OK);
        if 0 <= pricePercent < 0.70 then
            proposalWithFeedback.setPriceFeedback(OK);
        else if 0.70 <= pricePercent < 1.1 then
            proposalWithFeedback.setPriceFeedback(LOWER);
    end
    feedbackMessage.setContent(proposalWithFeedback);
end
```

4.3.2 Price Establishment

As stated in 4.1.1.3, the main attributes of the resource under negotiation are availability and price. Availability has already already explained and works the same way for both agents. The same principle applied to one resource's availability also applies to the same resource's price, which is comprehended between 150% and 300% of its leasing associated cost, as presented in equation (4.16).

$$\forall \rho : \ \rho \in [1.5 \times \zeta, \ 3 \times \zeta] \tag{4.16}$$

Algorithm 4: Evaluate Proposal Algorithm

```
Input: Proposal
if bestProposal = null then
    bestProposal := Proposal;
else
    bestProposalUtility := utilityCalc(bestProposal);
    currProposalUtility := utilityCalc(Proposal);
    if currProposalUtility > bestProposalUtility then
        bestProposal := Proposal;
    else if currProposalUtility = bestProposalUtility then
        if Proposal.price < bestProposal.price then
            bestProposal := Proposal;
end
```

where:

- ρ is the resource's price;
- ζ is the resource leasing associated cost.

Forcing the minimum price to be 150% of the leasing associated cost, ensures that sellers always have profit (otherwise the market would not be viable). For this reason price is calculated in order to a resource's availability, according to the following specifications. Considering an availability of 0 ms as the best possible scenario, it will be the most expensive as well. Regarding the worst possible scenario, the limit of availability as it tends to delay is the delay itself, and the price will be the cheapest accordingly, as figure 4.1 shows.

Figure 4.1: Price variation/Availability relation

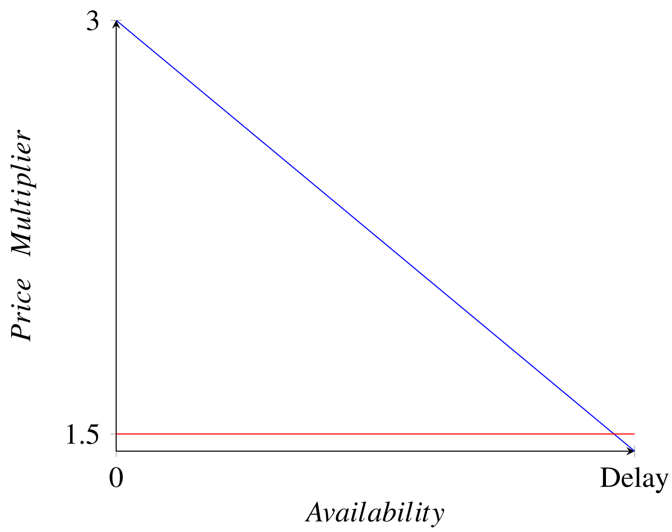


Table 4.9: Seller Data Structure

Attribute	Data Type	Unit	Explanation
Flights List	List	-	List with all available flights to be leased
Negotiation Historic	HashMap	-	Structure to save each round's proposal
Participants	List	-	List of all sellers registered in the EM
Experiences Record	String	-	Experiences record file name
Round	Integer	-	Round number

Assuming this, price and availability are inversely proportional and knowing that the price multiplier when availability is 0 ms, is 3 and 1.5 when availability is close to delay, the price variation in order to availability is expressed by a factor represented by the blue line segment, which is given by equation (4.17).

$$\xi = \frac{-\gamma}{\lambda} \times \alpha + \sigma \tag{4.17}$$

where:

- ξ is the price multiplier;
- γ is the minimum price multiplier;
- λ is the delay;
- α is the availability;
- σ is the maximum price multiplier.

Replacing with the minimum and maximum multiplier factors given in (4.16), the price factor is given by equation (4.18).

$$\xi = \frac{-1.5}{\lambda} \times \alpha + 3 \tag{4.18}$$

where:

- ξ is the price multiplier;
- λ is the delay;
- α is the availability;

So, the price asked in the market is the product between the price multiplier, obtained through the resource availability, explained in equation (4.18), and the resource leasing associated cost, as is shown in equation (4.19)

$$\rho = \xi \times \omega; \tag{4.19}$$

where:

- ρ is the price to be proposed;
- ξ is the price multiplier;
- ω is leasing associated cost;

The leasing associated cost is the cost that includes all costs that derive from the leasing, this is, average cost per nautical miles traveled, average maintenance cost per minute, average fuel cost per minute and airport handling cost. This cost is applied when an aircraft is needed these extra costs are calculated for the entire trip but when there is no need for an aircraft it is not necessary to apply this cost and the cost for the lessor is only the crew or crew member salary.

4.3.3 Seller Proposal Evaluation

As stated in 4.2.4, the measure of how preferable a proposal is given, in the case of seller, exclusively by its price, contradictory to what happens with buyer agents. This is materialized in the fact that seller does not need to fulfill any disrupted schedule but it just has to compensate the leasing associated cost. As the proposal is more profitable as the higher is the price, seller utility is simply a formalization given by equation (4.20).

$$\mu = \frac{\rho - \gamma \times \zeta}{(\sigma \times \zeta) - (\gamma \times \zeta)} \tag{4.20}$$

Where:

- μ is the proposal's utility;
- ρ is the price calculated;
- γ is the minimum price multiplier;
- σ is the maximum price multiplier;
- ζ is the leasing associated cost.

Replacing the price multipliers, that are already known, the seller utility is simplified as shown in equation (4.21) and its range in equation (4.22).

$$\mu = \frac{\rho - (1.5 \times \zeta)}{(3 \times \zeta) - (1.5 \times \zeta)} \tag{4.21}$$

Where:

- μ is the proposal's utility;
- ρ is the price calculated;
- ζ is the leasing associated cost.

with:

$$\forall \mu : \ \mu \in [0,1] \tag{4.22}$$

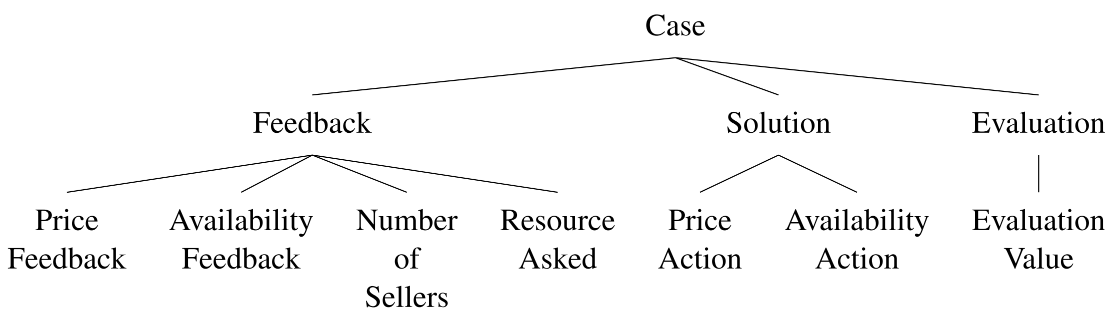
Where:

- μ is the proposal's utility;

4.3.4 Case-based Reasoning

The use of Case-based Reasoning helps sellers to decide what to do upon receiving some feedback over the proposal sent, consulting a record of previous experiences that have been classified according to its usefulness. The object that allows similar experiences identification is called case and it is represented by a list of strings. This means that an agent's record is an aggregation of all of its experiences, which is, an aggregation of lists of strings, each one representing one experience, with one exception. Every record's first line is common among agents, being an header with all the attributes' names for a better understanding of the record file. A case is further detailed as shown in figure 4.2.

Figure 4.2: Case Composition



CBR starts by retrieving similar cases to the one received. For this purpose, only features are used to compare cases and to identify equal ones. Although all features are used, they do not have the same preponderance on the task, because the feedback over a proposal is the most important due to its relevance on the action to be made and because the same feedback values may have a wide range of actions. So, to each feature is given a weight, as table 4.10 shows.

Similar cases are found trough the euclidean distance between them, a distance of 0 means that the case is identical, a distance greater than 0 means a different case. Being the Euclidean distance between points p and q is the length of the line segment connecting them, given by equation (4.23).

$$d(p,q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \tag{4.23}$$

Table 4.10: Feature's weights

Features	Weight
Price Feedback	10
Availability Feedback	10
Number of Sellers	2
Resource Asked	4

To ensure that features' weight has relevance in the distance calculation, the weight was added to the euclidean distance formula. So, by definition, the distance between two cases is the sum of the product between their features distances and the feature weight, as presented in equation (4.24).

$$d(\kappa,\chi) = \sum_{\eta=1}^{\psi} \sqrt{(\kappa\eta - \chi\eta)^2} \times \varepsilon \tag{4.24}$$

Where:

- κ is the case received;
- χ is one of the case in the data set;
- η is the current feature;
- ψ is the total number of features;
- ε is the current feature's weight;

Considering that most features are represented as text, a conversion was made, so the distance could be calculated. This conversion is explained in table 4.11.

Table 4.11: Case Conversion

Features	Data Type	Value	Conversion
Feedback	String	{OK, LOWER, MUCH LOWER}	{5,6,7}
Number of Sellers	Integer	[1. ∞[	-
Resource Asked	String	{Aircraft, Crew, Crew Member}	{0,1,2}

The algorithm compares the case received with all cases in the data set and returns a list with the respective lines indexes in the data set, where each line index identifies a similar case to the one received, as presented by algorithm 5.

To make possible the match between cases with different resources, when the resource asked attribute is empty, only the feedback is considered as feature to fetch for similar cases. When a new case is created, the default evaluation value is -1 because its value is comprehended in the interval specified in equation (4.25) .

$$\forall \Upsilon : \ \Upsilon \in [0,1] \tag{4.25}$$

Algorithm 5: CBR Algorithm

```
Input: Case Received
Output: Similar Cases
foreach case in DataSet do
    if euclideanDistance(caseReceived,case) = 0 then
        similarCases.add(case.getLineIndex());
    end
end
return similarCases;
```

Where:

- Υ is the evaluation value.

The latter the case, more important is its evaluation, due to the fact that over time, the EM paradigm may change, and to prevent obsolete cases, every time a experience is reproduced, its evaluation is updated as equation (4.26) shows.

$$\Upsilon = \Upsilon_{n-1} * 0.2 + \Upsilon_n * 0.8 \tag{4.26}$$

Where:

- Υ is the final evaluation value;
- Υ_{n-1} is the evaluation value read from record;
- Υ_n is the evaluation value for present round.

If there are no previous cases in the data set, the evaluation is given by the difference between feedback over previous round proposal and current round proposal, as presented in equation (4.27).

$$\Upsilon = \Delta\rho_{feedback} + \Delta\alpha_{feedback} \tag{4.27}$$

where:

- Υ is the final evaluation value;
- $\Delta\rho_{feedback}$ is the price's feedback variation;
- $\Delta\alpha_{feedback}$ is the availability's feedback variation.

In order make this possible, feedback is converted to integers and then subtracted, as table 4.12 presents.

Then, if the difference is greater than 0, evaluation is incremented by 0.5 for each attribute. This means that in the worst scenario, where feedback remains unchanged, evaluation is 0. If only one of feedback's values changed, evaluation is set to 0.5 and if both changed, best scenario, evaluation is set to 1.

Table 4.12: Feedback Conversion

Feedback	Integer
OK	0
LOWER	1
MUCH LOWER	2

4.3.5 Softmax

To help in the decision of what case to apply after the compilation of similar cases, seller calls softmax. This algorithm applies a probability to each similar case retrieved from CBR, where a case's probability is higher, the higher its classification value. So, a case probability to be select is given by equation (4.28).

$$P(\aleph) = \frac{e^{\aleph}}{\sum_{i=1}^{\varpi} e^{i}} \tag{4.28}$$

Where:

- $\aleph$ is the evaluation of current case;
- $P(\aleph)$ is probability of the item with $\aleph$ evaluation being selected;
- ϖ is the number of cases in the list.

In this case, the factor that resolves a case probability is it's evaluation value. After being assigned a probability to each case, a random value is generated and is selected the first case with cumulative probability greater than that random value. Algorithm 6 shows what is explained above for a better comprehension.

Algorithm 6: Softmax Algorithm

```
Input: Similar Cases
Output: Selected Case
probabilitiesList := new List;
foreach case in Similar Cases do
    probabilitiesList.add(softmax(case));
end
p := random;
cumulativeProbability := 0;
index := 0;
foreach probability in probabilitiesList do
    cumulativeProbability += probability.get(index);
    if p <= cumulativeProbability then
        return similarCases.get(index);
    end
end
```

4.3.6 Using Case-Based Reasoning

As soon as seller receives a feedback message, it will consult its record, searching for similar experiences. As explained in 4.3.4, all features must be equal in order to one case be considered similar to another one, so that is the first step. Similar experiences are found using CBR as presented in algorithm 7.

Algorithm 7: Find Similar Experiences Algorithm

```
Input: Proposal, Experience
if round > 1 then
    prevProposal := loadPreviousProposal(round);
    prevProposalEvaluation := evaluate(prevProposal,proposal);
    if recordEntries > 1 then
        similarExperiences := euclideanDistance(experience);
        if similarExperiences.size() > 1 then
            return similarExperiences;
        else
            noEqualCaseFound := -1;
            return noEqualCaseFound;
        end
    else
        recordIsEmpty := -2;
        return recordIsEmpty;
    end
end
```

If there are similar experiences present in the record, the seller loads all of them and uses softmax to select one to be applied to the current scenario. This scenario is represented by algorithm 8.

Algorithm 8: Process Similar Experiences Algorithm Part 1

```
Input: Experience
similarExperiences := findSimilarExperiences(experience);
if similarCases.size() > 1 then
    similarExperience := softmax(similarExperiences);
    foundEqualExperience := true;
else
    similarExperience := similarExperiences.getFirst();
    if similarExperience != -1 && similarExperience != -2 then
        foundEqualExperience := true;
    end
end
```

If there are no similar experiences in the seller's record, it will consult its record again but this time the search includes different parameters. The feature considered this time is only the feedback, so it can learn from experiences with other resources. If it finds experiences with the same feedback, it will load them and again, use softmax to select on to be applied to the current scenario, presented by algorithm 9.

Algorithm 9: Process Similar Experiences Algorithm Part 2

```
else if similarCases.size() = 1 && similarCases.getFirst() = -1 then
    resourceAffected := experience.getResource();
    experience.deleteResource();
    similarExperiences := findSimilarExperiences(experience);
    if similarCases.size() > 1 then
        similarExperience := softmax(similarExperiences);
        experience := similarExperience;
        experience.setResource(resourceAffected);
        experience.setEvaluation(-1);
        addToRecord(experience);
    end
end
```

A third hypothesis is that its record is empty and in this situation seller processes the feedback message (detailed in the next subsection) and creates an experience to add to its record. This scenario is presented by algorithm 10.

Algorithm 10: Process Similar Experiences Algorithm Part 3

```
else
    addToRecord(experience);
end
```

4.3.7 Handle Feedback

As stated in 4.1.1.3, availability is an immutable attribute, so, seller handles with availability's feedback in an extreme way. If it is asked to improve (lower) a resource's availability, seller will try to improve (lower) its price and if it is not possible, it will change the resource proposed. The price cannot be too low as explained in 4.3.2, so seller tries to decrease it between 10% and 25%, according to the feedback received and the proposal's utility with the new price. Table 4.13 illustrates the criterion of this price variation.

Table 4.13: Seller Feedback Processing

Feedback	Price Reduction	Minimum Acceptable Utility	Maximum Acceptable Utility
LOWER	10%	0.40	0.79
MUCH LOWER	25%	0.10	0.39

If, in each case, the reduced price does not fit into the acceptable utility interval or is lower than the minimum price established, the price is set to the minimum, explained in 4.3.2, to force that proposal's utility to be 0, explained in 4.3.3, and that way seller changes the resource to be proposed. This happens when the utility of the next resource in the list of resources is greater than the utility of the current resource being negotiated.

4.3.8 Seller Life Cycle

As the seller is launched, it registers itself in the DFS and waits for some buyer to register too. As soon as it receives a message from the buyer, containing the disruption, seller searches in its data set for similar resource(s) to the one asked, what can be one of three types:

- Aircraft;
- Complete Crew;
- Crew Member;

In the scenario where an aircraft is needed, seller must provide a complete flight, this is, an aircraft and respective crew to handle it. This is done by creating all possible crews for each available aircraft. A crew is different from another if their prices and availabilities are different and therefore the combination criteria are one crew member's rank, salary and availability. In the scenario where a complete crew is needed, the resource asked contains how many crew members are needed from each rank and after reading that information, seller generates all possible combinations with the available resources, for the crew asked. In the scenario where a crew member is needed, no additional precautions are necessary. For this reason the seller only searches in its data set for similar resources after processing the resource asked type. In all scenarios the seller creates a list of possible solutions to propose and sorts that list in order to maximize the profit, by descending order of utility. So, seller starts by proposing the best resource in its list, receives some feedback, checks if it has similar experiences in it record and acts accordingly. If there are similar experiences, it selects one and replicates it. If there aren't similar experiences, seller just follows the feedback given. Then it's time to compare utilities between the updated proposal and an hypothetical proposal composed by the next resource in the list of possible solutions. If the updated proposals keeps an higher utility, it is proposed, else seller creates a new proposal with the next resource in its list of possible solutions. This cycle repeats itself until seller has no further resources in its list, when it sends a refuse message (explained in 4.1.3) and stops negotiating. As all sellers stop negotiating, one of them will receive an accept message, containing the proposal accepted, while all others receive a reject. The seller who received the accept, responds with an inform message to notify the buyer of the resource it just leased (all data concerning the resources). The sellers who received the reject message reset their structure and wait for another buyer to register in the EM to negotiate again. The seller that received the accept message updates its data set with the resources leased (are unavailable now) and resets its structure and waits for another buyer to enter the EM to negotiate again. Seller life cycle is presented by algorithm 11.

Algorithm 11: Seller Life Cycle

```
registerInDFS();
message := receivesFirstMessage();
resourceAsked = extractData(message);
allResources := loadsNecessaryData(dataSetFile);
possibleSolutions := combineResouces(allResources, resourceAsked);
index := 0;
while index < possibleSolutions.size() do
    if index = 0 then
        proposal := resourceToProposal(possibleSolutions.get(index);
        sendProposal(proposal);
    else
        feedbackMessage := receiveMessage();
        proposal := applyFeedback(feedbackMessage);
    end
    index++;
end
refuseProposal := RREFUSE;
sendProposal(refuseProposal);
lastMessage := receiveLastMessage();
if lastMessage = ACCEPT then
    resourceAccepted := extractData(lastMessage);
    inform := createInform(resourceAccepted);
    sendProposal(inform);
    updateDataSet(dataSetFile);
    resetStructure();
else
    resetStructure();
end
```

4.4 Summary

This chapter can be divided in three parts, one about the EM itself, containing the proper explanation of the EM data structure and each component function, and two more about the agents (buyer and seller), containing the proper explanation of its data structure, each component function and the agent life cycle. After explained the implementation it is possible to conclude that the negotiation this dissertation's specific case is a distributive negotiation because both agents wish to maximize the profit (sellers wish to maximize the profit and buyer wishes to minimize its cost) to the detriment of the other agents. This is explained by the fact that agents do not cooperate to find the best mutual solution but bargain to find oneself best solution.

Chapter 5

Experiments and Results

Now that both EM working flow and its structure were presented it is time to test and validate it. For this purpose, the Co-supervisor provided some data regarding disruptions and solutions found by an expert in the disruption management area. Each scenario reflected a disruption and assorted solution possibilities. This chapter will start by introducing some information about the experiments conducted to validate the EM and the data used, followed by a presentation of each experiment results. Finally, it is presented the analysis of the results obtained.

5.1 Preliminaries

This section provides some information about the data used in the experiments, in 5.1.1 and how the availability was inserted in the EM structure in 5.1.2.

5.1.1 Battery Tests

Due to the extensive data that represents a disruption, table 5.1 shows the most important attributes in the EM perspective, as the missing resource, delay and disruption cost and the resource id code (resource affected). Each disruption is duly identified with a number to make further references less confusing.

The full data provided to validate the EM can be seen in A.

Table 5.1: Simplified Disruptions

Disruption Nº	Resource Affected	Delay (min)	Disrupted Cost	Missing Resource
1	CSTTJ	40	20130.70	aircraft
2	CSTQD	40	801.58	cpt
3	CSTTP	15	3336.11	complete crew
4	CSTJG	30	292.50	cab
5	CSTTK	20	3150.91	aircraft
6	CSTNL	25	9622.17	aircraft
7	CSTNJ	11	17293.23	aircraft
8	CSTNN	70	34844.67	aircraft
9	CSTTU	8	1569.50	complete crew
10	CSTJF2	22	258.03	cab
11	CSTNM	10	18566.14	aircraft
12	CSTJF1	15	9099.27	aircraft

5.1.2 Availability Conversion

For the purpose of testing the market and since the availability was not present for neither aircraft or crew members, it was decided that a random value should be generated. This value should be a percentage where 100% would mean a resource with an availability at the moment, and 0% would mean a resource with an availability very close to delay, as referred in 4.1.1.3. For this reason, the availability calculation, in time units (milliseconds) was made by converting the value read from the data set to a decimal one. Then, that decimal value was subtracted to 100% and then multiplied by the delay. This is illustrated below for a better comprehension in equations (5.1) and (5.2).

$$\vartheta = 1 - (\frac{\nu}{100}) \tag{5.1}$$

where:

- ϑ is a resource's availability percentage;
- ν is the availability read from data set.

and:

$$\alpha = \lambda \times \vartheta \tag{5.2}$$

where:

- α is the resource's availability in time units;

- λ is the delay;
- ϑ is the resource's availability percentage.

This means that the higher the value read from the data set, the lower the percentage will be and consequently, the lower the availability, which means, a better one.

5.2 Experiments

In order to validate the results obtained through the EM, a few parameters were considered to measure each scenario solution's benefit, as follows:

- Buyer utility;
- Seller utility;
- Solution's availability;
- Solution's price;
- Delay reduction;
- Price reduction.

As previously stated in 4.2.4, by default, both attributes have the same weight in buyer's utility. To evaluate price and utility variations, the same tests were executed with different weights in two additional experiments. The first experiment considered the weights by default. The second experiment valued the availability with a weight of 80% and the price with a weight of 20% in the utility calculation. The third experiment showed an inversion regarding the values of the second one, i.e. availability with a weight of 20% and the price with a weight of 80% in the utility calculation. In all experiences, disruption number 12 has no results to present because does not exists in seller's data set any similar resource to the one asked, so seller leaves the negotiation as soon as it receives an empty result from the query to data set.

5.2.1 Experiment 1 (Balanced Experiment)

As stated in 5.2, three experiments were conducted with different weights for price and availability in the buyer's utility calculation, being the first experiment the default case where both have the same attributes have the same weight, 50%. In this experiment it was expected to find a similarity between cost and delay reductions. Chart 5.1 presents the results obtained in this experiment.

Disruption number 6 has a negative value for buyer's utility and for this reason it is not a viable solution. The fact that there is no delay reduction or the seller's utility being 0 supports that statement, so this is a case that would never result in a leasing contract.

One interesting fact that can be extracted is that seller's utility and delay reduction lines are overlapped, which mean that the have the same values for every disruption. This is explained

Figure 5.1: Experiment 1 Results

by both agents' utility formulas, detailed in chapter 4.2.4 and 4.3.3. While buyer considers both attributes (price and availability), seller considers only price, so, the higher the price, the best the availability and consequently if a resource has an availability of 0 ms (best possible scenario) it implies a 100% delay reduction and a seller's utility of 1.

The points below 0 are solutions that had a price greater that the disruption cost, or in the utilities case, unfeasible solutions.

A detailed overview over the results is presented in table 5.2

Table 5.2: Experiment 1 Results Table

Disruption No.	**Resource Affected**	**Market Solution**					
		Availability (ms)	**Price (€)**	**Seller Utility**	**Buyer Utility**	**Delay Reduction(%)**	**Cost Reduction(%)**
1	CSTTJ	384000	22021.489	0.84	0.37	84.00	-9.39
2	CSTQD	767999	1201.2	0.68	0.09	68.00	-49.85
3	CSTTP	144000	2079.66	0.84	0.61	84.00	37.66
4	CSTJG	54000	172.868	0.97	0.71	97.00	40.90
5	CSTTK	239999	4892.147	0.80	0.12	80.00	-55.26
6	CSTNL	1500000	14279.014	0.00	-0.24	0.00	-48.40
7	CSTNJ	131999	15271.318	0.80	0.46	80.00	11.69
8	CSTNN	756000	21413.575	0.82	0.60	82.00	38.55
9	CSTTU	144000	2171.835	0.70	0.16	70.00	-38.38
10	CSTJF2	39600	152.478	0.97	0.71	97.00	40.91
11	CSTNM	108000	26075.549	0.82	0.21	82.00	-40.52
12	CSTJF1						

The results obtained for this experiment show that with equal weights in the utility calculation, all but one disruption had its delay minimized by at least 68%. Relatively to the cost reduction the results are not as good as for availability, what is explained by the need to minimize a flight's delay and considering this, if the delay is largely reduced, as showed, then buyer is willing to pay more than the disruption cost. Still regarding this experiment results, the means of the considered parameters are presented in table 5.3.

Table 5.3: Experiment 1 - Means

Seller Utility	**Buyer Utility**	**Delay Reduction (%)**	**Cost Reduction(%)**
0.75	0.35	74.91	-6.55

Although the mean delay reduction is good (74.91%), the fact that the mean cost reduction is negative (-6.55%) implies a cost increase instead of a reduction. For this reason, this experiment cost reduction is not the expected. Concerning to utilities, the mean seller utility reveals that the EM is highly useful, at least for this experiment. Regarding the mean buyer utility, it shows some improvement but in some cases at a great cost, what explains the considerable difference between seller and buyer utilities.

5.2.2 Experiment 2 (80/20 Experiment)

In the second experiment the weights were modified in order to understand the EM evolution in order to more urgent needs, i.e. weights were distributed 80% - 20% between availability and price attributes, respectively. In this experiment it was expected to find a great delay reduction to the detriment of cost reduction. Chart 5.2 presents the results obtained in this experiment.

Although in this experiment there were no negative utilities and the delay reduction varies between 80% and 100%, the costs are not reduced as wanted. In some cases it increases and in disruptions 2 and 5, the cost increased over 50% and in the specific case of disruption 6 it increased almost 200%. This colossal cost increase is explained by the fact that the price had only a weight of 20% in buyer's utility calculation, which values it as $\frac{1}{5}$ of its real ratio, i.e. in this experiment buyer was willing to pay 5 times the disruption cost for a total delay reduction. A detailed overview over the results is presented in table 5.4.

Still regarding this experiment results, the means of the considered parameters are presented in table 5.5.

Although agents' utilities are not very discrepant there is still a remarkable gap between both, which means that seller gets more useful deals. Besides that, the mean delay reduction is 88.55% what can be seen as a major delay reduction. In the other hand this experiment had a negative mean cost reduction, what means that there is no reduction but a cost increase of 28.10% instead, what meets the expected outcome.

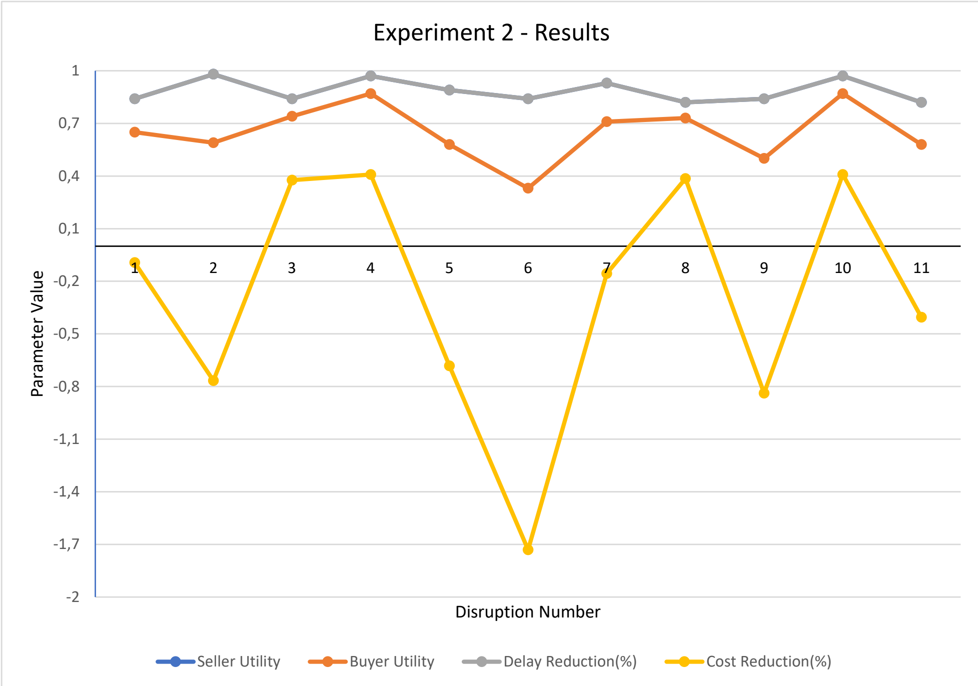


Figure 5.2: Experiment 2 Results

5.2.3 Experiment 3 (20/80 Experiment)

In the third and last experiment, in order keep exploring the EM evolution but this time to less urgent needs, the weights were distributed 20% - 80% between availability and price attributes, respectively. In this experiment it was expected to find a great cost reduction to the detriment of delay reduction. Chart 5.3 presents the results obtained in this experiment.

From this experiment it can be concluded that when price is preferred over availability, it is predictable on how the outcome will vary, i.e. for small departure intervals it will be one of the worse possible solutions, but for big departure intervals it will be the perfect choice. This perspective will be approached later.

The expected is to lease the resources with an availability tending to delay which would make the price to be much lower but that does not happens, because the departure intervals used for this experiment were small . For this reason the cost reduction is improved but it is not improved in the same measure as delay reduction is worsened, having a great range of cost reduction variation, from around 10% to nearly 100%, in the disruptions 5 and 10 respectively. Disruption number 6 has a negative value for buyer's utility and for this reason it is not a viable solution. The fact that there is no delay reduction or the seller's utility being 0 supports that statement, so this is a case that would never result in a leasing contract. This can be explained by the fact that the minimum price proposed will never be 0 because seller wants to have some profit from the leasing contract.

A detailed overview over the results is presented in table 5.6.

Still regarding this experiment results, the means of the considered parameters are presented in table 5.7.

Table 5.4: Experiment 2 Results

Disruption No.	Resource Affected	Market Solution					
		Availability (ms)	Price (€)	Seller Utility	Buyer Utility	Delay Reduction(%)	Cost Reduction(%)
1	CSTTJ	384000	22021.498	0.84	0.65	84.00	-9.39
2	CSTQD	48000	1415.7	0.98	0.59	98.00	-76.61
3	CSTTP	144000	2079.66	0.84	0.74	84.00	37.66
4	CSTJG	54000	172.868	0.97	0.87	97.00	40.90
5	CSTTK	131999	5299.222	0.89	0.58	89.00	-68.18
6	CSTNL	240000	26273.385	0.84	0.33	84.00	-173.05
7	CSTNJ	46199	19980.666	0.93	0.71	93.00	-15.54
8	CSTNN	756000	21413.575	0.82	0.73	82.00	38.55
9	CSTTU	144000	2171.835	0.84	0.50	70.00	-38.38
10	CSTJF2	39600	152.478	0.97	0.87	97.00	40.91
11	CSTNM	108000	26075.549	0.82	0.58	82.00	-40.52
12	CSTJF1						

In this experiment agents' utilities are close, yet are to low to be considered as a useful scenario for both. There is a mean delay reduction of 32.73% although only four disruptions had their delay minimized, what was not expected but it is explained by the cost-benefit relation. The mean cost reduction of 36.72% is to low for what was expected, i.e. as the availability tends to delay it was expected a greater cost reduction that the one obtained. This will be explained further in detail.

5.3 Results Discussion

5.3.1 First Comparison

In order to understand the relation among experiments and whether if one is better than the others in the EM perspective, the difference between each pair of experiments was made. The first pair is composed by the balanced experiment (with the same weight for both attributes,5.2.1) and the 80/20 distribution experiment (5.2.2). Chart 5.4 presents the difference between both experiments.

As shown, the only advantage in the balanced experiment (experiment 1) is in the cost reduction parameter but even with the extra cost, the buyer's utility is better in the 80/20 weight distribution experiment, which means that it is considered a better option for most of the disruptions. Even with and increased cost a bit higher than 120%, the buyer's utility shows and improvement of nearly 60% and a delay reduction of practically 90%, in disruption 6.

To help this comparison, a detailed overview is presented in table 5.8.

Still regarding the comparison between experiments, the means of the considered parameters are presented in table 5.9.

According to the data presented, it is possible to conclude that the 80/20 weight distribution experiment has an average improvement of 14% regarding the delay reduction parameter, when compared to the balanced experiment. Both agents have also an average improvement on their utilities in the same experiment, which points to 80/20 weight distribution experiment as a better one. Despite this, the balanced experiment has an average improvement of 22% on the cost reduction

Table 5.5: Experiment 2 - Means

Seller Utility	**Buyer Utility**	**Delay Reduction (%)**	**Cost Reduction(%)**
0.89	0.65	88.55	-28.10

parameter which shows that it is more balanced than the 80/20 weight distribution experiment, as expected.

5.3.2 Second Comparison

The second pair is composed by the balanced experiment (with the same weight for both attributes,5.2.1) and the 20/80 distribution experiment (5.2.3). Chart 5.5 presents the difference between both experiments.

As expected, the 20/80 weight distribution experiment benefits the cost reduction to the detriment of delay reduction, having an increase cost reduction varying from nearly 40% to approximately 70%, but having a decrease delay reduction between 60% and 85%. The buyer's utility is not much different in both experiments but still is better in the balanced experiment.

To help this comparison, a detailed overview is presented in table 5.10.

Still regarding the comparison between experiments, the means of the considered parameters are presented in table 5.11.

According to the data presented, it is possible to conclude that the 20/80 weight distribution experiment has an average improvement of 31.14% regarding the cost reduction parameter. All the other parameters are worsened, giving special focus to buyer's utility that has its average reduced to 9%.

5.3.3 Third Comparison

The third pair is composed by the 80/20 distribution experiment (5.2.2) and the 20/80 distribution experiment (5.2.3). Char 5.6 presents the difference between both experiments.

As expected, the 80/20 weight distribution experiment has a better delay reduction while the 20/80 weight distribution experiment has a better cost reduction and yet, even with a compared cost reduction over 50% over most disruptions, the buyer's utility remains better in the 80/20 weight distribution experiment. As stated before (in 5.2.2), this utility increase in favor of the 80/20 weight distribution experiment can be explained by the fact that buyer is willing to pay until 5 times the disruption cost, for a resource with a high delay reduction.

To help this comparison, a detailed overview is presented in table 5.12.

Still regarding the comparison between experiments, the means of the considered parameters are presented in table 5.13.

According to the data presented, it is possible to conclude that the 80/20 weight distribution experiment has an average improvement of 55.81% regarding the delay reduction parameter. Both agents utilities are also improved and the cost reduction is worsened in 52.69%.

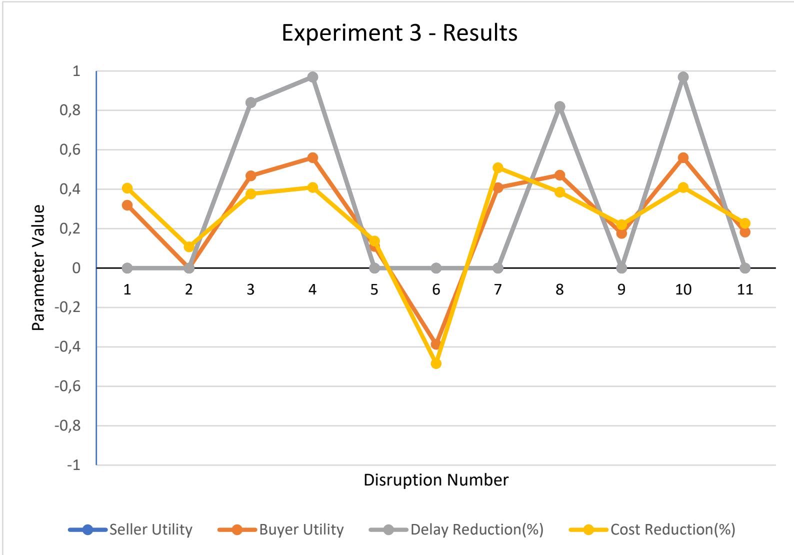


Figure 5.3: Experiment 3 Results

5.4 Human Specialist Evaluation

5.4.1 Methodology

After the EM returns the solutions found, there are some extra parameters that are not present in the market that must be in consideration when choosing the solution to be applied. As already said (5.2.1), the EM does not considers any costs unrelated to the disrupted resources, but passenger related costs must be considered, after the EM returns its solutions, in order to choose the best solution. For instance, the number of passengers that will miss a flight connection due to the delay carries an extra cost to the injured company (passenger cost) and will affect the passenger satisfaction, which also carries an extra cost to the company (passenger good will cost). These costs will be added to the aircraft and crew costs, being distributed as follows:

- Direct Costs: Aircraft cost plus crew cost plus passenger cost;
- Integrated Solution Costs: Passenger good will cost times passenger good will weight plus direct costs.

All these costs are considered by the human specialist (at the AOCC) when it must choose a solution to a disruption in its daily operation. This section intends to compare the solutions found by the EM to the ones chosen by a human specialist, by presenting the EM solutions to the human for analyze and validation. The passenger good will weight is, by default 5 according to the specialist.

In the table 5.14 it is possible to see how the three solutions obtained in the EM consequences in the flight delay and in the number of passengers missing the flight connection. The third scenario does not influence the delay and consequently has four passengers missing their connection flight and for this reason, this scenario is excluded.

Table 5.6: Experiment 3 Results

Disruption No.	**Resource Affected**	**Market Solution**					
		Availability (ms)	**Price (€)**	**Seller Utility**	**Buyer Utility**	**Delay Reduction(%)**	**Cost Reduction(%)**
1	CSTTJ	2400000	11968.205	0.00	0.32	0.00	41.55
2	CSTQD	2400000	715	0.00	0.00	0.00	10.80
3	CSTTP	144000	2079.66	0.84	0.47	84.00	37.66
4	CSTJG	54000	172.868	0.97	0.56	97.0	40.90
5	CSTTK	1200000	2717.858	0.00	0.11	0.00	13.74
6	CSTNL	1500000	14279.014	0.00	-0.39	0.00	-48.40
7	CSTNJ	660000	8484.058	0.00	0.41	0.00	50.94
8	CSTNN	756000	21413.575	0.82	0.47	82.00	38.55
9	CSTTU	480000	1224.00	0.00	0.18	0.00	22.01
10	CSTJF2	39600	152.478	0.97	0.56	97.00	40.91
11	CSTNM	600000	14327.225	0.00	0.18	0.00	22.79
12	CSTJF1						

In the table 5.15 it is possible to see the disruption and each solution cost, and its influence in the passenger and passenger good will costs. As the third solution does not improve the delay, it will have an increased cost due to the fact that some passengers lose their connection flight (as shown in 5.14).

In table 5.16 is possible to see the costs without the EM solutions (original direct costs) as the integrated solution cost without EM solutions (original integrated solution cost), calculated as explained before, i.e. the original integrated solution cost is the sum among aircraft,crew and passenger costs to which is added the result of the multiplication between passengers good will cost and its weight, as shown in equation (5.3).

$$IC = c_a + c_{cr} + c_{pax} + (c_{paxgw} \times w_{gw}) \tag{5.3}$$

where:

- IC is the integrated solution cost;
- c_a is the aircraft cost;
- c_{cr} is the crew cost;
- c_{pax} is the passenger cost;
- c_{paxgw} is the passenger good will cost;
- w_{gw} is the weight of good will.

The new integrated costs represent the integrated costs of the EM solutions while the original integrated costs represent the company solution integrated costs. As it is possible to see, the

Table 5.7: Experiment 3 - Means

Seller Utility	Buyer Utility	Delay Reduction (%)	Cost Reduction(%)
0.33	0.26	32.73	24.59

integrated costs of the solutions obtained in the market are considerably lower (around 40%), which points out to a better solution than the one found within the company own resources.

In the table 5.17 it is possible to see the savings provoked by each EM solution. The specialist always chooses the solution with a higher value of savings integrated, meaning that, in the demonstrated case two solutions could be picked, the one obtained when availability and price were equally valued or when availability had a weight of 80% and price had a weight of 20% in the buyer utility calculation.

5.4.2 Final Results

After being introduced the methodology used to calculate the Savings Integrated with a practical example, the results over all disruptions are presented in chart 5.7.

The table with all the data can be consulted in B. As shown, the EM solutions are much more cost-effective than the company's solutions, except in the disruption which is identified by CSTJF that has no similar resources in the EM. When comparing the chosen solution from the EM with the disruptive solution, the EM solutions present a mean delay reduction of 66.85% and a mean cost reduction of 63.51%.

5.5 Summary

This chapter presents the scenarios and data used to validate the EM. The results of the tests were explained and discussed, have been concluded that the outcome is very promising but there are some issues to be considered as, instead of using percentage values to represent resources' availability, the usage of time intervals may allow better results with 80/20 and 20/80 weight distribution, since both represent an urgency level (more urgent and less urgent, respectively).

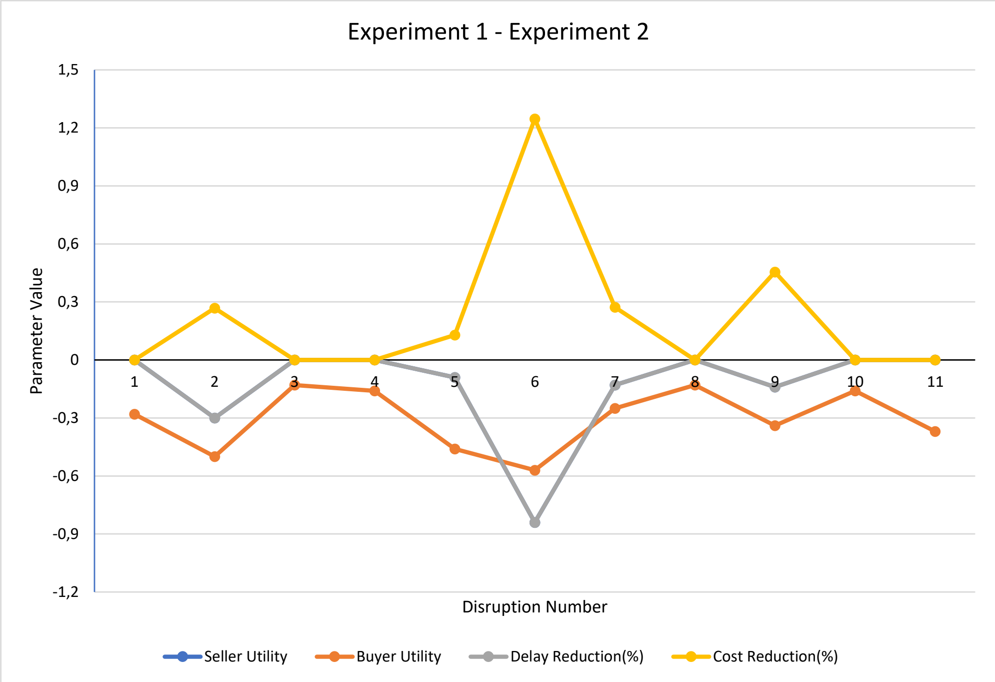


Figure 5.4: First Comparison

Table 5.8: First Comparison Table

50/50 minus 80/20					
Disruption Number	Resource Affected	Comparison			
		Seller Utility	Buyer Utility	Delay Reduction(%)	Cost Reduction(%)
1	CSTTJ	0.84	0.05	84.00	-50.00
2	CSTQD	0.68	0.09	68.00	-61.00
3	CSTTP	0.00	0.14	0.00	0.00
4	CSTJG	0.00	0.15	0.00	0.00
5	CSTTK	0.80	0.01	80.00	-69.00
6	CSTNL	0.00	0.15	0.00	0.00
7	CSTNJ	0.80	0.05	80.00	-39.00
8	CSTNN	0.00	0.13	0.00	0.00
9	CSTTU	0.70	-0.02	70.00	-60.00
10	CSTJF2	0.00	0.15	0.00	0.00
11	CSTNM	0.82	0.03	82.82	-63.00

Table 5.9: First Comparison Means

Seller Utility	Buyer Utility	Delay Reduction (%)	Cost Reduction(%)
-0.14	-0.30	-14.00	22.00

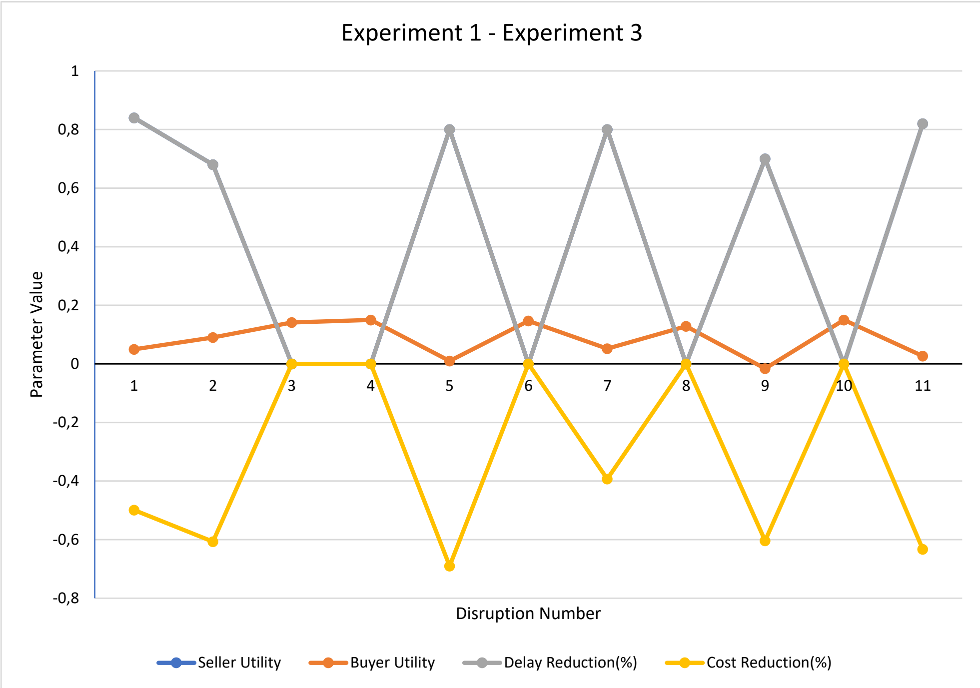


Figure 5.5: Second Comparison

Table 5.10: Second Comparison Table

50/50 minus 20/80					
Disruption Number	**Resource Affected**	**Comparison**			
		Seller Utility	**Buyer Utility**	**Delay Reduction(%)**	**Cost Reduction(%)**
1	CSTTJ	0.84	0.05	84.00	-50.00
2	CSTQD	0.68	0.09	68.00	-61.00
3	CSTTP	0.00	0.14	0.00	0.00
4	CSTJG	0.00	0.15	0.00	0.00
5	CSTTK	0.80	0.01	80.00	-69.00
6	CSTNL	0.00	0.15	0.00	0.00
7	CSTNJ	0.80	0.05	80.00	-39.00
8	CSTNN	0.00	0.13	0.00	0.00
9	CSTTU	0.70	-0.02	70.00	-60.00
10	CSTJF2	0.00	0.15	0.00	0.00
11	CSTNM	0.82	0.03	82.00	-63.00

Table 5.11: Second Comparison Means

Seller Utility	Buyer Utility	Delay Reduction (%)	Cost Reduction(%)
0.42	0.09	42.00	-31.14

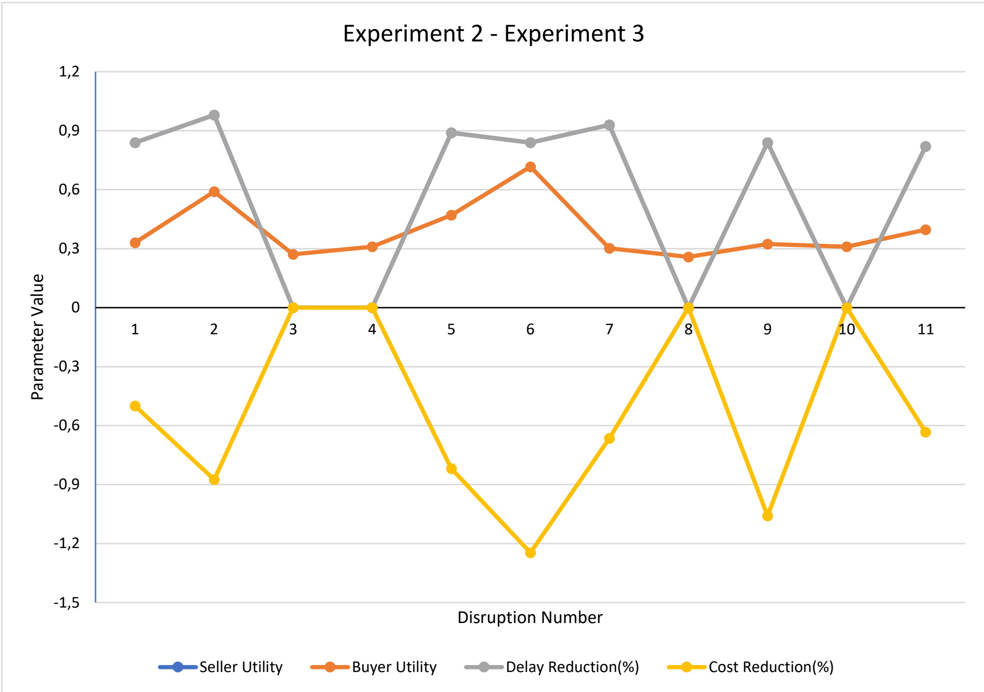


Figure 5.6: Third Comparison

Table 5.12: Third Comparison Table

80/20 minus 20/80					
Disruption Number	**Resource Affected**	**Comparison**			
		Seller Utility	**Buyer Utility**	**Delay Reduction(%)**	**Cost Reduction(%)**
1	CSTTJ	0.84	0.33	84.00	-50.00
2	CSTQD	0.98	0.59	98.00	-87.00
3	CSTTP	0	0.27	0.00	0.00
4	CSTJG	0	0.31	0.00	0.00
5	CSTTK	0.89	0.47	89.00	-82.00
6	CSTNL	0.84	0.72	84.00	-125.00
7	CSTNJ	0.93	0.30	93.00	-66.00
8	CSTNN	0	0.26	0.00	0.00
9	CSTTU	0.84	0.32	84.00	-106.00
10	CSTJF2	0	0.31	0.00	0.00
11	CSTNM	0.82	0.40	82.00	-63.00

Table 5.13: Second Comparison Means

Seller Utility	**Buyer Utility**	**Delay Reduction (%)**	**Cost Reduction(%)**
0.56	0.39	55.81	-52.69

Table 5.14: Solutions Delay and Consequences

EM Scenario	**Affected Resource**	**Missing Resource**	**Original Delay (min)**	**Proposed Delay (min)**	**Original Nº Passenger Miss Connection**	**Original Passenger Miss Connection by (min)**	**New Nº Passenger Miss Connection**
50-50	CSTTJ	aircraft	40	6.4	4	10	0
80-20				6.4			0
20-80				40			4

Table 5.15: Solutions Costs

EM Scenario	Resource Affected	Original Aircraft Cost	Original Crew Cost	Original Passenger Cost	Original Passenger Good Will Cost	Proposed Aircraft + Crew Cost	Proposed Crew Cost	New Passenger Cost	New Passenger Good Will Cost
50-50	CSTTJ	20131	2859	1000	3391	22021.489	0	0	542.488
80-20						22021.489	0	0	542.488
20-80						11968.205	0	1000	3390.55

Table 5.16: Cost Comparison

EM Scenario	Resource Affected	Original Direct Costs	Original Integrated Solution Cost	New Direct Costs	New Integrated Solution Cost
50-50	CSTTJ	23989	40942	22021	24734
80-20		23989	40942	22021	24734
20-80		23989	40942	12968	29921

Table 5.17: Solutions Savings

EM Scenario	Resource Affected	Savings Direct	Savings Good Will	Savings Integrated	Preferred by Specialist
50-50	CSTTJ	1968	2848	16208	**YES**
80-20		1968	2848	16208	**YES**
20-80		11021	0	11021	**NO**

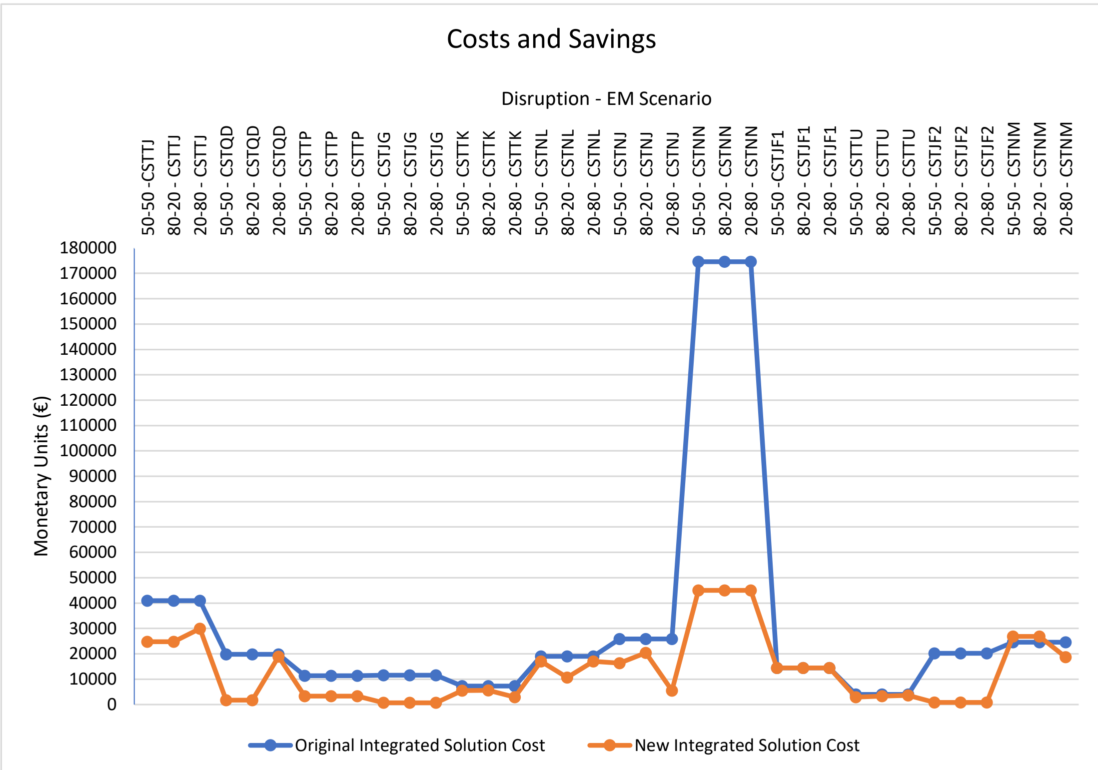


Figure 5.7: Costs and Savings

Chapter 6

Conclusions and Fututure Work

This chapter concludes the current dissertation by enumerating the main contributions of the work depleted and by suggesting some directions for future work.

6.1 Contributions

In chapter 1 the aim of this dissertation was defined as the development of a multi-agent based electronic market to expand airline companies' solutions space through the leasing of the required resources; and it was followed by the specification of each system component along with the EM workflow in chapter 3.

In chapter 4 all the EM structure was defined, from the negotiation object (flight) and its composition (aircraft and crew) to the negotiation data (proposal) and its function. The exchanged messages, their content and functions were also defined. Still in this chapter, both buyer and seller agents, entities representing the injured company and provider company respectively, were defined as well. In the buyer agent case, it was described how it evaluates and provides feedback to the received proposals. In the seller agent case, it was described how it learns from previous experiences and how that knowledge affects future negotiations, how it processes the feedback given by the buyer and how it evaluates the value of resources and proposals.

In chapter 5 the performed experiments were presented and discussed. Three approaches were introduced and tested over the disruptions provided. The results obtained from the tests performed were presented and analyzed along with a comparison between the three approaches. Following this comparison, the specialist choice was presented and explained along side with a general overview over the results obtained.

All aspects considered, the contribution given through this project was the expected one, that a multi-agent based electronic market is able to expand companies' solutions space regarding the disruption management problem.

6.2 Future Work

This project as a whole, was a proof of concept that a multi-agent based electronic would be a viable solution to help airline companies in their disruption management. As such, there are still many directions for future work.

First, the whole process of identifying previous similar experiences (by the seller) could be tested with different approaches by using machine learning and q-learning in order to understand how the agent learning process influences the negotiation, either in terms of proposals' price and availability or in terms of utility for each agent. The methodology used (CBR) can also be improved by creating better evaluation scenarios and benefiting the accepted proposal (or the tree of the proposals that lead to the accepted one).

Second, the embedding of the developed system into MASDIMA framework, referred in chapter 2, would be and interesting direction to go, once MASDIMA already has a proper environment regarding disruption management. This can be an improvement to the existent MASDIMA framework.

Third, the usage of heuristics to combine resources instead of doing all possible combinations would be an interesting feature. Or the usage of clustering algorithms to classify resources (where the parameters would be availability and/or price) in order to have a better and more efficient resource combination (where the resources would have similar availabilities).

Moreover, it would be worthy to use trust models to evaluate the EM outcome when considering the relations established between agents and, whether that trust measure would influence the agents' behaviour.

Another suggestion for further work is to create a graphical interface to make the application more intuitive for users.

Withstanding with the work developed during this dissertation, surely there is room for improvements on the results obtained. Such improvements depends on further testing of the developed system in terms of scale and variety of scenarios.

Furthermore, real test case scenarios consisting of real time-frames, i.e, using flights availability corresponding to a calendar-scheduled time-frame, should be done to evaluate the real application and use of the obtained results.

References

[Aha98] David W. Aha. The omnipresence of case-based reasoning in science and application. *Knowledge-Based Systems*, 11(5):261 – 273, 1998.

[AP94] Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1):39–59, 1994.

[BCG07] Fabio Luigi Bellifemine, Giovanni Caire, and Dominic Greenwood. *Developing multi-agent systems with JADE*, volume 7. John Wiley & Sons, 2007.

[BPR99] Fabio Bellifemine, Agostino Poggi, and Giovanni Rimassa. Jade–a fipa-compliant agent framework. In *Proceedings of PAAM*, volume 99, page 33. London, 1999.

[Cas08] Antonio J. M. Castro. Centros de controlo operacional: Organizacao e ferramentas. ISEC - Instituto Superior de Educacao e Ciencias, October 2008. Monograph for Post-graduation in Air Transport Operations.

[CCZ10] Xindu Chen, Xin Chen, and Xinhui Zhang. Crew scheduling models in airline disruption management. In *Industrial Engineering and Engineering Management (IE&EM), 2010 IEEE 17Th International Conference on*, pages 1032–1037. IEEE, 2010.

[CJGM97] John Collins, Scott Jamison, Maria Gini, and Bamshad Mobasher. Temporal strategies in a multi-agent contracting protocol. In *AAAI-97 Workshop on AI in Electronic Commerce*, 1997.

[CO00] Henrique Lopes Cardoso and Eugénio Oliveira. A platform for electronic commerce with adaptive agents. In *International Workshop on Agent-Mediated Electronic Commerce*, pages 96–107. Springer, 2000.

[CRO12] António J. M. Castro, Ana Paula Rocha, and Eugénio Oliveira. Towards an Autonomous and Intelligent Airline Operations Control. *15th International IEEE Conference on Intelligent Transportation Systems Anchorage, Alaska, USA, September 16-19*, pages 1429–1434, 2012.

[CRO14] António JM Castro, Ana Paula Rocha, and Eugénio Oliveira. *A new approach for disruption management in airline operations control*, volume 562. Springer, 2014.

[dC13] António Jesus Monteiro de Castro. *A Distributed Approach to Integrated and Dynamic Disruption Management in Airline Operations Control.* Phd thesis, Faculty of Engineering, University of Porto, Portugal, Porto, Portugal, July 2013.

[dOFSG99] Eugenio de Oliveira, Jose Manuel Fonseca, and Adolfo Steiger-Garção. Multi-criteria negotiation in multi-agent systems. *CEEMAS'99*, page 190, 1999.

[FIP17] The Foundation for Intelligent Physical Agents. Available at http://www.fipa.org/, June 2017.

[FSJB99] Peyman Faratin, Carlos Sierra, Nick R Jennings, and PHIL Buckle. Designing responsive and deliberative automated negotiators. In *Proc. AAAI Workshop on Negotiation: Settling Conflicts and Identifying Opportunities*, pages 12–18, 1999.

[Irr96] ME Irrang. The handbook of airline economics. *Air Transport Association of America*, pages 349–365, 1996.

[J2M17] JAVA PLATFORM, MICRO EDITION (JAVA ME). Available at http://www.oracle.com/technetwork/java/embedded/javame/index.html, June 2017.

[JAD17] Java Agent DEvelopment Framework. Available at http://jade.tilab.com/, June 2017.

[JFL+01] Nicholas R Jennings, Peyman Faratin, Alessio R Lomuscio, Simon Parsons, Michael J Wooldridge, and Carles Sierra. Automated negotiation: prospects, methods and challenges. *Group Decision and Negotiation*, 10(2):199–215, 2001.

[JSW98] Nicholas R Jennings, Katia Sycara, and Michael Wooldridge. A roadmap of agent research and development. *Autonomous agents and multi-agent systems*, 1(1):7–38, 1998.

[Kol14] Janet Kolodner. *Case-based reasoning*. Morgan Kaufmann, 2014.

[Lim16] Jorge Filipe Monteiro Lima. Argumentation in the resolution of passenger problems using mobile devices. Msc thesis, University of Porto, Faculty of Engineering, Porto, Portugal, September 2016.

[MSJ98] Noyda Matos, Carles Sierra, and Nicholas R Jennings. Determining successful negotiation strategies: An evolutionary approach. In *Multi Agent Systems, 1998. Proceedings. International Conference on*, pages 182–189. IEEE, 1998.

[Nil96] Nils J Nilsson. Artificial intelligence: A modern approach: Stuart russell and peter norvig,(prentice hall, englewood cliffs, nj, 1995); xxviii+ 932 pages, 1996.

[OR01] Eugenio Oliveira and Ana Rocha. Agents advanced features for negotiation in electronic commerce and virtual organisations formation process. *Agent Mediated Electronic Commerce*, pages 78–97, 2001.

[PSJ98] Simon Parsons, Carles Sierra, and Nick Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and computation*, 8(3):261–292, 1998.

[RRM02] Howard Raiffa, John Richardson, and David Metcalfe. *Negotiation analysis: the science and art of collaborative decision making*. Harvard University Press, 2002.

[RS13] Christopher K Riesbeck and Roger C Schank. *Inside case-based reasoning*. Psychology Press, 2013.

[RZ94] Jeffrey S Rosenschein and Gilad Zlotkin. *Rules of encounter: designing conventions for automated negotiation among computers*. MIT press, 1994.

[SB98] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[sma] The house that learns: Bringing artificial intelligence into the home. Available at http://bit.ly/2tl0OW2.

[SS97] Troy J. Strader and Michael J. Shaw. Characteristics of electronic markets. *Decision Support Systems*, 21(3):185 – 198, 1997. Electronic commerce.

[TAP17] TAP. Tap portugal relatório anual 2010, February 2017.

[VJ00] Nir Vulkan and Nicholas R Jennings. Efficient mechanisms for the supply of services in multi-agent environments. *Decision Support Systems*, 28(1):5–19, 2000.

[Wat98] Ian Watson. *Applying case-based reasoning: techniques for enterprise systems*. Morgan Kaufmann Publishers Inc., 1998.

[Wat99] Ian Watson. Case-based reasoning is a methodology not a technology. *Knowledge-based systems*, 12(5):303–308, 1999.

[Woo09] Michael Wooldridge. *An introduction to multiagent systems*, pages 15–23,105–111,129–148. John Wiley & Sons, 2009.

REFERENCES

Appendix A

Disrupted Flights

A.1 Disrupted Flights Data

Table A.1: Disrupted Flights 1

resource_affected	**resource_type**	**resource_cap**	**crew_res_type**	flight_date	flight_number	origin	destination	scheduled_time_of_departure	scheduled_time_of_arrival	estimated_trip_time	departure_delay_in_minutes
CSTTJ	319	126	NB	03/04/2017 00:00	431	ORY	LIS	2017-04-03 16:50:00	2017-04-03 19:15:00	3:05:00	40
CSTQD	320	156	NB	03/04/2017 00:00	661	AMS	LIS	2017-04-03 16:50:00	2017-04-03 19:50:00	3:40:00	40
CSTTP	319	138	NB	03/04/2017 00:00	1686	FNC	LIS	2017-04-03 16:50:00	2017-04-03 18:25:00	1:50:00	15
CSTJG	321	210	NB	03/04/2017 00:00	809	MXP	LIS	2017-04-03 17:05:00	2017-04-03 19:50:00	3:15:00	30
CSTTK	319	138	NB	03/04/2017 00:00	1904	FAO	LIS	2017-04-03 17:05:00	2017-04-03 17:50:00	1:05:00	20
CSTNL	320	156	NB	03/04/2017 00:00	551	MUC	LIS	2017-04-03 17:40:00	2017-04-03 20:50:00	3:35:00	25
CSTNJ	320	168	NB	04/04/2017 00:00	618	LIS	BRU	2017-04-04 08:35:00	2017-04-04 11:15:00	2:51:00	11
CSTNN	320	168	NB	04/04/2017 00:00	1674	FNC	LIS	2017-04-04 09:15:00	2017-04-04 10:50:00	2:45:00	70
CSTJF	321	210	NB	04/04/2017 00:00	803	MXP	LIS	2017-04-04 09:25:00	2017-04-04 12:15:00	3:05:00	15
CSTTU	319	132	NB	04/04/2017 00:00	436	LIS	ORY	2017-04-04 05:35:00	2017-04-04 08:00:00	2:33:00	8
CSTJF	321	210	NB	04/04/2017 00:00	802	LIS	MXP	2017-04-04 06:05:00	2017-04-04 08:35:00	2:52:00	22
CSTNM	320	156	NB	04/04/2017 00:00	834	LIS	FCO	2017-04-04 05:50:00	2017-04-04 08:40:00	3:00:00	10

Table A.2: Disrupted Flights 2

resource_affected	**bus_pax**	**econ_pax**	**total_pax**	**scheduled_cost_aircraft**	**scheduled_cost_crew**	**scheduled_cost_passenger**	number_pax_pnr	number_crew_members
CSTTJ	5	112	117	20113,013	2569,750	0,000	117	6
CSTQD	4	135	139	9301,813	1469,000	0,000	139	6
CSTTP	14	123	137	23069,246	3209,750	0,000	137	6
CSTJG	2	108	110	8941,719	1881,500	0,000	110	8
CSTTK	1	57	58	3149,554	1144,250	0,000	58	6
CSTNL	3	111	114	9619,174	1766,583	0,000	114	6
CSTNJ	2	91	93	17285,762	2969,917	0,000	93	6
CSTNN	3	166	169	34826,773	4651,583	0,000	169	6
CSTJF	3	82	85	9097,869	1807,750	0,000	85	8
CSTTU	5	109	114	14817,245	1522,167	0,000	114	7
CSTJF	13	195	208	17606,583	3490,500	0,000	208	8
CSTNM	5	154	159	18190,240	1536,333	0,000	159	7

Table A.3: Disrupted Flights 3

resource_affected	CPT	OPT	SCC	CC	CAB	cost_disr_aircraft	cost_disr_crew	cost_disr_pax	cost_disr_pax_gw	missing_resource
CSTTJ	1	1	0	1	3	20130,695	2858,535	0,000	3390,550	aircraft
CSTQD	1	1	0	1	3	9304,551	1666,684	0,000	3237,300	cpt
CSTTP	1	1	0	1	3	23070,814	3336,106	0,000	1611,000	all crew
CSTJG	1	1	0	1	4	8944,495	2072,636	0,000	1738,700	cab
CSTTK	1	1	0	1	3	3150,909	1249,885	0,000	586,650	aircraft
CSTNL	1	1	0	1	3	9622,167	1903,268	0,000	1500,000	aircraft
CSTNJ	1	1	0	1	3	17293,233	3078,611	0,000	1107,000	aircraft
CSTNN	1	1	0	1	3	34844,666	6079,931	0,000	26230,150	aircraft
CSTJF	1	1	0	1	4	9099,216	1895,771	0,000	687,500	aircraft
CSTTU	1	1	0	1	3	15075,645	1569,500	0,000	484,000	all crew
CSTJF	1	1	0	1	4	18793,323	3728,000	0,000	3285,700	cab
CSTNM	1	1	0	1	3	18566,140	1591,000	0,000	875,000	aircraft

A.2 Airports Identification

Table A.4: Acronyms and Cities

Acronym	**City(Airport)**
ORY	Paris (Orly)
AMS	Amterdam (Schiphol)
FNC	Madeira
MXP	Milan (Malpensa)
FAO	Faro
MUC	Munich
LIS	Lisbon
BRU	Brussels
FNC	Madeira
FCO	Roma (Fiumicino)

Appendix B

Specialist Choice

Table B.1: Specialist Choice 1

EM Scenario	Resource Affected	Missing resource	Original Delay	Proposed Delay	original nr pax miss conn	original pax miss conn by minutes	new nr pax miss conn
50-50				6.4			0
80-20	CSTTJ	aircraft	40	6.4	4	10	0
20-80				40			4
50-50				0.8			0
80-20	CSTQD	cpt	40	0.8	5	10	0
20-80				40			5
50-50				2.4			0
80-20	CSTTP	all crew	15	2.4	0	0	0
20-80				2.4			0
50-50				0.9			0
80-20	CSTJG	cab	30	0.9	2	5	0
20-80				0.9			0
50-50				3.999983333			0
80-20	CSTTK	aircraft	20	2.199983333	0	0	0
20-80				20			0
50-50				25			0
80-20	CSTNL	aircraft	25	4	0	0	0
20-80				25			0

Table B.2: Specialist Choice 2

EM Scenario	Resource Affected	Original Aircraft Cost	Original Crew Cost	Original pax cost	Original px gw cost	Proposed Aircraft + Crew Cost	Proposed Crew Cost	New pax cost	New pax gw cost
50-50	CSTTJ	20131	2859	1000	3391	22021.489	0	0	542.488
80-20						22021.489	0	0	542.488
20-80						11968.205	0	1000	3390.55
50-50	CSTQD	9305	801.58	2000	3237	9305	1201.2	0	64.746
80-20						9305	1415.7	0	64.746
20-80						9305	715	2000	3237.3
50-50	CSTTP	23071	3336	0	1611	23071	2079.66	0	257.76
80-20						23071	2079.66	0	257.76
20-80						23071	2079.66	0	257.76
50-50	CSTJG	8944	292.5	800	1739	8944	172.868	0	52.161
80-20						8944	172.868	0	52.161
20-80						8944	172.868	0	52.161
50-50	CSTTK	3151	1250	0	587	4892.147	0	0	117.3295111
80-20						0	0	0	64.53101113
20-80						0	0	0	586.65
50-50	CSTNL	9622	1903	0	1500	14279.014	0	0	1500
80-20						0	0	0	240
20-80						0	0	0	1500

Table B.3: Specialist Choice 3

EM Scenario	Resource Affected	Original Direct Costs	Original Integrated Solution Cost	New Direct Costs	New Integrated Solution Cost	Savings Direct	Savings GW	Savings Integrated	Preferred by Specialist
50-50	CSTTJ	23989	40942	22021	24734	1968	2848	16208	**YES**
80-20		23989	40942	22021	24734	1968	2848	16208	**YES**
20-80		23989	40942	12968	29921	11021	0	11021	**NO**
50-50	CSTQD	12106	28293	10506	10829	1600	3173	17463	**YES**
80-20		12106	28293	10720	11044	1386	3173	17249	**NO**
20-80		12106	28293	12020	28206	87	0	87	**NO**
50-50	CSTTP	26407	34462	25150	26439	1256	1353	8023	**YES**
80-20		26407	34462	25150	26439	1256	1353	8023	**YES**
20-80		26407	34462	25150	26439	1256	1353	8023	**YES**
50-50	CSTJG	10037	18730	9117	9378	920	1687	9352	**YES**
80-20		10037	18730	9117	9378	920	1687	9352	**YES**
20-80		10037	18730	9117	9378	920	1687	9352	**YES**
50-50	CSTTK	4401	7334	4892	5479	-491	469	1855	**NO**
80-20		4401	7334	0	323	4401	522	7011	**YES**
20-80		4401	7334	0	2933	4401	0	4401	**NO**
50-50	CSTNL	11525	19025	14279	21779	-2754	0	-2754	**NO**
80-20		11525	19025	0	1200	11525	1260	17825	**YES**
20-80		11525	19025	0	7500	11525	0	11525	**NO**

Table B.4: Specialist Choice 4

EM Scenario	Resource Affected	Missing resource	Original Delay	Proposed Delay	original nr pax miss conn	original pax miss conn by minutes	new nr pax miss conn
50-50	CSTNJ	aircraft	11	2.199983333	0	0	0
80-20				0.769983333			0
20-80				11			0
50-50	CSTNN	aircraft	70	12.6	10	20	0
80-20				12.6			0
20-80				12.6			0
50-50	CSTJF	aircraft	15	15	0	0	0
80-20				15			0
20-80				15			0
50-50	CSTTU	all crew	8	2.4	0	0	0
80-20				1.28			0
20-80				8			0
50-50	CSTJF2	cab	22	0.66	0	0	0
80-20				0.66			0
20-80				0.66			0
50-50	CSTNM	aircraft	10	1.8	0	0	0
80-20				1.8			0
20-80				10			0

Table B.5: Specialist Choice 5

EM Scenario	Resource Affected	Original Aircraft Cost	Original Crew Cost	Original pax cost	Original px gw cost	Proposed Aircraft + Crew Cost	Proposed Crew Cost	New pax cost	New pax gw cost
50-50						15271.318	0	0	221.3983227
80-20	CSTNJ	17293.233	3078.611	0	1107	0	0	0	77.48832273
20-80						0	0	0	1107
50-50						21413.575	0	0	4721.427
80-20	CSTNN	34844.666	6079.931	2500	26230.15	0	0	0	4721.427
20-80						0	0	0	4721.427
50-50						10994.987	0	0	687.5
80-20	CSTJF	9099.216	1895.771	0	687.5	10994.987	0	0	687.5
20-80						10994.987	0	0	687.5
50-50						15075.645	2171.835	0	145.2
80-20	CSTTU	15075.645	1569.5	0	484	15075.645	2885.58	0	77.44
20-80						15075.645	1224	0	484
50-50						18793.323	152.478	0	98.571
80-20	CSTJF2	18793.323	258.03	0	3285.7	18793.323	152.478	0	98.571
20-80						18793.323	152.478	0	98.571
50-50						26075.549	0	0	157.5
80-20	CSTNM	18566.14	1591	0	875	0	0	0	157.5
20-80						0	0	0	875

Table B.6: Specialist Choice 6

EM Scenario	Resource Affected	Original Direct Costs	Original Integrated Solution Cost	New Direct Costs	New Integrated Solution Cost	Savings Direct	Savings GW	Savings Integrated	Preferred by Specialist
50-50	CSTNJ	20371.844	25906.844	15271.318	16378.30961	5100.526	885.6016773	9528.534386	**NO**
80-20		20371.844	25906.844	0	387.4416136	20371.844	1029.511677	25519.40239	**YES**
20-80		20371.844	25906.844	0	5535	20371.844	0	20371.844	**NO**
50-50	CSTNN	43424.597	174575.347	21413.575	45020.71	22011.022	21508.723	129554.637	**NO**
80-20		43424.597	174575.347	0	23607.135	43424.597	21508.723	150968.212	**YES**
20-80		43424.597	174575.347	0	23607.135	43424.597	21508.723	150968.212	**YES**
50-50	CSTJF	10994.987	14432.487	10994.987	14432.487	0	0	0	**NO**
80-20		10994.987	14432.487	10994.987	14432.487	0	0	0	**NO**
20-80		10994.987	14432.487	10994.987	14432.487	0	0	0	**NO**
50-50	CSTTU	16645.145	19065.145	17247.48	17973.48	-602.335	338.8	1091.665	**YES**
80-20		16645.145	19065.145	17961.225	18348.425	-1316.08	406.56	716.72	**NO**
20-80		16645.145	19065.145	16299.645	18719.645	345.5	0	345.5	**NO**
50-50	CSTJF2	19051.353	35479.853	18945.801	19438.656	105.552	3187.129	16041.197	**YES**
80-20		19051.353	35479.853	18945.801	19438.656	105.552	3187.129	16041.197	**YES**
20-80		19051.353	35479.853	18945.801	19438.656	105.552	3187.129	16041.197	**YES**
50-50	CSTNM	20157.14	24532.14	26075.549	26863.049	-5918.409	717.5	-2330.909	**NO**
80-20		20157.14	24532.14	0	787.5	20157.14	717.5	23744.64	**YES**
20-80		20157.14	24532.14	0	4375	20157.14	0	20157.14	**NO**