Thesis

by Diederik van Krieken

Negotiation in a modular manufacturing process

Master Thesis v 0.5

Diederik van Krieken
Artificial Intelligence
Univeristy of Groningen, the Netherlands
d.r.j.van.krieken@student.rug.nl
S2009730

Internal supervisor:
Rineke Verbrugge (Artificial Intelligence, RUG)

November 2016

Abstract

In this thesis an application of a multi-issue multi-lateral negotiation is discussed. The

Contents

Acknowledgements

I would first like to thank my thesis advisor Prof. dr. Rineke (L.C.) Verbrugge of the Artificial Intelligence and Cognitive Engineering (ALICE) research institute at Groningen University. The little time she has, due to the sudden growth of the department, was unnoticeable. She always took the time to listen to my difficulties and or problems and was constantly open about it. She consistently allowed this paper to be my own work, and steered me in the right the direction whenever she thought I needed it.

Furthermore, I would like to thank Youri de Koster who has supported me at the office. His weekly reviews and personal advices has taught me more than any class and thesis ever could. Also Rob Goes, Dennis Kersten and Edwin Knoop have been a great aid, and continued to support me, even when I thought I could not continue.

I would also like to thank the experts who were involved in this research project: Ronghuo Zheng, Mathijs de Weerdt, and Tim Baarslag. Without their input, no thesis would have been written

I would also like to acknowledge my fellow interns, especially Marnix Montantus and Pedram Muurlink, who always listened to the problems and the 09:30hr coffee is forever in my mind.

Finally, I must express my very profound gratitude to my mother, sister and brother, and to my dearest Vivianne for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. At last I would like to thank my father, who has always been the greatest support during the studies, and unfortunately can not see what has been accomplished.

Diederik van Krieken

Chapter 1

Introduction

This thesis is written as part of the Master Artificial Intelligence at the University of Groningen on behalf of the Multi-Agent Systems (MAS) group. The MAS group is part of the Artificial Intelligence and Cognitive Engineering (ALICE) research institute. This group is led by L.C. (Rineke) Verbrugge.

1.1 Introduction in Production, AI and the use case environment

Currently there is a lot of research being conducted in Artificial Intelligence (AI) and how to apply this in business. One field of interest, which is researched in this thesis, is the combination of Multi-Agents Systems in Production using Negotiation.

1.1.1 Production and Manufacturing

Production is the process of converting inputs into outputs. It is one of the economic pillars on which the economic markets are driven. By creating extra value from basic commodities, a (perceived) contribution to the well-being of individuals is conceivable. Manufacturing is a specific subsidiary of production, and is the process of converting (raw) material into semi and/or finished end products by using various processes, machines and energy. Thus, every type of manufacturing can be production, but not every type of production is manufacturing. The production and manufacturing industry is and will be one of the wealth generators of the world economy (Monostori et al., 2006), and is characterised by the production of commodities that have value and contribute to the well-being of individuals.

In the industrial production world, a 4th revolution is going on, which enables the world to think about new production processes. The First Industrial Revolution was the use of steam power to mechanize production. In the second revolution, the use of electric power allowed for assembly lines, resulting in mass production. The third revolution used electronics and information technology to automate production. Now a fourth industrial revolution, also called Industry 4.0*, is building on the third, and is called the digital revolution. It is characterized by a fusion of technologies that is blurring the lines between the physical and digital (Leitao et al., 2016).

Throughout this thesis, the terms' production and manufacturing will be used interchangeably. This does not mean that the terms are interchangeable in general, since in the industry there is a difference. However, for this research, due to the similarity in the sense of the processes, no separation is needed. This is supported by the exchangeability of the terms in the literature.

1.1.2 Artificial Intelligence

The research will be based on an intelligent Multi-Agent System (MAS) which would consist of sensors, and processors which act and react on their environment in both a physical as in an IT way. For the intelligent agents it would be possible, by understanding the system, and by negotiating, to come up with a (near-) optimal production planning, taking in consideration possible maintenance and downtime, based on real-time data acquisition, analysis, negotiations and decentralized autonomous decision making. Such intelligence is an example of a typical MAS where artificial intelligence may include methodical, functional, and procedural approaches, algorithmic search and/or reinforcement learning.

1.1.3 Ecosystem of the case study

In this thesis a new model will be constructed based on negotiation in an intelligent Multi-Agent System. An application of this new model is tested and modelled based on a plant that creates de-mineralized water. By removing all the ions from common water, de-mineralized water is obtained. This water is used for many processes, and has many applications. In this plant specifically it is used for the steam turbines, which generate electricity. By burning the by-product, heat is generated, which creates steam to power the turbines.

Minerals are removed from water by multiple production steps. Most common, and as is implemented in the plant described, is to first remove the positively charged minerals in so called anions. After this, the negative charged ions are removed within a cation filter. To ensure that all ions are removed, a final

*This revolution has multiple terms in multiple countries. For example, Industrie 4.0 in Germany, Smart Manufacturing or Smart Industry in the Netherlands, or the Industrial Internet Consortium in the U.S.A. In this thesis the term Industry 4.0 will be used.

combined "mixbed" is used. Here a combination of an anion and cation filter removes the residues.

These filters have to be cleaned every few hours to ensure that proper demineralization occurs. By optimizing the production planning, real-time predicting where cleaning is necessary is possible, resulting in minimal waste.

1.2 Thesis outline

In Chapter 2 an overview of the problem is given. Chapter 3 will explain the literature regarding manufacturing and negotiation. This also includes an overview of the methods. From this a framework is concluded with, which is used to design and implement our model, the foundation of Chapter 3. In Chapter 4 the model is tested and evaluated by simulation. From this we can conclude and generalize for further use as described in Chapter 5.

1. Introduction
2. Problem definition
3. Literature Study & Theoretical framework
4. Design & Computational Implementation
5. Simulation comparison incl evaluations
6. Discussion & Conclusions

Chapter 2

Problem Definition & Research Goal

An overview of the problem will be given, and based on the findings, the research goal will be discussed. Important is to define the relevance and approach to the entire research.

2.1 Problem Analysis

Due to the 4th industrial revolution, new production and manufacturing methods are required which need new digital solutions to optimize the planning. One solution is centralized analysis, combining all the data in a central database and analysing this to optimize decision making. Another solution, which is decentralisation, analyses the data on several points, which independently create decisions. One of the decisions for the implementation of such a system requires many considerations. Currently it is not fully clear what requirements depend on the implementation (Leitao et al., 2016). Also, the practically of different negotiation frameworks is unknown (Fatima et al., 2014b). For example, a necessity might be the requirement that the process is subject to change. If expanded or changed, many modifications in a centralized system are required since the central database has to relearn the patterns, and new databases might have to be set up. This might, however, not be the case with decentralized solutions.

A second problem is that the amount of data nowadays is enormous and as a result large quantities of data are pouring on-line, waiting to be processed in the centralized database. Furthermore, much of the data is not processed from the sensor towards the centralized database, resulting in incomplete analysis. There is an overall consensus that the future of industry 4.0 lies with pre-aggregated data (Slaughter et al., 2015) which is obtained by having the sensors think and reason about the measurements before sending the processed information to a central database.

Thirdly, scheduling production problems are Non-deterministic Polynomial (NP)-hard problems that are very complex to solve using (mixed) integer programming and take a very long time to find an optimal solution. There is a consensus that Multi-Agent scheduling retrieves a (suboptimal)-solution in reasonable time (Konolige and Nilsson, 1980). Since scheduling is NP-hard, this solution does not have to be the optimal solution but a "good enough" result.

The new developments in the industries, like the use of Internet of Things (IoT) require manufacturers to rethink their production. An IoT is a network where many sensors are connected using different web protocols or protocols specifically designed for IoT. These sensors retrieve their data and share the information via this network and usually communicate with a centralized database, where the data of the sensors is analysed. After analysing, production can be planned resulting in lower down time of the asset and more efficient production. When these systems are embedded, they are also known as Cyber-Physical Systems (CPS).

2.2 Area of Application

Currently an industry leader in the production of steal is looking to optimize their de-mineralized water production. Currently their production process is done by hand, and no digital optimization method is currently in place. Furthermore, a substantial amount of some very costly materials is "discarded" due to legislative requirements. By using these materials instead of dumping them, cost can be reduced.

As the main scope of this research project is aimed at negotiation, the planning under consideration will undergo some idealization meaning, that it will not be too constrained. This leaves for example, specific training levels of the mechanics out of scope. Furthermore, the possible difficult operations are excluded. If time allows it, more constraints can be included.

2.3 Relevance

The research will be relevant for two different stakeholders, the academic and business world. Business has always been dependent on the academic world, and by connecting these, new valuable insights can be combined.

2.3.1 Scientific relevance

Currently there are not a lot of papers discussing the use of negotiation in a multi-agent solution for manufacturing. There comprehensive overviews, but the negotiation aspect is a commonly lacking subject (Leitão, 2009). In chapter 3 a comprehensive overview will be given. By researching and, importantly, and computationally implementing the use of negotiation in distributed production planning, the theory can be connected to real life cases. This is based on the classic artificial intelligence problem, which is the combination of information and objectives from different sources and will be solved with a Multi-Agent System.

This research is about the application of multi-agent system technology, negotiation, game theory and decision making. Knowledge from AI about negotiation will be used to obtain new insights in possible decentralized production solutions.

For me personally this research project would be a perfect way to find out how ideas and solutions in the AI literature can be used to describe and improve large-scale and real-world solutions.

2.3.2 Business relevance

The business has difficulty in the transformation to the new industrial pillars. Enormous amounts of data, and new requirements ask for "on top of the line" production systems. By computationally implementing one of the processes and optimizing these processes, these insight can be applied for further use.

2.4 Research Goal

The main goal is to create a production planner using a Multi-Agent System with negotiation. This is divided into the following sub-goals:

1. Provide a theoretical framework for negotiation in a Multi-Agent System in the context of manufacturing/production planning incl predictive maintenance and/or process control.
2. Create a demonstrator of this framework to show that a Multi-Agent System can be used for manufacturing/production planning incl predictive maintenance and/or process control.
3. Provide new maintenance management insights;
4. Determine what application contexts profit of the creation of such a new Multi-Agent System.

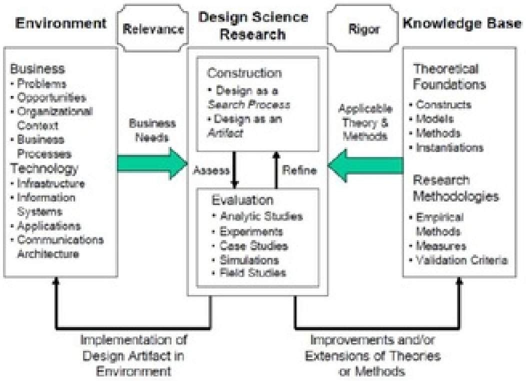

Figure 2.1: The Information System Research Framework as designed by Hevner and Chatterjee (2010)

2.5 Research Approach

Since this is an academic research project, a new MAS framework will be investigated and constructed. The working and exact results will be analysed by the use of a demonstrator. This falls under the computational implementation and modelling of a new MAS framework. This excludes the verification (use users to control your theory) & validation of the system.

The research framework used will be based on Hevner and Chatterjee (2010) and can be seen in Figure 2.1. The aim of the relevance cycle is to connect the real-world environment of the research project with the design science activities. Through this relevance cycle, opportunities for the improvement of practices can be identified.

The rigor cycle is used to assemble a knowledge base that consists of the relevant theoretical foundations and research methodologies. Prior research provides a starting point and benchmark for new artefacts. This knowledge base is necessary to establish theoretical appropriateness and relevance, achieving rigor.

In this research, a case-study is included to ensure the relevance of the new MAS framework. By comparing the model with a real-world situation, the new MAS framework can be assessed and maybe refined.

2.6 Research Process

Firstly a literature research was concluded to assess the current negotiation methods in agent manufacturing systems. Afterwards a mapping of the processes, clarification of the objectives and determination of the requirements related to the use-case was performed. It was important to define the boundaries of the context system and the evaluation method before the system was built.

Based on this framework, a mathematical model, to assess how the negotiation will concur in the multi-agent system, will be created. A simulator will be created to evaluate this method. After the creation of the model and simulator, the relevance will be assessed by it's performance.

The operational requirements should be clear then, and the theory can match the business expectations. From this, future prospects can be concluded.

2.6.1 Evaluation Method

To test the final theoretical framework, a virtual simulation is to be created. The sensors/agents can be shown, including what they know and for example their gauge values. In such a simulation, the negotiation can be visualized, and proof for the (near-)optimal outcome can be shown. By having a few variable sliders on the display, a user might change some variables which would show an increase/decrease in negotiation and optimal solution time.

Depending on the eventual data source, two evaluation methods are possible. If a "real" data source is available, the Key Performance Indicator (KPI) of the business will be checked. By minimizing e.g. the consumed base and acid during the production process, an improvement of the new system can be concluded.

If a "real" data source cannot be found, a second model is to be implemented. Using this other, probably heuristic or Linear solution, a comparison between the new method and the other method can be conducted and evaluated in aspects of speed, quality solution, and dynamicity.

2.7 Research Questions

From the research goals and process, the following research questions are concluded:

1. How can energy and manufacturing companies use the AI concept of intelligent multi-agent systems (MAS) for the optimization of production planning incl predictive maintenance and/or process control optimization?
   (a) What is the optimal MAS framework for the optimization of production planning?
      i. Theoretical: Which negotiation techniques, communication protocols, knowledge models and hierarchy/coalition to optimize decision making/scheduling.

ii. Simulation: Compare the new framework to an old use-case using simulation results.

(b) What is the roadmap for other industries within the industry 4.0?

i. Decentralized systems vs Centralized systems

ii. Negotiation in manufacturing

Chapter 3

Literature Study & Theoretical framework

The manufacturing industry is and will be one of the wealth generators of the world economy. A shift towards a modular production process, the 4th industrial revolution (also called Industry 4.0 transition), results in a demand for products with high quality at lower cost while being highly customized. This results in new way of controlling the production. High-performing computing, the internet, universal access and connectivity, and enterprise integration all contribute. Overall the consensus is that only the companies that fully leverage the information, its availability, the ability to exchange it seamlessly, and process it quickly, are the companies that can meet the high demand of the consumers (Monostori et al., 2006).

The so-called agent-based computation is a solution for many of the problems that arise from this new trend. By having autonomous agents, who can address changes adaptively and are distributed in nature, intelligent solutions are available (Monostori et al., 2006).

In this literature review, an overview of the manufacturing processes and current agent technologies/solutions are given. Using such a decentralized agent solution is only optimal when certain process and hardware-wise requirements, are realised on the manufacturing side. We will conclude the chapter with an overview of the framework. Abedin et al. (2014)

3.1 Manufacturing processes

A new paradigm shift in the discrete manufacturing world requires a production that is competitive but also sustainable. Most of these solutions lie in the field of Cyber-Physical systems. A Cyber-Physical entity is one that integrates its

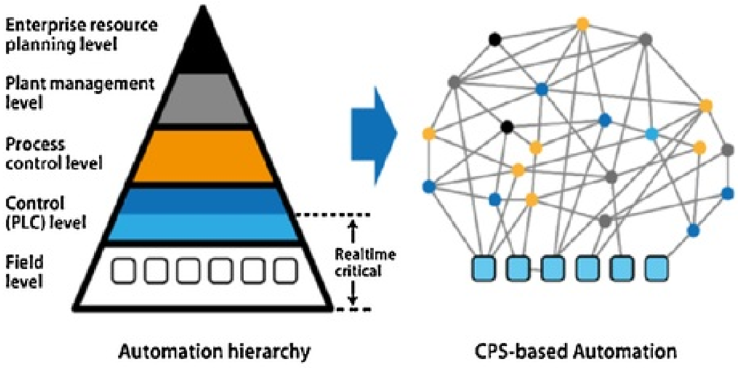

Figure 3.1: The breaking of the traditional automation pyramid, and future of a new more decentralized way of function. Image from Monostori et al. (2016).

hardware with a cyber-representation as a virtual representation. By doing so, it combines two worlds: the embedded systems and the software worlds. By doing so it breaks the traditional automation pyramid, and introduces a new more decentralized way of function (Leitao et al., 2016). This is also visualized in figure 3.1.

The traditional automation pyramid, is very similar to the multiple layers in the manufacturing process, which has been standardised by the American National Standards Institute (ANSI) (Harjunkoski et al., 2009). The integration of the planning and control in the manufacturing process is one that has many aspects. Below a short overview of manufacturing will be given in the ANSI structure. This goes from asset management using process control, to real time monitoring.

3.1.1 Asset management

Asset Management is the broad overview of the administration of assets. This includes the design, construction, use, maintenance, repair, disposal and recycling of assets. For most corporations and enterprises, the focus lies on the operational aspects of the assets, due to the fact that asset failures result in production or service delays. Therefore, insufficient asset management on one side results in loss of the asset itself, and on the other side loss due to productions delays and loss of service (Trappey et al., 2013). A lot is currently being researched, for example by Leitão (2009), on asset management, and especially the condition monitoring and prediction are in focus. This is due to the shift from reactive repair work to real-time condition monitoring, prediction, diagnostics and pre-scheduled maintenance. Also, traditional asset management approaches are poorly suited for current equipment failure solutions.

Traditional manufacturing control systems are unable to be sufficiently responsive, flexible, robust and reconfigurable due to the fact that they are built upon

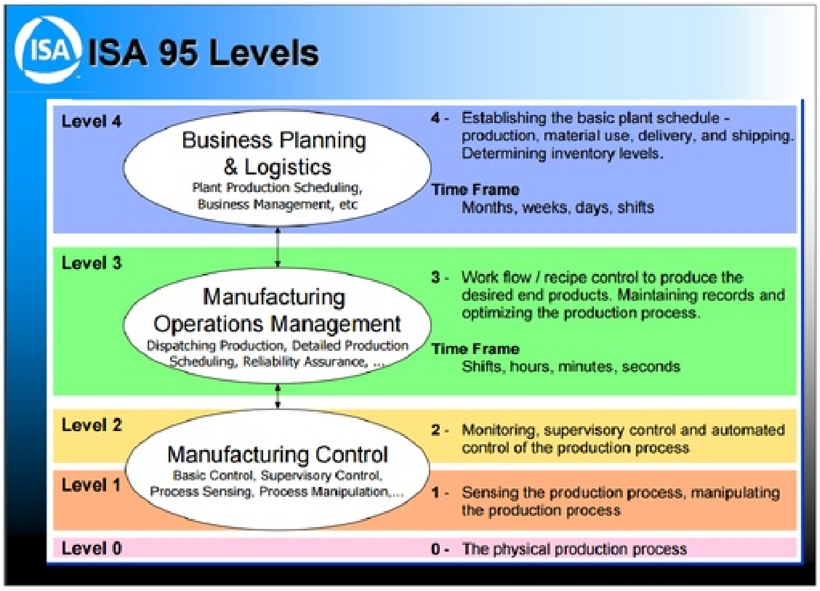

Figure 3.2: The manufacturing levels as described and defined by ANSI for the ISA-95 levels (Brandl and BR&L Consulting, 2008).

a centralised and hierarchical control structures. These are optimal for perfect optimization, but weakly responsive to change. Another consequence of this structure is that a single failure can shut down an entire system (Leitão, 2009). This requires a change to decentralized asset management, demanding for new process control methods.

Generally, researchers use agent-based technology to represent real world situations through the use of a computational simulation process, where agents can interact with each other to find a common goal. Typically, in these environments, agents have conflicting goals. In such circumstances, they will negotiate with each other in order to resolve conflicts (Rosa et al., 2009). These methods will be described in section 3.3.

3.1.2 Process control

There are three different processing methods: discrete, batch and continuous. Each process can be defined in terms of one or more of these methods. A discrete process method is when the production results in separate pieces. These are for example created in Industrial Robotic Solutions. Each robot produces a separate product in the manufacturing process. It is one of the most used manufacturing production application.

Batch production is when specific quantities of the materials have to be com-

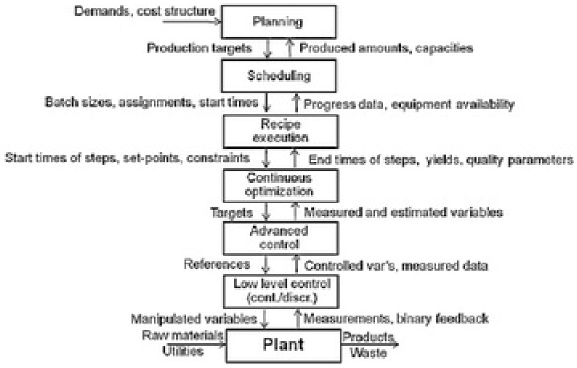

Figure 3.3: Typical process structure from Engell and Harjunkoski (2012)

bined in particular ways. These are typically food productions. An example is the beer production. In a specific batch, the ingredients are combined, and after a period we have our required product. The last process method is continuous production. This type of control is required if the variables are smooth and uninterrupted in time. The process of the creation of de-mineralized water is a continuous process. The water continuously flows through the system and finalizes in the required product with no interruptions.

An example from Engell and Harjunkoski (2012), which is displayed in figure 3.3 shows the typical process control method. This is in line with the ANSI standardisation described in the introduction.

Planning

Important when controlling a process is to optimize the planning. The forms of decision making used in optimization of planning play an important role in the performance of a production plant. By using different mathematical and heuristic methods, the limited resources can be correctly allocated. This optimization is essential such that the objectives and goals of a company are satisfied (or even better). By minimizing, for example, the time to complete the production, while satisfying the goals, efficiency is increased, which often results in cost reduction (Pinedo, 2005).

One of the largest difficulties when planning, is that of ensuring that the assets are always operational, or have (as short as possible) planned downtime. This is achieved with predictive maintenance.

3.1.3 Predictive maintenance systems

To prevent malfunctions, maintenance is necessary. However, this maintenance results in downtime, and is preferably left out, to keep operations running. This however results in the breakdown or wear-out of these systems. By using maintaining assets before they break by so called "preventive maintenance" this damage can be controlled.

The old fashioned model is corrective maintenance. Since maintenance results in the shut-down of production plans, most companies postpone the maintenance to the last moment possible. By ensuring to take as many hours as possible from the machine, the most is taken out of their investment. However, since the breakdown can happen any moment, they need a high inventory of spare parts and materials. And usually the repair is more expensive than maintenance.

Preventive maintenance is the alternative to corrective maintenance. Using predetermined fixed interval planned maintenance, the asset are maintained. However, this results in the not knowing whether maintenance is planned too early, or worse, too late. How can one be assured that the maintenance timing is optimal, due to the many factors of influence on the asset (wrong usage, or external surrounding like sun, dust and rain)? Often either maintenance is done to soon, resulting in extra cost, or to late which results in the breakdown of the asset.

Condition-based maintenance is a step in the right direction. By ensuring preventive maintenance on the right moment, the machines do not breakdown and there is no overkill on maintenance. On specific intervals, the machines are measured regarding their current status and using, for example, vibration measurements or oil samples, their current condition can be assessed. Parts that have a high probability of failure can be replaced in their next maintenance or production stop. However, this is not the optimal solution: measurements are sporadically done (not continuously) and there remains the chance of failure before the maintenance stop has occurred.

Using predictive maintenance it is possible to continuously, in real-time, monitor an installation. This can be done over a distance. Currently there are assets filled with sensors which produce data. This data is shared with people, other machines and servers. This allows for predication of failures, and real-time maintenance.

Currently a lot of research is conducted on this new form of maintenance (Muller et al., 2008). This central analysis is done by recognizing patterns in the data which allows for prediction of possible faults. This branch of maintenance is also known as e-maintenance (Yu et al., 2003), condition-based maintenance or intelligent maintenance (Vermaak and Kinyua, 2007).

3.1.4 Real-time Monitoring

To ensure that processes are running accordingly and continuous planning is applied, real-time monitoring is required. Essential in implementing a real-time plan or schedule is that it has to be generated in seconds on the available computer. This may be the case if rescheduling is required many times a day because of schedule deviations. This can be done in two ways. The first way is to review the overall processes and functions performed on the data in real time, or as it happens, through graphical charts and bars on a central interface/dashboard. The second method is by implementing a programmable logic controller. By automating the industrial electromechanical processes, in a predictable and repeating sequence by use of a logic ladder, a real-time controller is achievable.

Manufacturing with agents

When dealing with multiple processes, production and manufacturing wise, and have to keep real-time track of the assets with sensors, the most common solution lies in agent solutions Leitão et al. (2013); Monostori et al. (2016). This is often easier said than done. In the following section, an introduction in agent-solutions will be given with a focus on negotiation and manufacturing.

3.2 Agent Solutions

The new requirements in production ask for new manufacturing planning. This requires a new planning method, which is best implemented using distributed, decentralized structures (Parunak, 1999). The basis of a distributed method lies in Object-oriented programming (OOP) and Multi-Agent structures. Using these structures and negotiations planning can be optimized.

3.2.1 Object-oriented programming

Object-oriented programming (OOP) is a programming method based on the concept of "objects", which may contain data and code. For example an object can be a variable, a data structure, or a function, or a combination of these. The code that an object contains can be seen as the behaviour of the object, and as such it is easily interchangeable with an agent, since a method (or message) in OOP is a procedure associated with an object. An object is made up of data and behaviour, which form the interface that an object presents to the outside world (Shoham, 1993).

Agent-oriented programming is a method often used to implement a multi-agent system. In such a system anthropomorphic ideas, like beliefs, desires are used to model the objects, and thus called agents (Shoham, 1993). Agents will be discussed later in this overview.

3.2.2 Convex

A convex set is a region such that, for every pair of points within the region, every point on the straight line segment that joins the pair of points is also within the region. The boundary of a convex set is always a convex curve. And If a real-valued function defined on an interval is called a convex function if any two points on the graph of the function lies above or on the graph, in a Euclidean space (or more generally a vector space) of at least two dimensions. Equivalently, a function is convex if its epigraph (the set of points on or above the graph of the function) is a convex set. Well-known examples of convex functions include the quadratic function x^2 and the exponential function $e^x$for any real number x.

Convex functions play an important role in many areas of mathematics. They are especially important in the study of optimization problems where they are distinguished by a number of convenient properties. For instance, a (strictly) convex function on an open set has no more than one minimum. Even in infinite-dimensional spaces convex functions continue to be convex. Furthermore, convex functions are optimal when dealing with multiple convex sets, since the intersect will always be convex as well meaning that the properties of the new sets will

3.2.3 Multi-Agent Systems

Some terms used in the literature for data collection apparatus that aggregate the data are "Smart Objects", "Intelligent Gateways", "Collaborative Network", "Wireless Sensor Network" and "Industrial Agents". Most of these can be viewed as Multi-Agent Systems (MAS) where the sensors communicate with one another as decentralized intelligent agents for independent action performance depending on the context, circumstances or environments (sensor input) of the situation. From such MAS, Ambient Intelligence is conceivable: real-time decentralized decision making based on real-time data acquisition, analytics and negotiations. An example structure can be seen in Figure 3.4.

To define MAS, an agent needs to be defined. An agent is a system that is capable of independent action on behalf of its user or owner. As Wooldridge (2009) formulates it, "An *agent* is a computer system that is *situated* in some *environment*, and that is capable of *autonomous action* in this environment in order to meet its delegated objectives." This independent action execution is already a form of intelligence(Wooldridge, 2009). In the MAS the developer would most probably implement such intelligence by giving each agent "Beliefs, Desires, and Intentions" (Rao et al., 1995).

Multi-agent systems (MAS) have been identified as one of the most suitable technologies to contribute to the deployment of decentralized optimization that exhibit flexibility, robustness and autonomy(Vinyals et al., 2011). Currently there are a lot of relevant contributions regarding agent technologies to this emerging application domain. However, many challenges remain for the establishment of MAS as the key enabling technology (Vinyals et al., 2011). A few problems, like a lack of focus on multiple owners, decision making with only available local knowledge research and lack collective sensing strategies, are still subjects that require extensive research. They see these as the possibly most active MAS research topics. Many of these problems can be solved with negotiation, which will be covered later.

3.2.4 Holonic Systems

Multi-agent systems are composed of autonomous software entities *. They are able to simulate a system or to solve problems. In manufacturing the requirement linked to the real-time processes resulted in a new entity and control structure: Holonic systems (Giret, 2005). A holon, just like an agent, is an intelligent entity able to interact with the environment and to take decisions to solve a specific problem. Holon has the property of playing the role of a whole and a part at the same time. The first successfully implemented holonic structure was created by (Van Brussel et al., 1998). PROSA consisted of three types of basic holons: order holons, product holons, and resource holons. They were structured using the object-oriented concepts of aggregation and specialisation.

*The holonic structure used in our design will be explained in the chapter 4, including a visualization

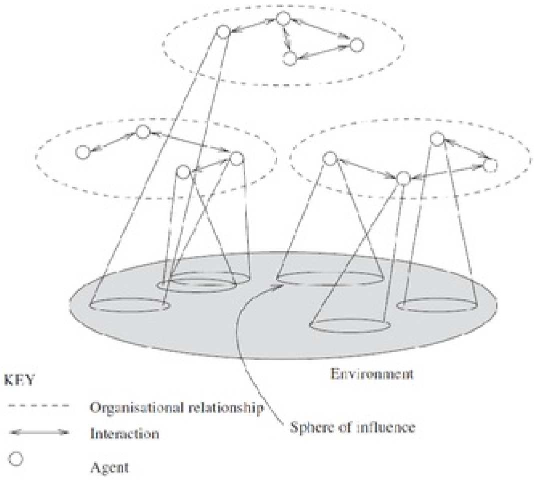

Figure 3.4: Typical structure of a Multi-Agent System (Wooldridge, 2009).

By decoupled the system structure from the control algorithm, logistical aspects could be decoupled from technical ones.

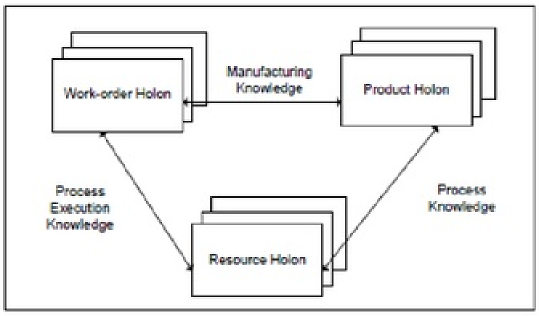

Figure 3.5: An example of an Holonic Manufacturing System (from Giret (2005), based on Van Brussel et al. (1998))

The concept of holon is based on the idea that complex systems will evolve from simple systems much more rapidly if there are stable intermediate forms than if there are not; the resulting complex systems in the former case will be hierarchic. Secondly, although it is easy to identify sub-wholes or parts, wholes and parts in an absolute sense do not exist anywhere (Van Brussel et al., 1998).

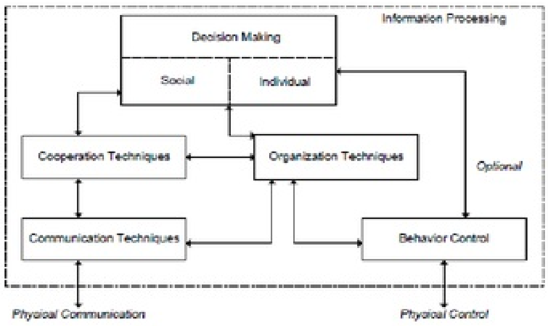

Figure 3.6: An Holonic agent based architecture Giret (2005)

3.2.5 Task/resource allocation

An example of resource allocation is when a set of agents shares a joint resource. Such a resource can be anything from indefinitely renewed continuous or discrete theoretical resource. By limiting the use of the resource to one agent at the time, negotiation is necessary to ensure that all the agents can use the resource. Usually and often crucial is the preference of the agents. Since the agents have different preferences regarding the resource, it is possible and feasible to divide the resource and create a schedule describing who has access to the resource and when (Fatima et al., 2014b).

The same principle applies to task allocation, where the agents want to achieve a common goal. To achieve this goal quickly the agents most divide different task, which overlap, and reach an agreement on the optimal planning.

3.2.6 Scheduling and planning

Since most Process Planning and Scheduling (PPS) problems are NP-hard problems, many MAS have also been deployed to "solve" such problems in reasonable time. NP-hard (nondeterministic polynomial) problems are those problems which are at least as hard as the hardest problems in NP. This means that it is possible to reduce the problems in NP to the original problem in polynomial time. Using the decentralized global optimization approach a (sub-optimal) solution can be found. This solution would be found faster than when using an (mixed) integer program (Feng et al., 2014). It does however depend on the practical application of the system to see whether it is an NP-hard problem. Furthermore, this does not guarantee an optimal solution, rather that a reasonable solution will be found in reasonable time.

Real-world scheduling problems are usually complex and involve many approaches to find sub-optimal rather than optimal solutions using reasonable computing resources. The Bus Maintenance Scheduling Problem (Zhou et al., 2004), which is distributed and dynamic in nature, has received less attention compared to scheduling problems in manufacturing. In the Bus Maintenance Scheduling Problem (Zhou et al., 2004) a MAS is proposed to heuristically solve the bus maintenance scheduling problem investigated here. It is shown that with equal optimality and less computing time without constraint violation it is comparable to the work of a mathematical programming approach.

It is also shown in Bruccoleri et al. (2005) that the agent based approach out performs the centralized mixed integer programming solution for the planning of a production.

Another example is the agile development with a MAS (Rabelo et al., 1999). Agile development is based on the idea that requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. Agile development promotes adaptive planning. By using a MAS for Agile planning, it has been shown that *"the scheduling agility can be extremely improved once it is based on the following key points:*

- *distributed and autonomous systems instead of the centralized and non-autonomous solutions;*
- *negotiation-based decision making instead of the totally pre-planned processes;*
- *application of different problem-solvers in the same environment instead of only one fixed problem solver;*
- *concurrent execution instead of the sequential processing" (Rabelo et al., 1999).*

Each agent is part of a heterogeneous system and processes its own information and has its own particular capabilities that it exchanges to the system. In this matter it contributes to finding a solution to the global problem which works very well in complex environments. Optimization of scheduling in such complex environments is highly constrained, with which advanced analytics also has great difficulty. Using the dynamic, flexible and intelligent relaxation of the constraints within the distributed knowledge of the agents, autonomous intelligent decision making as a Multi-Agent System is achieved (Rabelo et al., 1999).

3.3 Negotiation

Often discussed above is the negotiation of the agents in a multi-agent system. This branch of research, also called automated negotiation, is studied by Artificial Intelligence and Economics (Jennings et al., 2001). Concepts from fields such as decision theory and game theory can provide standards to be used in the design of appropriate negotiation and interaction environments (Jennings et al., 2001). It is used to reach an agreement that meets the constraints of two or more parties in the presence conflicting interests. And thus is a basic means of getting what you want from others (Fisher et al., 1987). It is back and forth communications designed to reach an agreement when you and the other side have some interests that are shared, and others that are opposed. Agents reason rationally and strategically. An agent's objective is to maximize the expected value of its own payoff.

The four components of a negotiation model are (Fatima et al., 2004):

1. The negotiation protocol.
2. The negotiation strategies.
3. The information state of agents.
4. The negotiation equilibrium.

Since negotiating situations occur when there is a conflict of interest, the first step will be to detect such a conflict. Agents will use communication channels and try to eliminate the conflicts. Conflicts may be about limited available resources, or may be a conflict between the beliefs of some agents. In the first case, optimization is the result, whereas, in the second case, one of the agents will have to change its beliefs (Shen et al., 2003). Often it is seen as maximizing the quality of the result. Two solutions are possible, one, the agents can try to achieve Pareto optimality, meaning that the outcome maximizes the product of the agents' utilities, or they try to reach a Nash equilibrium, meaning an stable state in the system, both which will be discussed in the evaluation method.

Negotiation is done by exchanging messages among agents. Since the process involves several messages, a discussion will take place in which each agent's belief and goals will be an important factor. These depend on the global situation. Clearly, to be able to negotiate, agents must be able to reason. Thus, negotiation is restricted to cognitive agents. Automated negotiation is essentially a distributed search in the space of potential agreements between the different negotiators represented by autonomous agents, which involves the exchange of relevant information and aims to find an agreement that is acceptable to all participants.

Negotiation domains, can be divided into task orientated domains (TODs), state orientated domains (SODs) and worth orientated domains (WODs). TODs are the simplest and an agent's activity is defined in terms of the set of tasks it has to achieve. It is assumed that all resources are available, the benefit of negotiation is the redistribution of tasks amongst a group of agents which results in a more efficient task order. A typical example is mail delivery where an agent may carry another agent's mail at little extra cost. It is sertain that the states come closer to a Pareto optimal solution as all agents can proceed with their original task list and be no worse off. SODs deal with problem where agents wish to change their environment from an initial state to some goal state. The classic AI Blocks World problem is a good example. There is the possibility of conflict and dead end, since the agents may have different goals, and it is not feasible to try and satisfy these goals all for all agents. In this situation, agents must be able to make concessions. WODs are domains where agents attach a worth to each potential state. This allows much more flexible goals to be set and allows concessions to be made on these goals. An example would be agents in an marketplace where the goal for a seller may be to obtain the highest price for x within time y. There is again the possibility of conflict and deadlock, but now within a more complicated bargaining environment (Anumba et al., 2003; Fatima et al., 2014a) .

3.3.1 Negotiation Protocol

Negotiation Protocol is the set of rules that govern the interaction and defines who are the actors of the negotiation, the states that characterize a trade (for example, when a negotiation has begins or ends), the events that determine the change of actors status, and messages that can be sent by the actors in a particular state. This however is no easy task, since there is no one size fits all solution. Some attempts have been made, by Marsa-Maestre et al. (2014) for example, and a collection of design rules which allow, given a particular negotiation problem, to choose the most appropriate protocol to address it. However, these problems are only derivable when (1) the negotiation domain, including the issues and possible issue values, (2) a scenario utility histogram, which defines the distribution of contracts in utility space, and (3) several structural parameters that specify the topography (e.g. ruggedness) of each agent's utility function are known. In the design of the system, this will be discussed.

The most important protocol is that of the alternating-offers protocol (Rubinstein, 1982) **Todo stein uitleggen** (1). It is based on a divisible pie, discrete or continuous, and is the most widely studied among game-theorist as well as MAS researchers (Fatima et al., 2014b). Other examples is the contract net protocol and the bargaining protocol.

A typical negotiation protocol is very similar to that of our negotiations in our everyday life and work. Thus, a negotiation typically proceeds over a series of rounds, with one or more proposals being made at each round. It also includes the rules that impose the constraints on the rounds and the rule that shows when a deal has been struck (Fatima et al., 2014b). Different negotiation mechanisms need to be developed to suit the different application environments of MAS. Unlike the negotiations between human beings which involve more complex human interactions than simple technical issues, the negotiation mechanisms between agents are rule-based or case-based. Yet, the human negotiation approaches and theories, which mainly include game theory and behaviour theory, provide sound bases for the negotiations between agents.

Another common protocol is the monotonic concession protocol. It is a proposal which has also been adapted for multi-lateral negotiation in (Endriss, 2006).

3.3.2 Negotiation Strategies

The strategy can be defined formally as an apparatus which allows the agent to determine the content of the action that it will perform consistently with the protocol. In general, for a given set of negotiation protocol there are many strategies compatible with it, each of which can determine a different action. This means that a strategy can work well with a given protocol, but does not work with others. So, the choice of strategy depends on the protocol in use and by the trading scenario (Di Nocera, 2015).

Often these strategies are private, meaning that not all the agents can see what the strategy of an agent is (Fatima et al., 2004).

Concession Strategy

Todo Hier concession uitleggen (2) When negotiating it is essential for the agents to concess. Initially each of the agents involved makes a proposal that has the highest utility to itself. If no concessions are made, the agents will never reach an agreement. By making concessions on the utility, a proposal towards the agents agreement-zone can be made, which is essential in finding an agreement. Furthermore as put by Endriss (2006): a concession should always be minimal with respect to the utility loss incurred by the agent making the concession.

There are multiple strategies when dealing with how to concede. Most of these are on or another way of a monotonic concession protocol, meaning that the desired utility of the agent will never increase. This means that the desired utility will always decrease, or stay the same. After each proposal there are 2 options. Either the agent refuses to make a concession and stick to proposal or it Make a concession and propose a new deal that is preferable to the other agent. The monotonic concession protocol is verifiable, and guaranteed to terminate. This is due to the conflict deal that would occur if no agents concede, which gives both agents a utility of 0 Endriss (2006).

An alternative is the one-step protocol Rosenschein and Zlotkin (1994) simplifies this matter. Here, each agent is allowed to make a single offer, and the proposal that yields the higher product of all the utilities of the agents accepted. The best strategy that agents can follow in this protocol is to propose the agreement that is best for themselves amongst those with maximal product of utilities. Essential however is that the utilities of the other agents must be known to ensure that the maximal product of utilities is calculated.

There are multiple options when dealing with concession to a multilateral negotiation (Endriss, 2006). Most of these have a relation to social welfare concepts, meaning that the agents will together try to maximize the utility of all the agents. This means however that the utilities of the other agents must be know since, it is not possible to discover the group maximum using private utilities.

Four concession strategies are given by Wu et al. (2009): An *amount of utility* meaning that an agent concedes a fixed amount utility each time. *Fraction of utility*, which means that an agent concedes a fraction of the desired utility. *Fraction of the difference* which means that the agent concedes a fixed fraction of the change in current desired utility and a reference point. And finally a *fraction of remains* which is a fixed fraction of the issues that no agreement has been made on yet. They found very little difference between the performances (distance from Pareto-optimum) of the concession strategies, although the fixed step was the quickest.

When dealing with private utility functions matter change completely. There is no way knowing whether the other agents has conceded. A solution to this is proposed by Zheng et al. (2015) using the reactive concession protocol.

Reactive Concession protocol The reactive concession protocol as proposed by Zheng et al. (2015) tries to solve the problem that occurs when dealing with private functions and ensuring that there is not one agent that does not stall its concessions. Zheng et al. (2015) show that by having an agent concider its own utility change resulting from another agents' offer, it can concede accordingly. There are two cases. If the change in utility that the other agetns' offer caused has resulted in an higher utility than the reservation utility, the agent will respond with the non-reactive concession strategy.

However, if the utility is lower than the reservation utility, the agent will concede by an amount based on the change the agent perceives. By checking the last best offer, the agent checks for the marginal perceived change of utility and the total perceived change from the original offer. This gives two values, of which the maximum will be the agents' concession.

3.3.3 Negotiation States

An agents information state describes the information it has about the negotiation game. There are two possibilities, states with complete information and those of incomplete information. The first category is basic and most common. In these games the players are assumed to know all the information about the rules of the game and the players their preferences. However, in the incomplete category, information may be lacking about a variety of factors in the problem (Fatima et al., 2004).

3.3.4 Evaluating /equilibrium solutions

When evaluating the dilemmas of a negotiation between agents, it is essential to know the Pareto-Frontier. Visualized in figure 3.7, it is used to determine whether an outcome of a negotiation is efficient. This means that no improvement can be achieved for all agents. In the figure we have the utility of agent$_i$

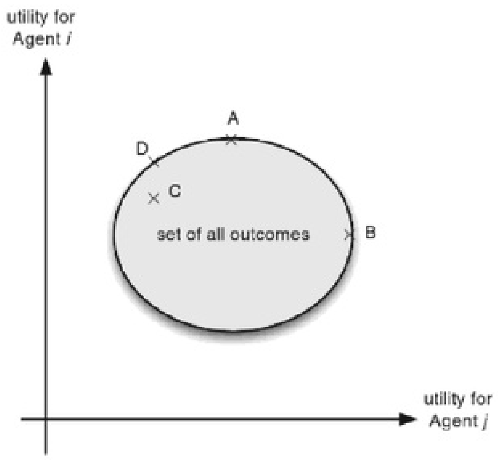

Figure 3.7: An example of Pareto optimal. Locations A and B are optimal, since no improvement, without loss for the agents, is possible. C and D are not optimal. From Fatima et al. (2014b).

Table 1 Major categories of game theory

	Static	Dynamic
Complete information	Static games of complete information: Nash equilibrium	Dynamic games of complete information: Subgame Perfect Nash Equilibrium
Incomplete information	Static games of incomplete information: Bayesian Nash equilibrium	Dynamic games of incomplete information: Perfect Bayesian Nash equilibrium

Figure 3.8: The four types of games in game theory from Trappey et al. (2013).

plotted against that of agent_j. In the set of all possible outcomes, lie all the possible agreements, which are all offers, that are acceptable by both agent. An offer is Pareto optimal if the agents can not choose an new offer for which at least one agent has a higher utility, while the other agents have the same utility. In the figure these are shown as A and B. Each other offer for these agents decreases at least the utility of one of the agents. Offers C and D are not Pareto optimal since both offers can be improved for at least one agent. The line of points in which an improvement of utility for one agent necessarily means a decrease in utility for the other agent is the Pareto-Frontier.

Formally: If we have agents $M = \{1, ..., m\}$, and issues $N = \{1, ..., n\}$ denoted as issue $j \in N$, than an offer $x = \{x_1, ..., x_j\}$ is Pareto optimal if the outcome of negotiation x has no feasible allocation x' such that $u_i(x') \geq u_i(x) \forall \text{agents} i \in M$.

The Nash equilibrium is the best reply to the other players strategies. This means that if both players play their Nash strategy, neither will have the incentive to change their method. Different equilibria are possible and shown in figure 3.8. The most common Nash equilibrium and probably most widely

A		i	
		defect (confess)	cooperate (¬ confess)
	defect (confess)	(-3, -3)	(0,-5)
	cooperate(¬ confess)	(-5, 0)	(-1,-1)

Table 3.1: The prisoners dilemma

known is that of the prisoners dilemma: There are two subjects of a crime, agent i and j. However, the evidence is not very convincing and therefore the prisoners are interrogated separately. If both confess, they get three years of prison. If both do not confess, they get a lighter sentence of 1 year. Finally, if one of them confesses to the crime and the other does not, the confessor will be freed, and the other will be jailed for five years. This can be visualized in a normal form pay-off matrix. It is common to refer to confessing as defection, and not confessing as cooperating. For agent i it is obvious to reason as follows. Suppose agent j cooperates. Then the best response is to defect. Suppose the agent j defects. Then the best response is to defect. In other words, defection for agent i is the best response to all possible strategies of the player j. This makes defection the dominant strategy for agent i. Since both agents reason the same way, the agent will both defect which results in the Nash equilibria (defect, defect).

In general, that two strategies s_1 and s_2 are in Nash equilibrium if: under the assumption that agent i plays s_i, agent j can do no better than play s_2, and under the assumption that agent j plays s_2, agent i can do no better than play s_l. Thus strategies s_i and s_j for agents i and j form a Nash equilibrium if they are the best response to each other (Wooldridge, 2009).

Important to note here is that in the prisoners dilemma, the Nash equilibrium is the only solution which is not Pareto optimum. THe optimal social solutions, which is if both agents do not confess, is different from the Nash equilibrium. Furthermore, it should be stated that not every interaction scenario has a Nash equilibrium and some interaction scenarios have more than one Nash equilibrium.

3.3.5 Principled Negotiation

An example of a common method for negotiation is principled negotiation. This method developed by Fisher et al. (1987) was founded on the idea that negotiators could reach better agreements by finding favourable agreements. By focussing on interests not positions and using objective criteria, an agreement is more likely to be reached. This method has successfully been deployed in a Multi-Agent System for air traffic management (Wangermann and Stengel, 1998). Emphasised is the fact that it is important to agree on objective criteria for assessing options (Fisher et al., 1987). If an agreement can be reached using this criteria, it is more likely that it is rational. Furthermore it is useful for systems in which no agent has global knowledge of the system.

The purpose of this type of negotiation is to help to reach agreement without jeopardizing the business relations. It was created by Fisher et al. (1987) and they refer to this kind of agreement as a wise agreement. Wise agreement is agreement that meets the interests of both parties to the extent possible, is long lasting, and also considers the interests of the larger society. The basis of this negotiation principle is to separate the relationship issues from the problem issues, to focus on interests not on positions, while trying to be creative in developing solutions.

3.3.6 Hierarchy and Voting

Voting is a form of group decision making. The agents participating in the voting will take into account their own preferences as well as those of others when making decision about how to vote. This will often have a strategic flavour. By aiming to rank or order the candidates, a group decision can be made.

Another option are auctions, a popular mechanism to reach an agreement within the allocation of resources to agents. Examples include English auctions, Dutch auctions, Vickrey auctions and First-price sealed-bid auctions (Wooldridge, 2009). Interaction between a large number of low-level agents results in a complex system behaviour which is difficult to understand, to control and to predict. Structuring the agents in a hierarchy is the appropriate solution to tackle this complexity (Van Brussel et al., 1998).

3.3.7 Mapping of negotiation protocols

An attempt at the visualization of the different negotiation techniques is strived at. Three variables are decided on. Single- vs Multi-Issue negotiation; bi- vs multi-lateral negotiation, and; perfect vs imperfect negotiation.

Single-Issue

Negotiation among self-interested agents has been studied from the perspective of game theory. This is most obvious when the agents negotiate on single issues. An example might be the price of a product. When dealing with a single issue there is only one goal for both agents and there must be a conflict. If there was no conflict, no negotiation would be necessary. Typical single issue methods are patient vs impatient players, two sided matching. Argumentation based methods, which are based on the beliefs of an agents are also included in the mapping.

Essential is that all these methods are a form of the alternating offers protocol. Depending on the sort of players, the method result in completely different

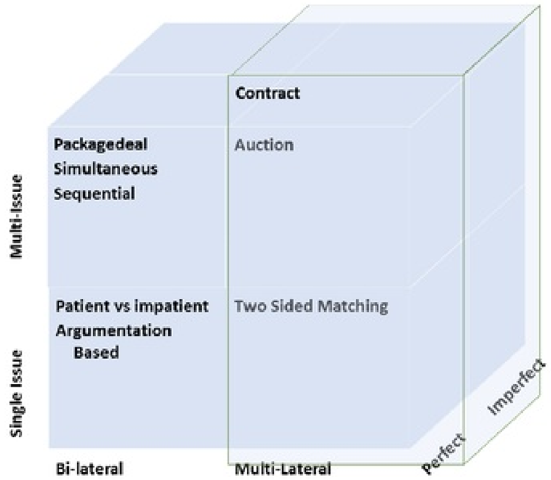

Figure 3.9: Current Negotiation overview

behaviours. These negotiation can either be complete or incomplete meaning that all information is known, or not all.

When the game is complete, all the agents know all the information about their states and the strategies of other agents. When not all is known, the game is incomplete. The idea of negotiation is that we have an incomplete game, since if the strategies are known, most negotiation would not be necessary.

When looking at perfect vs imperfect information, it means that either the information states of the agents is perfect, meaning that the agent is perfectly informed of all the events that have previously occurred and actions (like chess), or that not all actions are known. Depending on the implementation of the system, with for example public and private announcements, the difference is made.

When looking at single issue negotiation, depending on whether the negotiation happens between 2 (bilateral) or more (multi-lateral) agents, there are a few protocols possible. Bilateral negotiation can be either patient or impatient (Fatima et al., 2014a) meaning that an agent has a initiative to limit the time of negotiation. Most negotiation in the manufacturing are time restrained, thus impatient agents must be implemented (Kraus et al., 1995). In symmetric vs asymmetric the players are uncertain about the other players utility functions (as is the case in imperfect negotiation), but essential is that one agent might know more than the other in the asymmetric protocol.

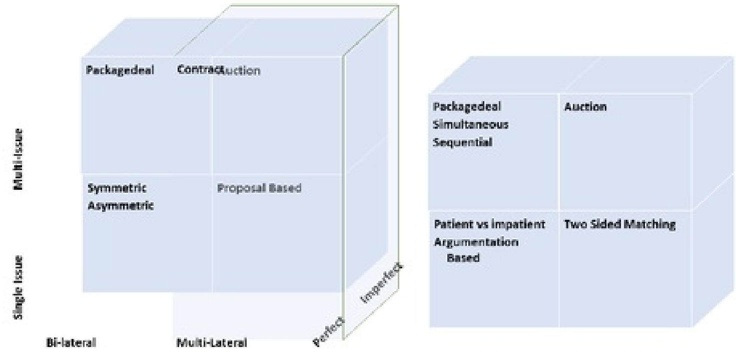

Figure 3.10: Current Negotiation overview

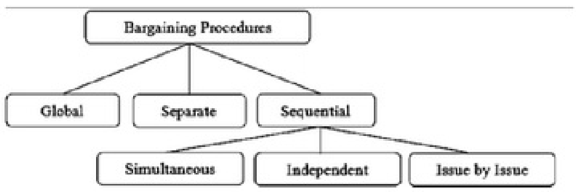

Figure 3.11: An overview of the different negotiation method for multi issues bargaining. From (Abedin et al., 2014).

Multi-Issue

When negotiating multi-issues, agents attempt to combine 2 or more issue in their discussing. An example is the typical seller, buyer relationship between two agents, as for example shown in Schramm and Morais (2013). Here a supply chain construction company is used to asses an method to support bilateral negotiation. Aspects like price, quality and lead-time are considered as issues, on which can be negotiated. Most used multi-issue method, for single-lateral negotiation is the package deal method. In this method, complete packages with all the issues are provided. These can be discussed either sequentially or simultaneously.

Agents can employ either an issue-by-issue (one-at-a-time) approach, or a packaged approach in the negotiation agenda (Fatima et al., 2004). In Abedin et al. (2014) a packaged approach for this is lack of knowledge about the opposing agents. As one issue is settled, the agent subsequently negotiates the other pending issues. This allows the agent to be cautious and opportunistic at the same time. For a multi-issue negotiation under incomplete information settings, the ideal solution is one that is Pareto optimal. A solution is said to be Pareto optimal if no agent can be better off without sacrificing the others utility as will be discussed in the evaluation.

When choosing the preferred method of negotiation, important to realize is the solution required. As explained above, the issue-by-issue approach has a higher chance of optaining the Pareto optimal solution. Furthermore, the majority of the existing work on multi-issue negotiations focuses on the negotiation strategy, assuming the agenda and the procedure to be predetermined (Fatima et al., 2004; Lai et al., 2004). Interesting to determine would be the influence of the domain and protocol since, depending on the scenario under which the negotiation is taking place a supervised agenda procedure can have a positive impact on the outcome of the negotiation when compared to a procedure without use of an agenda. (Abedin et al., 2014).

Multi-lateral

The most common used method for multilateral negotiations are contract based methods, most popular being the contract net protocol. Contract net protocol by Smith (1980) is based on the principle that agents, each with a distinct expertise, can solve sub problems that are required to solve the global problem. This form of cooperative distributed problem solving is based on the assumption that agents in a system implicitly share a common goal, and thus that there is no potential for conflict between them.

Each agent (manager) having some work to subcontract broadcasts an offer and waits for other agents (contractors) to send bids. After some delay, the best offers are retained and contracts are allocated to one or more contractors who process their subtasks. The contract-net protocol provides for coordination in task allocation.

The protocol is best suited to problems in which it is appropriate to define a hierarchy of tasks. Such problems lend themselves to decomposition into a set of relatively independent subtasks with little need for global information or synchronization. Individual subtasks can be assigned to separate processor agents. The main contribution of the contract net protocol is the mechanism it offers for structuring high-level interactions between nodes for cooperative task execution. Negotiation can be used at different levels of complexity. At one extreme, it is a means of achieving task distribution with distributed control and shared responsibility for tasks to maintain reliability. At the other extreme, the twoway transfer of information and mutual selection attributes of negotiation make possible a finer degree of control in making resource allocation and focus decisions than is possible with traditional mechanisms (Smith, 1980).

Since the contract net protocol has the uncertainty of matches being stable, the protocol of two-sided matching has been developed. Furthermore it is not certain that the matches are Pareto optimal. Using the two sided matching method, this uncertainty can be avoided, however, this protocol is harder to implement due to the fact that a clear allocation division is required. (Fatima et al., 2014b).

If the game is imperfect two sided matching does not work, and a proposal based protocol is the right fit (Rahwan et al., 2003).

Example of the three agents and three resources (3)

Heuristic methods in negotiation

Most of the negotiation in manufacturing can be seen as multi-lateral multi-issue negotiation. Three important distinctions are to be made, based on Lai et al. (2004).

1. issue by issue negotiation;
2. multi-issue cooperative negotiation;
3. multi-issue negotiation with heuristic methods.

The first aspect looks at the agreement which is built through a strategy, and examines this individually and interactively, and the parties are considered as non-cooperative and they are built for environments with incomplete and asymmetric information, where an agenda containing the order in which issues are treated as needed. For the second aspect a multi-issue concession strategy is used whose parties are considered cooperative and they have complete and symmetrical information about their environments. These two aspects have been discussed in the sections above. In the last type, an agreement is reached through a hybrid negotiation strategy, which uses the first two types of theoretical framework with the focus in automated models based on autonomous agents for multi-issue negotiation and in negotiation strategies tractable. This is also where possible learning methods are available (Schramm and Morais, 2013).

These heuristic methods are a lot more common in the implementation of negotiation, as discussed by Leitão et al. (2013); Monostori et al. (2006), since it does not require the through analysis of the states and protocol compared to the game theoretic methods. Also it allows for larger groups and learning in the agents.

Learning methods in Negotiation

When dealing with heuristic methods for negotiation, learning methods can be implemented. An overview can be seen in figure 3.12.

Based on the research conducted on heuristic methods and Jennings et al. (2001), it can be concluded that the optimal research in learning in heuristic methods is not yet known. They are often used however to decide on the optimal counter bid in Beheshti and Mozayani (2014). They show that efficient learning algorithms based on an statistical ranking algorithm and linear regression, all with linear time complexities. These characteristics allow our method to be used in real-world applications.

Table 1
Negotiation approaches in MAS based on the learning methods

Approach	Characteristics
Probabilistic Decision Theory	Selecting optimal decision
Possibilistic Decision Theory	Selecting optimal decision
Bayesian learning	Learning negotiation partner's type
Possibilistic CBR	Selecting most prospective negotiation partners
Constraint-based reasoning	Finding a solution that satisfies constraints of negotiating partners
Heuristic search	Determination of negotiation offer
Q-learning	Searching the set of potential strategies
Evolutionary computing	Searching the set of potential strategies

Figure 3.12: Overview of different learning methods for heuristic negotiation methods from Beheshti and Mozayani (2014)

3.4 Negotiation in Manufacturing

There are many applications of agent based solutions in the manufacturing world (Monostori et al., 2006). In these negotiations an overwhelming aspects is realised in the creation of intelligent individual agents, and less on the overall intelligence of the system. Often ignored is the specific negotiation method in these systems. This is where the problem lays, since conflicting interest, essential in the optimal decision making are left out. An example where the conflicting interest are well implemented is in Zheng et al. (2014). A cloud consumer usually prefers a high reliability, whereas a cloud provider may only guarantee a less than maximum reliability in order to reduce costs and maximize profits. If such a conflict occurs, a Service Level Agreement cannot be reached without negotiation. Automated negotiation occurs, when software agents negotiate on behalf of their human counterparts. However no learning occurs.

Rockwell Automation uses agents in its automation processes and is one of the industrial leaders in the implementation of agent based solutions (Vrba et al., 20[illegible] One of their future insights in the requirements of agent based solutions is to enhance the capabilities of agents for expressing and exchanging knowledge, and as a consequence, to increase the flexibility of control systems. In order to correctly do so, better insights in the negotiation is needed.

Overall, nearly all factory scheduling negotiations use some form of these market-based approaches (Monostori et al., 2006) to implement the solutions. Different version of the contract net protocol were used or other auction based methods. The problem with these methods is that no reasoning about another's interest and desires is achievable. If this is known, more efficient and better systems can be achieved. It is however shown in Bruccoleri et al. (2005) that the agent based approach using market auctions out performs the centralized mixed integer programming solution. This system uses bilateral simultaneous negotiation on the medium level of the production plant. It is however a form of auctions, where the agents simultaneously bid towards the goal. If this system already outperforms a centralized system, a non-auction based method might outperform even better.

Other examples of negotiation in a Multi-Agent System have been deployed in Smart Grids for optimal energy delivery (Pipattanasomporn et al., 2009), the collaborative design of light industrial buildings (Anumba et al., 2003), negotiation in an electronic market of water rights, and for example in the scheduling of Agile software development (Rabelo et al., 1999).

From the above, in comparison with the knowledge obtained, there are two gaps. Firstly, little multi-issue multi-lateral strategic (game theory wise) application has been implemented. An example from the theory is Wu et al. (2009) where a Pareto-optimal-search method for three-agent multilateral negotiation is developed. This however has not been implemented in any real usecase, and would be very interesting to implement. The other gap in the literature is the research into the optimal learning methods for heuristic methods. In de la Hoz et al. (2015) a wireless surveillance sensor networks is optimized using heuristic learning methods. This is limited to a bilateral negotiation protocol with a mediator, where negotiating agents (two access providers, each of them controlling a fraction of the access points in the scenario) negotiate. No multilateral application has been attempted. An attempt at generalizing multilateral heuristic learning has been made in Beheshti and Mozayani (2014), but this has not been applied to a real use case as of yet.

The last option, an application of multilateral heuristic learning, is the best fit on the usecase.

3.5 Framework

A framework to compare a centralized vs a decentralised solution is discussed here. Essential in the difference between these two possible solution spaces is the location of the processing power for the calculations. Centralised solutions have a single control unit where the information flows to, while decentralized solutions do not have this structure.

A popular comparison, discussed in Parunak (1999), is that of the original Roman army structures. Decisions where made at the top and dripped down, while the information stream went up. This method, has been deployed into most companies. Due to the fact that something can be computed on a single computer, and be optimized on this single program, an optimal can be found.

However, the increasing complexity of computer and information systems, combined with the increasing complexity of their applications exceed the level of conventional centralized computing. This is due to the processing of huge amounts of data, or data that origin from (geographically) distinct locations. To solve such difficulties computers have to act more as agents where each agent can solve part, or decides on part of the problem. This is where agent-based architectures are an ideal fit to such an organizational structure.

To push the decision making to the lowest level, excessive layers of management can be absolute. This allows for, sometimes, easier to understand and developing of problems, especially if the problem being solved is itself distributed.

By using principles of decomposition which is a classical optimization (reformulation) method Sharif and Huynh (2012) presents a comparative study of two contrasting approaches for modelling the yard crane scheduling problem: centralized and decentralized. It seeks to assess their relative performances and factors that affect their performances. They conclude that a centralized approach outperforms the decentralized approach by 16.5 % on average, due to having complete and accurate information about future truck arrivals. However, since the decentralized under performs the centralized, the decentralized approach can dynamically adapt to real-time dynamic changes, making it better suited for real-life operations.

To optimize these different types of resources allocation problems, there are different kinds of allocation problems, for which different solutions are feasible. The purpose here is to find what characteristics are optimal to use a centralized vs a decentralized solution.

3.5.1 Size and modularity

A **To do** Framework weghalen (4) critical aspect of the possibility to determine whether a centralized or decentralised solutions is preferred is the search space size of the problem. The size of the problem is seen as the number of resources that have to be allocated. If a clear structure is conceivable and a clear population is in place a centralized solution is feasible. This is due to the global overview .

In a decentralized structure, individual models are decoupled from one another, errors in one module impact only the modules that interact with it, leaving the rest of the system unaffected. This can be seen in figure 3.13. It shows however the importance of having a clear modular problem.

gure was created by Seiichi Yaskawa of Yaskawa Electric Corporation, Tokyo, Japan, and is used with his kind permission.)

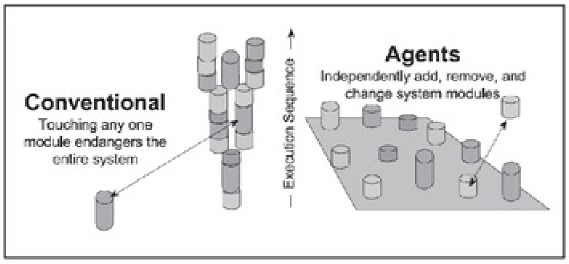

Figure 3.13: Comparison of a conventional control thread and an agent based, from Parunak (1999).

3.5.2 Dynamicity (Time Scale/Changeability)

In a centralized solution, the continuous monitoring of the state of the environment and typically the lack of complex decisions, a quick reaction to changes is possible. A high dynamical is the result.

Unfortunately, it is difficult to achieve real-time scheduling in traditional manufacturing systems because the scheduling algorithms used are executed on a single, centralized computer that becomes a computation (Duffie and Prabhu, 1994).

3.5.3 Solution Quality

Since agent-based approaches are distributed, they do not have a global view of the entire state of a system. A lot can reached through communication and negotiation, but for a truly optimal solution an entire view is necassary. For example, (Palmer et al., 2003) shows that this algorithm is not intended to find the optimal solution; it finds a good solution with less computation.

In the centralized approach the assumption of a complete information on supply and demand is made. This requires rescheduling to adapt with changes.

In the decentralized approach, no assumptions on the are necessary.

3.5.4 Complexity

Requirments are Simple relations.

	Centralised Solution	Decentralised Solution	Building Blocks
Size / Modularity	Small; No sub-problems	Large; Ill-Structured; Easily dividable; Independent Modules	Population; Holonicicity; # of resources: (decision variables, parameters & constraints)
Time scale / Changeability	Days - Weeks; Not subject to a lot of change	Real-time - Hour; Changeability	Adaptive Capability ; Degree of Re- and Pro-activeness
Solution quality	Perfect	(sub-)Optimal	Object and Solution Space
Complexity	Simple	Complex	Interaction between the set of elements; Communication

Chapter 4

Research Design & Application

As discussed in the literature review (Chapter 3), this research has a focus on the negotiation of agents. Using different methods and techniques, attempt is made to optimize a production process. The agents attempt to find an approximate-optimal solution while the optimal solution is unknown to the group.

The system in which the decentralized solution will be applied is a de-mineralized water plant as described in the introduction and problem chapters (Chapter 1).

As explained in the literature, a negotiation problem can be characterized by a negotiation domain (who negotiate and what do they negotiate about), an interaction protocol (which rules govern the negotiation process) and a set of decision mechanisms or strategies that guide the negotiating agents through every phase of the interaction protocol (Fatima et al., 2014b).

For the scope of this work, we assume a multi-attribute negotiation domain, where a deal or solution to the problem is defined as the set of attributes (issues), and each one of them can be in a certain range.

The coding will be done in Java from scratch. Multiple open-source systems are available, including Jadex, which is perfect for communication research in a Multi-Agent System (Kravari and Bassiliades, 2015). However since most of these systems are very comprehensive, little adjustments which are necessary for our research are not conceivable.

4.1 Demineralization of water

As discussed in the introduction, the usecase for which a Multi-Agent system for production will be implemented is a water demineralization plant. An application of this new model will be applied to a large plant that creates de-mineralized water. By removing all the ions from common water, de-mineralized water is obtained. This water is used for many processes and has many applications. In this plant specifically, it is used for the steam turbines, which generate electricity. By burning the by-product, heat is generated, which creates steam to power the turbines.

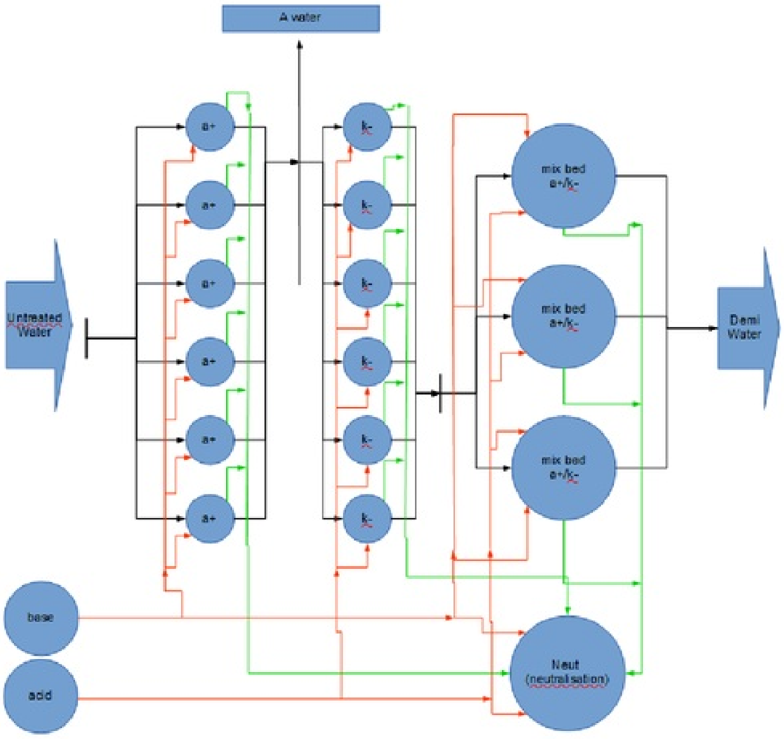

Figure 4.1: An overview of a water demineralization plant. We simplify the many anion and cation filters to a single anion and cation filter. The same for the mixbed filter.

Todo Make letters larger (5) Minerals are removed from water by multiple production steps. The most common method, which is implemented in the plant described, is to first remove the positively charged minerals in so-called anions. After this the negative charged ions are removed within a cation filter. To ensure that all ions are removed, a final combined "mixbed" is used. Here a combination of an anion filter and a cation filter removes the residues.

These filters have to be cleaned every few hours to ensure that proper demineralization occurs. For cleaning, acid and base are used. By filtering the anion with base, the ions that have been retrieved in the filtering are flushed. The residue, still of a base composition, is stored in a storage tank where the combination of the base and acids from the filters is neutralized. This storage tank is called the "Neut" as short.

So overall, we have three kinds of filters, the anions, the cations, the mixbed and the residue storage tank: the Neut. Each of the filters is exemplified by multiple items, but for simplification we only look at the allocation of resources. So overall there are 4 agents: 1. "Anion", 2. "Cation", 3. "Mixbed", 4. "Neut". Within each agent, the right amount of resources will be allocated. This will be done by negotiation and combined with the information that is retrieved from the experts knowledge.

4.2 Negotiation model

These four rational agents, $m = \{1, 2, 3, 4\}$, partition three issues $n = \{1, 2, 3\}$: the water that has to be delivered at the end of the process, and the base and acid for cleaning. This can be simplified to a buyer-seller negotiation.

Anion wants to buy as much base as possible, while it wants to minimize the amount of water. For the cation, as much acid as possible is required, while still as little water as possible should be produced. The mixbed requires as much as possible acid and base for cleaning. Since it the final production step, it requires the water to be delivered, forcing it to obtain as much water as possible from the cation. Finally there is the Neut, which wants as little base and acid as possible. Also the base and acid should be levelled out as much as possible to attempt to stay as close to a pH of 7 as possible.

Each of the above issues is translated to a unit interval $[0, 1]$ in $\mathbb{R}$. Since we have 3 issues, the unit hypercube $[0, 1]^3$ in $\mathbb{R}^3$. This results in a possible negotiation domain $\Omega = [0, 1]$ per issue.

The utility function for each agent $u_i(x)$ is convex, as described in Section 3.2.2, and normalized between $[0, 1]^3$. Each agent has a reservation utility ru_i. Any offer below this reservation utility is unacceptable. This means that the set of feasible offers is $A = \{x \in [0, 1]^3 \mid u_i(x) \geq ru_i\}$. Since the function is convex, A is also convex. The solution, if it exists, lies in the intersection of feasible offers, $\mathbb{Z} \cap_{i=1}^{M} A^i$.

The protocol used is that of the alternating offers protocol, based on (Rubinstein, 1982) as described in Section 3.3. The agents will propose in a fixed sequence, where the new offer is based on all previous offers given in the previous round. If all agents accept a current offer, the negotiation ends. The overall protocol is based on (Zheng et al., 2015).

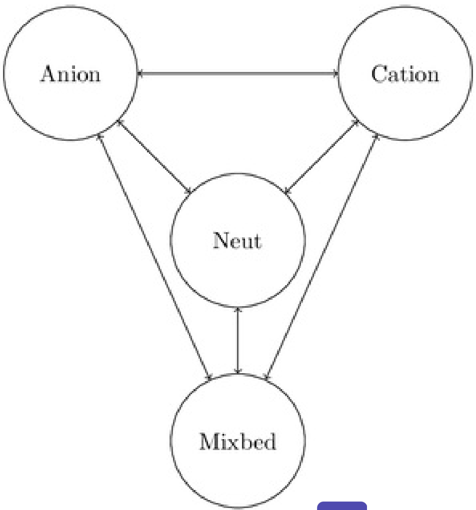

Figure 4.2: A simplified representation of the agents in the negotiation. See Figure 4.1 for the overview of the water plant on which the model is based.

It is a difficult task to determine whether the solution is a proper one, since the agents only have knowledge of their own utility function, which is private.

The negotiation takes place in rounds $n \in \mathbb{N}$. $x_i \in [0,1]^3$ denotes a bid of agent $i \in m$ in a round and $x_j \in x_i$ denote the amount of issue $j \in n$.

4.3 Details of the Agents

Each agent has its own characteristics on which the system will run. As shown in the introduction, the agents have difference preferences regarding the allocation of the resources. An offer has a different utility for each agent.

When the negotiations start, each agent will attempt to obtain as much utility as possible. Thus, when negotiations start, they propose the offer with the highest utility for them. During the negotiation, the required utility slowly decreases. An agent i's concession strategy is defined as the series of the agent's desired utility at time $0, 1, ..., t$. This is written as $s_i(0), s_i(1), s_i(2), ..., s_i(t)$. This is a monotonically decreasing function of the time/rounds t. The concession strategy s, is agent dependent. However, this utility will always be larger than the agent's minimum utility, or reservation curve: $s_i(t) \geq ru_i \forall t$.

While the utility decreases each round, the set of feasible offers increases as well. This means that at time t, agent i has the convex set $A_t^i = \{x \in [0,1]^3 \mid u_i(x) \geq s_i(t)\}$ as possible offers. Since $s_i(t)$ is monotonically decreasing, this means that $\forall t, A_1^i \subseteq A_2^i \subseteq ... \subseteq A^i$

4.3.1 Reactive concession strategy

The reactive concession protocol as described by Zheng et al. (2015), and also shown in Section 3.3.2, is a specific concession strategy to determine the amount of utility to concede. The non-reactive concession follows a predefined concession strategy $s_i^0(1), s_i^0(2), ..., s_i^0(T)$, where T is the maximum amount of rounds. The reactive concession, $(s_i(1), s_i(2), ..., s_i(T))$, amount is determined by each agent by the utility change resulting from other agents' offers. There are two options for an agent to concede. Either the change in utility that the other agents' caused is above the reservation function. In this case, an agent does an concession according to the non-reactive concession strategy. However if the change is below the reservation function the agents concedes by an amount based on the change the agent perceives in its own utility. This is shown in Algorithm 1, Line 9 to Line 16. **Todo explain reactive concession!!** (6)

Offer generation

When an agent i has offered $x_i \in [0,1]^3$, the next agent j will either accept the offer x_t^i, or create a new offer x_{t+1}^I.

To create the new offer, the agent will project offer x_t^i on the border of its feasible offer set A_t^i. This boundary, or indifference curve, is the set of points for which the utility is equal to the desired utility of agent j. The projection can be done using convex projection techniques as explained by Boyd and Vandenberghe (2004). In this method, a linear indifference curve will always be obtained, making convex projection calculations unnecessary. This will be done to ensure a high speed run-time, since the state of the world will change quickly.

4.3.2 Anion

The anion is the first filter where the untreated water will arrive. It needs base to clean the filters after water has been produced. The water that is produced by the anion filter will flow to the cation filter. The anion and cation filter both have a low interest in the production of water, and thus do not need to negotiate which each other.

Utility function

The utility function for the anion filter is set up as follows:

$$\text{Anion utility} = \frac{e^{-W+B}}{e^1}$$

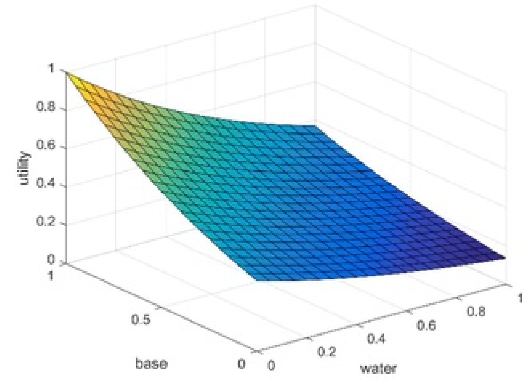

Figure 4.3: The utility function for the Anion.

where W is water ranging over ([0,1]) and B is base ranging over([0,1]). The division by e is done to ensure normalization. The utility is in [0,1]. It is visualized in Figure 4.3.

This function has been obtained after talks with industry experts. They confirmed that a high base wish, and low water wish is in line with the anion's utility. A consequence of this function is that it can be expressed as $(\ln(u) + 1) = -W + B$. This means that if the required utility is known, an indifference line is the result, as visualized in Figure 4.4.

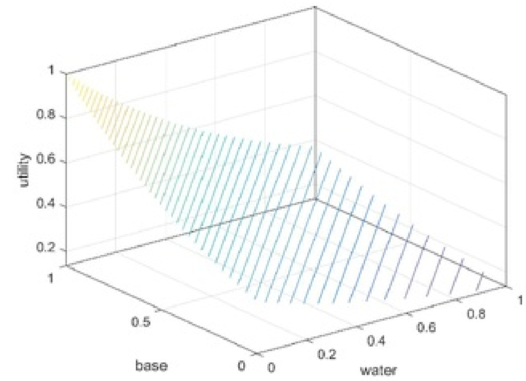

Figure 4.4: The visualization of the indifference curves for the Anion agent.

The reservation curve can be set as the curve where the utility = 0.3 e.g.. This means that any offer on the line $0 = B - W - (\ln(0.3)+1)$, or above, is acceptable for the anion in the stages of negotiation.

If the indifference line is know, a projection on this line is possible. At t, the agent i (anion) has the required utility s_t^i. Suppose $s_t^i = 0.4$ This means that there is an indifference curve $0 = B - W - (\ln(0.4) + 1)$. Suppose that offer $x(t-1)$ contains an offer for base 0.4 and water 0.7 (we can disregard the acid offer in this example, since the Anion has an indifference for the amount of acid used).

A projection to the indifference curve of the Anion is achieved with the following formulae. It has been thoroughly proven that if function is written as, $\mathtt{a}x + \mathtt{b}y + \mathtt{c} = 0$,the distance is calculated as follows:

$$\text{distance}(ax + by + c = 0, (x_0, y_0)) = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}}.$$

From this we get the point of the line, which is the line with the minimal distance:

$$x = \frac{b(bx_0 - ay_0) - ac}{a^2 + b^2} \text{ and } y = \frac{a(-bx_0 + ay_0) - bc}{a^2 + b^2}$$

This has been proven multiple times, for example by algebaric proof by finding the line that is perpendicular to the original line, and b/-a (the negative reciprocal) by letting it pass trough the point (x_0, y_0). We find that the distance is indeed as shown above, with the minimal distance point x and y.

The above is important since Zheng et al. (2015) has proven that if the projection is the point with the shortest minimum Eucilidean distance, the algorithm will converge for any concession strategy!

So given the situation above, we have the function $\mathtt{a}x + \mathtt{b}y + \mathtt{c} = 0$ where $\mathtt{a} = 1, \mathtt{b} = -1, \mathtt{c} = -(\ln(0.4) + 1)$. Given the information above, $x_0 = 0.4$ and $y_0 = 0.7$ this gives us the solution of the new $x = \frac{2.1+\ln(0.4)}{2}andy = \frac{2.1-\ln(0.4)}{2}$, which is on the indifference curve of the anion for an utility of 0.4. The above is visualized in Figure 4.5.

The projection is formally written as $P_A[x]$, which is the projection of point x on set A.

Important to note is that if the utility of the new point is larger that the desired utility, (the point is above the indifference curve), the proposal will not be accepted.

The advantage of using a linear indifference curve is that we don't have to deal with an minimization function ($P_A[x] = \arg\min_{\forall q \in A} ||q - x||$) which increases speed dramatically.

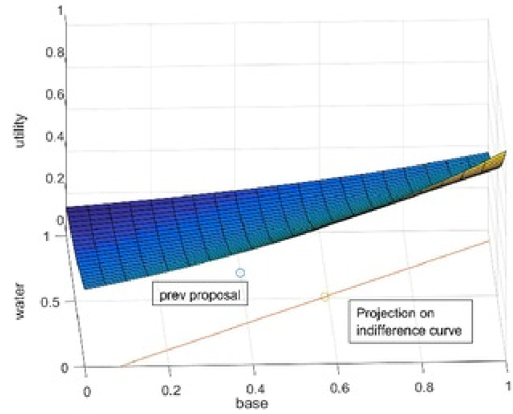

Figure 4.5: Example of the projection of a point to the indifference curve for an Anion. The utility here is 0.4, and the offer is base = 0.4; water = 0.7.

4.3.3 Cation

The cation is the second aspect of the water cleaning process and is the siode where the positively charged ions are removed. It cleans itself with acid.

The utility is very similar to that of the Anion, but a preference over acid instead of base is required. This results in the function:

$$\text{Cation utility} = \frac{e^{-W+A}}{e^{1}}$$

where W is water ranging over ([0,1]) and A is acid ranging over ([0,1]).

The reservation curve for the Cation is very similar to that of the Anion. The only difference lies in the requirement for acid instead of base. It can been set as the curve where the utility = 0.3. This means that any offer on the line $0 = B - W - (\ln(0.3) + 1)$, or above, is acceptable for the cation.

4.3.4 Neut

The Neut is the agent responsible for the allocation of the amount of acid and base. Since it wants to stay as pH-neutral as possible, it requires an even distribution of base and acid between the agents. This is not achieved in the utility function which is the same as the others, just different variables.

$$\text{Neut utility} = \frac{e^{-A-B}}{e^{3}}$$

where B is base ranging over ([0,1]) and A is acid ranging over ([0,1]).

The reservation curve, however, is a little different. It consists of two functions, namely $0 \geq -B - A - 0.2$ and $0 \leq -B - A + 0.2$. The value of 0.2 has been decided on after talks with experts.

4.3.5 Mixbed

The mixbed is where the final cleaning occurs. It is also the agent responsible for the end water delivery. Since it consists of a mixture of anion and base, it has three issues about it has desires. This is realized with the function below:

$$\text{mixbed utility} = \frac{e^{W+A+B}}{e^3}$$

where W is water ranging over ([0,1]), B is base ranging over ([0,1]) and A is acid ranging over([0,1]).

The reservation curve is given as $0 = A + B + W - (\ln(u) + 3)$. The projection of a point to this linear plane is calculated as follows:

$$x = \frac{x_0 + (\ln(u) + 3) - (x_0, y_0, z_0)}{3}$$

$$y = \frac{y_0 + (\ln(u) + 3) - (x_0, y_0, z_0)}{3}$$

$$z = \frac{z_0 + (\ln(u) + 3) - (x_0, y_0, z_0)}{3}$$

This is proven

Important to note her is that the ratio of water to the base and acid is disputed. Since only a residue of ions have to be removed, the amount of water desired can be much hire compared to the amount of acid and base used. This is solved with the introduction of a variable l with which to multiply the amount of water.

If the amount of water, compared to the amount of base and acid used is 10 i.e., we get the following function:

$$\text{mixbed utility} = \frac{e^{(10*W)+A+B}}{e^(10 + 2)}$$

The normalisation is still required, which is carried over to the projection to the plane in x for example:

$$x = \frac{x_0 + (\ln(u) + 12) - (x_0, y_0, z_0)}{12}$$

4.4 Negotiations among the agents

All in all, there are three kinds of sub-negotiations. These are shown in Figures 4.6, 4.7 and 4.8. To visualize the workings of these the solution space is shown in

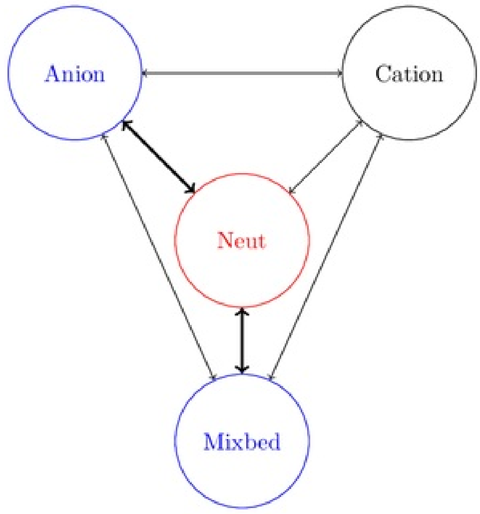

Figure 4.6: Base negotiation. Red indicates seller, blue buyer. The Neut sells the base, while the Anion and Mixbed agent try to obtain as much base as possible.

So although there is a multi-issue negotiation, the only agent that has interest in all three issues is the mixbed. This means that if an agent proposes 0.7 acid to the anion, the anion will not consider this part of the offer, as in since it can not project this to its' indifference curve. This means that the anions new proposal will also include 0.7 anion, and the adjusted values to the other issues.

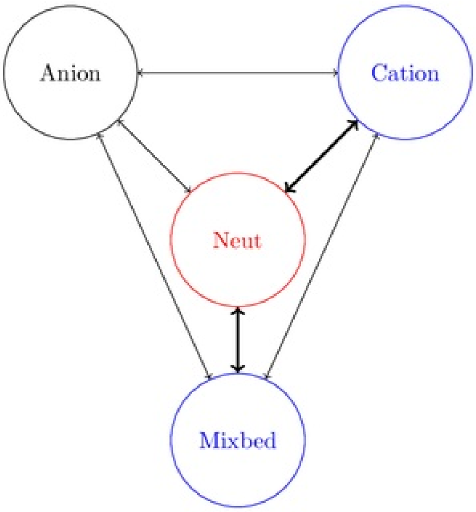

Figure 4.7: Acid negotiation. Red indicates seller, blue buyer. The Neut sells the acid, while the Cation and Mixbed agent try to obtain as much acid as possible.

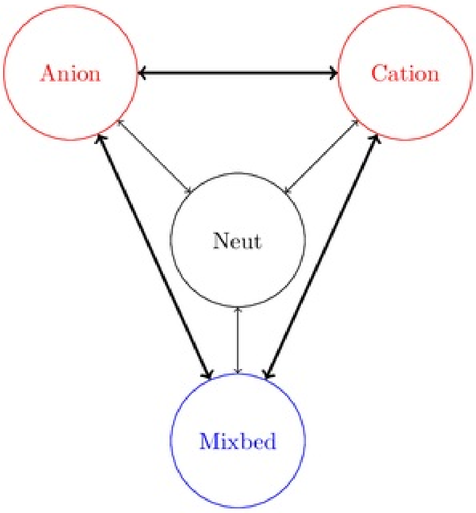

Figure 4.8: Water negotiation. Red indicates seller, blue buyer. The cation and anion sell water, while the mixbed agent tries to obtain as much water as possible

4.5 Algorithm

The algorithm, as programmed in Java is implemented using different objects. This was originally described in Chapter 3 as the optimal way to implement agents. The model consist of 4 different agents, which in turn can either be an anion, cation, mixbed or neut with their own characteristics.

The algorithm starts with each agent proposing their preferred offer, which is the offer with their highest utility. After each agent proposes, the first agent starts with the generation of the offer. Algorithm 1 shows the reactive concession strategy, but a non-reactive strategy is easily applied when changing Line 9 to if$true$.

Afterwards, depending on the concession protocol, their desired utility decreases, meaning that the proposal creeps towards the solution space as can be seen in Figure 4.9.

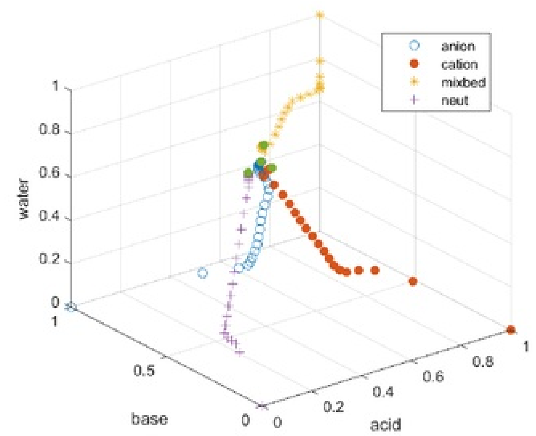

Figure 4.9: An example run of the agents searching for the solution. It is clearly seen that the agents start with their proposal in the corners, their optimal offers, and slowly make concessions towards eachother.

Data: Each agent's utility function $u_i(x)$, reservation utility ru_i, and concession strategy $s_i(t) = 1, 2, ..., T$

Result: Negotiation Agreement

initialization: Each agent proposes a preferred offer x_0^i;
$t \leftarrow 1$;
Set convergence tolerance: δ;
while $t \leq T$ *and IsConverge* $==$ *False* **do**
    Determine the agent to propose: i = mod(t,4);
    foreach $j \in 1, 2, 3, 4$ **do**
        if $j == i$ **then**
            foreach $k \in 1, 2, 3, 4$ **do**
                if $x^k_{[j,-1]} == \emptyset \parallel u_j(x_t^k) \geq ru_j$ **then**
                    $\Delta u_{jk} \leftarrow \Delta u_{j0}(t)$;
                else
                    $\Delta_1 u_{jk}(t) \leftarrow u_j(x_t^k) - u_j(x^k_{[j,-1]})$;
                    $\Delta_2 u_{jk}(t) \leftarrow u_j(x_t^k) - u_j(x_0^k) - (1 - u_j(x^j_{t-1}))$;
                    $\Delta u_{jk} \leftarrow \max\{\Delta_1 u_{jk}(t), \Delta_2 u_{jk}(t), 0\}$;
                end
                $\Delta u_j(t) \leftarrow \min\{\min_{k \in \Gamma_t(i)} \Delta u_{jk}(t), \Delta u_{j0}(t)\}$;
            end
            Agent i concedes by determining $s_i(t) \leftarrow s_i(t-1) - \Delta u_j(t)$;
            Agent i calculates: $w_{t-1} \leftarrow \frac{1}{m}\sum_{j=1}^{m} x^j_{t-1}$;
            Agent i proposes $P_{A_t^i}[w_{t-1}]$;
        else
            $x_i^j \leftarrow x^j_{t-1}$;
            if $u_j(x^i)_{t-1} \geq u_j(x^i_{[j,-1]})$ **then**
                $x^i_{[j,-1]}(t) \leftarrow x^i_{t-1}(t-1)$;
            else
            $x^j_{[i,-1]}(t) \leftarrow x^j_{[i,-1]}(t-1)$;
        end
    end
    if $\max_{j \in 1,2,...,m} \parallel x_t^j - w_{t-1} \parallel < \delta$ **then**
        IsConverge $\leftarrow$ True;
    else
        $t \leftarrow t + 1$;
    end
end

Algorithm 1: Basic algorithm structure modified from (Zheng et al., 2015). Applied to four agents.

4.6 What does result mean? 0.6, 0.4, 0.8 eg

Means that of the total amount of resources available, the allocation will be devided among 2. **Todo What does result mean?!** (7)

Allocate the total possible to the agents. e.g. the final offer is 0.7 water, meaning that the anion and cation produce 0.7 of the total possible water production.

If water l is set to 10, it also requires 10 times as little base than the anion filter i.e..

The allocation of the base to the anion and mixbed is dependent on de water l. This is a fixed separation. Let's say the group agree to 0.6 usage of base. If l is set to 1, we allocate as much base to the mixbed as the anion (0.3 each). However, if l is set to 10, this means that the mixbed will receive 1/11th of the base while the anion receives 10/11th.

Another possiblity is that the reservation funtion of the mixbed could be higher than the others.

4.7 Proof of convergence

If there is an intersect of the agreement zone, the agents will find this. This is already proven in the Zeuten strategy, which shows that no agreement is always worse than a bad agreement. In infinite time this will happen. **Zeuten Strategy** (8)

Furthermore, we have shown that the projection gives us the point closest to the agreement-set of an agent. Since it was proven by Zheng et al. (2015) that if x is projected on to the agreement-set ($P_A[x]$), this also can be said of the projection on the linear line with the method used.

Chapter 5

Simulation comparison & evaluations

In the previous chapter we have described our model, with the reactive concession strategy. Here the reactive concession strategy is compared to a non-reactive concession strategy. Furthermore different values for the reservation utility are checked, and different values for the mixbed agent water requirements are compared.

5.1 Parameters

The parameters that are checked are compared to the baseline non-reactive strategy. Firstly the Nash solution is described to compare to the optimal solution. The parameters will be described in more detail.

5.1.1 Nash Bargaining Solution

The Nash bargaining solution is found using the product of agent's utility, maximizing the joint utility: $\prod_{i=i}^{m} u_i(x)$. We can calculate this since to us the utilities of the agent is known. This information is unknown to the agents since they only know their own utility curve. The joint utility gives the global maximum, and optimal Nash Solution. The utility functions of the agents are convex, which means that the solution is Pareto optimal and maximizes the product of the utilities (Nash Jr, 1950; Roth, 1977; Lensberg, 1988).

$$
\begin{aligned}
\text{maximize} \quad & \prod_{i=1}^{m} u_i(x) \\
\text{subject to} \quad & u_i(x) \geq ru_i, i = 1, ..., m \\
& 0 \leq x_j \leq 1, \; j = 1, ..., n
\end{aligned}
$$

Using the GAMS solver, the solutions are calculated. The limit for the reservation curve is also found, $\forall i, ru_i = 0.3182$. This means that there is no solution space, if $\forall i, ru_i > 0.3182$. However, if only a single agent were to have $ru_i > 0.3182$, there still would be a solution. This obviously depends on the reservation curve values of the other agents. **show the nash solution visualised?** (9)

5.1.2 Non-reactive concession strategy

As explained in Section 3.3.2 the concession strategy determines whether a solution will be found. If no concession is made during the negotiation, and the agent stay on their initial utility, no agreement can be made. In this result the non-reactive strategy is used as a base line to compare to other methods. As described, there are a large number of different methods, and a weak concession is used in here, since the utility functions of the other agents are unknown. The non-reactive concession strategy used is $s_i(t) = \max\{s_0(t) - t * 0.01, ru_i\}$. This monotonic decreasing concession is a linear function until the reservation value. Described by Wu et al. (2009), it is an *amount of utility*, where Agent $i \in N$ concedes a fixed amount utility au. Since the utility functions are private, utilitarian concessions are not possible (Endriss, 2006).

This means that the minimum utility value is reached after 100 rounds, if only the non-reactive concession strategy is used. Then each agent makes $\frac{100}{4 \text{agents}} = 25 \text{proposals}$. Since we combine it with the reactive concession strategy, it is possible for the agents to negotiate for more rounds.

5.1.3 Reactive Concession strategy

The reactive concession strategy (see Section 4.3.1 for an explanation) is compared to the non-reactive concession strategy. Similar to Zheng et al. (2015), however here different reservation utilities are checked, while comparing the reactive to non-reactive strategy.

5.1.4 Reservation curve

The curve, as shown in Section 4.2, is not really a curve, but a linear limit. The values can differ from $ru_i = \{0.05, 0.10, 0.15, 0.20, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65\}$. We have seen that in the initial case, there is no agreement zone if the ru_i is larger than 0.3182. But, since we try different kind of parameters, it could be the case that if one of the reservation curves were to be different, the rest could as well.

5.1.5 Distance

For the algorithm Algorithm 1 to finish, there are two option. Either the distance from the offer and weight is smaller than a threshold, or max number of rounds are reached. This distance, $\max_{j \in 1,2,\ldots,m} \parallel x_t^j - w_{t-1} \parallel$ gives the maximum distance from the agents to the weight. It tells something about the final solution and is thus used in the results to determine how the efficiency of the solutions.

If the distance is larger than the threshold, it means that the agents have not found an agreement and the max number of rounds has been reached. The final solution will be the average of all proposals. Two options are possible. Either one or more agent(s) has not conceded and thus not moved to the agreement-zone. Another option is that the reservation utilities is too high, meaning that there is no agreement-zone, and thus no agreement possible.

Since we know where the agreement zone lies, we can see when the agent(s) do not concede, and thus refrain from agreement if the distance is larger than the threshold.

Threshold & maximum number of rounds

As shown in the algorithm, there is a threshold required to decide on the value and whether an agreement is reached. For this simulation this is set to $\delta = 0.05$. The maximum number of rounds is set to 200.

Since the non-reactive concession give a maximum of 100 rounds until zero is reached, 200 seems as an overkill. However, since we are dealing with the combination of different concession strategies, it is useful to check whether a solution is found afterwards. This since there might change something in the proposals due to the reactive concession method. If the non-reactive method were to be changed, it would be important to change the number of rounds as well.

5.2 Reactive concession compared to non-reactive

When comparing the reactive to the non-reactive strategy, as shown in Figure 5.1, it is obvious that the Nash limit indeed lies at $ru_i = 0.3182$. However, unexpectedly, the non-reactive strategy consequently finds the solution closer to the optimal Nash Bargaining Solution then the reactive strategy. Nash lies at acid 0.571, base 0.571, water 0.714.

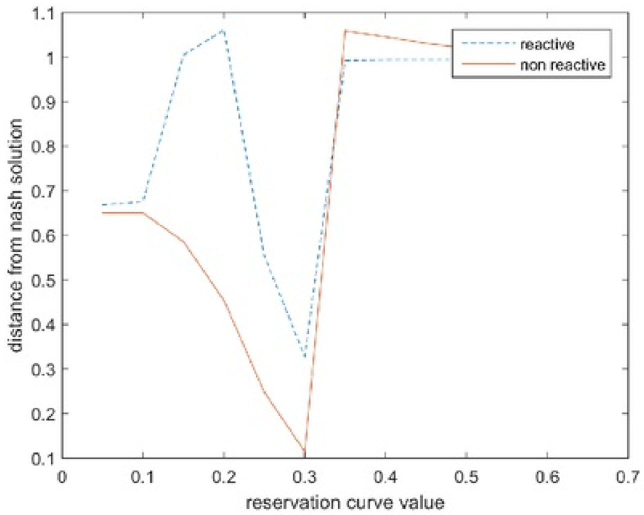

Figure 5.1: Distance from the Nash bargaining solution for the reactive and non-reactive concession strategies.

When looking at the distance and the amount of rounds necessary to find an solution (Table 5.1), we see that although our Nash limit lies at 0.3182, this solution is not found if the reactive concession strategy is used.

It is very interesting to see that the non-reactive concession strategy halts after 23 rounds, which means that it is faster than the reactive concessions strategy. It also has some distance, which means that there is no real agreement, since the distance is larger than 0. However, when looking at the distance from the reactive concession, it is zero, which means that the average is exactly the same as the proposals. Interestingly when looking at the distance from the optimal solution, Figure 5.1, it is very large. This unexpected result will be discussed later.

	Reactive concession		Non-reactive concession	
reservation utility	distance	# of rounds	distance	# of rounds
0.05	0.0000	26	0.0296	23
0.10	0.0000	26	0.0296	23
0.15	0.0000	38	0.0296	23
0.20	0.0000	50	0.0296	23
0.25	1.0693	199	0.0296	23
0.30	1.1751	199	0.0485	66
0.35	1.1716	199	0.1277	199
0.40	1.1675	199	0.2799	199
0.45	1.1675	199	0.4254	199
0.50	1.1675	199	0.5555	199
0.55	1.1675	199	0.6733	199
0.60	1.1675	199	0.7807	199
0.65	1.1675	199	0.8745	199

Table 5.1: The distance in the final proposal and number of rounds of a simulation.

5.3 Reactive mixbed vs non-reactive

Although the reactive concession strategy performed worse when compared to the non-reactive concession strategy, a comparison to is made when only the mixbed agent uses the reactive strategy. The mixbed agent is the most important agent since it “produces” the final demi water product.

In the comparison of the Nash Bargaining Solution, in Figure 5.2 it is seen that the method initially performs better than the reactive strategy.

The table of the distance and number of rounds is very repetitive. Exactly the same, although the end solution, which is the average of all offers, is close to the Nash solution. It does not find the solution within the 200 rounds however.

5.4 Changing the l value for the mixbed

In the design it was stated that the water ratio to the base and acid could change for the mixbed. Here an example is given where the mixbed water to base and acid ratio is 2:1:1 and 10:1:1. Here a new Nash solution has to be calculated, since the utility functions have changed. So in the graph comparison, the mixbed water with the updated ratio, is checked against the reactive and non-reactive concession strategies.

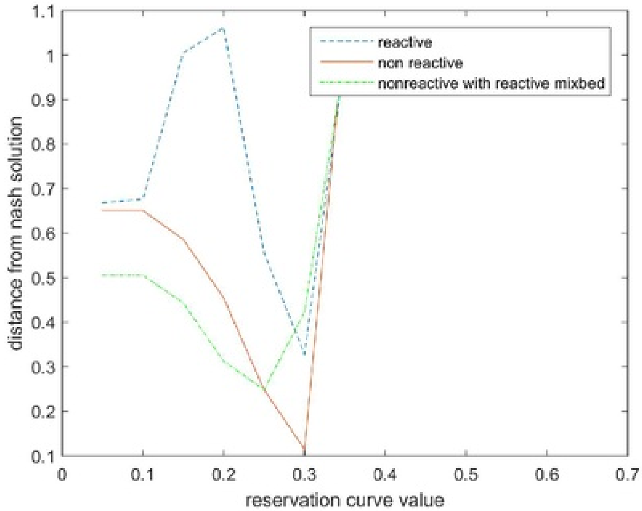

Figure 5.2: Comparison of the reactive and non-reactive vs only reactive mixb

5.4.1 Mixbed 2 water

When looking at the first ration of 2:1:1, we get very similar result as that of the comparison. The minimum rui lies way lower however, and has an maximum of $\forall i, ru_i = 0.301$. The Nash solution lies at acid = 0.600, base = 0.600, water = 0.800. It is interesting to note that the new original end proposal is a lot nearer to the Nash solution than the base line situation when the reservation utility is low as can be seen in Figure 5.3. Again the non-reactive concession strategy comes closer to the Nash solution ultimately, however initially the reactive concession strategy is better.

5.4.2 Mixbed 10 water

When using an ratio of 10:1:1 we have an max $ru_i = 0.274$. The Nash solution lies at acid = 0.647, base = 0.647, water = 0.941. The large increase to the water demand makes this an interesting solution. It is very interesting to note that although the reservation minimum lies at 0.274, the algorightm still finds an solution very close to the Nash optimum when the reservation utility is 0.3.

reservation utility	distance	# of rounds
0.05	1.186 9	199
0.10	1.186 9	199
0.15	1.186 9	199
0.20	1.186 9	199
0.25	1.186 9	199
0.30	1.186 9	199
0.35	1.186 9	199
0.40	1.186 9	199
0.45	1.186 9	199
0.50	1.186 9	199
0.55	1.186 9	199
0.60	1.186 9	199
0.65	1.186 9	199

Table 5.2: The distance in the final proposal and number of rounds of a simulation. This is where only the mixbed makes reactive concessions, and the other agents make non-reactive concessions.

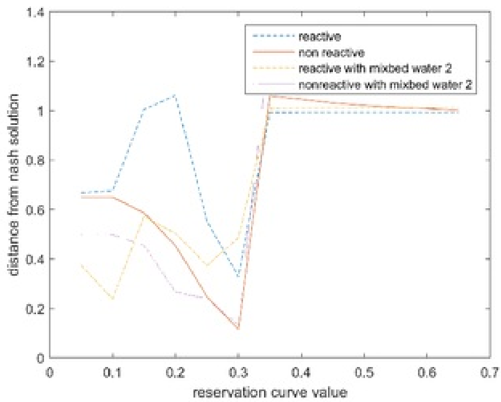

Figure 5.3: Comparison of the reactive and non-reactive strategy for a 2 water value mixbed.

	Reactive concession		Non-reactive concession	
reservation utility	distance	# of rounds	distance	# of rounds
0.05	0.0000	26	0.0000	26
0.10	0.0000	30	0.0000	26
0.15	0.0000	43	0.0000	26
0.20	0.9507	199	0.0000	26
0.25	1.1133	199	0.0000	26
0.30	1.2415	199	0.0472	71
0.35	1.2400	199	0.2830	199
0.40	1.2360	199	0.4522	199
0.45	1.2360	199	0.6774	199
0.50	1.2360	199	0.7797	199
0.55	1.2360	199	0.8625	199
0.60	1.2360	199	0.9268	199
0.65	1.2360	199	0.9859	199

Table 5.3: Here mixbed is water 2.

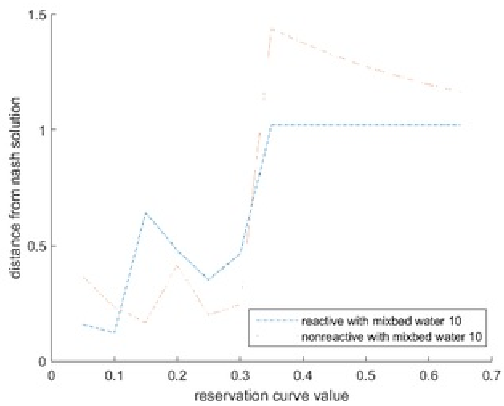

Figure 5.4: Comparison of the reactive and non-reactive strategy for a 10 water value mixbed.

	Reactive concession		Non-reactive concession	
reservation utility	distance	# of rounds	distance	# of rounds
0.05	0.000 0	34	0.000 0	26
0.10	0.000 0	38	0.000 0	30
0.15	0.000 0	59	0.000 0	30
0.20	1.010 2	199	0.037 6	47
0.25	1.184 5	199	0.040 4	59
0.30	1.283 0	199	0.196 7	199
0.35	1.300 0	199	0.516 8	199
0.40	1.300 0	199	0.620 7	199
0.45	1.300 0	199	0.712 3	199
0.50	1.300 0	199	0.794 2	199
0.55	1.300 0	199	0.868 3	199
0.60	1.300 0	199	0.936 0	199
0.65	1.300 0	199	0.998 3	199

Table 5.4: Here mixbed is water 10.

5.5 Discussion

5.5.1 reactive vs non-reactive

A possible solution is that the answer can not be found as is stated by Zheng et al. (2015) as Lemma 2. "If Agent i deliberately stops conceding before reaching the agent's own reservation utility from time period t onward. and all other agents use the reactive concession strategy the negotiation will stall; i.e. other agents will reactively stop conceding and there will be no agreement. if $\Delta_j < s_j(t) - u_j(x^*_{ru_i})$ and $u_j(x^*_{s_i(t)}) < ru_j$".

This is although there is a nonempty zone of agreement it does not find it. This can be clearly seen when we analyze the situation for the reservation curve at 0.2.

Situation at the 0.2 ru

So as shown in the results. there is a unexpected distance from the Nash solution in the situation of the 0.2 reservation curve. This can mostly be attributed to the stalling of the agents while trying to find a solution.

Looking at the proposals made. it is clearly seen that in the reactive process. the Neut and Mixbed do not react to the proposals of the Anion and Cation. After multiple proposals they finally adjust

The solution space

5.6 Solution space

Below the solution space for the agents is shown, when $ru_i = 0.2$ for all agents. Each agent has 2 or three subjects.

5.6.1 reactive mixbed vs non-reactive

Nash solution lies in 0 water with 1 base and 1 acid. This is close to the start of mixbed in 1.1.1

This is however the result of the stalling of the mixbed. and not the result of the solution being found. This is obvious when looking at the table. Although the reservation curve changes in the offers. the mixbed does not move. This can also been seen in the proposal figure

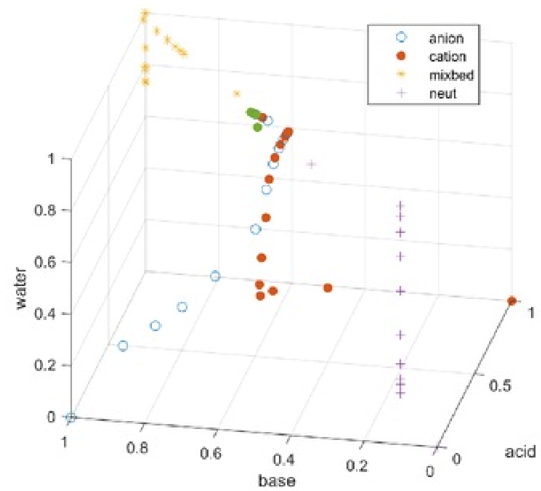

Figure 5.5: The proposals made by the 4 agents. We see that the mixbed and neut "stall" in de finding of the solution.

Todo Hier proposal figure toevoegen (10)

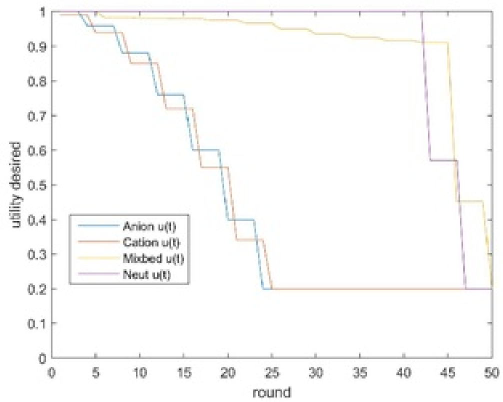

Figure 5.6: The desired utility for the agents. The stall can be clearly seen for the mixbed and neut agent. Only towards the end do they start conceding.

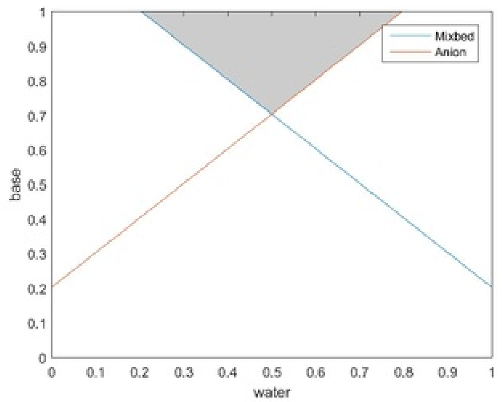

Figure 5.7: The solution space for the combination of base and water, which are negotiated over by the anion and the mixbed agent.

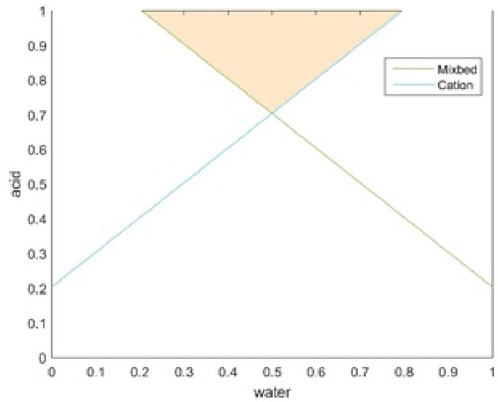

Figure 5.8: The solution space for the combination of acid and water

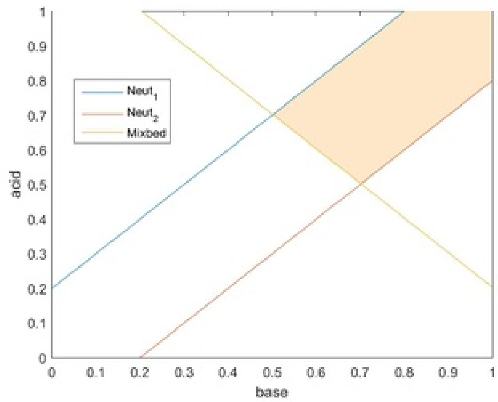

Figure 5.9: The solution space for the combination of acid and base

Chapter 6

Conclusion and Further Work

6.1 Conclusion

In this thesis we have given an overview of agent solutions used in the manufatcturing world. We found that a gap lies in the "real" negotiation, which excludes the use of auctions and or contract net protocol.

By using the alternating offer protocol it is checked whether an optimal solution can be found.

At the moment, it seems as if the reactive concession strategy, as described in Zheng et al. (2015) still has some difficulties. This can be clearly seen in Figure 5.1.

6.2 Evaluation

Although the alternating offer has been used, it is not usable at al yet at a real case. The largest difficulty lies in the realistic portrayal of the utility function. The requirement for a convex utility function makes it even more difficult. However, if only the non-reactive strategy was used, it should be possible to use a non-convex function.

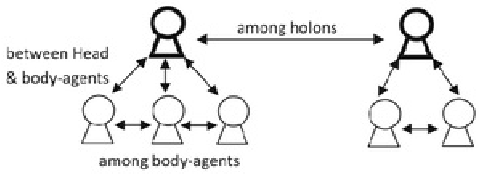

Figure 6.1: An example of the different negotiation between holons from Beheshti et al. (2016).

6.3 Further research

A lot of further research could be done on the alternating offer protocol to be used in manufacturing world. The agents could be improved to allow reasoning, using an holonic structure for example. Furthermore other strategies can be used, while the utility functions can be changed as well. Extra negotiation could be applied using bilateral negotiations, and to finalize heuristic learning methods can be applied.

6.3.1 Holonic agents

This structure is that of a holon as can be seen in figure fig. 6.1. As shown in the literature it is based on PROSA by Van Brussel et al. (1998).

The following facts and rules are part of the Anion.

1. Knowledge of anion head about the sub-agents:
   - $\{A_1, ..., A_6\}$ can process a amount of water
   - $\{A_1, ..., A_6\}$ needs to be cleaned after b water
   - $\{A_1, ..., A_6\}$ has filtered c amount of water
   - $\{A_1, ..., A_6\}$ needs d base to clean
   - $\{A_1, ..., A_6\}$ needs e time to clean
2. Currently x amount of water being filtered
3. Currently $Z \subseteq \{A_1, ..., A_6\}$ filter being used for water filtering
4. Currently $Y \subseteq \{A_1, ..., A_6\}$ filter being used for cleaning
5. Currently w amount of base being used for cleaning

The use of these holons could allow an agent to reason about the environment, and act upon it accordingly.

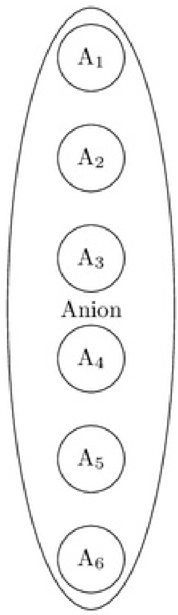

Figure 6.2: Anion head and sub-agents

6.3.2 Utility function

The requirement of a convex function forced the use of a specific, and highly theoretical function. This could be perfected using more expert input.

Reservation curve

We have used a linear reservation curve for simplicity, since this eliminated the minimization that would have been necessary if a truly curved function was used. An example is shown in Section 6.3.2.

6.3.3 Continue negotiation after group agreement

After the group has an agreement, the agents now allocate the resource as predefined (see Section 4.6). This could be optimized to bilateral negotiation between the Mixbed and Anion for example.

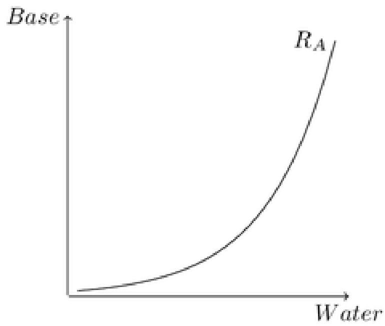

Figure 6.3: The reservation function for the Anion filter: the more water is filtered and given, the more base it requires.

6.3.4 Extra concession strategies

The many concession strategies, as shown in Section 3.3.2 allow for many strategies to be used. Further research could include using fraction, which was suggested as a good solution by Wu et al. (2009).

6.3.5 Heuristic learning methods

As shown in **??** 3, heuristic learning methods can also be used to learn the desired utility function. This has not been done since the requirement of a convex utility functions forced the usage of predefined functions.

To do...

- ☐ 1 (p. 22): **Todo stein uitleggen** Rubenstein uitleggen
- ☐ 2 (p. 23): **Todo Hier concession uitleggen** Concession uitleggen!!
- ☐ 3 (p. 31): **Example of the three agents and three resources** Example of the Three agents and three resources,
- ☐ 4 (p. 34): Framework weghalen
- ☐ 5 (p. 38): **Todo Make letters larger** Make letters larger
- ☐ 6 (p. 41): **Todo explain reactive concession!!** Explain reactive Concession!
- ☐ 7 (p. 50): **Todo What does result mean?!** What does the outcome mean?
- ☐ 8 (p. 50): **Zeuten Strategy** Zeuten Strategy
- ☐ 9 (p. 52): **show the nash solution visualised?** Show the nash solution visualised?
- ☐ 10 (p. 60): **Todo Hier proposal figure toevoegen** Hier proposal figure toevoegen!!

Bibliography

Fahmida Abedin, Kuo-Ming Chao, and Nick Godwin. An agenda based multi issue negotiation approach. *Journal of Ambient Intelligence and Humanized Computing*, 5(4):455–473, 2014.

CJ Anumba, Z Ren, A Thorpe, OO Ugwu, and L Newnham. Negotiation within a multi-agent system for the collaborative design of light industrial buildings. *Advances in Engineering Software*, 34(7):389–401, 2003.

Rahmatollah Beheshti and Nasser Mozayani. Homan, a learning based negotiation method for holonic multi-agent systems. *Journal of Intelligent & Fuzzy Systems*, 26(2):655–666, 2014.

Rahmatollah Beheshti, Roghayeh Barmaki, and Nasser Mozayani. Negotiations in holonic multi-agent systems. In *Recent Advances in Agent-based Complex Automated Negotiation*, pages 107–118. Springer, 2016.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Dennis Brandl and BR&L Consulting. What is isa-95? industrial best practices of manufacturing information technologies with isa-95 models, May 2008. URL `http://apsom.org/docs/t061_isa95-04.pdf`.

M Bruccoleri, G Lo Nigro, G Perrone, P Renna, and S Noto La Diega. Production planning in reconfigurable enterprises and reconfigurable production systems. *CIRP Annals-Manufacturing Technology*, 54(1):433–436, 2005.

Enrique de la Hoz, Jose Manuel Gimenez-Guzman, Ivan Marsa-Maestre, and David Orden. Automated negotiation for resource assignment in wireless surveillance sensor networks. *Sensors*, 15(11):29547–29568, 2015.

Dario Di Nocera. *Multi-agent automated negotiation for management of services in smart cities*. PhD thesis, 2015.

Neil A Duffie and Vittaldas V Prabhu. Real-time distributed scheduling of heterarchical manufacturing systems. *Journal of Manufacturing Systems*, 13 (2):94–107, 1994.

Ulle Endriss. Monotonic concession protocols for multilateral negotiation. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 392–399. ACM, 2006.

Sebastian Engell and Iiro Harjunkoski. Optimal operation: Scheduling, advanced control and their integration. *Computers & Chemical Engineering*, 47:121–133, 2012.

Shaheen Fatima, Sarit Kraus, and Michael Wooldridge. The negotiation game. *IEEE Intelligent Systems*, 29(5):57–61, 2014a.

Shaheen Fatima, Sarit Kraus, and Michael Wooldridge. *Principles of Automated Negotiation*. Cambridge University Press, 2014b.

Shaheen S Fatima, Michael Wooldridge, and Nicholas R Jennings. An agenda-based framework for multi-issue negotiation. *Artificial Intelligence*, 152(1): 1–45, 2004.

Qiang Feng, Songjie Li, and Bo Sun. A multi-agent based intelligent configuration method for aircraft fleet maintenance personnel. *Chinese Journal of Aeronautics*, 27(2):280–290, 2014.

Roger Fisher, William Ury, and Bruce Patton. *Getting to Yes*. Simon & Schuster Sound Ideas, 1987.

Adriana Giret. A multi agent methodology for holonic manufacturing systems. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 1375–1375. ACM, 2005.

Iiro Harjunkoski, Rasmus Nyström, and Alexander Horch. Integration of scheduling and control – Theory or practice? *Computers & Chemical Engineering*, 33(12):1909–1918, 2009.

Alan Hevner and Samir Chatterjee. *Design Science Research in Information Systems*. Springer, 2010.

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.

Kurt Konolige and Nils J Nilsson. Multiple-agent planning systems. In *AAAI*, volume 80, pages 138–142, 1980.

Sarit Kraus, Jonathan Wilkenfeld, and Gilad Zlotkin. Multiagent negotiation under time constraints. *Artificial intelligence*, 75(2):297–345, 1995.

Kalliopi Kravari and Nick Bassiliades. A survey of agent platforms. *Journal of Artificial Societies and Social Simulation*, 18(1):11, 2015.

Guoming Lai, Cuihong Li, Katia Sycara, and Joseph Giampapa. Literature review on multi-attribute negotiations. *Robotics Inst., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-RI-TR-04-66*, 2004.

Paulo Leitão. Agent-based distributed manufacturing control: A state-of-the-art survey. *Engineering Applications of Artificial Intelligence*, 22(7):979–991, 2009.

Paulo Leitão, Vladimír Mařík, and Pavel Vrba. Past, present, and future of industrial agent applications. *IEEE Transactions on Industrial Informatics*, 9(4):2360–2372, 2013.

Paulo Leitao, Stamatis Karnouskos, Luis Ribeiro, Jay Lee, Thomas Strasser, and Armando W Colombo. Smart agents in industrial cyber–physical systems. *Proceedings of the IEEE*, 104(5):1086–1101, 2016.

Terje Lensberg. Stability and the nash solution. *Journal of Economic Theory*, 45(2):330–341, 1988.

Ivan Marsa-Maestre, Mark Klein, Catholijn M Jonker, and Reyhan Aydoğan. From problems to protocols: Towards a negotiation handbook. *Decision Support Systems*, 60:39–54, 2014.

L. Monostori, B. Kadar, T. Bauernhansl, S. Kondoh, S. Kumar, G. Reinhart, O. Sauer, G. Schuh, W. Sihn, and K. Ueda. Cyber-physical systems in manufacturing. *CIRP Annals – Manufacturing Technology*, 65(2):621–641, 2016.

László Monostori, József Váncza, and Soundar RT Kumara. Agent-based systems for manufacturing. *CIRP Annals-Manufacturing Technology*, 55(2):697–720, 2006.

Alexandre Muller, Adolfo Crespo Marquez, and Benoît Iung. On the concept of e-maintenance: Review and current research. *Reliability Engineering & System Safety*, 93(8):1165–1187, 2008.

John F Nash Jr. The bargaining problem. *Econometrica: Journal of the Econometric Society*, pages 155–162, 1950.

Daniel Palmer, Marc Kirschenbaum, Jon Murton, Kelly Zajac, M Kovacina, and Ravi Vaidyanathan. Decentralized cooperative auction for multiple agent task allocation using synchronized random number generators. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pages 1963–1968. IEEE, 2003.

H Van Dyke Parunak. Industrial and practical applications of dai. *Multiagent systems: a modern approach to distributed artificial intelligence*, pages 337–421, 1999.

Michael Pinedo. *Planning and scheduling in manufacturing and services*, volume 24. Springer, 2005.

Manisa Pipattanasomporn, Hassan Feroze, and S Rahman. Multi-agent systems in a distributed smart grid: Design and implementation. In *Power Systems Conference and Exposition, 2009. PSCE'09. IEEE/PES*, pages 1–8. IEEE, 2009.

Ricardo J Rabelo, Luis M Camarinha-Matos, and Hamideh Afsarmanesh. Multi-agent-based agile scheduling. *Robotics and Autonomous Systems*, 27(1):15–28, 1999.

Iyad Rahwan, Sarvapali D Ramchurn, Nicholas R Jennings, Peter Mcburney, Simon Parsons, and Liz Sonenberg. Argumentation-based negotiation. *The Knowledge Engineering Review*, 18(04):343–375, 2003.

Anand S Rao, Michael P Georgeff, et al. Bdi agents: From theory to practice. In *ICMAS*, volume 95, pages 312–319, 1995.

M Rosa, V Miranda, M Matos, G Sheble, and AM Silva. Intelligent agent-based environment to coordinate maintenance schedule discussions. In *Intelligent System Applications to Power Systems, 2009. ISAP'09. 15th International Conference on*, pages 1–7. IEEE, 2009.

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

Alvin E Roth. Individual rationality and nash's solution to the bargaining problem. *Mathematics of Operations Research*, 2(1):64–65, 1977.

Ariel Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica: Journal of the Econometric Society*, pages 97–109, 1982.

Fernando Schramm and Danielle Costa Morais. A bilateral and multi-issue negotiation framework to support a supply chain of construction industry. *Pesquisa Operacional*, 33(3):491–512, 2013.

Omor Sharif and Nathan Huynh. Yard crane scheduling at container terminals: A comparative study of centralized and decentralized approaches. *Maritime economics & logistics*, 14(2):139–161, 2012.

Weiming Shen, Douglas H Norrie, and Jean-Paul Barthès. *Multi-agent systems for concurrent intelligent design and manufacturing*. CRC press, 2003.

Yoav Shoham. Agent-oriented programming. *Artificial intelligence*, 60(1):51–92, 1993.

Andrew Slaughter, Gregory Bean, and Anshu Mittal. Connected barrels, august 2015. URL `http://dupress.com/articles/internet-of-things-iot-in-oil-and-gas-industry/`.

R Smith. Communication and control in problem solver. *IEEE Transactions on computers*, 29(12):1104–1113, 1980.

Amy JC Trappey, Charles V Trappey, and Wei-Chun Ni. A multi-agent collaborative maintenance platform applying game theory negotiation strategies. *Journal of Intelligent Manufacturing*, 24(3):613–623, 2013.

Hendrik Van Brussel, Jo Wyns, Paul Valckenaers, Luc Bongaerts, and Patrick Peeters. Reference architecture for holonic manufacturing systems: Prosa. *Computers in industry*, 37(3):255–274, 1998.

Herman Vermaak and Johnson Kinyua. Multi-agent systems based intelligent maintenance management for a component-handling platform. In *Automation Science and Engineering, 2007. CASE 2007. IEEE International Conference on*, pages 1057–1062. IEEE, 2007.

Meritxell Vinyals, Juan A Rodriguez-Aguilar, and Jesus Cerquides. A survey on sensor networks from a multiagent perspective. *The Computer Journal*, 54(3):455–470, 2011.

Pavel Vrba, Pavel Tichỳ, Vladimír Mařík, Kenwood H Hall, Raymond J Staron, Francisco P Maturana, and Petr Kadera. Rockwell automation's holonic and multiagent control systems compendium. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(1):14–30, 2011.

John P Wangermann and Robert F Stengel. Principled negotiation between intelligent agents: a model for air traffic management. *Artificial Intelligence in Engineering*, 12(3):177–187, 1998.

Michael Wooldridge. *An Introduction to Multiagent Systems*. John Wiley & Sons, 2009.

Mengxiao Wu, Mathijs de Weerdt, and Han La Poutré. Efficient methods for multi-agent multi-issue negotiation: allocating resources. In *International Conference on Principles and Practice of Multi-Agent Systems*, pages 97–112. Springer, 2009.

Ren Yu, Benoit Iung, and Hervé Panetto. A multi-agents based e-maintenance system with case-based reasoning decision support. *Engineering Applications of Artificial Intelligence*, 16(4):321–333, 2003.

Ronghuo Zheng, Nilanjan Chakraborty, Tinglong Dai, and Katia Sycara. Automated multilateral negotiation on multiple issues with private information. *Available at SSRN 2677729*, 2015.

Xianrong Zheng, Patrick Martin, Kathryn Brohman, and Li Da Xu. Cloud service negotiation in internet of things environment: a mixed approach. *Industrial Informatics, IEEE Transactions on*, 10(2):1506–1515, 2014.

R Zhou, B Fox, HP Lee, and AYC Nee. Bus maintenance scheduling using multi-agent systems. *Engineering Applications of Artificial Intelligence*, 17 (6):623–630, 2004.

Thesis

ORIGINALITY REPORT

18%

SIMILARITY INDEX

EXCLUDE QUOTES	OFF
EXCLUDE BIBLIOGRAPHY	ON

Thesis

WRITECHECK REPORT

PAGE 1
PAGE 2
PAGE 3
PAGE 4
PAGE 5
PAGE 6
PAGE 7
PAGE 8
PAGE 9
PAGE 10
PAGE 11
PAGE 12
PAGE 13
PAGE 14
PAGE 15
PAGE 16
PAGE 17
PAGE 18
PAGE 19
PAGE 20
PAGE 21
PAGE 22
PAGE 23
PAGE 24
PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51

PAGE 52

PAGE 53

PAGE 54

PAGE 55

PAGE 56

PAGE 57

PAGE 58

PAGE 59

PAGE 60

PAGE 61

PAGE 62

PAGE 63

PAGE 64

PAGE 65

PAGE 66

PAGE 67

PAGE 68

PAGE 69

PAGE 70

PAGE 71

PAGE 72

PAGE 73

PAGE 74

PAGE 75

PAGE 76

PAGE 77

PAGE 78

PAGE 79