

# Agent Negotiation in a Manufacturing Process

## *Master Thesis*

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

First Supervisor:  
Prof. dr. Rineke Verbrugge  
(Artificial Intelligence, University of Groningen)

Second Evaluator:  
Prof. dr. Bart Verheij  
(Artificial Intelligence, University of Groningen)

External supervisor:  
ir. Youri de Koster

December 2016

## **Abstract**

Businesses around the world change from a centralized hierarchical structure to a decentralized structure. This change, part of Industrie 4.0, requires new methods which ensure that the manufacturing and production processes are still optimized. A possible method is the use of multi-agent systems. Central in the implementation of such a system is the communication, such as negotiation. With different negotiation techniques, processes optimization is achievable.

In this research the possible negotiation techniques that can be used for the agents to communicate are discussed. Some of these are desirable to optimize processes. A possible solution is the use of multi-issue multilateral negotiation, with private utility functions. Using the alternation projection method to negotiate, process optimization should be possible.

This is tested with a use case and, using the reactive compared to the non-reactive concession strategy, the optimal concession strategy is discussed. It is found that the reactive concession strategy is not as well performing as the non-reactive in respect to the systems optimal (Nash and Pareto) solution, since it can stall while the agreement-zone is non-empty. However, if a single agent uses the reactive strategy, the system performs well. A possible solution could be the use of different concession strategies, and future research steps could clarify these.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Introduction in Production, AI and the Use Case Environment . . .	1
1.1.1	Production and Manufacturing . . . . .	1
1.1.2	Artificial Intelligence . . . . .	2
1.1.3	Ecosystem of the Case Study . . . . .	2
1.2	Thesis outline . . . . .	3
<b>2</b>	<b>Problem Definition and Research Goal</b>	<b>4</b>
2.1	Problem Analysis . . . . .	4
2.2	Area of Application . . . . .	5
2.3	Relevance . . . . .	5
2.3.1	Scientific Relevance . . . . .	6
2.3.2	Business Relevance . . . . .	6
2.4	Research Goal . . . . .	6
2.5	Research Approach . . . . .	7
2.6	Research Process . . . . .	8
2.6.1	Evaluation Method . . . . .	8
2.7	Research Questions . . . . .	8

<b>3</b>	<b>Literature Study and Theoretical Framework</b>	<b>9</b>
3.1	Manufacturing Processes . . . . .	10
3.1.1	Asset Management . . . . .	10
3.1.2	Process Control . . . . .	12
3.1.3	Predictive maintenance systems . . . . .	13
3.1.4	Real-time Monitoring . . . . .	14
3.2	Agent Solutions . . . . .	15
3.2.1	Object-Oriented Programming . . . . .	15
3.2.2	Multi-Agent Systems . . . . .	15
3.2.3	Holonic Systems . . . . .	16
3.2.4	Task and Resource Allocation . . . . .	17
3.2.5	Scheduling and Planning . . . . .	18
3.3	Framework of a centralized and decentralized system . . . . .	19
3.3.1	Size and Modularity . . . . .	20
3.3.2	Dynamicity (Time Scale/Changeability) . . . . .	21
3.3.3	Solution Quality . . . . .	21
3.3.4	Complexity . . . . .	21
3.3.5	Framework Overview . . . . .	21
3.4	Negotiation . . . . .	23
3.4.1	Negotiation Domain . . . . .	24
3.4.2	Negotiation Protocol . . . . .	25
3.4.3	Negotiation Strategies . . . . .	26
3.4.4	Evaluation and Equilibrium Solutions . . . . .	28
3.4.5	Principled Negotiation . . . . .	30
3.4.6	Negotiation using a Mediator . . . . .	30

3.4.7	Mapping of Negotiation Protocols . . . . .	31
3.5	Negotiation in Manufacturing . . . . .	36
<b>4</b>	<b>Research Design and Application</b>	<b>38</b>
4.1	Demineralization of Water . . . . .	39
4.2	Negotiation Model . . . . .	40
4.3	Details of the Agents . . . . .	43
4.3.1	Reactive Concession Strategy . . . . .	43
4.3.2	Offer Generation . . . . .	44
4.3.3	Anion . . . . .	44
4.3.4	Cation . . . . .	47
4.3.5	Neut . . . . .	47
4.3.6	Mixed . . . . .	48
4.4	Negotiations among the Agents . . . . .	49
4.4.1	The Solution Space . . . . .	52
4.5	Algorithm . . . . .	53
4.6	Result Interpretation . . . . .	55
4.7	Proof of Convergence . . . . .	55
<b>5</b>	<b>Simulation Comparison and Evaluations</b>	<b>56</b>
5.1	Parameters . . . . .	56
5.1.1	Nash Bargaining Solution . . . . .	56
5.1.2	Non-Reactive Concession Strategy . . . . .	57
5.1.3	Reactive Concession Strategy . . . . .	57
5.1.4	Reservation Curve . . . . .	58
5.1.5	Distance . . . . .	58

5.2	Reactive Compared to Non-Reactive Concession Strategy . . . . .	59
5.3	Reactive Mixbed with the other Agents Non-Reactive . . . . .	60
5.4	Changing the water ratio for the Mixbed . . . . .	62
5.4.1	Mixbed Ratio 2:1:1 . . . . .	62
5.4.2	Mixbed Ratio 10:1:1 . . . . .	63
5.5	Evaluation . . . . .	65
5.5.1	Reactive compared to the non-reactive concession strategy	65
5.5.2	Reactive Mixbed compared to non-reactive other agents .	67
5.5.3	Different Mixbed ratios . . . . .	67
<b>6</b>	<b>Conclusion and Further Research</b>	<b>68</b>
6.1	Conclusion . . . . .	68
6.2	Discussion . . . . .	69
6.2.1	Discussion of the Reactive Concession Strategy . . . . .	70
6.3	Further Research . . . . .	72
6.3.1	Extra concession strategies . . . . .	72
6.3.2	Holonic Agents . . . . .	72
6.3.3	Collaboration . . . . .	74
6.3.4	Utility function . . . . .	74
6.3.5	Continue negotiation after group agreement . . . . .	75

## 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 gave optimal advice. She consistently allowed this paper to be my own work, and slightly steered me in the right the direction whenever 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 have 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 Weerd, 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 made the internship even more enjoyable and the 09:30 hr coffee is forever in my mind. Furthermore, I would like to thank my friends for their continuous support.

I must express my deep gratitude to my mother, sister and brother, and to my dearest Vivianne for providing me with constant support and endless encouragement throughout my years of study and through the process of researching and writing this thesis. This achievement would not have been possible without them. Finally, I would like to thank my father, who has always been the greatest support during the studies, and unfortunately cannot 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 Prof. dr. L.C. (Rineke) Verbrugge.

### 1.1 Introduction in Production, AI and the Use Case Environment

Currently a lot of research is conducted in Artificial Intelligence (AI) and how to apply this in business context. One field of interest, which is researched in this thesis, is the usage of a multi-agents system in production and manufacturing 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 making use of 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 fourth industrial 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 industrial 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 worlds, and the convergence of IT and OT (Leitão 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 required. This is supported by the exchangeability of the terms in the literature.

### **1.1.2 Artificial Intelligence**

The research is based on an intelligent multi-agent system (MAS) which consists of agents which act and react on their environment in both a physical and an IT way. For the intelligent agents it is possible, by understanding the system and by negotiating, to come up with a (near-) optimal production plan. Furthermore it can optimally allocate resources 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 is 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 multiple processes and 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 in multiple production steps. Most common, and as is implemented in the plant described, is to first remove the positively

---

\*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.

charged minerals in so called anions. After this, the negatively charged ions are removed with a cation filter. To ensure that all ions are removed, a final combined “Mixed” 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 and/or resource allocation, real-time usage of the cleaning resources is possible, including the optimal location, 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. Based on these methods a framework is introduced, after which a knowledge gap is determined. This knowledge gap, focussing on negotiation, is used to design and implement our model, the foundation of Chapter 4. In Chapter 5 the model is tested and evaluated. From this it is possible to conclude and detect further use as described in Chapter 6.

1. Introduction
2. Problem Definition and Research Goal
3. Literature Study and Theoretical Framework
4. Research Design and Application
5. Simulation comparison and Evaluation
6. Conclusion and Further work

## Chapter 2

# Problem Definition and Research Goal

An overview of the problem will be given, and based on the essentials of this problem, the research goal will be discussed. It is important to define the relevance and approach to the entire research.

### 2.1 Problem Analysis

Due to the fourth industrial revolution, new production and manufacturing methods are required which need new digital solutions to optimize their production. One solution is *centralized* analysis: combining all the data in a central database and analysing this to optimize decision making. Another solution, namely *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 (Leitão et al., 2016). Also, the practicality of different negotiation methods 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 (Leitão and Karnouskos, 2015).

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 and resource allocation production problems are typical Non-deterministic Polynomial (NP)-hard problems that are very complex to solve using (mixed) integer programming and take a long time to find an optimal solution. There is a consensus that multi-agent systems retrieve 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.

### **IoT and CPS**

The new developments in the industries, like the use of Internet of Things (IoT), require manufacturers to rethink their production process. An IoT is a network in which 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 steel 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.

Because the main scope of this research project is aimed at negotiation, the process 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 are comprehensive overviews of agent-based manufacturing, but the negotiation aspect is a commonly lacking subject (Leitão, 2009). In Chapter 3 a comprehensive overview will be given. By researching and, importantly, 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 by the way of a multi-agent system.

This research is about the application of multi-agent system technology, negotiation, game theory and decision making. Knowledge from artificial intelligence 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 require “on top of the line” production systems. By computationally implementing one of the processes and optimizing these processes, these insights can be applied for further use. An obvious solution lies in Multi-Agent Systems, but the exact implementation is difficult.

Furthermore, the insights of negotiation are very useful in every aspect of a business. A little more knowledge on how to optimize one’s negotiation helps optimize your business professionally.

## 2.4 Research Goal

The main goal is to optimize a process 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.
2. Create a simulator to show that a multi-agent system can be used for manufacturing/production planning.

3. Determine what steps are necessary for a business to make use of negotiation in such a new multi-agent system.

## 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) and 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.

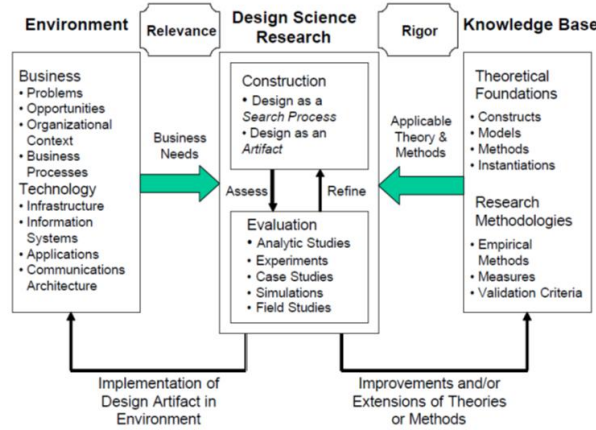


Figure 2.1: The Information System Research Framework as designed by Hevner and Chatterjee (2010)

In this research, a case-study is done to check the working of negotiation in this new MAS framework. By comparing the model with a real-world situation, the new MAS framework can be assessed and maybe refined. Furthermore, it can be determined whether this negotiation method can be used in a business context.

## 2.6 Research Process

Firstly a literature research was concluded to assess agent-based solution in the manufacturing world. Furthermore, the current negotiation methods in agent solutions were reviewed. From the literature a knowledge gap was found, which could be used in future manufacturing processes. Based on this knowledge gap, a mathematical model was created, to assess how the negotiation will concur in the multi-agent system. A simulator was created to evaluate this method. After the creation of the model and simulator, the relevance was assessed by its performance.

### 2.6.1 Evaluation Method

To test the final theoretical framework, a virtual simulation is created. By having the agents negotiate regarding the optimal resource allocation, and by using different negotiation methods, it can be shown that negotiation can be applied to find a possible (near-) optimal outcome. The model is to be evaluated using the known Nash solution. Since all the utilities are known, the optimal solution of the group can be determined. From this the effectivity of the method, and an evaluation of the model in aspects of speed, quality solution, and dynamicity can be made.

## 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?
  - (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 can be used to optimize decision making for production processes?
    - ii. Simulation: How does this new framework compare using an existing use case using simulation results?
  - (b) What is the general framework within the Industry 4.0?
    - i. What is the difference between a decentralized systems and a centralized system?
    - ii. How can negotiation be used in manufacturing?

## Chapter 3

# Literature Study and 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, called the fourth industrial revolution, results in a demand for products with high quality at lower cost while being highly customized. This results in new ways 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 to 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 is given. Using such a decentralized agent solution is only optimal when certain process and design requirements are realised on the manufacturing side. We will derive a framework on the basis of which one can decide between a centralized or decentralized system. After the framework is explained, we will give an overview of negotiation solutions in agent systems. From this we find a new approach to design a multi-agent system for a business process.



## 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 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 (Leitão et al., 2016). This is visualized in Figure 3.1.

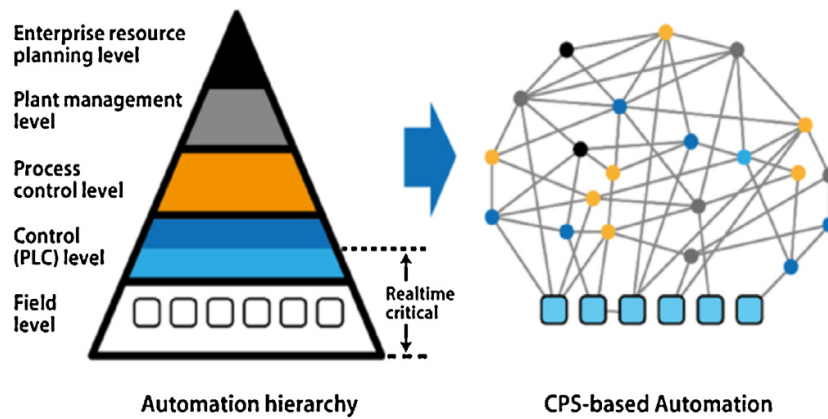


Figure 3.1: The breaking of the traditional automation pyramid (left) and the future of a new more decentralized way of function (right). Image from (Monostori et al., 2016).

The traditional automation pyramid is very similar to the multiple layers in the manufacturing process, which have been standardised by the American National Standards Institute (ANSI) (Harjunkoski et al., 2009). The integration of the planning and control in the manufacturing process 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.

By creating an overview of the different layers in the manufacturing process, an understanding of how optimization problems in these layers can be solved with agent solutions can be given.

### 3.1.1 Asset Management

On the top of the ANSI structure is the overview of the manufacturing process, also known as Asset Management, which 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

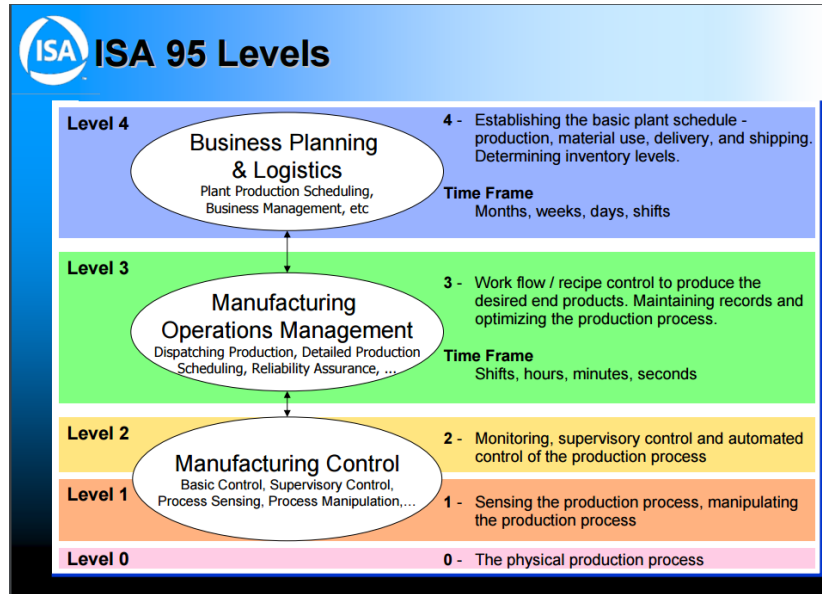


Figure 3.2: The manufacturing levels as described and defined by ANSI for the ISA-95 levels (Brandl and BR&L Consulting, 2008).

management on the one hand results in loss of the asset itself, and on the other hand results in loss due to production 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 of assets are well researched. This focus on the asset management 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. The asset integrity is crucial, and where most data is obtained, making it a preferred research domain (Lee et al., 2013).

Traditional manufacturing control systems are unable to be sufficiently responsive, flexible, robust and reconfigurable due to the fact that they are built upon 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 new process control methods which require new solutions to operate.

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 (da Rosa et al., 2009). These methods will be described in Section 3.4.

### 3.1.2 Process Control

One level lower than asset management is the order of process control, where 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 occurs 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 applications. The production of a car is for example a discrete process, since each part can be produced separately.

Batch production occurs when specific quantities of the materials have to be combined in particular ways. These are typically food production. An example is 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 results in the required product with no interruptions.

An example from Engell and Harjunkski (2012), which is displayed in Figure 3.3, shows the typical process control method. This is in line with the ANSI standardisation described in Chapter 1.

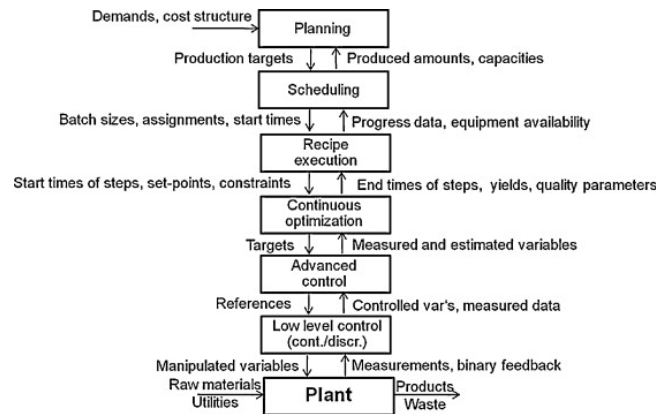


Figure 3.3: Typical process structure from (Engell and Harjunkski, 2012)

### Planning and Resource Allocation

When controlling a process, it is important 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 opti-

mization is essential in order to achieve the objectives and goals of a company. 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).

An important aspect of the planning is the allocation of resources. One can optimize the production process, but without the correct resources at the right location at the right time, the optimization is limited. This is a crucial aspect of planning, which is often done manually in large production plants due to the often unexpected and complex decision-making processes involved (Pinedo, 2005). By automating the resource allocation, a part of the overall planning can be automated.

Another large difficulty when planning, is that of ensuring that the assets are always operational, or that they 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. There are two possible ways of maintaining: planned and unplanned. These are often currently both included in a planning, since a lot of maintenance is done unplanned (Dey, 2004). However, all maintenance results in downtime, and is preferably left out, to keep operations running. This however results in the breakdown or wear-out of the systems. By 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 plants, 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, such companies need a high inventory of spare parts and materials. And usually the repair is more expensive than preventive maintenance.

Preventive maintenance is the alternative to corrective maintenance. Using predetermined fixed-interval planned maintenance, the assets are maintained. However, this results in uncertainty 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 environment like sun, dust and rain)? Often either maintenance is done too soon, resulting in extra cost, or too 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 break down and there is no overkill on maintenance. On specific intervals, the machines are measured regarding their current status and using, for example, vibration mea-

surements 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 still not the optimal solution: measurements are sporadically done (not continuously) and there remains the chance of failure before the maintenance stop has occurred. This method also depends on checking a single threshold value, and whether it has been reached.

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 prediction of failures and real-time maintenance. It does not require a specified threshold to be reached. Thus, it is more accurate since a combination of variables which individually have not reached a threshold, but together might cause failure, can be detected.

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), or intelligent maintenance (Vermaak and Kinyua, 2007).

### **3.1.4 Real-time Monitoring**

To ensure that processes are running according to plan and that 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. This may be the case if rescheduling is required multiple times a day because of schedule changes. 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 through graphical charts and bars on a dashboard. This, however, requires manual input, or an algorithm that comprehends all the data. 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. However, when using a programmable logic controller, the decision process is done on a very low level and optimization is difficult.

## **Manufacturing with Agents**

When dealing with multiple processes, in production and manufacturing, and when having 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 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 in combination with communication, planning can be optimized. This structure is also similar to that of the ANSI. We have a high-level object which can consist of multiple lower-level objects. First some terms have to be discussed, after which we can link the manufacturing processes to the agent-based solutions.

### 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 in OOP is an activity 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, and thus very similar to an agent (Shoham, 1993).

Agent-oriented programming is a method often used to implement a multi-agent system, see (Mahar and Bhatia, 2012) for a thorough overview. In such a system anthropomorphic ideas, like beliefs, desires are used to model the objects, and thus called agents (Shoham, 1993).

### 3.2.2 Multi-Agent Systems

Some terms used in the literature for data collection apparatus that aggregate the data are “Smart Objects”, “Intelligent Gateways”, “Collaborative Networks”, “Wireless Sensor Networks” 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 is shown in Figure 3.4.

To define MAS, an agent needs to be defined more precisely. 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

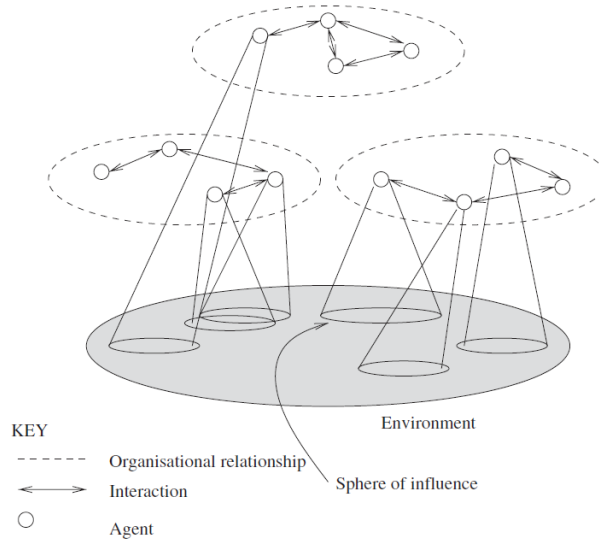


Figure 3.4: Typical structure of a multi-agent system (Wooldridge, 2009).

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 and Georgeff, 1995).

Multi-agent systems (MAS) have been identified as some 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, such as a lack of focus on multiple owners, decision making with only available local knowledge research and lack of collective sensing strategies, are still subjects that require extensive research. Vinyals et al. (2011) see these as the possibly most active MAS research topics. Many of these problems can be solved with negotiation, which will be covered in Section 3.4.

### 3.2.3 Holonic Systems

Another way of looking at agents, and more convenient in the manufacturing world, is by looking at MAS as holonic systems. Multi-agent systems are composed of autonomous software entities which allow them to be 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 is an intelligent entity, just like an agent, and able to interact with the environment. This allows the holon to take decisions to solve a specific problem. The holon has the property of playing the role of a whole system and a single part at the same time. The first success-

fully implemented holonic structure was created by Van Brussel et al. (1998). PROSA, the name given to the holonic structure, consisted of three types of basic holons: order holons, product holons, and resource holons. Van Brussel et al. (1998) structured the system using the object-oriented concepts of aggregation and specialisation. By decoupling the system structure from the control algorithm, logistical aspects could be decoupled from technical ones. After they compared the holonic system with existing manufacturing control approaches, they concluded that the holonic system was able to cover all aspects of both the traditional pyramid and decentralized control architectures, meaning that it could be regarded as a generalisation of the two.

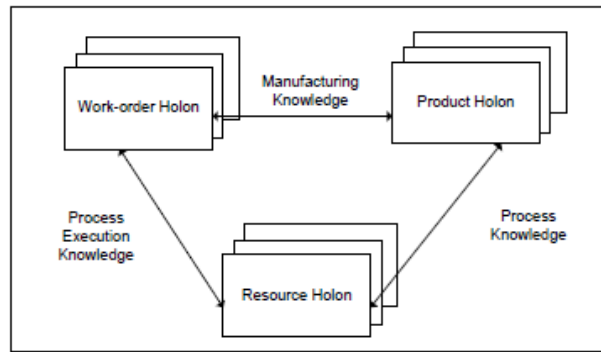


Figure 3.5: An example of an Holonic Manufacturing System (from Giret (2005), based on Van Brussel et al. (1998))

The concept of holon is based on the idea that complex systems will evolve from simple systems more rapidly if there are stable intermediate forms, than if there are not. This means that the resulting complex systems will be traditional pyramids. However, although it is easy to identify sub-wholes or parts, holons do not really exist anywhere, making them decentralized (Van Brussel et al., 1998).

### 3.2.4 Task and 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 or limited. Furthermore, it can be a 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. The preference of the agent is often crucial. 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 at which time (Fatima et al., 2014b).

The most common example of resource allocation is that of a pie. How should the pie be divided among the agents? Many strategies have been designed for



solving this issue. Another example could be the allocation of energy. Which processor gets how much energy? These resources can be continuous or discrete.

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

By optimizing the allocation of the resources, a more efficient production can be achieved, with less waste. See Section 3.4 for the difference in the task or resource allocation when dealing with negotiation.

### 3.2.5 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 (Hromkovič, 2013). This means that it is possible to reduce the problems in NP to the original problem, such as SAT (propositional satisfiability), 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 as for example applied in (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, (Feng et al., 2014) shows that decentralization 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. This is often done using a mathematical programming approach. Zhou et al. (2004), try to use a MAS to heuristically solve the bus maintenance scheduling problem. It is shown that with equal optimality and less computing time without constraint violation, the MAS solution is comparable to the work of a mathematical programming approach.

It is also shown in Brucoleri et al. (2005) that the agent-based approach outperforms 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 within 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; this is a context in 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 can be achieved (Rabelo et al., 1999).

### **3.3 Framework of a centralized and decentralized system**

When looking at the traditional pyramid, which is fully centralized, it seems difficult, if not impossible to translate this to a decentralized solution as shown in Section 3.1. It should be possible to determine whether a centralized or decentralized solution, using a MAS or holonic system, should be implemented at a business process. A framework to compare a centralized versus 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 by Parunak (1999), is that of the original Roman army structures. Decisions were made at the top and dripped down, while the information stream went up. This method has been deployed in most companies. Due to the fact that something can be computed on a single computer, and be optimized on this single program, an optimal decision 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 originate from different locations. To solve such difficulties, computers have to act more like agents where each agent can solve, or decide on part of the problem. This is where agent-based architectures are an ideal fit to such a decentralized organizational structure.

To push the decision making to the lowest level, excessive layers of management can be obsolete. 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.

### 3.3.1 Size and Modularity

A 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 or task that have to be allocated. If a clear structure is conceivable and a clear population is in place a centralized solution is infeasible. This is due to the global overview. However, the high sensitivity to size and complexity makes a centralized solution impracticable.

In a decentralized structure, individual models are decoupled from one another, errors in one module impact only those modules that interact with it, leaving the rest of the system unaffected. This can be seen in Figure 3.6. It shows however the importance of having a clear modular problem.

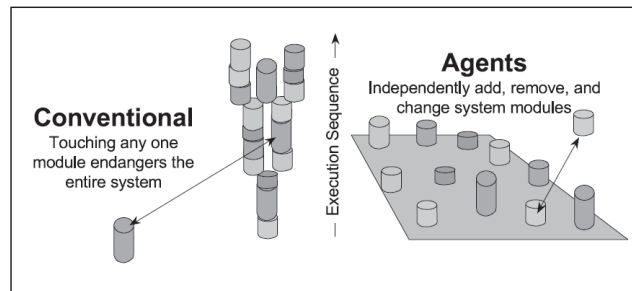


Figure 3.6: Comparison of a conventional control thread and an agent-based control, from (Parunak, 1999).

### **3.3.2    Dynamicity (Time Scale/Changeability)**

In a decentralized 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 computational incredibly difficult (Duffie and Prabhu, 1994).

### **3.3.3    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 be reached through communication and negotiation, but for a truly optimal solution, an entire view is necessary. 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 complete information is necessary.

### **3.3.4    Complexity**

Since an agent can execute actions only on its own surrounding, it is dependent on its local parameters. However, the agent can use information sent by its neighbours to adapt (Pujolle, 2006). This interaction between the elements makes the complexity of a solution many times higher and more difficult than a centralized solution.

### **3.3.5    Framework Overview**

Below a summary of the points above is given, with respect to the structure given. It is obvious that a decentralized solution is preferred, if the problem can be divided into sub problems. However, the real difficulty then lies in the complexity. Since in the system the communication becomes essential, the complexity increases.

	Centralised Solution	Decentralised Solution	Building Blocks
Size / Modularity	Small; No sub-problems	Large; Ill-Structured; Easily dividable; Independent Modules	Population; Holonic; number of resources: (decision variables, parameters & constraints) (Lang and Fink, 2015)
Time scale and Changeability	Days - Weeks; Not subject to a lot of change	Real-time - Hour; Changeable	Adaptive Capability ; Degree of Re- and Pro-activeness (Parunak, 1999)
Solution quality	Perfect	(sub-)Optimal	Object and Solution Space (Sharif and Huynh, 2012)
Complexity	Simple	Complex	Interaction between the set of elements; Communication (Pujolle, 2006)

## Negotiation in a Decentralized Structure

By decomposing the problem in smaller sub problems that a single agent can compute, and solve, the communication of the agents is essential. In order to integrate the solutions of the sub problems into the overall solution, the agents, which might not be cooperative, need to use negotiation.

### 3.4 Negotiation

The negotiation of the agents in a multi-agent system has often been discussed above. This branch of research, also called automated negotiation, is studied by both artificial intelligence and economics (Jennings et al., 2001). Concepts from fields such as decision theory and game theory are used in the design of appropriate negotiation and interaction environments (Jennings et al., 2001). Negotiation is used to reach an agreement that meets the constraints of two or more parties in the presence of conflicting interests. And thus it 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 information state of agents and domain;
2. The negotiation protocol;
3. The negotiation strategies;
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 there 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 negotiation is seen as maximizing the quality of the result. Two types of optimization are possible: one, the agents can try to achieve Pareto optimality, meaning that the outcome maximizes the product of the agents' utilities, or two, they try to reach a Nash equilibrium, meaning a stable state in the system. Both ways will be discussed in the evaluation of the model Chapter 5.

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.

### 3.4.1 Negotiation Domain

As discussed before, planning can be seen as concerning multiple different tasks, task allocation and resource allocation. The same holds for the negotiation domains, which can be divided into task oriented domains (TODs), state oriented domains (SODs) and worth oriented domains (WODs) (Rosenschein and Zlotkin, 1994). 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 unlimitedly available, and the advantage of negotiation is that it allows for the redistribution of tasks amongst a group of agents which can results in a more efficient task order. A typical example is that of mail delivery where an agent may carry another agent's mail at little extra cost. It is certain 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 (Rosenschein and Zlotkin, 1994).

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 classic example. Here the agents have to place as many blocks vertically as possible. However, the catch is that the agents must sometimes remove a block to access another block. This gives the possibility of conflict and dead end, since the agents may have different goals, and it is not feasible to try to satisfy all these goals for all agents. This means that the agents must be able to make concessions in order to reach an optimum. These concessions can be in the form of a joint plan (Rosenschein and Zlotkin, 1994).

WODs are where agents attach a worth to each potential state, using for example a utility function. This allows more flexible goals to be set and allows concessions to be made on these goals. An example would be agents in a marketplace where the goal for a seller may be to obtain the highest price for  $x$  within time  $y$ , while the buyer tries the to obtain the lowest price. 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).

Utility Function
A utility function is a way of mapping the desirability of a state to an agent. So the higher the utility of a state, the more desirable this state is. When making concessions, an agent accepts states that are less desirable.

### 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 research. 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 information category, information may be lacking about a variety

of factors in the problem (Fatima et al., 2004). The incomplete information state is of course most common in applied negotiations since it is not possible to include all the possible information of the world. Furthermore, private utility functions can be desired to not give away ones intention.

### 3.4.2 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 determinable when (1) the negotiation domain, including the issues and possible issue values, (2) a scenario utility histogram, which defines the distribution of the scenarios, and (3) several structural parameters that specify the configuration of each agent's utility function are known.

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 proposals 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 a MAS. Unlike the negotiations between human beings, which involve more complex human interactions than those about simple technical issues, the negotiation mechanisms between agents are rule-based or case-based due to these clear protocols. However, the human negotiation approaches and theories, which mainly include game theory and human behavioural theories, provide a proper foundation for the negotiations between agents.

The most important protocol is that of the alternating-offers protocol (Rubinstein, 1982). It is based on a divisible pie, discrete or continuous, and is the most widely studied among game theorists as well as MAS researchers (Fatima et al., 2014b). Each agent is allowed to make a single offer, and the proposal that yields the higher product of all the utilities of the agents is 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. Another example is the contract net protocol (explained in Section 3.4.7).

The most common protocol to ensure concession is the monotonic concession protocol. It is a proposal which has also been adapted for multilateral negotiation in (Endriss, 2006), and can cope with different strategies.



### 3.4.3 Negotiation Strategies

A negotiation 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 protocols. In general, for a given set of negotiation protocols 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 on the negotiation 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).

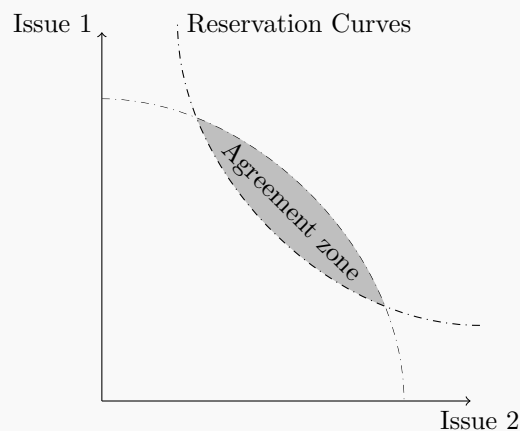
#### Concession Strategy

When negotiating, it is essential for the agents to make concessions, only in TODs it is unnecessary, as explained. 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.

#### Definition: Reservation curve and Agreement zone

When an agent has a utility, it usually has a minimum value to which it will concede. This curve of all points that have the minimum utility value is called the reservation curve.

All the points on the reservation curve are states that the agent values the same, and thus have the same utility.



Multiple agents have different reservation curves. The subset of all the agents reservation curves is called the agreement zone. The agreement zone must be non empty to be able to find a solution.

When dealing the requirement of conceding, there are multiple strategies. Most of these are one version or another 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 two options. Either the agent refuses to make a concession and sticks to the previous proposal or it makes a concession and proposes a new deal that is less preferable to the 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).

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 known since it is not possible to discover the group maximum using private utilities.

Four concession strategies are given by Wu et al. (2009):

**Amount of utility** an agent concedes a fixed amount utility per time.

**Fraction of utility** an agent concedes a fraction of the desired utility.

**Fraction of the difference** the agent concedes a fixed fraction of the change in current desired utility and a reference point.

**Fraction of remains** the agent concedes a fixed fraction of the issues that no agreement has been made on yet.

Wu et al. (2009) found very little difference between the performances (distance from Pareto-optimum) of the concession strategies, although the fixed step was the quickest.

#### Definition: Social welfare

When an agent makes a Pareto improvement, it proposes a new offer that harms nobody but benefits at least one member of society. This is seen as social welfare, since it results in an improvement for the entire group. Endriss (2006) defines two concession methods specifically that assess the well-being of a group instead of the individual.

An improvement in the *utilitarian social welfare* enjoyed by a group, for instance, is defined as the sum of the utilities of its members. The Utilitarian concession as a result is as follows: Make a proposal such that the sum of utilities of the other agents increases.

Another option is the *Egalitarian concession*: Make a proposal such that the minimum utility amongst the other agents increases. This is a form of egalitarian social welfare where the overall system performance is measured by the agent with the lowest utility level.

It is obvious that in order to make social concessions, the utility functions of the other agents must be known.

When dealing with private utility functions matters change completely. There is no way of knowing whether the other agents have conceded. A solution to this is proposed by Zheng et al. (2016) using the reactive concession protocol.

### Reactive Concession Protocol

The reactive concession protocol as proposed by Zheng et al. (2016) 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. (2016) show that by having an agent consider its own utility change resulting from another agent's offer, it can concede accordingly. There are two cases. If the change in utility that the other agents' offer caused has resulted in a 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 agent's concession. Exact details will be discussed in Section 4.3.1.

### 3.4.4 Evaluation and Equilibrium Solutions

When evaluating the dilemmas of a negotiation between agents, it is essential to determine 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<sub>i</sub> plotted against that of agent<sub>j</sub>. In the set of all possible outcomes, all the possible agreements are situated. These are all offers that are acceptable by both agents.

An offer is Pareto optimal if the agents cannot choose a new offer for which at least one agent has a higher utility, while the other agents have at least the same utility. In the figure these are shown as *A* and *B*, and each point on the arc inbetween. 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

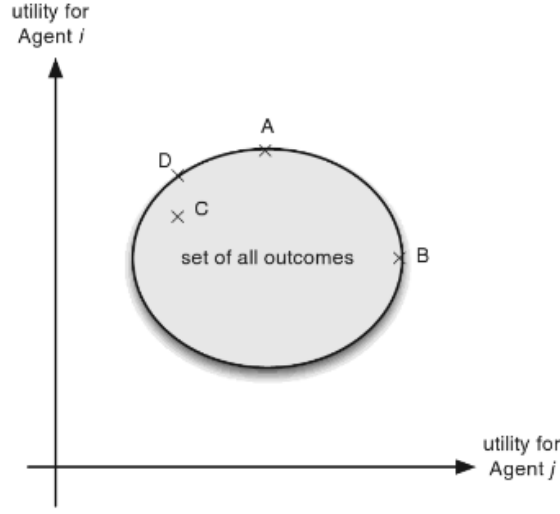


Figure 3.7: An example of Pareto optimality for two agents. Locations A and B are optimal, since no improvement for at least one agent, without loss for the other agent, is possible. The points on the outer arc between A and B are Pareto optimal as well. C and D are not optimal since they utility of the agents can be increased. From (Fatima et al., 2014b).

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 called the Pareto-Frontier.

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

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 given in (Trappey et al., 2013).

The most common Nash equilibrium and probably most widely known is that of the prisoner's 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

		<i>Agent i</i>	
		defect (confess)	cooperate ( $\neg$ confess)
<i>Agent j</i>	defect (confess)	(-3, -3)	(0,-5)
	cooperate( $\neg$ confess)	(-5, 0)	(-1,-1)

Table 3.1: The prisoners dilemma

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  $s_1$ . 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 prisoner’s dilemma, the Nash equilibrium is the only solution which is not a Pareto optimum. The optimal social solutions (see page 27 for an explanation), 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.

### 3.4.5 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). Fisher et al. (1987) emphasize 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 criterion, it is more likely that it is rational. Furthermore, principled negotiation is useful for systems in which no agent has global knowledge of the system.

### 3.4.6 Negotiation using a Mediator

A mediator can be used when negotiating. It is often used in typical hierarchy, voting, and auction based negotiations.

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).

However, in mediated negotiation, the agents have no desire to make concessions, and an (sub-)optimal solution cannot be guaranteed.

### 3.4.7 Mapping of Negotiation Protocols

An attempt at the visualization of the different negotiation techniques is strived at. Three variables are decided on. Single- versus Multi-Issue negotiation; bi- versus multilateral negotiation, and; perfect versus imperfect information negotiation. The overview given here will be explained in the following sections.

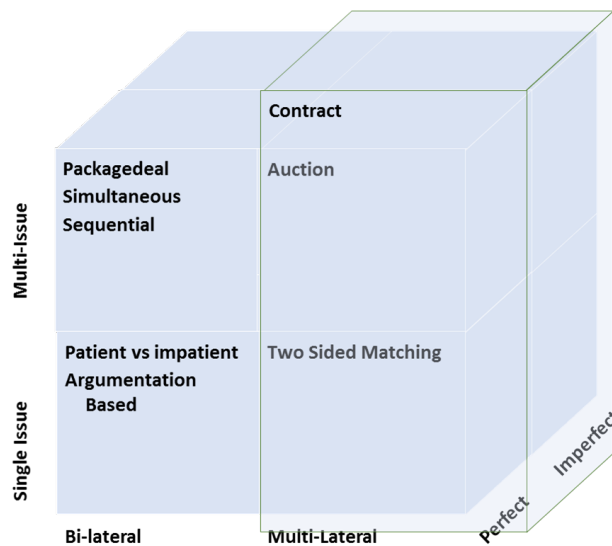


Figure 3.8: Overview of the negotiation protocols used and researched in the literature. The usage of a proposal-based protocol for a multilateral multi-issue situation is missing in the manufacturing literature.

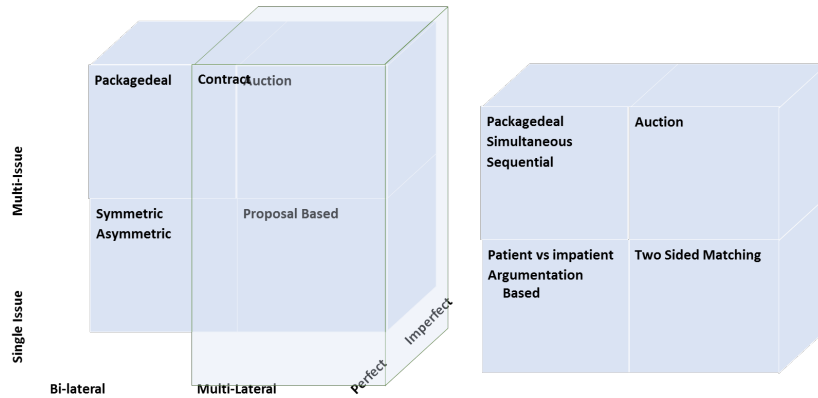


Figure 3.9: For completeness, the same figure as in Figure 3.8, but with the display of the imperfect information protocols which is more applicable in the manufacturing industries.

### Single-Issue Negotiation

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 is 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 versus impatient players and two sided matching (Fatima et al., 2014b). Argumentation based methods, which are based on the beliefs of an agent are also included in the mapping (Li et al., 2013).

Essential is that all these methods are forms of the alternating offers protocol. Depending on the sort of players, the method result in completely different behaviours. These negotiations can either be complete or incomplete meaning that all information is known, or not all.

When the game is ‘complete information’, 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 information’. The idea of negotiation is that we have an incomplete game, since if the strategies are known, most negotiations would not be necessary.

Looking at perfect versus 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.

In single issue negotiation, depending on whether the negotiation happens between 2 (bilateral) or more (multilateral) agents, there are a few protocols pos-

sible. 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 negotiations in the manufacturing are time restrained, thus impatient agents must be implemented (Kraus et al., 1995). In symmetric versus asymmetric negotiation the players are uncertain about the other player's utility functions (as is the case in imperfect information negotiation), but essential is that one agent might know more than the other in the asymmetric protocol.

## Multi-Issue Negotiation

When negotiating about multi-issues, agents attempt to combine two or more issues in their discussion. 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 is the package deal method. In this method, complete packages with all the issues are provided. These can be discussed either sequentially or simultaneously.

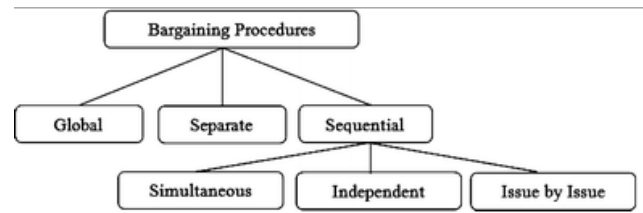


Figure 3.10: An overview of the different negotiation method for multi-issues bargaining. From (Abedin et al., 2014).

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). Abedin et al. (2014) think that a packaged approach is optimal since the agent lack knowledge about the opposing agent. 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.

When choosing the preferred method of negotiation, important to realize is the solution required. As explained above, Abedin et al. (2014) say that the issue-by-issue approach has a higher chance of obtaining the Pareto optimal solution. However, since the utility of an offer is not simply a sum of the utilities of the individual issues, some prefer to use the package deal (Zheng et al., 2016).

## Multilateral

The most commonly used methods for multilateral negotiations are contract based methods, most popular being the contract net protocol. The 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, called a manager, that has some work to be subcontracted, broadcasts an offer and waits for other agents, the contractors, to send bids. After some delay, the best offers are saved 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 for problems in which it is appropriate to define a hierarchy of tasks. Since such problems allow to be decomposed into a set of relatively independent subtasks, there is little need for global information or synchronization. These subtasks can be assigned to separate agents. The contract net protocol main contribution is the mechanism that it offers for structuring high-level interactions between nodes for cooperative task execution (Smith, 1980). With these contracts task allocation is possible. The allocation of resources gives difficulties however.

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

## **Heuristic methods in Negotiation**

Most of the negotiation in manufacturing can be seen as multilateral 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 is needed. For the second aspect a multi-issue concession strategy is

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

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 a context 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 in (Leitão et al., 2013; Monostori et al., 2006), since they do not require the thorough analysis of the states and protocol compared to the game-theoretic methods. Also they allow 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 3.11. This can be implemented in the negotiation of agents by having the agents learn their utility functions, for example. However, based on the research conducted on heuristic methods by Jennings et al. (2001), it can be concluded that the optimal research in learning in heuristic methods is not yet known. These methods are often used however, to decide on the optimal counter bid. In (Beheshti and Mozayani, 2014) it is shown that efficient learning algorithms based on a statistical ranking algorithm and linear regression are a suitable learning method and all have linear time complexities. These characteristics allow for a method to be used in real-world applications.

### 3.5 Negotiation in Manufacturing

There are many applications of agent-based solutions in the manufacturing world (Monostori et al., 2006). In the applications an overwhelming aspect 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 lies, since conflicting interest, essential in the optimal decision making, are left out. An example where these 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.

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., 2011). One of their insights on 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 are needed.

Overall, nearly all factory scheduling negotiations use some form of these market-based approaches (Monostori et al., 2006) to implement the solutions. Different versions of the contract net protocol were used, as well as 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 outperforms the centralized mixed integer programming solution. This system uses bilateral simultaneous negotiation on the medium level of the production plant. The system uses 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 exceed even more.

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 in the literature, there are two gaps. Firstly, only a few multi-issue multilateral strategic applications have 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 has not been implemented in any real use case, 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 network 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 yet.

From these knowledge gaps we try to determine whether it is possible to build a multi-issue, multilateral negotiation where the utility functions are private. Since it is unknown to the agent whether a concession is made by the other agents, we compare the reactive concession strategy by Zheng et al. (2016) to a typical non-reactive concession strategy.

The option of having a private utility function allows many more applications, since it allows for negotiations between competitors. This is in line with the optimized principled negotiation, as discussed.

## Chapter 4

# Research Design and Application

As discussed in the literature review (Chapter 3), this research has a focus on negotiation between agents. Using different methods and techniques, an 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. This knowledge is applied to a reference model as explained in Section 2.5 to ensure that the research has relevance.

The system of which a decentralized solution will be simulated is a de-mineralized water plant as described in the introduction and problem chapters (Chapters 1 and 2). This continuous production process is optimal to check the workings of negotiation in a multi-agent system. There are limited resources, that have to be allocated towards competing components. Currently there is no optimization in place at this plant, and every decision is made manually. By optimizing this plant, fewer materials will be wasted, having an environmental impact and cost improvement.

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, an assumption in a multi-attribute negotiation domain is made, 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 and Jade (Bellifemine et al., 2007), which is optimal for communication research in a multi-agent system (Kravari and Bassiliades,

2015). However, since most of these systems are very comprehensive, small adjustments which are necessary for our research are not conceivable. Secondly, specific requirements, which were designed specifically for this use case, such as the projections and reactive concession, have not been commonly implemented yet.

## 4.1 Demineralization of Water

As discussed in the introduction, the use case 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.

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” shorthand.

So overall, there are 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 only looked is 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. The resources are the acid that is needed for cleaning, the base for cleaning, and the water that has to be delivered at the end of the process. Thus, the resources are:

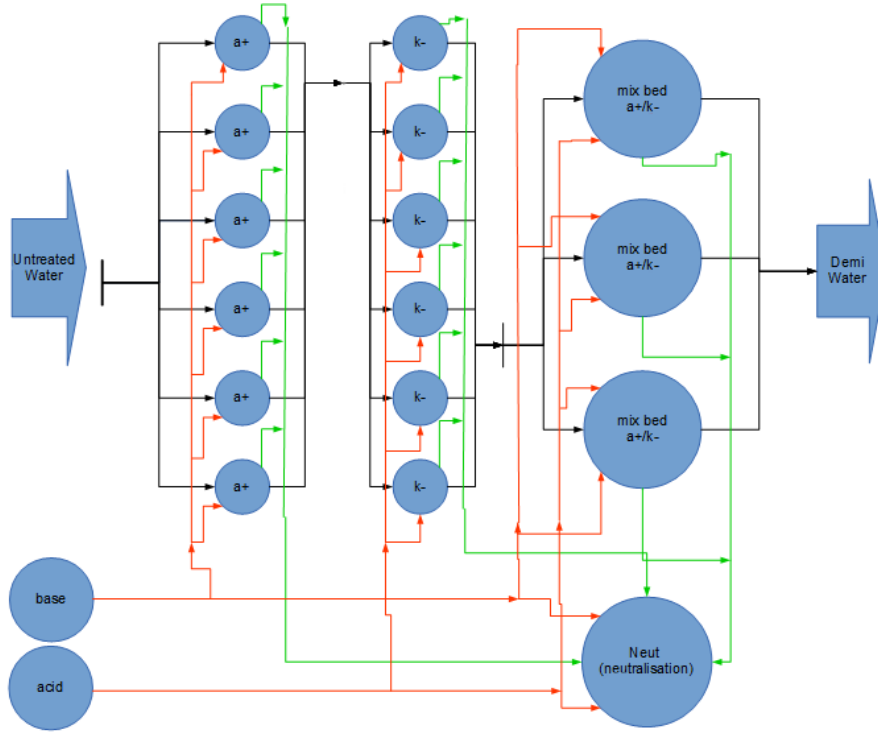


Figure 4.1: An overview of a water demineralization plant. The many anion and cation filters are simplified to a single anion and cation filter. The same for the Mixbed filter. Result is Figure 4.2

1. Acid
2. Base
3. Water

The decision on the amount of resources used will be done by negotiation and combined with the information that is retrieved from the experts knowledge.

## 4.2 Negotiation Model

The four rational agents defined in Section 4.1, namely  $m = \{1, 2, 3, 4\}$ , partition the three issues defined  $n = \{1, 2, 3\}$ . This can be simplified to a multilateral buyer-seller negotiation.

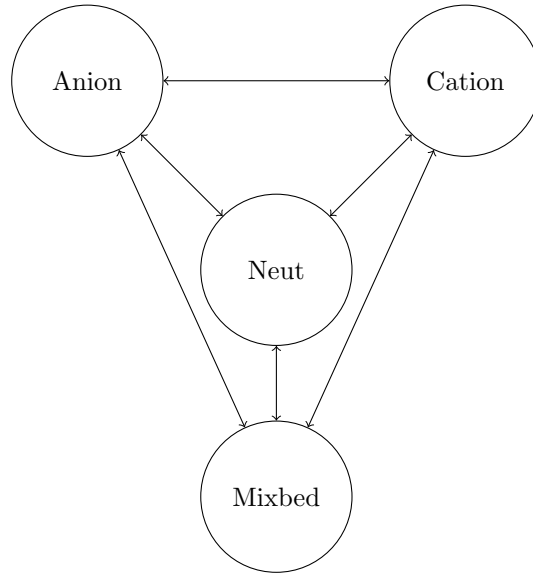


Figure 4.2: A simplified representation of the four agents in the negotiation. See Figure 4.1 for the overview of the water plant on which the model is based.

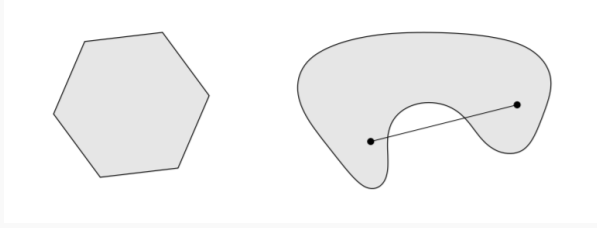
Anion, agent 1, wants as much base as possible, while it wants to minimize the amount of water. For the cation, agent 2, as much acid as possible is required, while still as little water as possible should be produced. The Mixbed, agent 3, requires as much as possible acid and base for cleaning. Since it is 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, agent 4, 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 there are 3 issues, the unit hypercube  $[0, 1]^3$  in  $\mathbb{R}^3$  is the result. This results in a possible negotiation domain  $\Omega = [0, 1]$  per issue.

**Definition: Convexity**

A convex set is a region such that if one were to connect two points in this set by a straight line segment, each point on that line segment is also within that set. Or formally:  $C$  is a convex set if the line segment between any two points in  $C$  lies in  $C$ , i.e., if for any  $x_1, x_2 \in C$  and any  $\theta$  with  $0 \leq \theta \leq 1$ , it is that  $\theta x_1 + (1 - \theta)x_2 \in C$ .





This can be clearly seen in the figure above. Two sets. The left is convex, while the right, kidney shaped is not, as indicated with the line outside the set. From (Boyd and Vandenberghe, 2004).

The boundary of a convex set is always a convex curve. A function is called a convex function if the set of points on or above the graph of the function is a convex set. Well-known examples of convex functions are the quadratic function  $x^2$  and the exponential function  $e^x$  for any real number  $x$ .

Convex functions play an important role in multiple areas of mathematics. They are especially important in optimization problems since they have some convenient properties. For instance, a convex function has no more than one minimum. Even in  $n$ -dimensional spaces convex functions continue to be convex. Furthermore, convex functions are optimal when dealing with multiple convex sets, since the intersection will always be convex as well meaning that the properties of the new sets will contain the same properties (Boyd and Vandenberghe, 2004).

The utility function for each agent  $u_i(x)$  is convex, as described in the definition box, 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, or the agreement-zone, 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,  $\mathbb{Z} \cap \bigcap_{i=1}^M A^i$ .

The protocol used is that of the alternating offers protocol, based on (Rubinstein, 1982) as described in Section 3.4. 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., 2016).

It is a difficult task to determine whether the solution is a Pareto-optimal solution, 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 Section 4.2, 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 a time series of the agent's desirable utility at time  $0, 1, \dots, t$ . This is written as  $s_i(0), s_i(1), s_i(2), \dots, s_i(t)$ . 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:  $\forall t, s_i(t) \geq ru_i$ .

Since 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  $\forall t, A_1^i \subseteq A_2^i \subseteq \dots \subseteq A^i$ . In this model, the *amount of utility monotonic concession protocol* as described in Section 3.4.3, is used since it performs well, and has the advantage of finding a solution in a specific time.

#### 4.3.1 Reactive Concession Strategy

The reactive concession protocol as described by Zheng et al. (2016), and also shown in Section 3.4.3, is a specific concession strategy to determine the amount of utility to concede. The non-reactive concession follows a predefined concession strategy  $s_i^0(1), s_i^0(2), \dots, s_i^0(T)$ , where  $T$  is the maximum amount of rounds. The reactive concession,  $(s_i(1), s_i(2), \dots, s_i(T))$ , amount is determined by each agent by the utility change resulting from other agents' offers. The nonreactive concession made at time  $t$  is  $\Delta u_{i0} = s_i^0(1) - s_i^0(t - 1)$ .

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 a concession according to the non-reactive concession strategy as calculated above.

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 18, which is found on page 54

The perceived change of utility for agent  $i$  from agent  $j$  is defined as the difference between the utility of the current offer and that of the previous best offer:  $u_i(x_t^j) - u_i(x_{[t-1]}^j)$ .

The total perceived change since the beginning of the negotiation is equal to the difference between the utility of the current offer and that of the first offer:

$u_i(x_t^j) - u_i(x_0^j)$ . Since an agent only want to concede compared to their own concessions, this perceived change is subtracted from our change:  $(1 - u_i(x^i(t - 1)))$ .

The maximum of these function is the amount the agent is willing to concede per agent:  $\max\{\Delta_1 u_{jk}(t), \Delta_2 u_{jk}(t), 0\}$

This value is determined for each agent, whom then takes the minimum amount of all these values, to determine its concession  $\min\{\min_{k \in \{1,2,3,4\}} \Delta u_{jk}(t), \Delta u_{j0}(t)\}$ .

The convex, see the box on page 41 for an explanation, requirement makes it obvious why the reactive concession can be used. If a non-convex utility function has been used, it could not have been determined whether an agent concedes if it moves.

### 4.3.2 Offer Generation

The sequential projection strategy for multilateral negotiation to generate the offers is used. Zheng et al. (2016) have shown that this method converges if all agents have an incentive to concede.

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

To create the new offer, the agent  $i$  will first calculate the weight of all the offers by other agents ( $j$ ). This is done by  $\sum_{j=1}^m a_t^{i,j} x_t^j$  where  $\sum_{j=1}^m a_t^{i,j} = 1$ . Where  $a_t^{i,j} = \frac{1}{m} = \frac{1}{4}$ . It could be argued to divide by three, since each resource is negotiated over by three agents. However, by dividing by four, a global overview is obtained which is preferable.

This weighted average of the offers is projected by on the border of its feasible offer set  $A_t^i$  by the agent who has to propose,  $x_{t+1}^j$ . This boundary, or indifference curve, is the set of points for which the utility is equal to the desired utility of agent  $i$ . 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.

### 4.3.3 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.

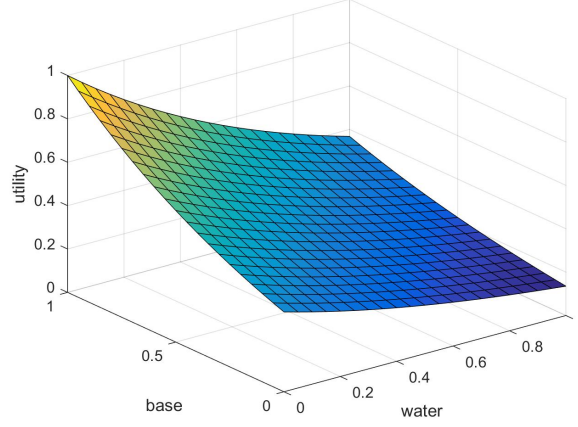


Figure 4.3: The utility function for the Anion.

### Utility Function

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

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

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 chosen based on interviews 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 choosing 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.

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

If the indifference line is known, 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 (the acid offer can be disregarded in this example, since the Anion has an indifference for the amount of acid used).

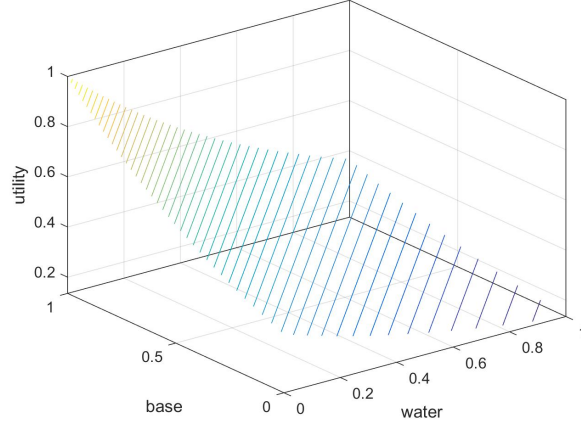


Figure 4.4: The visualization of the indifference curves for the Anion agent.

A projection to the indifference curve of the Anion is achieved with the following formulae. It has been thoroughly proven that if a function is written as,  $ax + by + c = 0$ , 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 the point of the line is obtained, 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 can be proven by algebraic proof, for example, by finding the line that is perpendicular to the original line, and  $-\frac{b}{a}$  (the negative reciprocal) by letting it pass through the point  $(x_0, y_0)$ . Then it is obvious that the distance is indeed as shown above, with the minimal distance point  $x$  and  $y$ .

The above is important since Zheng et al. (2016) have proven that if the projection is the point with the shortest minimum Euclidean distance, the algorithm will converge for any concession strategy.

So given the situation above, the function is  $ax + by + c = 0$  where  $a = 1, b = -1, c = -(\ln(0.4) + 1)$ . Given the information above,  $x_0 = 0.4$  and  $y_0 = 0.7$  this gives us the solution of the new  $x = \frac{2.1 + \ln(0.4)}{2}$  and  $y = \frac{2.1 - \ln(0.4)}{2}$ , 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 than the desired utility, (the point is above the indifference curve), the proposal will not

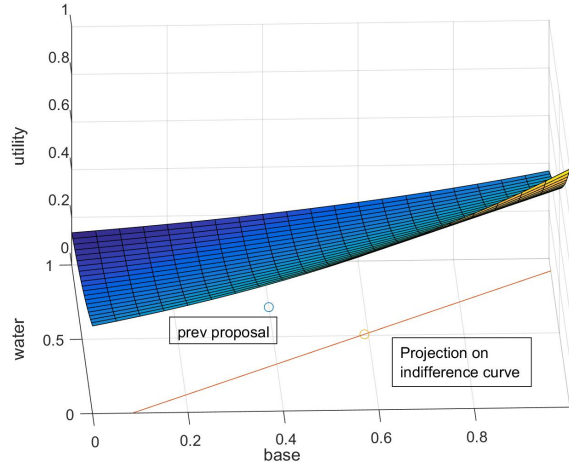


Figure 4.5: Example of the projection of a point to the indifference curve for an Anion. The utility here is 0.4, and the offer is base = 0.4; water = 0.7.

be accepted. The advantage of using a linear indifference curve is that we don't have to deal with an minimization function ( $P_A[x] = \arg \min_{q \in A} ||q - x||$ ) which increases speed dramatically. If a point is projected outside the  $[0,1]$  boundary, the intersection of the indifference curve and the boundary is the closest point.

#### 4.3.4 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  $0 = -W + A - (\ln(0.3) + 1)$ , or above, is acceptable for the cation.

#### 4.3.5 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.

$$\text{Neut utility} = \frac{e^{-A-B}}{e^0} = e^{-A-B}$$

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 expert interviews. These two reservation curves are shown in Figure 4.6.

Furthermore does the Neut have the third reservation curve, dependent on the reservation utility. If the reservation utility is 0.3 the third reservation curve is:  $0 = -A - B - (\ln(0.3))$ ,

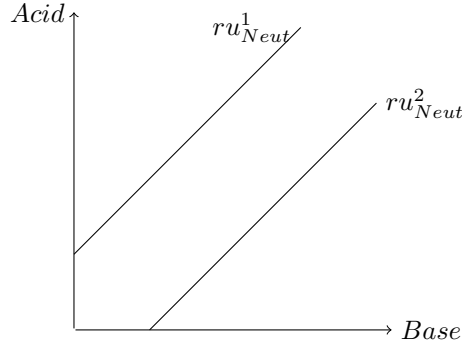


Figure 4.6: The neut reservation curve of the Neut. The requirement of the acid and base staying near each other is achieved.

### 4.3.6 Mixed

The Mixed 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 cation, it has three issues about it has desires. This is realized with the function below:

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

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:

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

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

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

The shortest distance from a point to a plane is along a line perpendicular to the plane. By calculating the normal vector, which is perpendicular to the plane, the line that is parallel to the normal vector is found, which goes through point  $(x_0, y_0, z_0)$ .

Important to note here 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 higher 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.

$$\Delta_{ij} = \frac{1}{e^4} + \sum_{i=1}^m x_i^2$$

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

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

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 4.7, 4.8 and 4.9.

So although there is a multi-issue negotiation, the only agent that has interest in all three issues is the Mixed. This means that if an agent proposes 0.7 Acid to the anion, the anion will not consider this part of the offer, since it cannot project this to its' indifference curve. This means that the anion's new proposal will be the average as shown in the offer generation (Section 4.3.2) for the acid, and the adjusted values to the other issues.



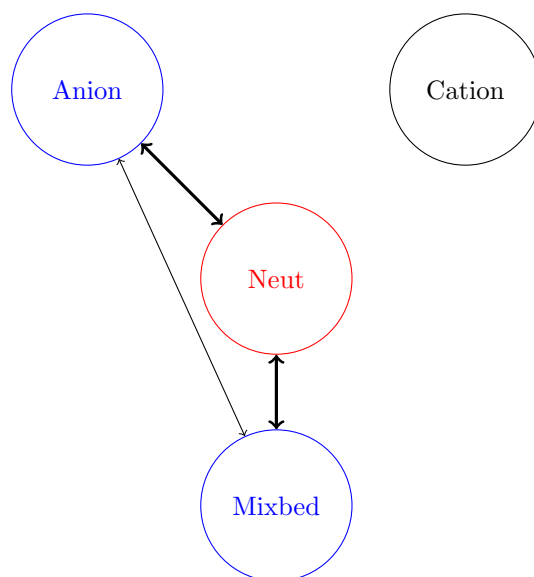


Figure 4.7: Base negotiation. Red indicates seller, blue buyer. The Neut sells the base, while the Anion and Mixbed agent try to obtain as much base as possible. The thick arrows show conflicting desires.

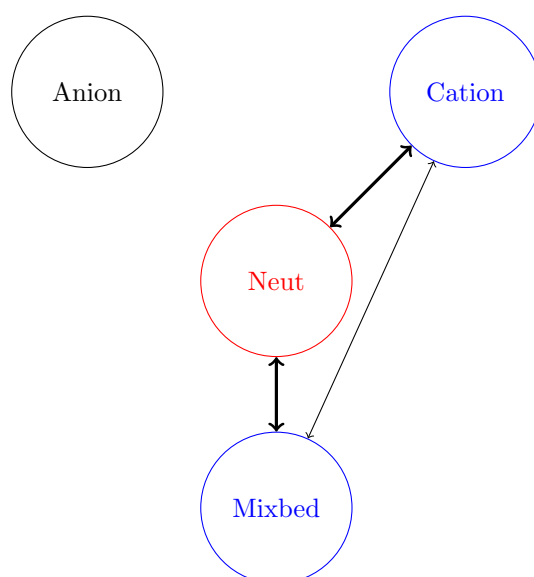


Figure 4.8: Acid negotiation. Red indicates seller, blue buyer. The Neut sells the acid, while the Cation and Mixbed agent try to obtain as much acid as possible.

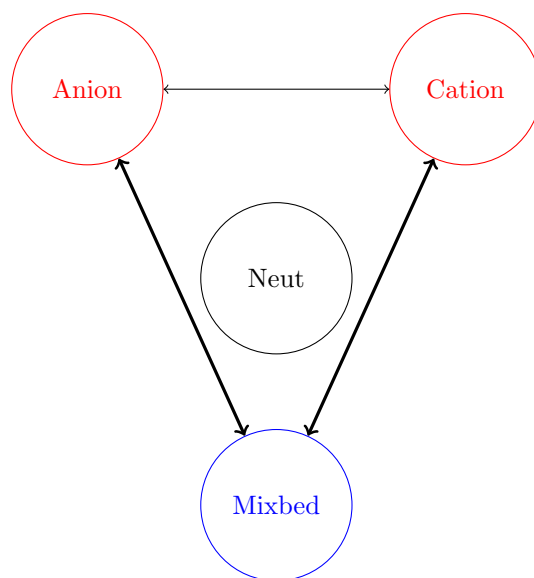
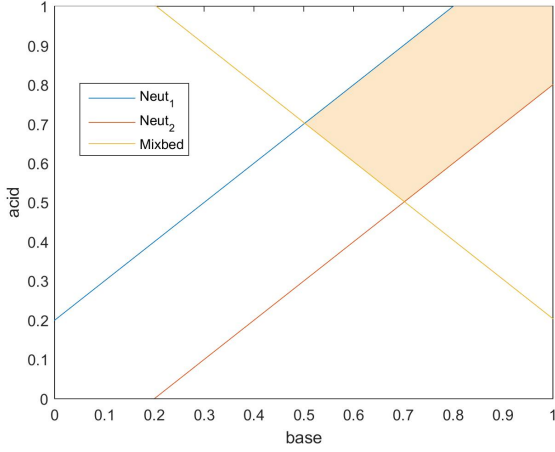


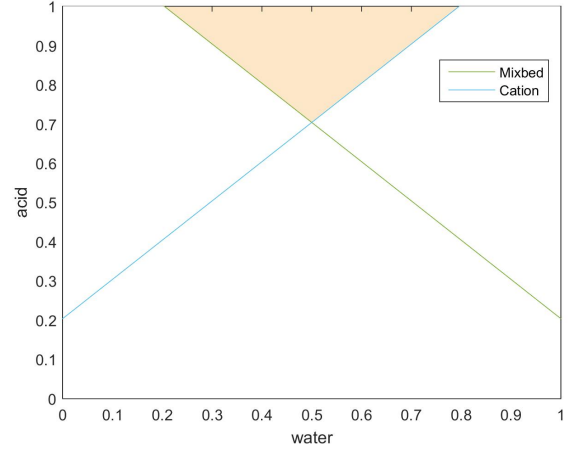
Figure 4.9: Water negotiation. Red indicates seller, blue buyer. The cation and anion sell water, while the Mixbed agent tries to obtain as much water as possible

#### 4.4.1 The Solution Space

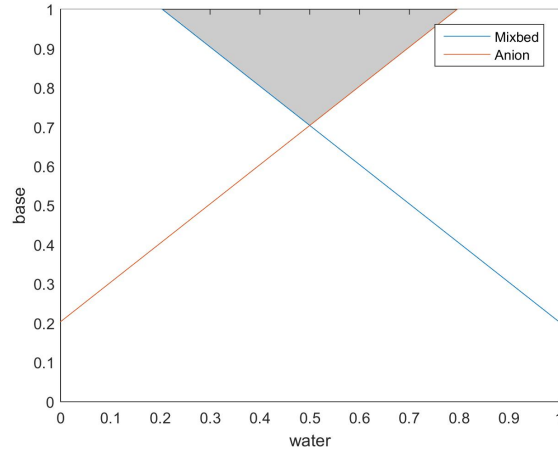
Below the solution space for the agents is shown, when  $ru_i = 0.3$  for all agents. These solution spaces are dependent on the reservation value, and within these solution spaces lies the agreement-zone.



(a) The solution space for the combination of acid and base. This is shown for the Neut and Mixbed.



(b) The solution space for the combination of acid and water



(c) The solution space for the combination of base and water, which are negotiated over by the Anion and the Mixbed agent.

Figure 4.10: The solution spaces for the different resources

## 4.5 Algorithm

The algorithm, which is programmed in Java, is implemented using different objects. This was originally described in Chapter 3 as the optimal way to implement agents. The model consists of 4 different agents, which in turn can either be an anion, cation, Mixed or neut, each 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 11 to if *true*.

Afterwards, depending on the concession protocol, each agents' desired utility decreases, meaning that the proposal creeps towards the solution space as can be seen in Figure 4.11.

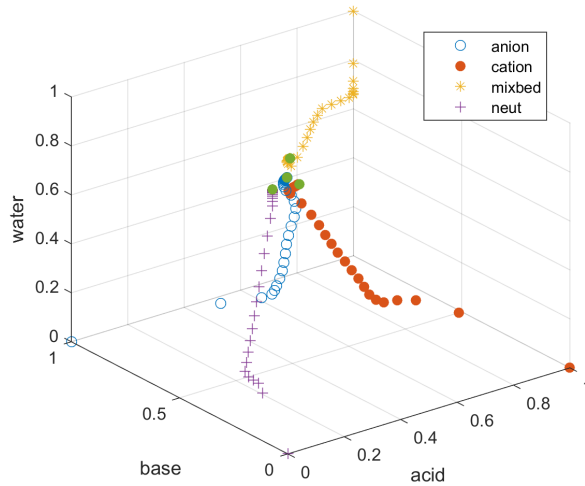


Figure 4.11: An example run of the agents searching for the solution. It is clearly seen that the agents start with their proposal in the corners, their optimal offers, and slowly make concessions towards each other.

**Data:** Each agent's utility function  $u_i(x)$ , reservation utility  $ru_i$ , and non-reactive concession strategy  $s_i(t) = 1, 2, \dots, T$

**Result:** Negotiation Agreement

```

1 initialization: Each agent proposes a preferred offer  $x_0^i$ ;
2 Each agent initializes these as previous best offer  $x_{[j,-1]}^k$ ;
3  $t \leftarrow 1$ ;
4 Set convergence tolerance:  $\delta$ ;
5 while  $t \leq T$  and  $IsConverge == False$  do
6   Determine the agent to propose:  $i = \text{mod}(t, 4)$ ;
7   foreach  $j \in 1, 2, 3, 4$  do
8     if  $j == i$  then
9        $\Delta u_{j0} = s_j^0(t) - s_j^0(t-1)$ ;
10      foreach  $k \in 1, 2, 3, 4$  do
11        if  $u_j(x_t^k) \geq ru_j$  then
12           $\Delta u_{jk} \leftarrow \Delta u_{j0}(t)$ ;
13        else
14           $\Delta_1 u_{jk}(t) \leftarrow u_j(x_t^k) - u_j(x_{[j,-1]}^k)$ ;
15           $\Delta_2 u_{jk}(t) \leftarrow u_j(x_t^k) - u_j(x_0^k) - (1 - u_j(x_{t-1}^j))$ ;
16           $\Delta u_{jk} \leftarrow \max\{\Delta_1 u_{jk}(t), \Delta_2 u_{jk}(t), 0\}$ ;
17        end
18       $\Delta u_j(t) \leftarrow \min\{\min_{k \in 1,2,3,4} \Delta u_{jk}(t), \Delta u_{j0}(t)\}$ ;
19    end
20    Agent  $i$  concedes by determining  $s_i(t) \leftarrow s_i(t-1) - \Delta u_j(t)$ ;
21    Agent  $i$  calculates:  $w_{t-1} \leftarrow \frac{1}{m} \sum_{j=1}^m x_{t-1}^j$ ;
22    Agent  $i$  proposes  $P_{A_t^i}[w_{t-1}]$ ;
23  else
24     $x_i^j \leftarrow x_{t-1}^j$ ;
25    if  $u_j(x_{t-1}^i) \geq u_j(x_{[j,-1]}^i)$  then
26       $x_{[j,-1]}^i(t) \leftarrow x_{t-1}^i(t-1)$ ;
27    end
28     $x_{[i,-1]}^j(t) \leftarrow x_{[i,-1]}^j(t-1)$ ;
29  end
30 end
31 if  $\max_{j \in 1,2,\dots,m} \|x_t^j - w_{t-1}\| < \delta$  then
32    $IsConverge \leftarrow True$ ;
33 else
34    $t \leftarrow t + 1$ ;
35 end
36 end

```

**Algorithm 1:** Basic algorithm structure modified from (Zheng et al., 2016). Applied to the four agents which are used in this use case.

## 4.6 Result Interpretation

The outcome of the negotiation has to be translated to a value. We know the maximum production. Thus, 0.5 water means for example that 0.5 of the maximum water will be produced by the anion, cation and Mixbed. When an outcome of 0.7 for the acid is negotiated, this means that 0.7 of the possible acid has to be divided over the cation and Mixbed. This amount is then divided equally over the cation and Mixbed. However, if the Mixbed's water  $l$  is set to 10, it also requires 10 times as little base than the anion filter i.e..

Thus, the allocation of the base to the anion and Mixbed is dependent on the water  $l$ . This is a fixed separation. Suppose that 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.

## 4.7 Proof of Convergence

If there is a non-empty intersection of the agreement zone, the agents will find this. This has already been proven in the Zeuthen strategy (Rosenschein and Zlotkin, 1994), which shows that no agreement is always worse than a bad agreement. In infinite time this will happen since each agent has an incentive to concede (Zheng et al., 2016).

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. (2016) 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 and Evaluations

In the previous chapter a have described of the model is given, 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 simulations done 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:  $\prod_{i=1}^m u_i(x)$ . This optimum can be calculated if all the utility functions of the agents are known. This information is unknown to the agents since they only know their own utility functions. 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, 1950; Roth, 1977; Lensberg, 1988).

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

Using the non-linear COUENNE program (Belotti et al., 2013) implemented in GAMS (Corporation, 2013), the solutions are calculated. The limit for the reservation curve is also found. This is dependent on the utility functions, but our default utility give a  $\forall i, ru_i = 0.3182$ . This means that there is no solution space, if  $\forall i, ru_i > 0.3182$ . However, if only a single agent were to have a  $ru_i > 0.3182$ , there still would be a solution. This obviously depends on the reservation curve values of the other agents.

### 5.1.2 Non-Reactive Concession Strategy

As explained in Section 3.4.3, the concession strategy determines whether a solution will be found. If no concession is made during the negotiation, and the agents 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 many methods, and a weak concession strategy 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 strategy is a linear function until the reservation value. Described by Wu et al. (2009) and in Section 3.4.3, it is an *amount of utility*, where Agent  $i \in N$  concedes a fixed amount utility  $au$ . Wu et al. (2009) found that it was well performing, and has the advantage of ending after a known number of rounds. 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  $\frac{100}{4_{\text{agents}}} = 25$  proposals. Since we combine it with the reactive concession strategy, it is possible for the agents to negotiate for more rounds. This means that a simulation with more than 100 rounds is preferred.

### 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. (2016), 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  $ru_i = \{0.05, 0.10, 0.15, 0.20, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65\}$ . In the initial case, there is no agreement zone if the  $ru_i$  is larger than 0.3182. But, since different kind of parameters are tried, 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 Algorithm 1, (page 54) to finish, there are two option. Either the distance between the current offer and weight is smaller than a threshold, or the maximum number of rounds is reached. This distance,  $\max_{j \in 1, 2, \dots, m} \|x_t^j - w_{t-1}\|$  gives the maximum distance from the agents' offer to the weight. The weight, as shown in Section 4.3.2, is the average of all proposals. 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 maximum 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 are too high, meaning that there is no agreement-zone, and thus no agreement possible. For a definition of the agreement zone see page 26.

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 and 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 a combination of different concession strategies are compared, it is useful to check whether a solution is found afterwards. This happens since something might change 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.

So if the number of rounds is equal to 199, no agreement has been made and

the average of all proposals will be used to determine the distance of the Nash optimum.

## 5.2 Reactive Compared to Non-Reactive Concession Strategy

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  $ru_i = 0.3182$ . Since the system does not find a solution when the reservation value is 0.35 or higher. However, unexpectedly, the non-reactive strategy consequently finds the solution closer to the optimal Nash Bargaining Solution than the reactive strategy. For completeness, the Nash solution lies at acid = 0.571, base = 0.571, water = 0.714.

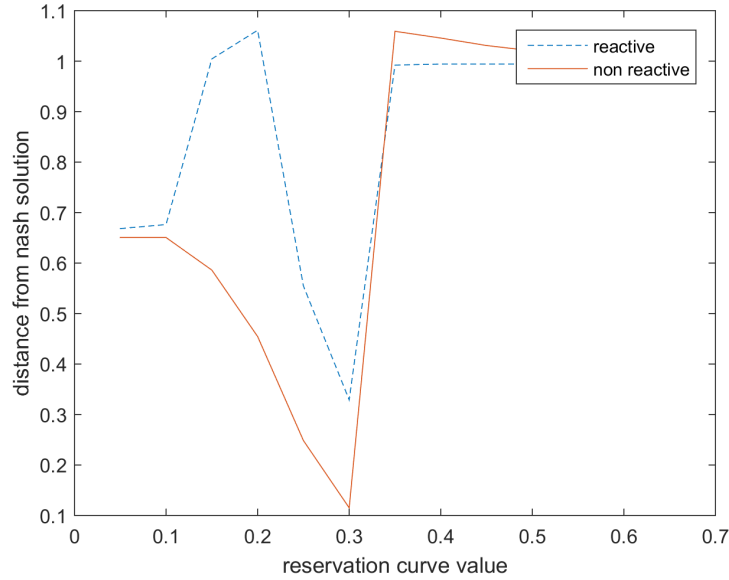


Figure 5.1: Distance from the Nash bargaining solution for the reactive and non-reactive concession strategies. Although a solution is shown in the figure, it does not mean that the agents have found an agreement since the weight is used to calculate the end solution.

In Table 5.1 it can be seen that the non-reactive concession strategy halts after fewer rounds than the reactive concession strategy, which means that it finds a solution faster. The reactive concession strategy has a larger distance than the threshold, which means that there is no agreement, since the distance is larger than 0.05, and the number of rounds is 199. 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

reservation utility	Reactive concession		Non-reactive concession	
	distance	# of rounds	distance	# of rounds
0.05	0.000 0	26	0.029 6	23
0.10	0.000 0	26	0.029 6	23
0.15	0.000 0	38	0.029 6	23
0.20	0.000 0	50	0.029 6	23
0.25	1.069 3	199	0.029 6	23
0.30	1.175 1	199	0.048 5	66
0.35	1.171 6	199	0.127 7	199
0.40	1.167 5	199	0.279 9	199
0.45	1.167 5	199	0.425 4	199
0.50	1.167 5	199	0.555 5	199
0.55	1.167 5	199	0.673 3	199
0.60	1.167 5	199	0.780 7	199
0.65	1.167 5	199	0.874 5	199

Table 5.1: The distance in the final proposal and number of rounds of a simulation. As can be seen, the agents do not find an agreement when the reservation utility is 0.25 or larger when using the reactive concession strategy, while the non-reactive concession correctly find an agreement.

distance from the optimal solution in Figure 5.1, it is larger than the non-reactive strategy.

### 5.3 Reactive Mixbed with the other Agents 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, the most important resource of the production line.

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

In the table it is shown that when only the Mixbed makes reactive concessions, the negotiation performs better than when all the agent use the reactive concession protocol. It performs very similarly to the situation where all the agents make non-reactive concessions. However, as shown in Figure 5.2 the overall solution found by the system when all agents make non-reactive concessions, continuous to find a solution closer to the overall optimum.

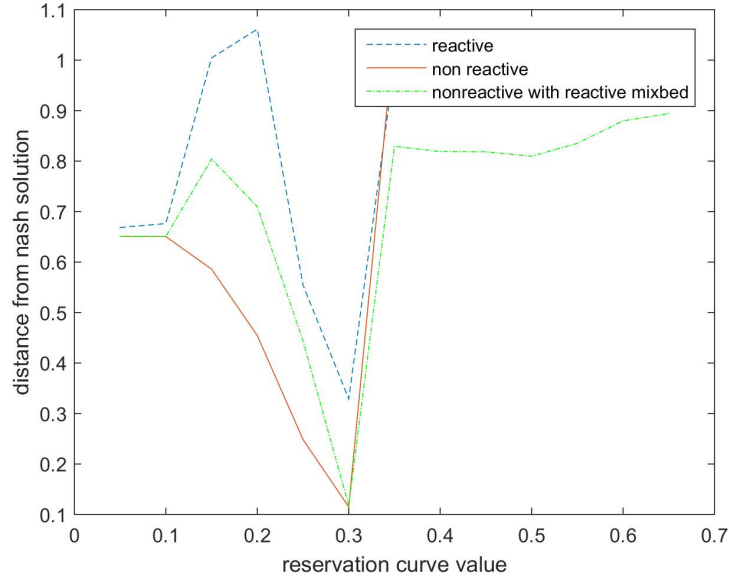


Figure 5.2: Comparison of the reactive and non-reactive concession strategy compared to the situation where only the Mixed makes reactive concessions.

reservation utility	distance	# of rounds
0.05	0.029 6	23
0.10	0.029 6	23
0.15	0.000 0	27
0.20	0.048 2	38
0.25	0.044 3	46
0.30	0.034 9	54
0.35	0.563 1	199
0.40	0.685 6	199
0.45	0.789 5	199
0.50	0.926 3	199
0.55	1.029 3	199
0.60	1.113 2	199
0.65	1.130 5	199

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

## 5.4 Changing the water ratio for the Mixbed

In the design (Section 4.3.6) 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. This means that  $l$  is respectively 2 and 10, and thus the ratio is 2 or 10 water for each acid and base. 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.

### 5.4.1 Mixbed Ratio 2:1:1

When looking at the first ration of 2:1:1 for water:base:acid, similar result as that of the original ratio are obtained. The minimum  $ru_i$  lies lower however, and has a 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.

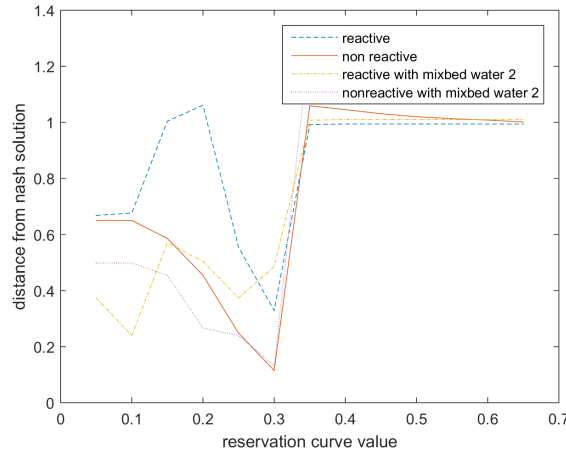


Figure 5.3: Comparison of the reactive and non-reactive strategy for a 2:1:1 ratio for the Mixbed. This compared to the original ratio of 1:1:1

reservation utility	Reactive concession		Non-reactive concession	
	distance	# of rounds	distance	# of rounds
0.05	0.000 0	26	0.000 0	26
0.10	0.000 0	30	0.000 0	26
0.15	0.000 0	43	0.000 0	26
0.20	0.950 7	199	0.000 0	26
0.25	1.113 3	199	0.000 0	26
0.30	1.241 5	199	0.047 2	71
0.35	1.240 0	199	0.283 0	199
0.40	1.236 0	199	0.452 2	199
0.45	1.236 0	199	0.677 4	199
0.50	1.236 0	199	0.779 7	199
0.55	1.236 0	199	0.862 5	199
0.60	1.236 0	199	0.926 8	199
0.65	1.236 0	199	0.985 9	199

Table 5.3: Here Mixbed ratio is water 2:1:1. The minimum reservation utility is 0.301, meaning that the agreement-zone is non-empty for any value above. Using the non-reactive concession strategy, the agents find this solution, while with the reactive method not even a solution is found when the reservation utility is 0.20.

#### 5.4.2 Mixbed Ratio 10:1:1

When using a ratio of 10:1:1, the maximal reservation curve is  $ru_i = 0.274$ . The Nash solution lies at acid = 0.647, base = 0.647, and 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 algorithm still finds a solution very close to the Nash optimum when the reservation utility is 0.3.

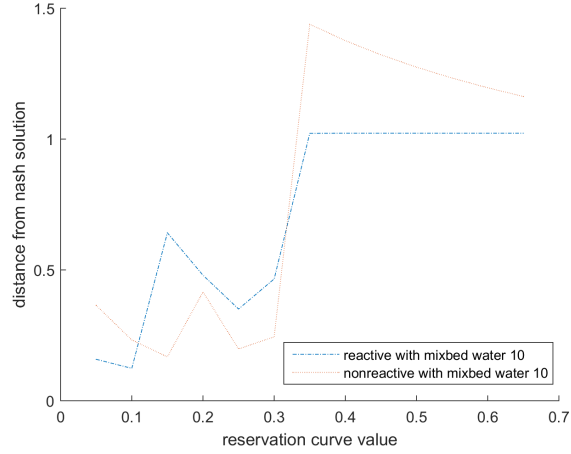


Figure 5.4: Comparison of the reactive and non-reactive strategy for a 10:1:1 ratio for the Mixbed

reservation utility	Reactive concession		Non-reactive concession	
	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: The distance in the final proposal and number of rounds of a simulation. Here the Mixbed ratio is 10:1:1.

## 5.5 Evaluation

Unexpectedly the reactive concession does not find a solution when the agreement-zone is non-empty. Possible explanations are given here.

### 5.5.1 Reactive compared to the non-reactive concession strategy

It is interesting that the reactive concession strategy does not find a solution although there is an agreement zone. A possible solution is that the answer cannot be found as is stated by Zheng et al. (2016) 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  $u_j(x_{s_i(t)}^*) < ru_j$ ”.*

As discussed, it can be seen that the reactive concession strategy takes more rounds to find a solution. This can be simply attributed to fact that the concession made each round is the  $\min\{\min_{j \in m} \Delta u_{ij} \Delta u_i\}$ , meaning that it is the minimum of the reactive and non-reactive concession. Thus, the concession will always smaller or equal to the non-reactive concession, and thus it will take longer to find a solution.

The stalling of the negotiation in the reactive case can be attributed to the perceived concession of an agent. A solution will be given in the discussion.

#### Situation when the Reservation Utility is 0.2

As shown in the results, there is an unexpected distance from the Nash solution in the situation of the 0.2 reservation curve for the reactive concession strategy. This can mostly be attributed to the stalling of the agents while trying to find a solution.

Looking at the proposals, Figure 5.5, 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 the Neut and Mixbed finally start conceding, but not before the anion and cation have started to move towards the average of the Neut en Mixbed.

The stalling of the Neut and Mixbed can also clearly be seen when looking at the desired utility of the agents. No concessions are made by the Neut and Mixbed until the end.



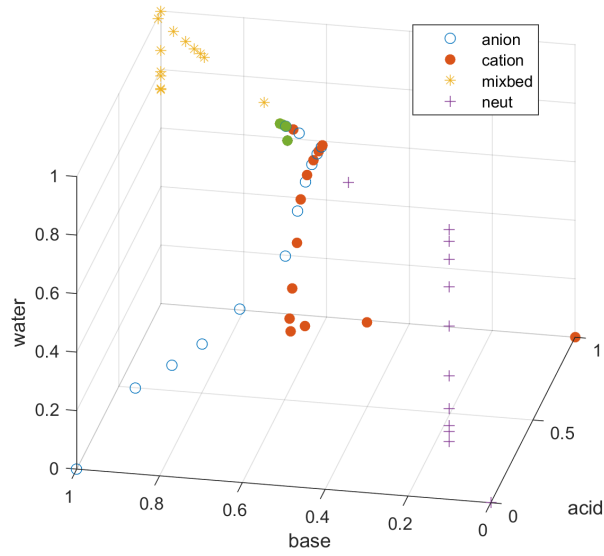


Figure 5.5: The proposals made by the 4 agents. We see that the Mixbed and neut “stall” in de finding of the solution. Only towards the end do they start conceding.

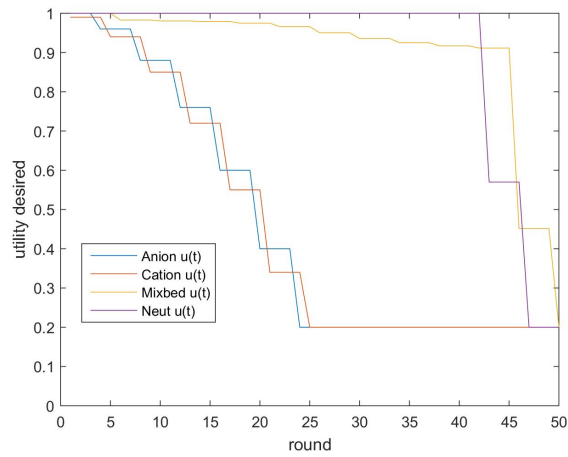


Figure 5.6: The desired utility for the agents. The stall can be clearly seen for the Mixbed and neut agent. Only towards the end do they start conceding.

### 5.5.2 Reactive Mixbed compared to non-reactive other agents

When looking at the solution which is found when only the Mixbed makes reactive concessions, it can be seen that this is a lot more optimal than when all the agents use the reactive concession strategy.

When this use case is to be implemented in the real situation, using the reactive Mixbed is a realistic option. This due to the fact that the Mixbed is the systems most important agent since it is responsible for the desired output.

### 5.5.3 Different Mixbed ratios

When looking at the different values of  $l$ , the exact same pattern is seen as in the original reactive versus the non-reactive. So although the Nash solution moves, a sub-optimal solution is still obtained, where the non-reactive method outperforms the reactive method on distance to Nash, average and rounds necessary.

This means that the usage of ratios for the Mixbed are a very realistic solution to solve the problem of desire, when a sudden increase of output is necessary. This also solves the realistic problem of the Mixbed requiring a lot less cleaning than the anion and cation.

## Chapter 6

# Conclusion and Further Research

To conclude an answer to the initial research question is given, and the results are discussed. An improved reactive concession strategy is given, and to finalize further research options are shown.

### 6.1 Conclusion

In this thesis an overview of agent solutions used in the manufacturing world is given. It is found that a gap lies in the “real” negotiation, which excludes the use of auctions and the 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. (2016) still has some difficulties. This can be clearly seen in Figure 5.1.

So although the reactive concession strategy ensures that the agents only concede when the other agents concede, it under-performs. However, when looking at the individual utility of an agent, it might be the best protocol to implement since it ensures that an agent only concedes if the other agents do as well.

The usage of private utility functions allows for competing companies to use automated negotiation. This is in line with the idea of Industry 4.0, where companies specialize, but require products from other producers. By implementing automated negotiation, without the requirement of releasing the utility functions, competing businesses can flourish together. This conform to the principled negotiation ideology, which allows an optimal outcome.

So, how can energy and manufacturing companies use the AI concept of intelligent multi-agent systems (MAS) for the optimization of production process?

Most importantly is that the production is shifting towards a decentralized control system. As discussed, the complexity of the system increases enormously. This means that the different nodes of the system have to communicate, while the nodes might have conflicting ideas and (sub-) solutions. Using negotiation these conflicts can be solved.

The optimal framework, using the sequential projection method with a monotonic concession protocol, allows for a system to find a solution, as shown with the use case.

From here on more domains can use negotiation, not only in the manufacturing. An example is the interaction between pedestrians, cars, public transport at an intersection when most of these get automated. Who gets the right of way when most central control structures, like the traffic lights, might disappear? Future asks for automatic negotiation \*.

## 6.2 Discussion

Although the alternating offer protocol has been used, it is not usable at all yet in 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.

In the literature the concept of principled negotiation is discussed. Addressed is the importance of retaining a private utility function, but making it very important to create these utility functions to ensure objective negotiation. This we have achieved, and thus we can say that the negotiation in this thesis is a form of principled negotiation.

In the problem statement, the option of optimizing a production process using negotiation is discussed. It can be concluded that this has not fully been achieved. However, when looked at the implementation of multilateral multi-issue negotiation, without a mediator, and using a very simplified theoretical solution, this is achieved.

Originally the preferred reference would have been to use the real data to compare this to the use case's current performance. This was not conceivable unfortunately due to the complexity of the system. Instead of using the Key Performance Indicator (KPI) of the business, an theoretical situation was checked.

The use case was not an entirely optimal situation for the research. It had all the requirements for a multilateral multi-issue negotiation, but the fact that the agents had no purpose to keep their utility private was not a requirement. This addition would have had more impact when looking at processes that require

---

\*For further research, the model is available at <https://github.com/diederikvkrieken/graduate-usecase>

privacy.

### 6.2.1 Discussion of the Reactive Concession Strategy

A difficulty is that no concession is seen if an agent concedes in a matter that the other agents have no desire in. So if, for example, the Anion concedes in the amount of water to produce (which means that it will produce more water), this concession will not be “seen” by the Neut, since it has no desire in this utility.

Even worse is the opposite, an agent can perceive that an agent wants an higher utility than initially.

Take for example the Anion and Neut from the use case. When the Neut looks at the concessions that the Anion does, he perceives an “anti-concession”, since the Anion increases the Acid in the offer.

In the beginning the Anion proposes 0 acid and 1 base. In the first round the Anion already proposes 0.5 acid. Since it takes the average of all acid proposals (both the Mixbed and Cation start with 1 acid), the weight of the Acid proposal is 0.5. This is the value that the Anion will propose, and the Neut perceives a decrease in the offer made by the Anion. This while the Anion actually does concede by increasing the water offer and decreasing the base.

A possible solution is as follows: Since an agent’s offer is equal to the weight of the proposals for issues it is indifferent about, the other agents can check. The agent knows the weight of the proposals and the offer of the agent. If the offer is equal to the weight for an issue, the proposal will be equal to the offer. This means that the agent that has proposed is indifferent for an issue.

There are two situations where the proposal is equal to the weight for an issue. Either the agent is indifferent for that issue, and does not project it onto its indifference curve, or the weight is above the agents’ indifference curve. Meaning that an agreement to the proposal is made.

If an agent is indifferent to the issue, the other agent must ignore the proposal the agent has done on the issue when looking at the concession. This can be done by setting the proposed value to the agents proposal at initiation.

From this it is possible to update the reactive concession strategy as shown in Algorithm 2. For clarification, we notate an issue  $l$  within  $x$  as  $a_l \in x$  instead of the notation  $x_j \in x$  as described in Section 4.2. An attempt to keep the algorithm as similar as possible to the original is attempted, since converge was already proven.

The weight is commonly known, since it is the average of all proposals, and thus the average value of each issue is also known. This is notated as  $w_l \in w_t$ .

```

1  $\Delta u_{j0} = s_j^0(t) - s_j^0(t-1);$ 
2 foreach  $k \in 1, 2, 3, 4$  do
3   if  $u_j(x_t^k) \geq ru_j$  then
4      $\Delta u_{jk} \leftarrow \Delta u_{j0}(t);$ 
5   else
6     foreach  $a_l \in x_t^k$  do
7       if  $a_l == w_l$  then
8         Issue is indifferent for agent  $k$ ;
9         Set to agent's initial value;
10         $z_l^k = x_0^k;$ 
11      else
12        Issue is important for agent  $k$ ;
13         $z_l^k = x_t^k;$ 
14      end
15    end
16     $\Delta_1 u_{jk}(t) \leftarrow u_j(z^k) - u_j(x_{[j,-1]}^k);$ 
17     $\Delta_2 u_{jk}(t) \leftarrow u_j(z^k) - u_j(x_0^k) - (1 - u_j(x_{t-1}^j));$ 
18     $\Delta u_{jk} \leftarrow \max\{\Delta_1 u_{jk}(t), \Delta_2 u_{jk}(t), 0\};$ 
19  end
20   $\Delta u_j(t) \leftarrow \min\{\min_{k \in 1,2,3,4} \Delta u_{jk}(t), \Delta u_{j0}(t)\};$ 
21 end
22 Agent  $i$  concedes by determining  $s_i(t) \leftarrow s_i(t-1) - \Delta u_j(t);$ 
Algorithm 2: Updated reactive concession strategy of Algorithm 1 on
page 54. By checking whether the issue is indifferent for the agent, the ap-
propriate concession is seen.

```

## 6.3 Further Research

As discussed, more research has to be conducted before negotiation can commonly be used in production and manufacturing. Some further research subjects are discussed here. Especially a lot of further research could be done on the alternating offer protocol to be able to use this method in manufacturing by looking at more concession strategies for example. Another option is that the agents could be improved to allow reasoning, using a holonic structure for example. Furthermore, other strategies can be used, while the utility functions can be changed as well. Also possible is the implementation of coalitions. Extra negotiation could be applied using bilateral negotiations, and to finalize heuristic learning methods can be applied.

### 6.3.1 Extra concession strategies

The many concession strategies, as shown in Section 3.4.3 allow for other strategies to be used. Further research could include using the utilities as described in that section, and compared to each other. Especially the *fraction of utility* method should be considered, which was suggested as another good solution by Wu et al. (2009).

### 6.3.2 Holonic Agents

The structure of a holonic agent is that of a holon as can be seen in Figure 6.1. As shown in the literature (Chapter 3), it is based on PROSA by (Van Brussel et al., 1998). Ideally this could be implemented in this use case to make use of the different parts of the anion filter.

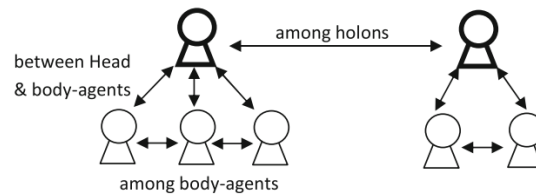


Figure 6.1: An example of the different negotiation between holons from Beheshti et al. (2016).

The sub-agents of the Anion would consist of the different filters, as where originally shown in Section 4.1. A consequence of the simplification made in this research, is that an assumption is made that the Anion can clean (receive

base) and produce (make water) at the same time. This however is not always possible, since this depends on the current state of the filter.

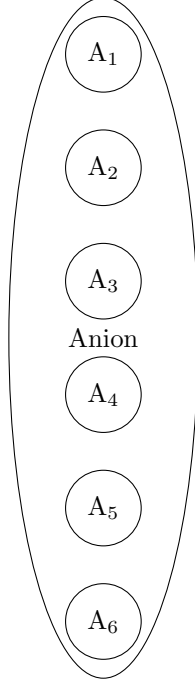


Figure 6.2: Anion head and sub-agents

The following facts and rules would then be part of the Anion, meaning that a beginning towards a representation of Beliefs, desires, and intentions, (BDI) in the agent can be made (Rao and Georgeff, 1995).

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

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



### 6.3.3 Collaboration

Not used and discussed, but an interesting research is the collaboration between agents which share a common interest. For example the Anion and Cation could collaborate against the Mixbed, since they both have no interest in producing water. By combining collaboration with the learning methods it could be possible to learn who shares the same desires.

### 6.3.4 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. Although an attempt was made, using a variable in the Mixbed ratio, this of course still is a highly theoretical and unrealistic representation.

#### Reservation curve

A linear reservation curve was used for simplicity, since this eliminated the minimization (mixed integer programming) that would have been necessary if a truly curved function was used. An example is shown in Section 6.3.4. The downside of this reservation curve is that the projection method would be a minimization problem, which is computationally heavy.

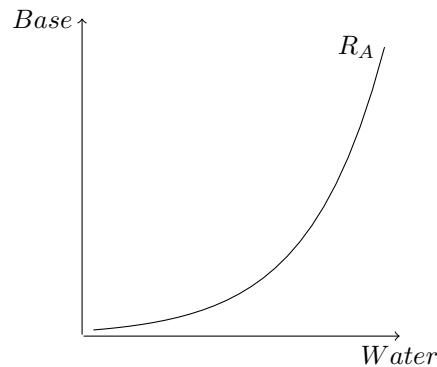


Figure 6.3: A more realistic reservation curve for the Anion filter: if more water is filtered and given, the more base it requires.

#### Learning methods

As shown in Section 3.4.7, learning methods can also be used to learn the desired utility function. This has not been done since the requirement of a convex utility function forced the usage of predefined functions.

It would be very interesting to see how learning could be implemented in combination with the reactive concession strategy. Would there still be an incentive to concede if the agents learn that if other agents concede they can obtain their optimum? Or do the agents learn that without concession, no solution is found?

### **6.3.5 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 agents. Take the Mixbed and Anion for example: if the group agrees to an allocation of 0.5 base, the Mixbed and anion can further negotiate how much of this 0.5 should be allocated to whom. This new bilateral negotiation is simpler, and due to the fact that it only contains a single issue, allows for quick determination.

# 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.
- Chimay Anumba, Zhaomin Ren, Antony Thorpe, Oneugbu Okoronkwo Ugwu, and Leonard Newnham. Negotiation within a multi-agent system for the collaborative design of light industrial buildings. *Advances in Engineering Software*, 34(7):389–401, 2003.
- Tim Baarslag, Katsuhide Fujita, Enrico H Gerding, Koen Hindriks, Takayuki Ito, Nicholas Jennings, Catholijn Jonker, Sarit Kraus, Raz Lin, Valentin Robu, et al. Evaluating practical negotiating agents: Results and analysis of the 2011 international competition. *Artificial Intelligence*, 198:73–103, 2013.
- 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.
- Fabio Luigi Bellifemine, Giovanni Caire, and Dominic Greenwood. *Developing multi-agent systems with JADE*, volume 7. John Wiley & Sons, 2007.
- Pietro Belotti, Christian Kirches, Sven Leyffer, Jeff Linderoth, James Luedtke, and Ashutosh Mahajan. Mixed-integer nonlinear optimization. *Acta Numerica*, 22:1–131, 2013.
- 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](http://apsom.org/docs/t061_isa95-04.pdf).
- Manfredi Bruccoleri, G Lo Nigro, Giovanni Perrone, Paolo Renna, and S Noto La Diega. Production planning in reconfigurable enterprises and reconfigurable production systems. *CIRP Annals-Manufacturing Technology*, 54(1): 433–436, 2005.

GAMS Development Corporation. General Algebraic Modeling System (GAMS) Release 24.2.1. Washington, DC, USA, 2013.

Mauro da Rosa, Vladimiro Miranda, Manuel Matos, Gerald Sheble, and Armando da 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.

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.

Prasanta Kumar Dey. Decision support system for inspection and maintenance: A case study of oil pipelines. *Engineering Management, IEEE Transactions on*, 51(1):47–56, 2004.

Dario Di Nocera. *Multi-agent Automated Negotiation for Management of Services in Smart Cities*. PhD thesis, Università degli Studi di Napoli Federico II, 2015.

Neil Duffie and Vittaldas 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 Harjunoski. Optimal operation: Scheduling, advanced control and their integration. *Computers & Chemical Engineering*, 47:121–133, 2012.

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

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.

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.
- Juraj Hromkovič. *Algorithmics for hard problems: introduction to combinatorial optimization, randomization, approximation, and heuristics*. Springer Science & Business Media, 2013.
- Nicholas Jennings, Peyman Faratin, Alessio Lomuscio, Simon Parsons, Michael Wooldridge, and Carles Sierra. Automated negotiation: Prospects, methods and challenges. *Group Decision and Negotiation*, 10(2):199–215, 2001.
- Kurt Konolige and Nils 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.
- Fabian Lang and Andreas Fink. Collaborative machine scheduling: Challenges of individually optimizing behavior. *Concurrency and Computation: Practice and Experience*, 27(11):2869–2888, 2015.
- Jae-Hoon Lee, Myung-Soo Lee, Sang-Hoon Lee, Se-Ghok Oh, Bo-Hyun Kim, Sung-Ho Nam, and Joong-Soon Jang. Development of computerized facility maintenance management system based on reliability centered maintenance and automated data gathering. *International Journal of Control and Automation, SERSC*, 6(1):1–12, 2013.
- 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 and Stamatis Karnouskos. *Industrial Agents: Emerging Applications of Software Agents in Industry*. Morgan Kaufmann, 2015.
- 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 Leitão, Stamatis Karnouskos, Luis Ribeiro, Jay Lee, Thomas Strasser, and Armando 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.

- Minyi Li, Quoc Bao Vo, Ryszard Kowalczyk, Sascha Ossowski, and Gregory Kersten. Automated negotiation in open and distributed environments. *Expert Systems with Applications*, 40(15):6195–6212, 2013.
- Sapna Mahar and Pradeep Bhatia. Agent oriented software development. *International Journal of Software Engineering Research and Practices*, 1(4):17–23, 2012.
- Ivan Marsa-Maestre, Mark Klein, Catholijn Jonker, and Reyhan Aydoğan. From problems to protocols: Towards a negotiation handbook. *Decision Support Systems*, 60:39–54, 2014.
- Laslo Monostori, Botond Kadar, Thomas Bauernhansl, Shinsuke Kondoh, Soundar Kumar, Gunther Reinhart, Olaf Sauer, Gunther Schuh, Wilfried Sihn, and Kenichi 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 Nash. The bargaining problem. *Econometrica: Journal of the Econometric Society*, pages 155–162, 1950.
- Daniel Palmer, Marc Kirschenbaum, Jon Murton, Kelly Zajac, Mike 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*. Springer, New York, NY, 2005.
- Manisa Pipattanasomporn, Hassan Feroze, and Saifur 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.
- Guy Pujolle. An autonomic-oriented architecture for the internet of things. In *Modern Computing, 2006. JVA'06. IEEE John Vincent Atanasoff 2006 International Symposium on*, pages 163–168. IEEE, 2006.
- Ricardo Rabelo, Luis 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 Jennings, Peter Mcburney, Simon Parsons, and Liz Sonenberg. Argumentation-based negotiation. *The Knowledge Engineering Review*, 18(04):343–375, 2003.
- Anand Rao and Michael Georgeff. BDI agents: From theory to practice. In *ICMAS*, volume 95, pages 312–319, 1995.
- Jeffrey Rosenschein and Gilad Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*. MIT press, 1994.
- Alvin 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 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/>.
- Reid Smith. Communication and control in problem solver. *IEEE Transactions on Computers*, 29(12):1104–1113, 1980.
- Amy Trappey, Charles 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 Wyls, 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 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 Tichý, Vladimír Mařík, Kenwood Hall, Raymond Staron, Francisco 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 Wangermann and Robert 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 Weerd, 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, Tinglong Dai, Katia Sycara, and Nilanjan Chakraborty. Automated multilateral negotiation on multiple issues with private information. *INFORMS Journal on Computing*, 28(4):612–628, 2016.
- 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.
- Rong Zhou, Bud Fox, Heow Pueh Lee, and Andrew YC Nee. Bus maintenance scheduling using multi-agent systems. *Engineering Applications of Artificial Intelligence*, 17(6):623–630, 2004.