Multi-objective Optimisation using Agent-based Modelling

Chris Franklin

Thesis presented in fulfillment of the requirements for the degree of Master of Science of Industrial Engineering at Stellenbosch University

Study leader: J Bekker

December 2012

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Abstract

It is very seldom that a decision-making problem concerns only a single value or objective. The process of simultaneously optimising two or more conflicting objectives is known as multi-objective optimisation (MOO). A number of metaheuristics have been successfully adapted for MOO. The aim of this study was to investigate the feasibility of applying an agent-based modelling approach to MOO.

The (s, S) inventory problem was chosen as the application field for this approach and *Anylogic* used as model platform. Agents in the model were responsible for inventory and sales management, and had to negotiate with each other in order to find optimal reorder strategies. The introduction of concepts such as agent satisfaction indexes, aggression factors, and recollection ability guided the negotiation process between the agents.

The results revealed that the agents had the ability to find good strategies. The Pareto front generated from their proposed strategies was a good approximation to the known front. The approach was also successfully applied to a recognised MOO test problem proving that it has the potential to solve a variety of MOO problems.

Future research could focus on further developing this approach for more practical applications such as complex supply chain systems, financial models, risk analysis and economics.

Opsomming

Daar is weinig besluitnemingsprobleme waar slegs 'n enkele waarde of doelwit ter sprake is. Die proses waar twee of meer doelwitte, wat in konflik staan met mekaar, gelyktydig optimiseer word, staan bekend as multi-doelwit optimisering (MOO). 'n Aantal metaheuristieke is al suksesvol aangepas vir MOO. Die doelwit van hierdie studie was om ondersoek in te stel na die lewensvatbaarheid van die toepassing van 'n agent gebasseerde modelerings benadering tot MOO.

As toepassingsveld vir hierdie benadering was die (s, S) voorraad probleem gekies en *Anylogic* was gebruik as model platform. In die model was agente verantwoordelik vir voorraad- en verkope bestuur. Hulle moes onderling met mekaar onderhandel om die optimale bestelling strategieë te verkry. Konsepte soos agentbevrediging, aggressie faktore en herinneringsvermoëns is ingestel om die onderhandeling tussen die agente te bewerkstellig.

Die resultate het gewys dat die agente oor die vermoë beskik om met goeie strategieë vorendag te kom. Die Pareto fronte wat gegenereer is deur hul voorgestelde strategieë was 'n goeie benadering tot die bekende front. Die benadering was ook suksesvol toegepas op 'n erkende MOO toets-probleem wat bewys het dat dit oor die potensiaal beskik om 'n verskeidenheid van MOO probleme op te los.

Toekomstige navorsing kan daarop fokus om hierdie benadering verder te ontwikkel vir meer praktiese toepassings soos komplekse voorsieningskettingstelsels, finansiële modelle, risiko-analises en ekonomie.

Acknowledgements

First and foremost, I want to honour God for giving me the opportunity to pursue my post graduate studies.

I would like to express my sincere gratitute to my supervisor, Mr James Bekker, for his guidance, support and patience throughout my thesis.

My post graduate studies have offered me the opportunity to get to meet and befriend many special people. In particular I would like to thank the Van der Merwe family for providing me with a home away from home whenever I spent time in Stellenbosch.

I am grateful to colleagues and friends who often acted as soundboards with whom I could test my ideas. Izak Janse van Rensburg deserves special mention for his advice during the initial phase of my thesis.

My parents' continuous encouragement inspired me throughout my academic career. I appreciate their motivation knowing that without it I would not have been able to complete my thesis. I would also like to thank my mother devoting many hours to proofread my thesis.

My wife is a pillar of strength in my life, and I truly value her love and support. It is to her that I dedicate this thesis.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

ABM	Agent-based modelling.
EOQ	Economic order quantity.
MOEA	Multi-objective evolutionary algorithm.
MOO	Multi-objective optimisation.
NSGA-II	Nondominated Sorting Genetic Algorithms II.
PAES	Pareto archived evolution strategy.

Greek Symbols

β	Interarrival time.
ε	Agent recollection ability.
λ	Mean arrival rate.
ς	Range of change.
τ	Agent aggression factor.

Other Symbols

$\mathcal{P}^*$	Pareto approximation set.
$\mathcal{P}^*_T$	Pareto front.

Roman Symbols

bm	Benchmark strategy.
C	Total inventory cost.
K	Order cost.
CV	Pareto front convergence indicator.
D	Demand per year.

Nomenclature

d_i	Euclidian distance.
ΔI_H	Hyper-area difference.
GD	Generation distance indicator.
i	Customer demand.
I_H	Hyper-volume quality indicator.
h	Inventory holding cost.
I_t	Inventory level at time t.
ME	Maximum Pareto front error indicator.
n_{it}	Number of negotiation iterations.
q	Reorder quantity for the (r, q) Continuous review policy.
r	Reorder point for the (r, q) Continuous review policy.
s	Reorder point for the (s, S) Continuous review policy.
S	Required inventory level after reordering for the (s, S) Continuous review policy.
SI_{IM}	Satisfaction index of the inventory manager.
SI_{SM}	Satisfaction index of the sales manager.
SL	Service level.
SP	Pareto front spacing indicator.
ST_i	Stockout experienced by customer i.

Chapter 1

Introduction

A short background on the project is provided in this chapter. The objectives of the project and an overview of the layout of this document are also given.

1.1 Project Background

Agent-based modelling (ABM) has become very popular for modelling and understanding complex systems. The characteristics of agents make this a useful tool to model the complex social interactions found in humans. ABM has been applied increasingly often in the field of social sciences, where the agents represent people and agent relationships represent the processes of social interaction. The interactions relating to negotiation between agents are of specific importance to this project because of its potential to model the human decision-making process.

Everyday decision-making often comprises of conflicting objectives that need to be optimised. Humans have the ability to weigh up different alternatives and perform a simple trade-off analysis whenever they encounter these problems. During this process it is almost as if two or more alter-egos negotiate with each other to come up with a solution that is satisfactory to all of them. This process of simultaneously optimising two or more conflicting objectives is known as multi-objective optimisation (MOO).

There are numerous different metaheuristics available that can be used in multi-objective optimisation. The purpose of this study is to determine if agent-based modelling can be used as a metaheuristic for multi-objective optimisation. Multiple agents are created, each representing one of the conflicting objective functions. The agents need to negotiate with each other in order to generate solutions that are attractive to all of them. A satisfaction index, which drives their negotiation, is defined for each of the agents.

Inventory management often provides an appropriate context for multi-objective optimisation. In the theoretical (s, S) inventory problem the vendor is confronted with two conflicting objectives. He needs to keep his inventory costs as low as possible, but keep enough inventory in stock to ensure that his service level is adequate. This inventory problem has therefore been selected as the context in which the agent-based approach to multi-objective optimisation is applied in this study.

An agent-based model of the inventory problem was developed in Anylogic. In addition to the basic inventory problem functionality, the model contains two agents – a sales manager and inventory manager – responsible for the two objective functions. A simulation model was developed to determine if the agents are capable of finding good solutions. The success of the approach is determined by making use of a set of performance metrics. Possible application areas for the research are highlighted and the potential for further research identified.

1.2 Project Objectives

The following project objectives have been identified for this research:

- The primary aim of the project is to investigate if it is feasible to use agent-based modelling as a metaheuristic for multi-objective optimisation.
- Establishing a knowledge of agent-based modelling at the Department of Industrial Engineering, Stellenbosch University.

- Providing input into the study leader's research in multi-objective optimisation.

1.3 Overview of the Document Structure

The diagram in Figure 1.1 serves as a *road map* to the study, explaining how the document is structured. It will appear at the beginning of every chapter to help the reader find his way through the document.

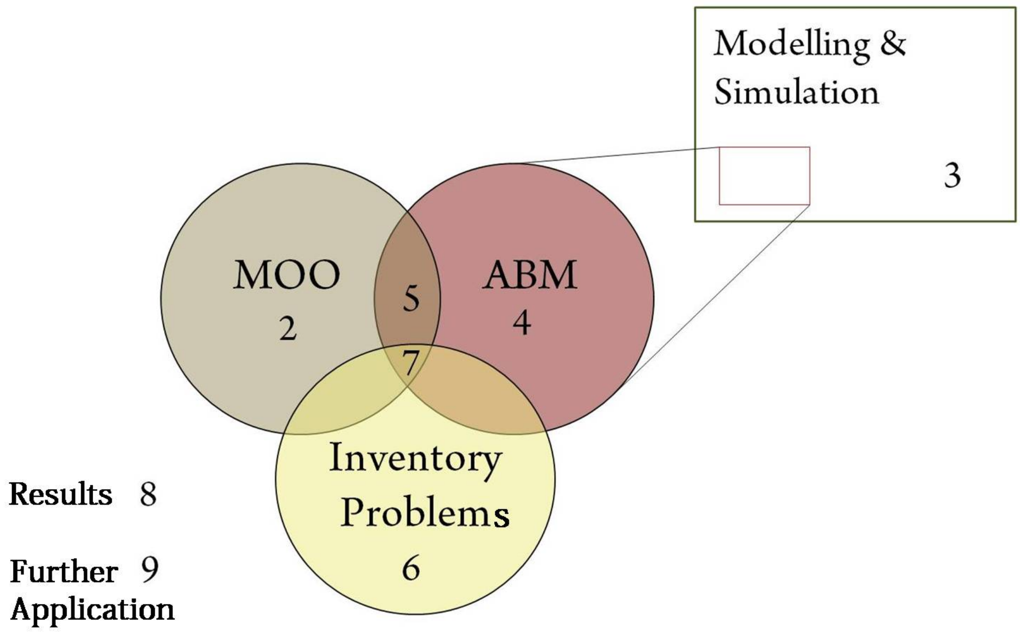


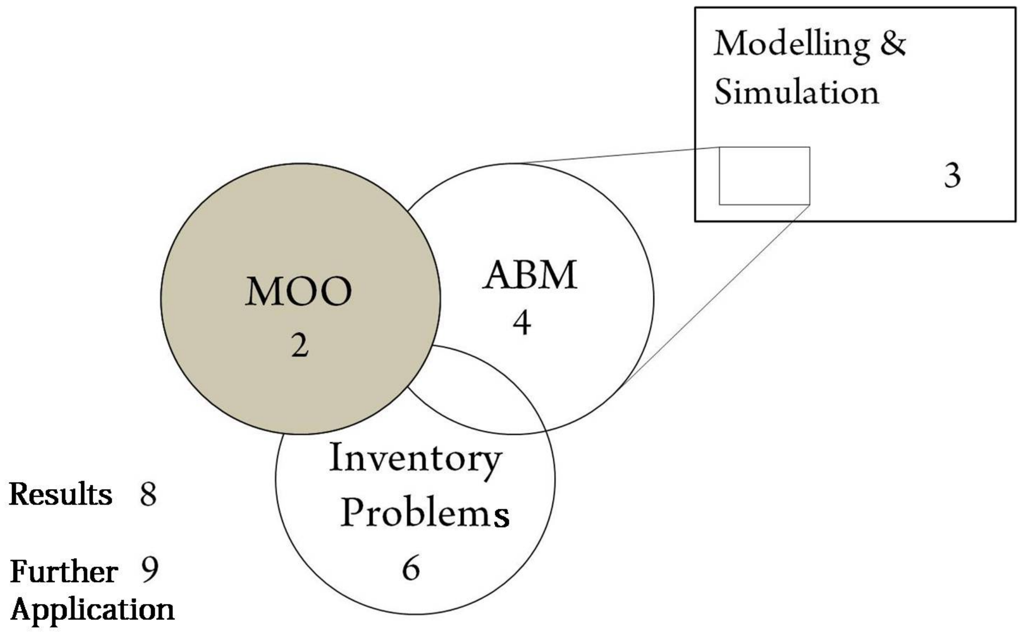
Figure 1.1: Road map of the document.

This chapter provided background information, an introduction to the problem studied and objectives of the study. In Chapter 2 a brief overview of multi-objective optimisation is given, with specific focus on a few important definitions relating to it. Chapter 3 provides an outline of the basic concepts relating to modelling and simulation. The reader is thereafter introduced to agent-based modelling as presented in Chapter 4, and examples of its use in multi-objective optimisation is given in Chapter 5. The focus then shifts to inventory problems in

Chapter 6 which have been identified as the application area in which the agent-based approach to multi-objective optimisation is evaluated. The agent-based model developed for this purpose is described in detail in Chapter 7. Chapter 8 describes how the performance of the approach can be measured and different scenarios compared. In Chapter 9 the results of the study are presented and analysed. Finally, conclusions are drawn from the study.

Chapter 2

Multi-objective Optimisation



In everyday decision-making, it is very rare for us to encounter problems where only one objective is concerned. This chapter aims to provide the reader with a brief introduction to multi-objective optimisation. A number of important definitions pertaining to MOO will be described. A summary of the different metaheuristics available for MOO will also be given.

2.1 Introduction to Multi-objective Optimisation

Optimisation and decision-making methods presented in graduate courses are usually focussed on linear programming techniques where only a single objective is optimised. However, in real-world decision-making a trade-off generally needs to be made between conflicting objectives. To complicate matters further these objectives are often measured in different units.

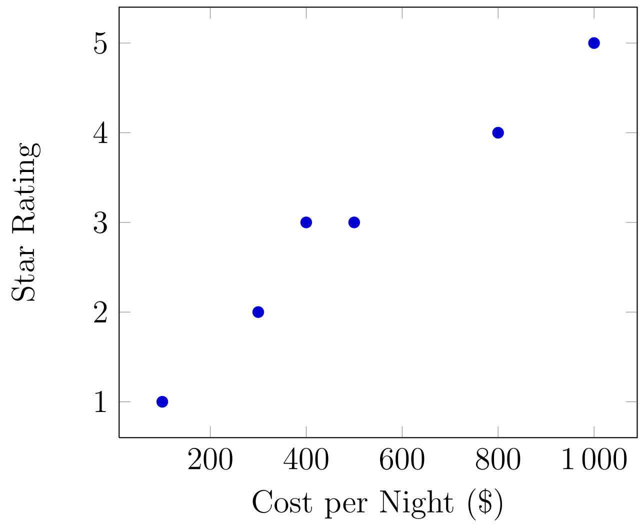


Figure 2.1: Hypothetical trade-off scenario in choosing a hotel.

A classic example of this can be found in a tourist choosing a hotel for the night (Branke *et al.*, 2008). Hotel rooms are available with the cost per night ranging between \$100 for a one-star hotel and \$1000 for a five-star hotel, as shown in Figure 2.1. If cost is the only objective to be taken into account, the tourist will choose the one-star hotel. However, it is expected that the one-star hotel is less comfortable than a higher rated hotel. If the tourist is very rich and comfort is his only concern, then the five-star hotel will be his optimal choice. The tourist however has many other options between these two extremes, but he will have to consider a trade-off between cost and comfort. In this example there are two three-star hotels that each charge a different rate. The one costs \$400 per night and the other \$500. By considering both objectives it is clear that the \$400 hotel is optimal in this case. These trade-off solutions provide a clear front on the objective space. This front is known as the *Pareto front* and the set of solutions is called the *Pareto approximation set*.

Multi-objective optimisation is defined as the process of simultaneously optimising two or more conflicting objectives which are subject to certain constraints. The terms *multi-objective* or *multi-criteria* indicate that the notion of optimality is quite ambiguous in these problems because decisions which optimise one objective do not necessarily optimise the others. There are two general approaches to multi-objective optimisation (Konak *et al.*, 2006). The first approach is to combine all the individual objectives into a single function by using techniques such as the weighted sum method. The problem can then be solved by simple linear programming. The problem with this approach is that it is often very difficult to precisely and accurately choose the weights to apply to the different objectives. As in the above example of the hotels, the stars cannot be simply converted into a monetary value. The second general approach is to determine an entire Pareto optimal solution set of non-dominated alternatives.

The following definitions pertaining to Pareto optimality are defined by Coello Coello (2009):

Definition 1: Given two vectors $\mathbf{u}$ and $\mathbf{v} \in \mathbb{R}^m$, then $\mathbf{u} \leq \mathbf{v}$ if $u_i \leq v_i$ for $i = 1, 2, \ldots, m$, and that $\mathbf{u} < \mathbf{v}$ if $\mathbf{u} \leq \mathbf{v}$ and $\mathbf{u} \neq \mathbf{v}$.

Definition 2: Given two vectors $\mathbf{u}$ and $\mathbf{v} \in \mathbb{R}^m$, then $\mathbf{u}$ dominates $\mathbf{v}$ (denoted by $\mathbf{u} \prec \mathbf{v}$) if $\mathbf{u} < \mathbf{v}$.

Definition 3: A vector of decision variables $\mathbf{x}^* \in \Omega$ (Ω is the feasible region) is *Pareto optimal* if there does not exist another $\mathbf{x} \in \Omega$ such that $\mathbf{f}(\mathbf{x}) \prec \mathbf{f}(\mathbf{x}^*)$.

Definition 4: The *Pareto approximation set* $\mathcal{P}^*$ is defined by $\mathcal{P}^* = \{\mathbf{x} \in \Omega | \mathbf{x} \text{ is Pareto optimal}\}$.

Definition 5: The *Pareto front* $\mathcal{P}^*_T$ is defined by $\mathcal{P}^*_T = \{\mathbf{f}(\mathbf{x}) \in \mathbb{R}^n | \mathbf{x} \in \mathcal{P}^*\}$.

2.2 Metaheuristics for Multi-objective Optimisation

There are a number of metaheuristics that have been successfully adapted for multi-objective optimisation. A summary of the metaheuristics identified by Bekker (2012) is given in Table 2.1.

Metaheuristic	**Author**	**Summary**
Evolutionary algorithms	(Coello Coello, 2009)	Inspired by natural selection in the biological world, poor solutions are weeded out from a population of solutions.
Simulated annealing	(Kirkpatrick *et al.*, 1983)	Locates a good approximation to the global optimum of a given function in a large search space.
Tabu search	(Glover & Laguna, 1997)	Enhances the performance of a local search method by using memory structures that describe the visited solutions. Once a potential solution has been determined, it is stored in a tabu list so that the algorithm does not visit that possibility repeatedly.
Ant systems	(Dorigo, 1992)	Inspired by real ants foraging for food, an optimal route is established by an increasing number of artificial ants following the same route.
Particle swarm optimisation	(Kennedy & Eberhart, 2002)	Iteratively tries to improve a candidate solution with regard to a given measure of quality, simulating the movement of organisms in a flock of birds or a school of fish.
Hill climbing techniques (extended to MOO)	(Weise, 2008)	A single solution is initially created. Thereafter it attempts to improve the solution by incrementally changing a single element of the solution.
Differential evolution	(Storn & Price, 1997)	Optimises a problem by creating a new candidate solution through a combination of existing ones.

Metaheuristic	Author	Summary
Artificial immune systems	(Bersini & Varela, 1991)	Exploits the immune system's characteristics of learning and memory to solve a problem.
Memetic algorithms	(Moscato, 1989)	Population-based approach for problem search with separate individual learning or local improvement procedures.
Evolution strategy	(Beyer & Schwefel, 2002)	Search operators are applied in a loop, with a sequence of iterations (generations) continued until a termination criterion is met.
Firefly algorithm	(Yang, 2008)	Inspired by the flashing behaviour of fireflies where brighter flashes attract others, the algorithm associates the brightness with the objective function.

Table 2.1: Metaheuristics for multi-objective optimisation.

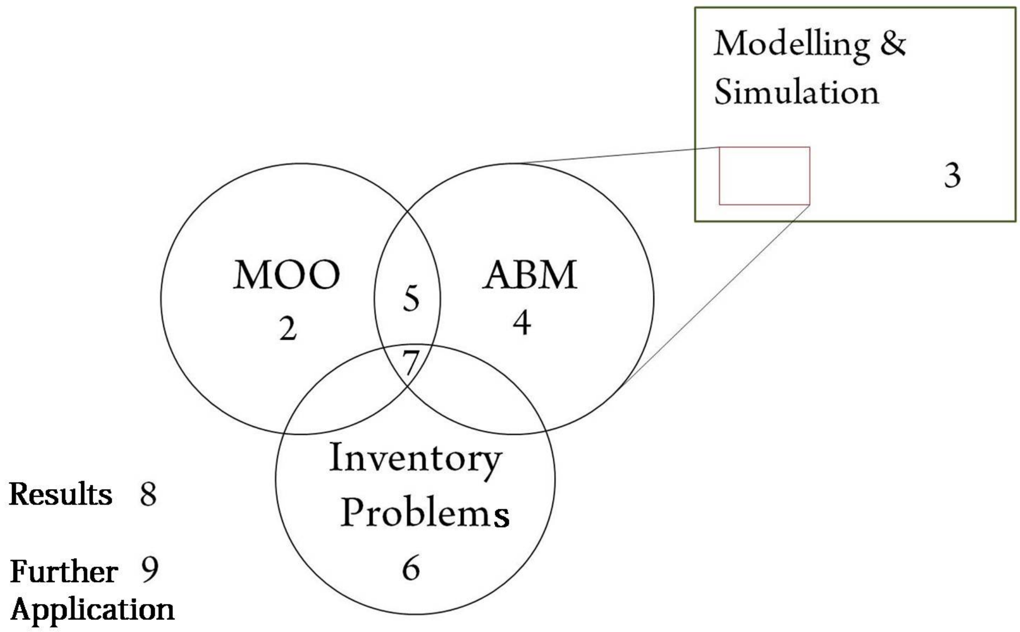
2.3 Concluding Remarks: Chapter 2

A brief overview of multi-objective optimisation was presented in this chapter. The purpose was to introduce the reader to this research field and explain some of the important definitions related to it. Different metaheuristics that can be applied in MOO were also summarised.

In the next chapter the focus turns to modelling and simulation, and how it can be effectively applied in a decision-making process.

Chapter 3

Modelling and Simulation



Modelling and simulation is a powerful tool which can be used to assist with complex decision-making. An overview of modelling and simulation will be given in this chapter. A number of different approaches and paradigms will be discussed and an outline given of the general steps to be followed during a simulation study.

3.1 Introduction to Modelling and Simulation

3.1.1 What is Modelling and Simulation?

A simulation model is a simplified representation of a real-world system. From a practical viewpoint, Kelton *et al.* (1998) describes simulation as the process of designing and creating a computer model of a real or proposed system. The purpose of the model is to conduct a number of virtual experiments to gain a better understanding into the behaviour of the system.

3.1.2 Why Use Simulation Modelling?

George Box expressed a very important concept of simulation modelling (Box & Draper, 1987):

> *Essentially, all models are wrong, but some are useful.*

A model that provides a sufficient representation of reality has many benefits. Some of the main uses of simulation identified by Banks (1999) and Kelton *et al.* (1998) are described below:

- It provides users with practical feedback regarding the effectiveness and efficiency of a design before the system is constructed. The typical cost of a simulation study is often significantly lower than the cost for redesign or modifications to a system after design.
- It allows the user to evaluate alternative designs and to explore new control philosophies, operating procedures and methods.
- Simulation helps to establish where the constraints lie in the system to ensure that it is properly managed.
- The significance of certain parameters can be determined by performing a sensitivity analysis.
- It enlightens the user why certain phenomena are occurring in the real system.

- Models where animation is provided can be used as an effective means for illustrating concepts relating to the system.

3.2 Simulation Modelling Approaches

According to Kelton *et al.* (1998) a simulation model can be classified along the following three dimensions:

Static versus Dynamic A system can be modelled independent of time, or with time playing a significant role. A static simulation model describes the behaviour of a system at a specific point in time. On the other hand, a dynamic simulation model simulates the changing behaviour of a system over a period of time. Although static models can be developed in a spreadsheet, specialised software is often required to develop dynamic models.

Deterministic versus Stochastic Very few real-world systems are completely free from the influence of random variation. A simulation model that is deterministic ignores this randomness. A stochastic simulation model uses random values from statistical distributions in some of its parameters to make provision for random variation in the system. It is often necessary to run multiple replications of the same scenario in a stochastic model to ensure that the results and findings are statistically relevant.

Discrete versus Continuous The way that the model deals with changes in the state of the system is another way in which a model can be classified. As described by Banks (1999) the system state variables are the collection of all the information needed to define what is happening within the system. The contrast between discrete models and continuous models is based on the variables that are needed to track the state of the system. In a discrete model the changes in the system state variables occur only at specific points defined as event times. The system state variables in continuous models are defined by differential or difference equations that change continuously over time.

3.3 Modelling Paradigms

A number of different modelling paradigms exist, each preferable to a relevant area of application. Borshchev & Filippov (2004) distinguish between the following modelling paradigms:

Systems Dynamic Modelling System dynamic modelling is a useful tool to determine how organisational structure, amplification in policies and time delays in decisions and actions interact to influence the success of the business. A system dynamic model describes the system as a number of causal loops and stock-flow diagrams that represent the relationships between the variables in the model. From a mathematical point of view the model consists of a system of differential equations.

Dynamic Systems Modelling Dynamic systems modelling can be seen as a mathematical representation of the dynamics between the inputs and the outputs of a dynamic system. Graphical modelling languages like Matlab-Simulink are typically used with the model, consisting of a number of state variables and differential equations of various forms.

Discrete Event Modelling The operation of a system is represented as a chronological sequence of events in discrete event modelling. The state changes in the model occur over randomly spaced discrete points in time and takes place as a result of activity times, delays, and entities that compete for system resources.

Agent-based Modelling In an agent-based model, agents are used to model behaviour at an individual level, with the global behaviour emerging as a result of their behaviour rules and interactions. Agent-based modelling is the preferred modelling paradigm used in this study and will be described in detail in Chapter 4.

3.4 Steps in a Simulation Study

The following steps are suggested by Law & Kelton (1991) to perform a simulation study:

1. *Problem Formulation and Definition*: Ensure that there is a clear understanding of the problem, the goals, purposes and expectations of the study.

2. *Planning*: Compile a project plan taking into account personnel, hardware, software, funding, and time requirements.

3. *Defining the Boundaries of the Study*: The boundaries determine what is included and excluded from the model. The purpose is primarily to simplify the study by reducing the amount of detail required.

4. *Conceptualisation*: Pseudo-code or block diagram format is used to construct the proposed model in order to gain a better understanding of the model, to establish the first order logic of the model, and to verify the level of detail and assumptions.

5. *Preliminary Experiment*: This step comprises the establishment of the level of confidence for the confidence intervals, model time span, input variables, measures of performance, data requirements, entity definitions, entity attributes and model resources.

6. *Parameter Selection*: Select the parameters that will be investigated to obtain the desired information.

7. *Input Data Requirements*: Collect and process the input data required for the study.

8. *Translation of the Model to a Simulation Language*: Develop the computer simulation model.

9. *Verification*: Debug the model to ensure that the computerised model works correctly.

10. *Validation*: Confirm that the model is an adequate representation of the real world system.

11. *Rework*: The model is reworked in order to address the potential problems identified during the verification and validation steps. It is an iterative process which is repeated until the model is of an acceptable standard.

12. *Initial run*: A set of replications of the base case is run which can be used for statistical analysis.

13. *Statistical Analysis*: Determine the preliminary confidence intervals to determine the number of replications required.

14. *Model Execution*: Execute the production runs.

15. *Documentation*: The study needs to be documented from the start. At this point the results, interpretations and conclusions are added.

16. *Implementation, Maintenance and Monitoring*: The results are implemented and maintained and feedback is obtained from the client to evaluate the success of the study.

3.5 Application Areas

Simulation is very versatile with many different application areas:

- Manufacturing systems
- Health care
- Military
- Mining
- Transportation systems
- Construction systems
- Supply chains and logistics

- Business process re-engineering
- Computer system performance
- Communications
- Environmental studies
- Financial decision support systems

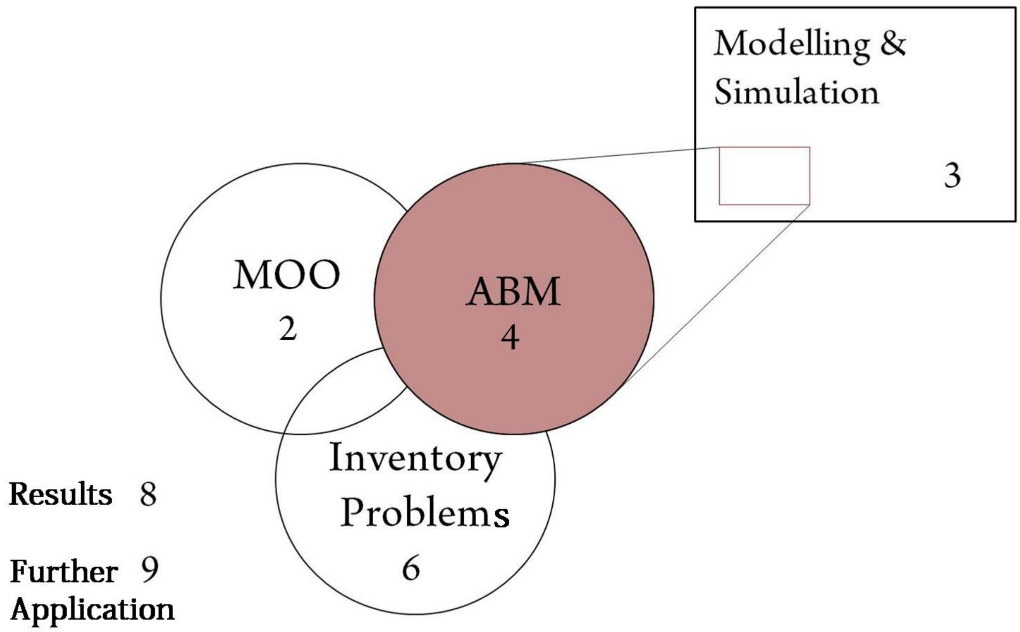
3.6 Concluding Remarks: Chapter 3

In this chapter the reader was introduced to a few basic concepts regarding modelling and simulation. It was shown that simulation modelling is a powerful problem-solving tool to assist with system design and to analyse "what-if" scenarios.

In the next chapter the reader will be introduced to a specific modelling paradigm – Agent-based modelling.

Chapter 4

Agent-based Modelling



In the previous chapter an introduction was given into modelling and simulation. Various different modelling paradigms were described. This chapter focusses on one of these paradigms, namely agent-based modelling. The chapter starts off with an overview of agents, which includes a review of literature on negotiations between agents. Thereafter the basic principles of agent-based modelling are discussed. Some background on the history of ABM is provided, with specific reference to previous applications in social sciences and supply chain management. Finally, the basic steps in building an agent-based model are described.

4.1 Agents

4.1.1 Definition of an Agent

The term *agent* has a diverse range of definitions in different fields of study. The agents relevant in this study – as applied in agent-based modelling – are commonly referred to as *software agents* in computer science. Although there is no universal agreement on the precise definition for this type of agent, most researchers agree with Wooldridge & Jennings (1995) that any object, computer system, or program can be classified as an agent if it has the following properties:

1. *Autonomy*: It should have some control over its actions and should work without human intervention.

2. *Social ability*: It should be able to communicate with other agents and/or with human operators.

3. *Reactivity*: It should be able to perceive and react to changes in its environment.

4. *Pro-activeness*: It should have reasoning capacity and be able to learn from experience. It should not only respond in reaction to certain stimuli, but it should take initiative as part of a more complex goal directed behaviour.

There are a few other attributes of agents which are generally agreed to be optional. Ingham (1999) lists the following optional attributes:

1. *Adaptation*: Agents may attempt to adapt themselves to better suit their new or changing environment to deal with new or changing goals. An agent usually follows a set of predefined rules and then applies them. Casti (1997) and Papazoglou (2001) argue that for an agent to be deemed intelligent it

also requires an additional high-level set of "rules to change the rules". The base-level rules are applied in response to the environment, while the high-level rules enable the agent to learn and adapt to changes in the environment.

2. *Mobility*: Agents may in some instances move from one system to another. Gupta *et al.* (2001) makes use of mobile agents to facilitate access to data required for improved supply chain decision making. In their system, mobile agents act as local representatives for remote services and provide interactive access to data they accompany.

3. *Cooperation and collaboration*: Agents may in some circumstances work together due to a specific event, or in order to achieve a specific goal. Each agent usually benefits from this cooperation.

4. *Negotiation*: Agents may be able to negotiate with each other, usually in some form of cooperation. Cooperation and negotiation between agents will be discussed in further detail in the next section.

4.1.2 Cooperation and Negotiation Between Agents

Agents are able to do more than just communicate; they are able to cooperate and negotiate with each other. Numerous research have been focussed on these complex social interactions between agents. Multi-agent software was developed by Sycara (1998) that allows agents to collaborate with each other to manage information. The agents form adaptive teams to solve decision-making and information management tasks delegated by users. The work of Axelrod (1997) shows that sustainable cooperative behaviour between agents can be established by applying a simple *tit-for-tat* strategy of reciprocal behaviour towards individuals.

Wong *et al.* (1997) introduces the *Concordia* infrastructure for the development and management of mobile agent applications for accessing information – anytime, anywhere and on any device. The infrastructure extends the notion of simple agent interaction with support for agent collaboration, which in this case allows the agents to interact and modify external and internal agent states.

Raiffa (1982) developed a basic model for bilateral negotiation between autonomous agents. The negotiations are based on a set of mutually influencing *two parties, multiple issues* negotiations, where offers and counter-offers are generated by linear combinations of simple functions. Faratin *et al.* (1998) builds on the work of Raiffa, introducing the framework for a service-oriented negotiation model. The model defines a range of strategies and tactics that agents can employ to generate initial offers, evaluate proposals and offer counter proposals.

Krishna & Ramesh (1998) present an approach for designing intelligent agents that are capable of negotiating on behalf of their human counterparts and then suggest market strategies that the human counterpart can implement. They detail a new negotiation protocol that does not require the agents to share any trustworthy information.

Chapelle *et al.* (2002) makes use of agent satisfaction measures to facilitate cooperation between agents. Two generic agent satisfaction measures are defined: *Personal satisfaction*, which evaluates the progression of the agent's actions, and *interactive satisfaction*, which evaluates the effect of the neighbouring agent's actions on the agent's task. Reinforcement learning is used to ensure that the agents learn to select behaviours that are well adapted to their neighbours' activities.

State diagrams, first introduced by Booth (1967), are often used in agent-based models to provide a graphical representation of the behaviour of the different agents. It provides a clear and intuitive approach to model negotiation and cooperation between agents (Kendall *et al.*, 1996).

4.2 Characteristics of an Agent-based Model

According to Sanchez & Lucas (2002) an agent-based model is a model where multiple entities sense and stochastically respond to conditions in their local environments, mimicking complex large-scale system behaviour. Agents are used to represent the entities that interact with each other and to the environment according to a set of rules that govern their actions and decisions (Chatfield *et al.*, 2007).

One of the key characteristics of any agent-based model is that it is decentralised (Garifullin *et al.*, 2007). The focus is placed on the individual behaviour rules of the agents, with the global behaviour emerging as a result of many individual activities. Bonabeau (2002) uses a simple game as an example to explain this *emergent phenomenon*:

Protector game: The game is played with a group of 10–40 people in the audience. Each member i in the audience randomly selects two individuals, person A_i and person B_i. They are instructed to move so that they always keep person A_i between them and B_i so A_i is their protector from B_i. Everyone in the room will walk about in a seemingly random fashion. They are then instructed to move so they always keep themselves inbetween A_i and B_i, thereby making themselves the protector. The result is quite remarkable: The whole room will almost instantaneously implode with everyone clustering in a tight knot. In this example simple individual rules are defined for the agents which lead to clear collective behaviour from the group. Small changes in the rules can make a dramatic impact on the global behaviour of the system. The group's collective behaviour is an emergent phenomenon.

Agent-based modelling has become a popular tool to model and understand complicated systems. It is one of the most natural methods to simulate systems that contain entities which exhibit complex behaviour. It enables one to realistically predict the global impact of small changes in individuals' behaviour. Therefore it is often used in the following application areas (Bonabeau (2002):

1. *Flows*: Evacuation, traffic, and customer flow management.
2. *Markets*: Stock market, shopbots and software agents, and strategic simulation.
3. *Organisations*: Operational risk and organisational design.

4. *Diffusion*: Diffusion of innovation and adoption dynamics.

In a passage in the fantasy novel by Pratchett (2007) a fictional device, the *Glooper*, is described. He provides a very unique explanation of what an agent-based model can be used for:

> "Mr. Hubert believes that this device is a sort of crystal ball for showing the future," said Bent, and rolled his eyes.
>
> "Possible futures. Would Mr. Lipstick like to see it in operation?" said Hubert, vibrating with enthusiasm and eagerness.[...]
>
> "The Glooper, as it is affectionately known, is what I call a quote analogy machine unquote. It solves problems not by considering them as a numerical exercise but by actually duplicating them in a form we can manipulate: in this case, the flow of money and its effects within our society become water flowing through a glass matrix the Glooper. The geometrical shape of certain vessels, the operation of valves and, although I say so myself, ingenious tipping buckets and flow-rate propellers enable the Glooper to simulate quite complex transactions. We can change the starting conditions, too, to learn the rules inherent in the system. For example, we can find out what happens if you halve the labour force in the city by the adjustment of a few valves, rather than by going out into the streets and killing people."
>
> "A big improvement! Bravo!" said Moist desperately, and started to clap. No one joined in.

Similar to the Glooper, an agent-based model can give the user the ability to understand the dynamics of complex systems.

4.3 Background on Agent-based Modelling

Heath & Hill (2010) consider the origins of agent-based modelling to lie hundreds of years back when scientists first began discovering and attempting to explain the emergent and complex behaviour seen in nonlinear systems. Scholars such as Macal & North (2006) believe that it has its direct historical roots in *complex adaptive systems* concerning the question of how complex behaviours arise in nature among myopic, autonomous agents. Chatfield *et al.* (2007) argues that the concepts of agent-based modelling were developed out of a sub-field of *distributed artificial intelligence* work which focussed on the coordination of multiple autonomous or semi-autonomous agents (Bond & Gasser, 1988). In fact, agent-based modelling has its roots in various different fields of study including economics, system dynamics, computer science, management science, social science, game theory and traditional modelling and simulation. Agent-based modelling draws on these fields for its theoretical foundations, its conceptual world view and philosophy, and for applicable modelling techniques.

Some application areas of agent-based modelling are discussed next.

4.3.1 Agent-based Modelling in the Social Sciences

Agent-based modelling has been growing in recognition and popularity over the past thirty years, specifically driven by its increased application in the field of social sciences. In these applications agents represent people, and agent relationships represent processes of social interaction (Gilbert & Troitzsch, 1999). One of the first social agent-based models was developed by Schelling (1978) who studied housing segregation patterns, trying to determine if segregated settlement patterns would still emerge if most of the population was colour-blind. His model proved "that patterns can emerge that are not necessarily implied or even consistent with the objectives of the individual agents".

4.3.2 Agent-based Modelling of Supply Chains

Supply chain management intrinsically deals with coordination between different business entities, which makes an agent-based model, based on explicit communication between the agents, a natural choice for supply chain management. Agents are able to capture the distributed nature of supply chain entities (e.g. customers, manufacturers, inventory managers etc.) in order to mimic their business behaviours and to collaboratively plan the supply chain operations. According to Fox *et al.* (2000) an agent-based model of a supply chain will have the following features:

1. *Distributed*: The functions of supply chain management are divided among agents.
2. *Dynamic*: Each agent performs its functions asynchronously as required.
3. *Intelligent*: Each agent is an expert in its function applying artificial intelligence and operations research problem-solving methods.
4. *Integrated*: Each agent is aware of and can access the functional capabilities of other agents.
5. *Responsive*: Each agent can ask for information from or a decision from other agents.
6. *Reactive*: An agent is able to respond to events as it occurs.
7. *Cooperative*: An agent can cooperate with other agents to find a solution.
8. *Reconfigurable*: The supply chain system must be reconfigurable for different scenarios.

A review of scholarly literature yields a number of examples where agent-based models were applied in supply chain management. Swaminathan *et al.* (1997) developed a multi-agent framework to enable rapid development of customised decision tools for supply chain management. Their work focusses on building a supply chain library of agents and control elements which could be used when developing a new model of a supply chain.

Julka *et al.* (2002) also developed a framework for modelling, monitoring and managing supply chains. The framework is specifically developed for application in the supply chain of a refinery and is focussed on providing decision support.

The work of Verdicchio & Colombetti (2002) explores how *social contracts* between companies in a supply chain can be modelled. The authors refer to these set of rules as *commitments* that a company makes with respect to others.

Chen *et al.* (2008) presents an inventory scheduling model for a supply chain system based on an agent-oriented Petri net. A conceptual framework for agent-based modelling of distributed supply chains is proposed by De Santa-Eulalia *et al.* (2008). Their work also includes specific methods for understanding and modelling simulation problems at the initial phase of the modelling effort.

Valluri & Croson (2005) uses agent-based modelling to study the performance of a supplier selection model.

An interesting application of a multi-agent system in supply chains is highlighted in the work of Pan *et al.* (2009) on reorder decision-making in the apparel supply chain. In their model they make use of an inventory manager agent who is responsible for controlling inventory and making decisions about reorder strategies and price setting. A client agent collects sales information from their market, forecasts the future customer demand and provides feedback to the inventory manager. The authors apply fuzzy knowledge to determine reorder points by taking into consideration the market changes and fashion trends. A genetic algorithm is applied to forecast the reorder volume with the aim of minimising the total cost in the supply chain. The model considers fashion trends, seasonal distribution, sales records, and point of sales data to adapt to the changing market. An important contribution of their work is proving how information sharing between entities in the supply chain can be used to optimise reorder strategies.

4.3.3 Agent-based Modelling as a Tool for Multi-objective Optimisation

Agent-based modelling has been applied quite effectively in the field of multi-objective optimisation. A review of the literature on this topic is described in Chapter 5.

4.4 Building an Agent-based Model

There are a number of modelling platforms available which can be used for agent-based modelling. Some of the most popular ones identified by Allan (2009) and Gilbert & Bankes (2002) are:

- ABLE
- Anylogic
- Breve
- Cougaar
- iGen
- JADE
- LSD
- MASON
- Netlogo
- RePast
- SDML
- SugarScape
- Swarm
- VisualBots
- Xholon
- Zeus

Macal & North (2006) expands on the standard model building tasks identified in Chapter 3 by identifying the following steps required for agent-based modelling:

1. *Agents*: Identify the agent types and other objects (classes) along with their attributes.

2. *Environment*: Define the environment the agents will live in and interact with.

3. *Agent Methods*: Specify the methods by which agent attributes are updated in response to either agent-to-agent interactions or agent interactions with the environment.

4. *Agent Interactions*: Add the methods that control which agents interact, when they interact, and how they interact during the simulation.

5. *Implementation*: Implement the agent model in computational software.

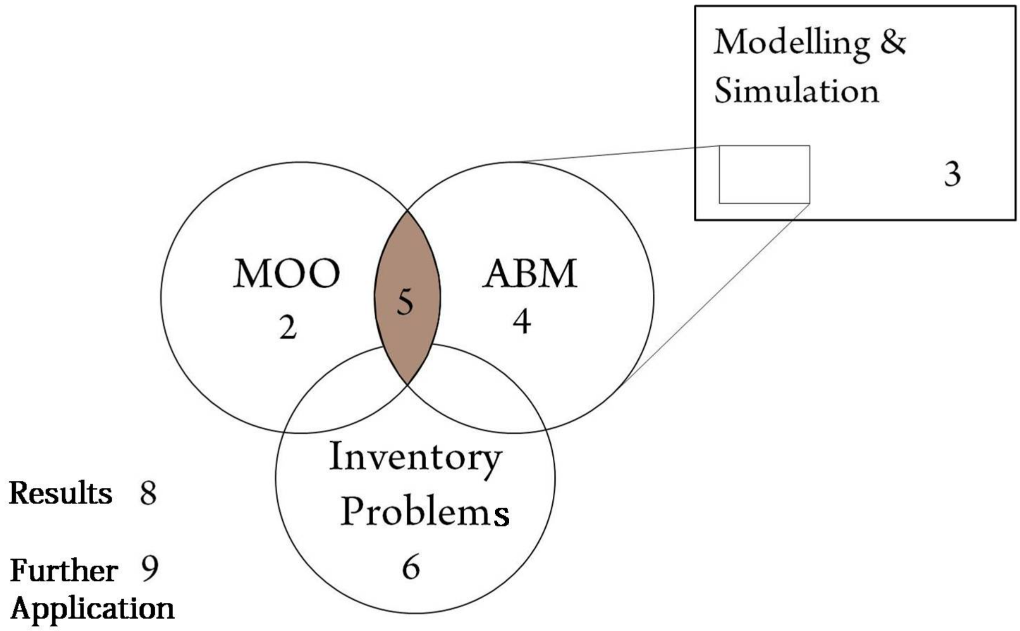
4.5 Concluding Remarks: Chapter 4

The purpose of this chapter was to introduce the reader to agents and agent-based modelling. A review of literature on ABM showed many different applications in social sciences and supply chain management. Of specific interest to this study was the work of Pan *et al.* (2009) which investigated how ABM could be applied to assist with reorder decision-making in the apparel supply chain. The chapter concluded with a list of modelling platforms for ABM and basic steps to be followed when constructing such a model.

The application of ABM in multi-objective optimisation will be discussed in the next chapter.

Chapter 5

Agent-based Modelling in Multi-objective Optimisation



In Chapter 2 a summary was given of some metaheuristics that can be applied to multi-objective optimisation. The attention in this chapter is shifted to how agent-based modelling – described in Chapter 4 – can be applied in multi-objective optimisation. The chapter starts off with an overview of scholarly literature on the application of ABM in MOO. The suitability of an ABM approach to MOO is also discussed.

5.1 Review of Literature on Agent-based Modelling in Multi-objective Optimisation

A literature survey on applications, where agent-based modelling is used as a metaheuristic in multi-objective optimisation, reveals limited research on the subject. In most instances where these two fields of study meet, MOO is employed to improve the accuracy and performance of agent-based models.

5.1.1 Multi-objective Optimisation as a Calibration Tool for Agent-based models

Running an agent-based simulation is quite easy, but the analysis is often more difficult. Terano & Naitoh (2004) list the following difficulties:

1. The model becomes too complex to be manually calibrated for accuracy
2. There are few similarities between the simulation results and the real-world phenomena
3. The results are too difficult to interpret
4. It is difficult to validate the parameters of the model after the simulation

MOO is a valuable tool that can be used to address the first two issues. For example, in the research of Terano & Naitoh (2004) they develop an agent-based model to explore optimal marketing strategies for a specific market. The customers are represented by agents with different purchasing attitudes. In order to ensure that the model is an accurate representation of the real system, they make use of genetic algorithms to tune the agents' parameters.

In another theoretical application Rogers & von Tessin (2004) use a multi-objective evolutionary approach to calibrate an agent-based model of a financial market.

The objective functions to be optimised in both these examples are the mean and the variance of the simulated model with respect to the real data.

5.1.2 Multi-objective Optimisation of Emerging Behaviour in Agent-based Models

In a research paper by Narzisi *et al.* (2006) the authors propose the use of *multi-objective evolutionary algorithms* (MOEAs) to optimise the emergent global behaviour in agent-based models. They apply their research to the selection of emergency response plans in disaster management. A comprehensive agent-based model was developed to simulate large-scale urban disasters. The system parameters at the local level of the agent behaviour rules must be tuned in order to achieve some desirable global objectives. The multiple objectives to be optimised are the following: Minimise the number of casualties, fatalities, the average ill-health of the population, and the average waiting time at the hospital, and maximise the average time taken by a person to die, and the utilization of resources at the different locations. Economic, legal and ethical issues also contribute to the objective functions. Two well-known MOEAs, the *Nondominated Sorting Genetic Algorithm II* (NSGA-II) (Deb *et al.*, 2000) and the *Pareto Archived Evolution Strategy* (PAES) (Knowles & Corne, 2000), were applied to estimate the Pareto front of the problem.

The Gantt diagram's optimisation in the job-shop scheduling problem can be considered an NP-hard problem (Graham, 1966). Cardon *et al.* (2000) developed a dynamic agent-based model to simulate the behaviour of entities that collaborate to optimise the Gantt diagram. Multiple objectives exist in that the delay and the advance of the set of jobs need to be minimised. Genetic algorithms are used to determine a set of good heuristics for a given benchmark and new schedules obtained with agent negotiations. The study has opened up an interesting research area by introducing genetic patrimony to agents.

5.1.3 Agent-based Modelling as a Heuristic in Multi-objective Optimisation

There are very few instances in literature where agent-based heuristics were used in multi-objective optimisation. In most of these instances the heuristics are not necessarily purely agent-based, but often have its roots in other classical approaches.

Socha & Kisiel-Dorohinicki (2002) present an agent-based evolutionary approach to search for a Pareto optimality set within a multi-objective optimisation problem. In their agent-based approach the evolution process is decentralised, allowing the search space to be intensively explored to find the Pareto front. A valuable outcome of their research is showing how the introduction of the *crowd principle* discourages the agents from creating large bunches of similar solutions at some points of the Pareto front. The algorithm of the crowd mechanism is the following:

1. One of the agents (Agent A) initiates the communication by requesting the solution from another agent (Agent B).

2. Agent B presents its solution to the problem to Agent A.

3. Agent A then compares its solution to the one obtained and calculates the similarity level of the two solutions described as a distance (in square metric) between the two solutions.

$$d(x^A, x^B) = \sum_{i=0}^{N_c} |x_i^A - x_i^B| \tag{5.1}$$

with N_c the number of dimensions of the problem, x_i^A the i-th coefficient of the solution owned by Agent A and x_i^B the i-th coefficient of the solution owned by Agent B.

4. Agent A checks if the other solution is to be considered similar, i.e. if the distance computed in the previous step is less than the crowding factor. If so, Agent A receives some energy from Agent B. The flow of energy causes that some of the similar agents are more likely to be eliminated.

An algorithm based on reinforcement learning was developed by Mariano & Morales (2000) for use in multi-objective optimisation. A family of agents is assigned to each objective function. Each agent proposes a solution for the objective function to which it is assigned. They leave traces while they construct solutions considering the traces made by other agents. The proposed solutions are evaluated for Pareto optimality. The algorithm is able to produce a relatively good approximation of the Pareto front for a wide range of multiple objective optimisation problems reported in the literature.

5.2 Suitability of an Agent-based Approach to Multi-objective Optimisation

Although the agent-based approaches to multi-objective optimisation reviewed in the previous section all find their roots in other metaheuristic approaches, agent-based modelling – as a stand-alone technique – also has the potential to be an effective metaheuristic.

Humans are continuously faced with decisions that need to be made which involve multiple criteria. Whether we are investing, choosing a career or even deciding what to have for dinner, we are very rarely faced with problems which concern only a single objective. Humans have the instinctive ability to do trade-off analysis in everyday decision-making.

Therefore, if some of the basic elements of the human decision-making process can be modelled, such a model should be able to perform multi-objective optimisation. As shown in Chapter 4, the characteristics of agents lend itself to the realistic modelling of complex social interactions as found in humans. It is therefore the researcher's view that agent-based modelling can be used as a metaheuristic to perform multi-objective optimisation.

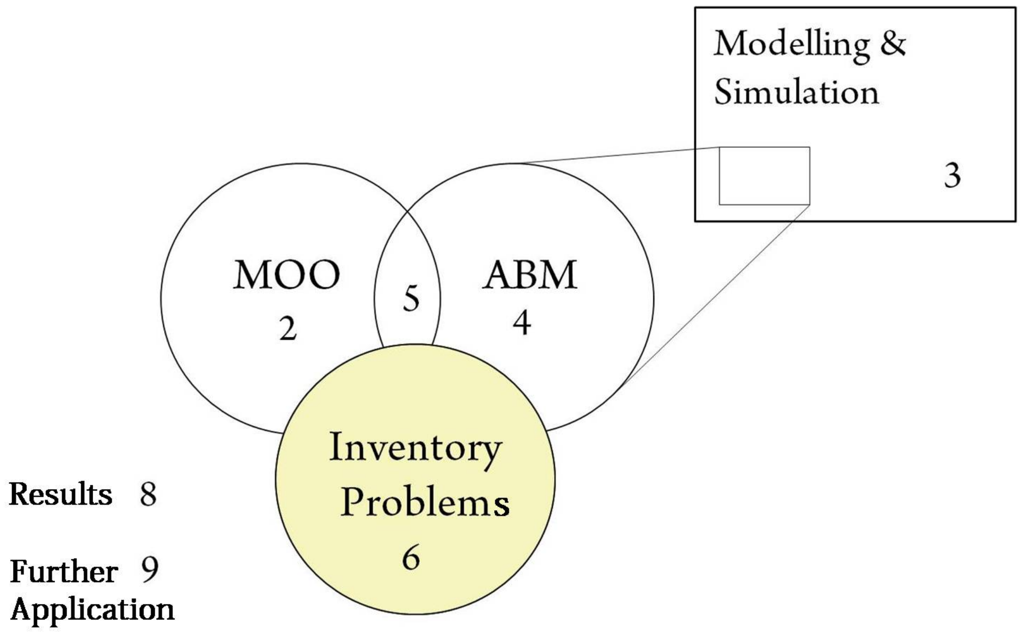
5.3 Concluding Remarks: Chapter 5

In this chapter some previous applications of agent-based modelling in multi-objective optimisation, and vice-versa, were reviewed. At the end of the chapter specific focus was given to the decision-making capabilities of agents. The chapter concluded with the researcher's view on why an agent-based approach is suitable for multi-objective optimisation.

In the next chapter inventory problems will be discussed, with specific focus on the inventory problem on which the agent-based multi-objective optimisation is applied.

Chapter 6

Inventory Problems



The previous four chapters were focussed on the concepts of agent-based modelling and multi-objective optimisation. The purpose of this study is to determine if an agent-based modelling approach can be used as a metaheuristic for multi-objective optimisation. The inventory problem has been identified as a suitable subject area to which this approach is applied. In this chapter several examples of inventory problems will be discussed. The specific inventory problem, on which the agent-based multi-objective optimisation is applied, will also be described.

6.1 Introduction to Inventory Problems

6.1.1 Inventory Management

Inventory management is one of the most important functions of supply chain management. According to Coyle *et al.* (2002) managing inventory involves the following four fundamental questions:

1. When should an order be placed from an upstream supplier and/or their plants
2. How large should each order be
3. Where the inventory should be held
4. What specific line items should be available at specific locations

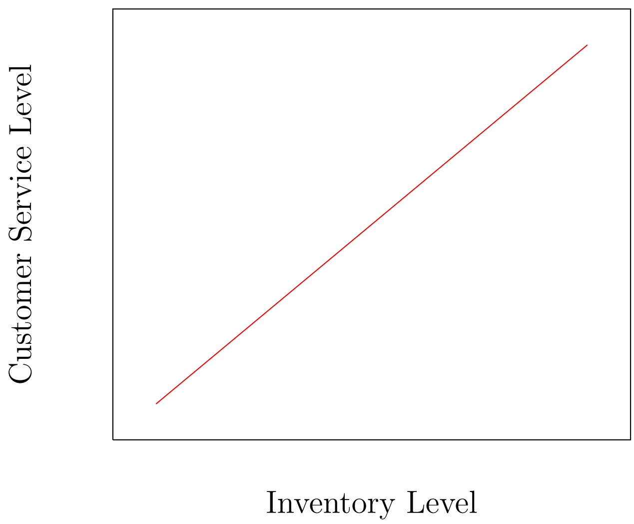


Figure 6.1: Relationship between inventory level and customer service level.

Inventory decision-making usually has a major impact on issues regarding cost and customer service requirements. There is a general relationship between the amount of inventory in stock and customer service levels as illustrated in Figure

6.1. It highlights the fact that it is often necessary for a business to increase its investment in inventory before it will be able to achieve a desired customer service level.

The key to successful inventory management is balancing the supply of inventory with the demand for inventory (Coyle *et al.*, 2002). It would be ideal for a company to have enough inventory to meet the demands of its customers without losing any sales due to inventory stockouts. On the other hand, the company does not want to have too much inventory on hand because of the cost of carrying inventory.

6.1.2 Description of a Basic Inventory Problem

Inventory problems are used to model a basic inventory management system. Inventory problems contain a vendor, or a similar type of agent, that needs to supply a number of customers with a single product or commodity. There are often variabilities associated with the customer arrival rate and individual demands. Stochastic lead times – the time it takes between placing an order and the inventory replenishment – must also be taken into account.

There are costs associated with keeping products in stock (holding costs) and a fixed cost is incurred every time an order is placed. In some inventory problems backorders are allowed whenever a product runs out of stock. These backorders however come at a higher cost for the vendor than normal orders. These costs are typically referred to as shortage costs (Iglehart, 1963).

The main responsibility of the vendor in inventory problems is to manage the inventory level in order to keep the total inventory cost as low as possible. Backorder costs or shortage costs are applied to penalize the vendor from having a low service level as a result of running out of stock. Keeping a sufficient inventory level, yet minimising the inventory cost, are the key factors that the vendor needs to keep into consideration. The vendor manages the inventory level by applying a specific replenishment strategy. The replenishment strategy specifies the inventory level at which the vendor needs to place a new order, as well as the number of units that needs to be ordered.

6.2 Variations on the Basic Inventory Problem

There are several variations to the basic inventory problem described in 6.1.2. A few of these variations will be discussed in this section.

6.2.1 Deterministic Inventory Problems

In deterministic inventory problems it is possible to make optimal inventory decisions because the demand is known in advance. The classic economic order quantity (EOQ) model was developed for these cases by Harris (1913). In an EOQ model a fixed order quantity is automatically ordered once the inventory drops below a predetermined level, the *reorder point*. This approach is sometimes referred to as a *two-bin system* (Coyle *et al.*, 2002). When the inventory in the first bin is empty, the firm places an order. The amount of inventory in the second bin represents the number of items required until the new order arrives. This implies that the firm will reorder or produce stock when the amount of inventory on hand decreases to some predetermined level.

6.2.1.1 Basic EOQ Model

In the basic EOQ model the demand is deterministic and occurs at a constant rate. There is also a zero lead time for each order. Shortages and backlogged inventory are not allowed. The economic order quantity (EOQ), which minimises the total cost can be calculated with

$$q^* = (\frac{2KD}{h})^{\frac{1}{2}} \tag{6.1}$$

where D is the number of units demanded per year, K is the order cost and h is the inventory holding cost per unit per year.

6.2.1.2 EOQ Model with Quantity Discounts

Suppliers sometimes reduce the unit purchasing price when large orders are placed. These price reductions are known as quantity discounts. The order points where a price change occurs is referred to as price break points, $b_1, b_2, ..., b_{k-1}$. Winston (2004) suggests that beginning with the lowest price, the order quantity that minimises the total annual costs for each price should be calculated, e.g. $b_{i-1} \leq q_i^* < b_i$. $q_k^*, q_{k-1}^*, ...$ should be calculated until one of the q_i^*'s is admissible. This will mean that $q_i^* = EOQ_i$. The optimal order quantity will therefore be the member of $q_k^*, q_{k-1}^*, ..., q_i^*$ with the smallest total cost.

6.2.1.3 Continuous Rate EOQ Model

In a company where goods are internally produced, it is not always possible to produce, for example, 5 000 trucks in one instance. Therefore, it is not always appropriate to work with the traditional EOQ model where the assumption is that each order arrives at the same instance. The Continuous Rate EOQ Model is more appropriate to use in these scenarios. The demand is still assumed to be deterministic and shortages are still not allowed. It is assumed that a firm can produce goods at a rate of r units per time period. In any period of length t, the firm can produce rt units. Winston (2004) defines that the optimal run size, which is equal to the economic order quantity, can be calculated using

$$EOQ = (\frac{r}{r-D})^{\frac{1}{2}} \tag{6.2}$$

where D is the annual demand for the product.

6.2.1.4 EOQ Model with Backorders

Inventory shortages are often a reality, and there are costs associated with not being able to meet demand on time. Sometimes these costs can be directly calculated, for example when a sale is lost or backorders are made at a premium price. Shortage costs can however take the form of a loss of future goodwill which cannot be calculated so easily. In an EOQ model where backorders are allowed and no sales are lost, let s be the cost of being one unit short for one period of time. We define q as the order quantity and $q - M$ as the maximum shortage that occurs under an ordering policy. The optimal order policy, where the total cost is minimised, can be determined with

$$\begin{aligned} q^* &= (\frac{2KD(h+s)}{hs})^{\frac{1}{2}} \\ M^* &= (\frac{2KDs}{h(h+s)})^{\frac{1}{2}} \\ \text{Maximum shortage} &= q^* - M^*. \end{aligned} \tag{6.3}$$

6.2.2 Stochastic Inventory Problems

In the inventory problems described in 6.2.1 conditions of certainty existed with regards to the demand and the lead time. This is however quite unrealistic and does not represent the usual operating situation for most firms. Coyle *et al.* (2002) mentions a few factors which could lead to the demand and lead time being uncertain, or stochastic:

Demand variations The demand for a product could vary depending on weather, social needs, psychological needs and many other factors.

Order processing time variations Order processing is not necessarily always a smooth process. Problems with order systems, or even poor corporate governance could create undesirable backlogs.

Transit time variations The lead time can also be influenced by transit time variations as a result of traffic, breakdowns and general delays.

Damage Inventory lost in transit or damaged could result in a stockout situation.

Some examples of inventory problems where stochastic demand and lead times are involved are discussed next.

6.2.2.1 The News Vendor Problem

Inventory problems where q is the predetermined order quantity, a demand of d units occurs with probability $p(d)$, and a cost $c(d, q)$ is incurred, are often called news vendor problems (Winston, 2004). A vendor must decide how many newspapers to order each day from the newspaper plant. If he orders too many papers he will be left with worthless papers at the end of the day. If he does not order enough he will run out of stock, lose out on possible profit, and disappoint customers. Marginal analysis can be used to solve these problems. If it is assumed that $F(q) = P(D \leq q)$ is the demand distribution function, it can be shown that

$$F(q^*) \geq \frac{c_u}{c_o + c_u} \tag{6.4}$$

where c_u is the *understocking cost* and c_o is the *overstocking cost*.

6.2.2.2 EOQ Model with Uncertain Demand and Lead Time

Determining the optimal order strategy for an EOQ model where the demand and the lead time are random will now be discussed. For this model it is assumed that demand can be backlogged. In addition to the earlier variable definitions, the following is also defined:

$D =$ the annual demand, with mean $E(D)$, variance var D and standard deviation σ_D

$c_B =$ the cost incurred for each unit short

$X =$ the demand during lead time, with mean $E(X)$, variance var X and ‘ standard deviation σ_X

Winston (2004) shows that the expected cost is minimised by q^* and r^* given by

$$\begin{aligned} q^* &= (\frac{2KE(D)}{h})^{\frac{1}{2}} \\ P(X \geq r^*) &= \frac{hq^*}{c_B E(D)}. \end{aligned} \tag{6.5}$$

If we assume that demand cannot be backlogged, thus all stockouts result in lost sales, the reorder point is calculated differently. Let c_{LS} = the cost incurred for each lost sale. The expected cost is minimised by q^* and r^* given by

$$\begin{aligned} q^* &= (\frac{2KE(D)}{h})^{\frac{1}{2}} \\ P(X \geq r^*) &= \frac{hq^*}{hq^* + c_{LS}E(D)}. \end{aligned} \tag{6.6}$$

6.2.2.3 Other Stochastic Inventory Problems

There are many other examples of stochastic inventory problems. Clark & Scarf (1960) investigate how the inventory policies for a *multi-echelon inventory system* can be optimised. This system consists of a supply chain containing several entities. Different lead times and stock in transit are also taken into account in this model. The Clark-Scarf model analyses the system under periodic review, whereas Bodt & Graves (1985) introduce continuous-review control policies. Chen (2000) generalises the Clark-Scarf model to derive optimal policies for multi-echelon systems where the materials flow in fixed batches between the entities.

Baker & Urban (1988) evaluate an interesting inventory system in which the demand rate of the product is a known function of the inventory level. The model assumes that the probability of making a sale decreases as the inventory level decreases, thereby lowering the "attractiveness" of the product for the customer. Giri & Chaudhuri (1998) extend the work of Baker and Urban to include a *deterioration function* to model perishable items.

6.2.3 MIT Beer Game

The MIT Beer Game is a simulation game developed by the Systems Dynamic Group at the Massachusetts Institute for Technology, under the guidance of Forrester (1999). It has subsequently been used in numerous undergraduate and postgraduate courses to demonstrate key principles of inventory management across a supply chain.

6.2.3.1 Description

The game involves a simple production and distribution system for a single brand of beer. There are four players in the supply chain, which are sequentially arranged:

- Manufacturer
- Distributor
- Wholesaler
- Retailer

Beer is produced by the manufacturer and delivered downstream through the supply chain until it reaches the external customer. Each of the four players has to fulfill incoming orders of beer by placing orders from its next upstream supplier. The players are not allowed to communicate with each other. The only information they are allowed to exchange is the order amount. There is no transparency as to what the other players' inventory levels or actual customer demand is, and only the retailer knows the external demand of the customer.

Each player incurs both inventory holding costs, and penalty costs for backlogged items. The primary aim for each player is to keep his total inventory cost as low as possible. The optimal strategy is therefore to keep the inventory as low as possible without running into backlog.

The forecast-driven distribution system found in the Beer Game leads to an oscillation demand magnification developing upstream. This phenomenon is known as the Bullwhip Effect and is caused by three underlying problems:

A lack of information Due to the fact that only order amounts are conveyed upstream through the supply chain, information about the customer demand is quickly lost. With no actual customer data available, forecasting relies only on incoming orders at each player.

The supply chain structure The structure of the supply chain prolongs the lead time. It therefore takes longer for an order to travel upstream and the subsequent delivery to travel downstream, which aggravates the Bullwhip Effect.

A lack of cooperation Each agent tries to optimise his inventory costs locally. Some agents employ batch ordering which also aggravates the Bullwhip Effect, because very little can be derived about actual customer demand from such orders.

6.2.3.2 The Use of Agents in the Beer Game

Kimbrough *et al.* (2002) investigated the effectiveness of computers, in the form of artificial agents, in playing the Beer Game. They replaced human players with artificial agents playing the Beer Game to see if artificial agents could learn and discover good and efficient order policies in supply chains. By learning rules via genetic algorithms it was found that individual agents were able to find fully co-operative solutions, even in stochastic situations. The artificial agents were even capable of better performance than undergraduate and postgraduate students.

6.3 Inventory Policies

In all of the inventory problems defined in 6.2.1 and 6.2.2 the assumption was made that an order could be placed at exactly the time that the inventory level reached the reorder point r. A reorder quantity q was also determined. This is known as an *(r, q) Continuous review policy.*

Suppose that a demand occurs for more than one unit at a specific time. It may then happen that an order could be triggered at an inventory level which is lower than the reorder point r. For example, assume that the current inventory level is 30 and $r = 25$. If a customer demand for 10 units arrives, the order will only be placed once the inventory level is 20. The calculations for the reorder quantity will not necessarily minimise the annual cost. It may even be possible for the order replenishment to fail to raise the inventory level above the reorder point.

In situations where the demand is often greater than one unit it has been shown that an *(s, S) Continuous review policy* is optimal (Arrow, 1958) and (Scarf, 1960). In this policy an order is placed once the inventory level is less than or equal to s. The size of the reorder is calculated to ensure that the inventory level will be raised to S, assuming zero lead time. If a $(25, 50)$ policy were to be implemented for example, and the inventory level were to suddenly drop from 30 to 20, an order would be placed for $50 - 20 = 30$ units.

Iyer & Schrage (1992) show historical demand data can be used to directly generate inventory control parameters for use in future inventory control. Instead of fitting a distribution model to the historical demand data, they provide a polynomial time algorithm to solve the (s, S) inventory problem.

6.4 Formulation of the Inventory Problem used in this Study

An inventory problem with an (s, S) review policy was used in this study to investigate the feasibility of applying an agent-oriented approach to multi-objective optimisation. The formulation, assumptions, and parameters of this specific inventory problem will be discussed in this section.

A single, discrete commodity is sold to customers who arrive according to a Poisson process, with a mean arrival rate of $\lambda = 2$. The interarrival times of the customers are thus exponentially distributed with parameter $\beta = 0.5h$. It is further assumed that the demand of customer i is distributed $\lfloor 20 \cdot beta(2, 1) \rfloor$ and the order lead time is $unif(1h, 2h)$. The following notation applies:

$$\begin{aligned}
I_t &= \text{Inventory level at time } t \text{ when customer } i \text{ arrives} \\
I_0 &= \text{Starting inventory at time } 0 \\
S &= \text{Required inventory level after reordering} \\
s &= \text{Reorder point} \\
SL &= \text{Service level} \\
ST_i &= \text{Stockout experienced by customer } i \\
D_i &= \text{Demand of customer } i \\
N_C &= \text{Number of customers arriving in period } T \\
i &= \text{Customer number at time } t
\end{aligned}$$

The service level is calculated with

$$SL = \frac{\sum_{i=1}^{N_C}(D_i - |ST_i|)}{\sum_{i=1}^{N_C} D_i} 100\%. \tag{6.7}$$

Stockouts are calculated using

$$ST_i = \min(0, I_t - D_i). \tag{6.8}$$

Further assumptions:

1. The system operates for 8 hours per day, *i.e.* $T = 8h$. The inventory at the end of the day is carried over to the following day.
2. The initial inventory is $I_0 = 100$ units
3. No backlog is allowed in this system. If $D_i > I_t$ and $I_t > 0$, the customer takes I_t units and after that I_t becomes 0. If $I_t = 0$ and a customer arrives, I_t remains 0, but the stockout is adjusted according to (6.8). When the replenishment quantity arrives then $I_t \leftarrow I_t + S$.
4. The following costs apply:
   (a) The cost to reorder S items is R100/order.
   (b) The overall holding cost per item is R10/item/day.

This formulation will be used for further analysis.

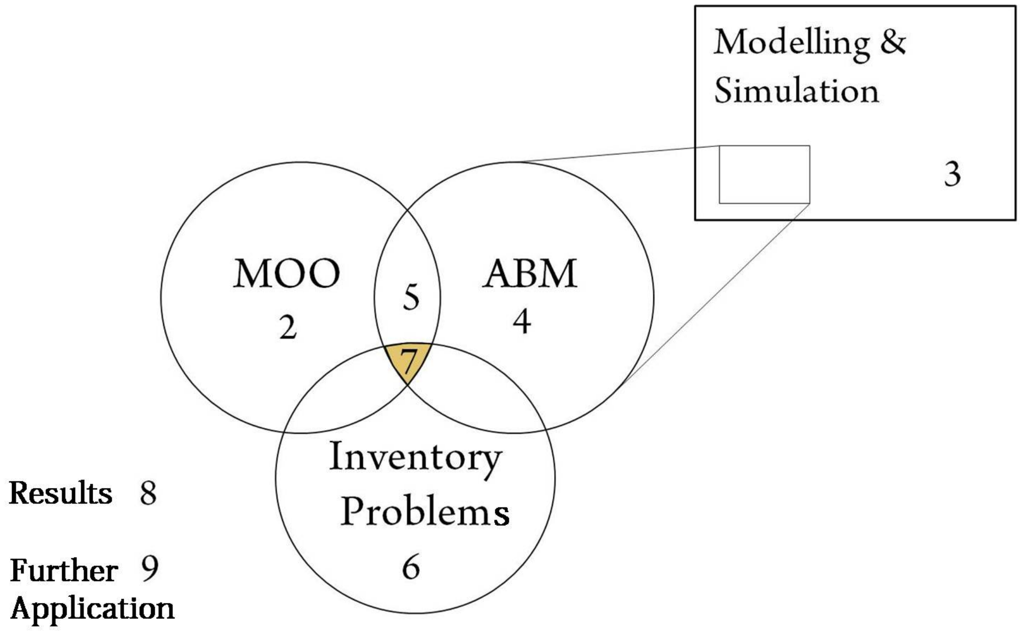
6.5 Concluding Remarks: Chapter 6

Several examples of inventory problems were described in this chapter. The formulation of the inventory problem to be used in this study was also given.

In the next chapter the application of ABM and MOO to the inventory problem will be described.

Chapter 7

Application of ABM and MOO to the Inventory Problem



The inventory problem formulated in Section 6.4 presents a good opportunity for the application of multi-objective optimisation by making use of agent-based modelling. In this chapter an overview will be given into the agent-based model developed for this purpose. The technical design of the model will be described in detail.

7.1 Model Overview

In Section 6.4 an inventory problem was formulated describing a system whereby a single, discrete commodity is sold to customers. The customers arrive according to a Poisson distributed arrival rate, each having a specific demand for the commodity. The formulas for determining the inventory cost and service level were also given.

The vendor in this inventory problem is faced with two conflicting objectives. He needs to keep the inventory level as low as possible to minimise the total inventory cost. However, he also needs to keep enough inventory in stock to be able to maintain a high service level. Therefore, in order to ensure successful inventory management, the vendor's inventory replenishment strategy needs to be optimised by making use of multi-objective optimisation.

The nature of this inventory problem lends itself to apply an agent-based modelling approach to the multi-objective optimisation. For the purpose of this study it is assumed that there are two agents, each with its own agenda relating to each of the objectives of the vendor. These agents, an inventory manager and a sales manager, are employed by the vendor to assist him with finding the ideal inventory replenishment strategy. The inventory manager is responsible for keeping the inventory costs as low as possible. The sales manager on the other hand tries to keep the customers happy by ensuring a high service level. These two managers are considered to be intelligent agents, able to make informed decisions based not only on facts, but also emotion. They negotiate with each other to come up with an inventory replenishment strategy that will meet the needs of the business.

The negotiation process between the inventory manager and sales manager is driven by each of their emotional states, represented by *satisfaction indexes* in the model. For example, if the last few replenishment strategies that had been applied resulted in extremely high inventory costs, the inventory manager will become very dissatisfied. The next time they need to decide on a new strategy the inventory manager will be quite aggressive in his negotiation and will strive to lower the re-order quantity and re-order point quite significantly. Likewise, if the sales manager is used to being able to achieve a high service level, but this has suddenly taken a dip, he will have a high incentive to ensure that the service level improves again.

The human factor involved in decision making is modelled by making use of these intelligent agents. The feasibility of the agent-based approach to multi-objective optimisation can be explored in this model by experimenting with different scenarios and agent logic.

7.2 Model Platform

In Section 4.4 a list of modelling platforms available for agent-based modelling was given. *Anylogic* (`www.anylogic.com`, cited on 18 August 2012) was selected as the platform most suitable for this inventory problem. Anylogic was developed by a Russian software company, XJ Technologies. One of the unique features of Anylogic is that it allows the user to combine various different simulation modelling paradigms in the same model. The four paradigms that Anylogic facilitates are Discrete-Event, Agent-Based, System Dynamics, and Dynamic Systems. Figure 7.1 provides an overview of these paradigms.

Anylogic's multi-method functionality provides a high degree of flexibility during model development as different processes within the same system can often be more effectively modelled by different paradigms. The agent-based approach in Anylogic can be applied without difficulty by making use of state diagrams. An example of an agent's state diagram is given in (Figure 7.2) where each block represents a state through which the agent loops during the model.

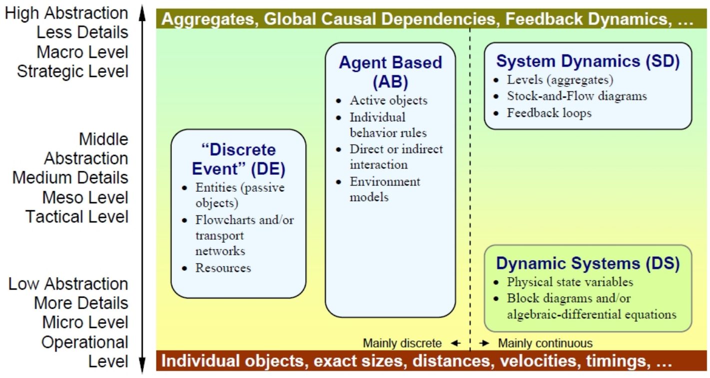


Figure 7.1: Paradigms in simulation modelling on an abstraction level scale (Borshchev & Filippov, 2004).

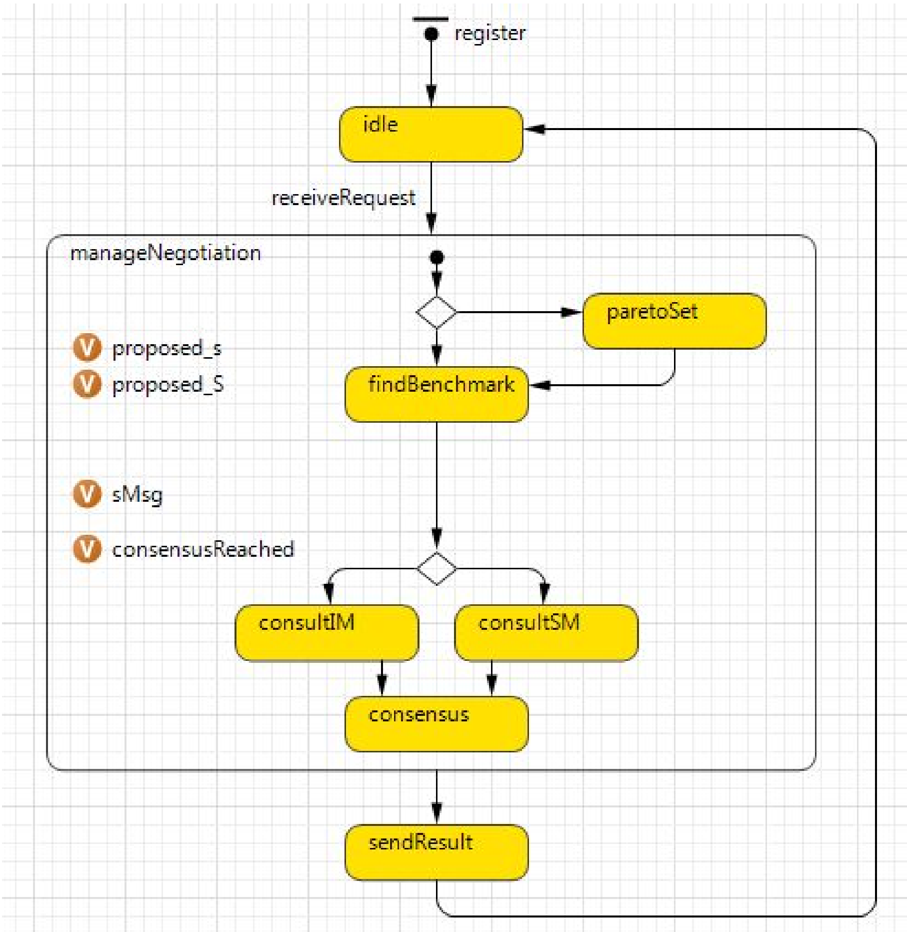


Figure 7.2: State diagram for an agent in Anylogic.

7.3 Technical Design

The inventory model developed in Anylogic consists of different objects interacting with one another. The interaction between the customers and the vendor forms part of the business transaction taking place when the customer arrives at the store. The vendor periodically interacts with his inventory manager and sales manager, which are intelligent agents, to review and make changes to the current inventory replenishment strategy. All these interactions are controlled by state diagrams, representing the decision making functions of the objects, and by messages which are transmitted between the objects.

7.3.1 Customer and Vendor Interaction

The model is driven by the periodic arrival of customers at the vendor's business. Each of these customers have a specific demand for the vendor's merchandise. If the vendor has enough inventory on hand to meet the customer's demand, the transaction is completed and the merchandise is handed over to the customer who leaves the store satisfied. However, if the number of units in stock is lower than the demand, the last few units are sold to the customer and the remainder of the demand is treated as a lost sale, impacting the service level of the vendor. The possibility also exists that the vendor experiences a complete stockout during which the customer cannot be serviced at all.

7.3.2 Inventory Replenishment

An inventory replenishment strategy governs all inventory related decision-making of the vendor. The vendor checks the inventory level after every customer transaction. If the inventory level is equal to or lower than the reorder point specified by the current replenishment strategy, and an order for new merchandise has not yet been placed, the vendor orders a number of units quantified by

$$\text{OrderSize} = S - I_t \tag{7.1}$$

where S is the reorder quantity of the current replenishment strategy and I_t is the current inventory level. Once an order has been placed there is a lead time before the inventory is available at the vendor's store.

7.3.3 Inventory Replenishment Strategy Review

After a predetermined number of reorders the vendor needs to review and possibly make changes to the current replenishment strategy.

In the model 30 random strategies are initially attempted by the vendor to get a few benchmark points which the inventory and sales managers can use to understand what constitutes a successful strategy. These random strategies are

$$s = \text{round}(200 * U_1) \tag{7.2}$$

$$S = \text{round}(200 * U_2) \tag{7.3}$$

where s is the reorder point, S is the reorder quantity, U_1 and U_2 are uniformly distributed random values between 0 and 1. The value of s needs to be higher than that of S because the reorder quantity cannot be a negative value. These values are returned to the vendor and implemented as the new replenishment strategy.

Once the initial sample of strategies has been populated, the vendor starts to consult with his inventory and sales managers through an *interaction agent* to determine the new replenishment strategies. The interaction agent is a theoretical agent that has been introduced into the model to facilitate the interaction between the vendor and his inventory manager and sales manager. The interaction agent also facilitates the negotiation process between the inventory manager and the sales manager. One of the core responsibilities of the interaction agent is to keep a database of all previous replenishment strategies applied by the vendor. This information is vital to the inventory manager and sales manager to assist them with making informed decisions on future replenishment strategies.

The following process is followed:

1. Populate the *Pareto approximation set*, $\mathcal{P}^*$, containing the most feasible strategies that have been attempted up to this point in time.
2. For each strategy in $\mathcal{P}^*$:
   (a) Make this strategy the *currentBenchmark*.
   (b) Determine the *satisfactionIndex* for both the inventory manager and the sales manager.
   (c) Allow the manager with the lowest satisfaction to choose the new replenishment strategy.
   (d) Implement the new strategy and wait for the next strategy review meeting.
   (e) Determine if the new strategy is now the new *currentBenchmark*.
   (f) Repeat Steps (b) to (e) until 10 new strategies have been attempted.
3. Repeat Steps 1 and 2 for n_{it} iterations, whereafter the replication is complete and the model terminates.

7.3.3.1 Pareto Approximation Set

The interaction agent identifies the *Pareto approximation set* containing all acceptable solutions from the current strategy database. This represents the most feasible strategies that have been attempted up to this point in time. Definitions pertaining to Pareto optimality were presented in Section 2.1 on 6.

7.3.3.2 Satisfaction Index and Recollection Ability

The *satisfaction index* of the inventory manager SI_{IM} and the sales manager SI_{SM} are parameters that represent each of their emotional states and dictates how aggressively they will negotiate during the next strategy review. They have a *recollection ability* ε which means that their satisfaction is dependent on the success of the last few strategies that have been implemented, as well as the current benchmark strategy. This means that the agents will take ε number of negotiation rounds before they disregard a strategy that had a negative effect on their respective objectives.

For example, if the inventory cost of the last few strategies was much higher than the benchmark, the inventory manager will become upset about it. The next time he meets with the sale manager, he will be determined to lower the reorder point and reorder quantity used for the new strategy.

In order to implement these principles, the satisfaction indexes need to be quantified, and the following notation applies:

$$\begin{aligned}
SI_{IM} &= \text{Satisfaction index of the inventory manager} \\
SI_{SM} &= \text{Satisfaction index of the sales manager} \\
SL_{bm} &= \text{Service level of the benchmark strategy} \\
SL_0 &= \text{Service level of the current strategy} \\
SL_1 &= \text{Service level of the previous strategy} \\
SL_\varepsilon &= \text{Service level of } \varepsilon \text{ strategies back} \\
C_{bm} &= \text{Total inventory cost of the benchmark strategy} \\
C_0 &= \text{Total inventory cost of the current strategy} \\
C_1 &= \text{Total inventory cost of the previous strategy} \\
C_\varepsilon &= \text{Total inventory cost of } \varepsilon \text{ strategies back}
\end{aligned}$$

The satisfaction index of the inventory manager is calculated with

$$SI_{IM} = \frac{x\sum_{n=0}^{\varepsilon}(\varepsilon - n + 1)(100)(\frac{C_{bm} - C_n}{C_{bm}})}{\sum_{n=0}^{\varepsilon} n} \tag{7.4}$$

and with

$$SI_{SM} = \frac{\sum_{n=0}^{\varepsilon}(\varepsilon - n + 1)(100)(\frac{SL_{bm} - SL_n}{SL_{bm}})}{\sum_{n=0}^{\varepsilon} n} \tag{7.5}$$

for the sales manager.

The agent with the lowest satisfaction index will have the upper hand during the next strategy review session and will be able to choose the new reorder point and reorder quantity to be implemented. The lower the agent's satisfaction index, the more aggressive he will be in choosing the new strategy. The range of change ς of the new strategy is calculated by a step function shown in Figure 7.3. In Section 8.4.2 an agent aggression factor will be introduced which will scale this step function.

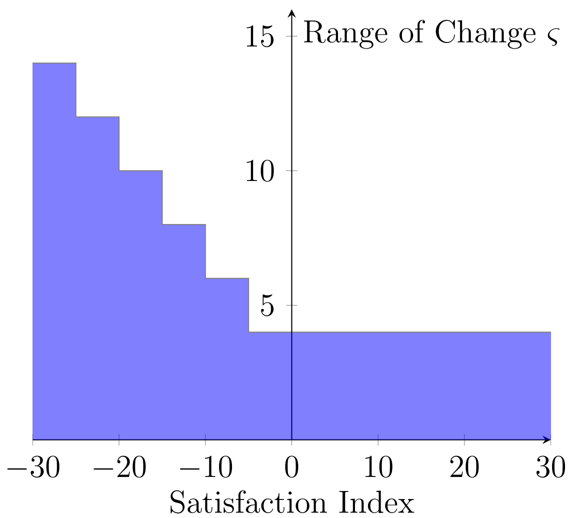


Figure 7.3: Impact of Satisfaction Index on Reorder Strategy.

Two random integer values, i_s and i_S are generated from a uniform distribution between 0 and ς which are then used to increase or decrease the previous strategy's reorder point s' and reorder quantity S' with $s = s' - i_s$ and $S = S' - i_S$ if the **inventory manager** chooses the new strategy, or with $s = s' + i_s$ and $S = S' + i_S$ if the **sales manager** chooses the new strategy.

A practical example of how the satisfaction indexes are used to determine the new strategy will now be given:

1. Assume that the inventory manager's satisfaction index (-12.76) is lower than the sales manager's satisfaction index (2.35). Therefore the inventory manager will be able to choose the new strategy.

2. From Figure 7.3 it can be seen that $\varsigma = 8$.

3. The two random integer values, i_s and i_S are calculated by rounding off uniform$(0, 8)$. Assume that the samples generated are $i_s = 3$ and $i_S = 6$.

4. The inventory manager prefers a lower reorder point and reorder quantity, so the new strategy's parameters are calculated with $s = s' - 3$ and $S = S' - 6$.

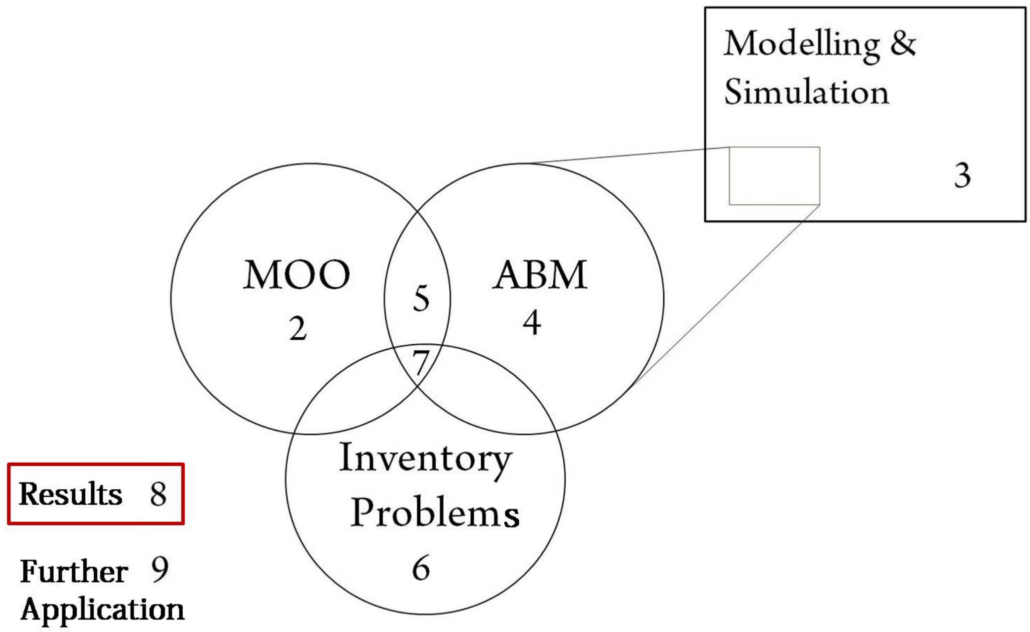
7.4 Concluding Remarks: Chapter 7

The purpose of the inventory model described in this chapter is to determine if it is possible to make use of an agent-based modelling approach to perform multi-objective optimisation. Through the interaction between the inventory manager and sales manager a number of different replenishment strategies will be attempted. The performance of these strategies will indicate if these intelligent agents have the ability to effectively reason between two conflicting objectives, thereby achieving good results with few evaluations.

The technical design of the agent-based inventory model was described in detail in this chapter. An overview was given into the architecture of the model, discussing the role of the customer, vendor, inventory manager and sales manager. In Section 7.3.3 important concepts such as the satisfaction index of the agents were explained. This forms the backbone of the negotiation process between the inventory manager and the sales manager.

Chapter 8

Results and Analysis



The inventory model described in Chapter 7 was developed to determine if it is feasible to apply an agent-based approach to multi-objective optimisation. In this chapter, the performance measures identified for the purpose of this study are discussed. The evaluation method to be used for comparing different scenarios is also described. At least 100 independent replications of various scenarios were run to ensure that the results are statistically plausible. The results from these simulation runs are analysed in this chapter.

8.1 Important Terms

There are a few important terms relating to the model that first need to be clarified in order to interpret the results. Some of the terms may have been used before, but are now formally defined in Table 8.1.

Term	**Description**
Strategy	A specific (s, S) reorder strategy attempted by the vendor.
Negotiation iteration	One iteration is completed every time the *Pareto approximation set* is repopulated, as described in Section 7.3.3.
Scenario	Each "what-if" scenario contains a unique set of parameters which are fixed throughout the run, e.g. Agent aggression.
Replication	100 replications of each scenario is run to ensure that variability in the model is taken into account when the results are analysed.

Table 8.1: Important terms.

8.2 Performance Measures Used to Test the Performance of the Agent-based Approach

The performance of the agent-based approach, and different scenarios relating to it, can be evaluated by making use of a set of metrics discussed in Bekker & Aldrich (2011). The Pareto front created by the replenishment strategies attempted, $\mathcal{P}_K$, and the true Pareto front of the inventory problem, $\mathcal{P}_T$, are required to use these metrics. $\mathcal{P}_T$ was obtained (see Figure 8.1) by running the inventory model for all replenishment strategies where $s = \{1, 2, ..., 200\}$ and $S = \{s, s+1, ..., 200\}$.

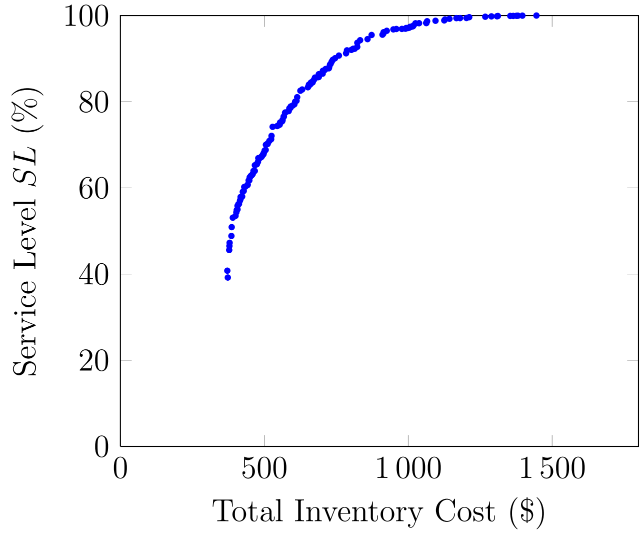


Figure 8.1: True Pareto front for the inventory problem.

The following performance measures are proposed:

1. The hyper-volume indicator I_H (Coello *et al.*, 2007) measures the portion of objective space that is weakly dominated by an approximation set A and is to be maximised. In this document I_H will be referred to as "hyper-area" since the problem presented here has two objectives. The objective space must be bounded, or a *strictly dominated* reference point must be provided. A simple example is shown in Figure 8.2, with maximisation Pareto set $\{(50, 50), (100, 80), (500, 90)\}$, $I_H = 173\,500$ and the reference point at $(2000, 0)$. Intuitively, one can see that the I_H indicator measures spread and proximity ("closeness") of the approximation set to the true Pareto front.

   The hyper-area difference ΔI_H is considered to be the absolute value of the difference between hyper-area indicator for $\mathcal{P}_T$ and the hyper-area indicator for $\mathcal{P}_K$, where the same reference point is used.

2. Generation distance (GD) (Coello *et al.*, 2007), which measures the average distance between $\mathcal{P}_K$ and $\mathcal{P}_T$. It is defined as

$$GD \triangleq \frac{(\sum_{i=1}^{|\mathcal{P}_K|} d_i^2)^{\frac{1}{2}}}{|\mathcal{P}_K|} \tag{8.1}$$

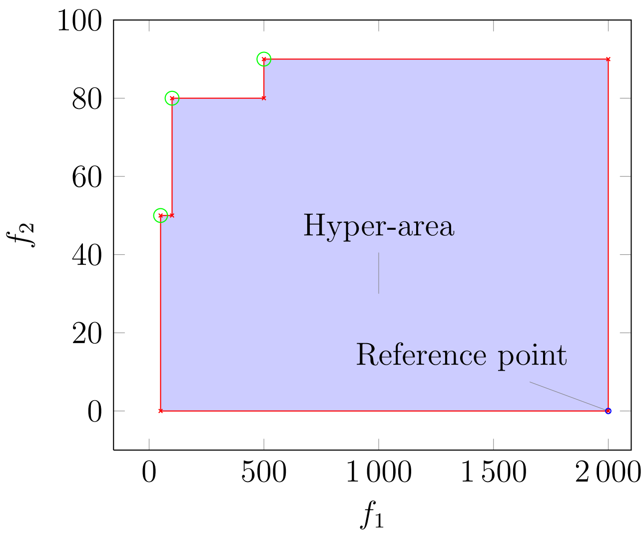


Figure 8.2: Example of a hyper-area and reference point.

with d_i the Euclidean distance between solution value i of $\mathcal{P}_K$ and the closest member in $\mathcal{P}_T$ to the solution i. When $\mathcal{P}_K = \mathcal{P}_T$, then $GD = 0$.

3. Spacing (SP) (Coello *et al.*, 2007), which numerically describes the spread of the vectors in $\mathcal{P}_K$. It is defined as

$$SP \triangleq \sqrt{\frac{1}{|\mathcal{P}_K| - 1} \sum_{i=1}^{|\mathcal{P}_K|} (\bar{d} - d_i)^2} \tag{8.2}$$

and

$$d_i = \min_j \sum_{k=1}^{K} |f_k^i(\mathbf{x}) - f_k^j(\mathbf{x})| \tag{8.3}$$

with $i, j = 1, \ldots, |\mathcal{P}_K|$, K the number of objectives, and $\bar{d}$ is the mean of all d_i. The members of the approximation front are equally spaced if $SP = 0$. The true Pareto front is not required for this test measure.

4. Maximum Pareto front error (ME) (Coello *et al.*, 2007), which measures how well two vector sets conform in terms of shape and distance apart. It

is determined with

$$ME \triangleq \max_j \left\{ \left\{ \min_i \left(\sum_{k=1}^{M} |f_k^i(\mathbf{x}) - f_k^j(\mathbf{x})|^2 \right)^{1/2} \right\} \right\}. \tag{8.4}$$

5. The convergence CV (Deb & Jain, 2002) is a running indicator calculated while executing a MOO algorithm. $\mathcal{F}(t)$ is the non-dominated set of population t. The smallest normalised Euclidean distance from each point i in $\mathcal{F}(t)$ to the true Pareto set $\mathcal{P}_T$ is calculated, with f_k^{max} and f_k^{min} the maximum and minimum function values of objective k in $\mathcal{P}_T$:

$$d_i = \min_{j=1}^{|\mathcal{P}_T|} \sqrt{\sum_{k=1}^{K} \frac{f_k(i) - f_k(j)}{f_k^{max} - f_k^{min}}} \tag{8.5}$$

   Now calculate the convergence CV by averaging the normalised distance for all points in $\mathcal{F}(t)$ with

$$CV = \frac{\sum_{i=1}^{|\mathcal{F}(t)|} d_i}{|\mathcal{F}(t)|}. \tag{8.6}$$

8.3 Scenario Comparison

All the performance measures listed above will be used to evaluate the performance of the agent-based approach. However, Knowles *et al.* (2006) showed that although these performance measures give a good indication of the quality performance of the algorithm, not all of them are suited to comparing algorithms. As recommended by Knowles *et al.* (2006) the *hyper-area difference* ΔI_H will be used for this purpose as it is Pareto compliant and quite simple to estimate.

At least 100 independent replications of each scenario must be run in the model to ensure that the results are statistically significant. ΔI_H will be calculated for each of these replications. It is proposed that *box plots* are used as a visual representation of the results for each scenario. First created by Tukey (1977), box plots are commonly used to display the distribution of a batch of data. The interpretation of a box plot is also relatively straightforward with the configuration shown in Figure 8.3.

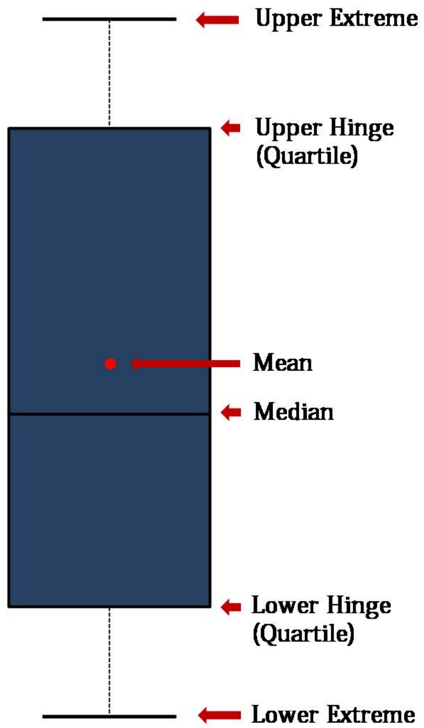


Figure 8.3: Configuration of a box plot.

The box itself represents 50% of the data set. The lower boundary of the box indicates the 25th percentile of the data, and the upper boundary locates the 75th percentile. The median of the data set is indicated by a line within the box. The *whiskers* of the plot indicate where the lower and upper limits of the data set lie.

The different scenarios can be compared by visual inspection of the box plot. The plot will not only show which results can be expected for each of the scenarios, but will also indicate the spread of the results. This allows the researcher to identify which scenarios give the best results. An example of how the box plot will be used in this study is given in Figure 8.4. Note that fictitious data is used in this example. The base scenario – in this case Scenario 2 – will always be highlighted in red.

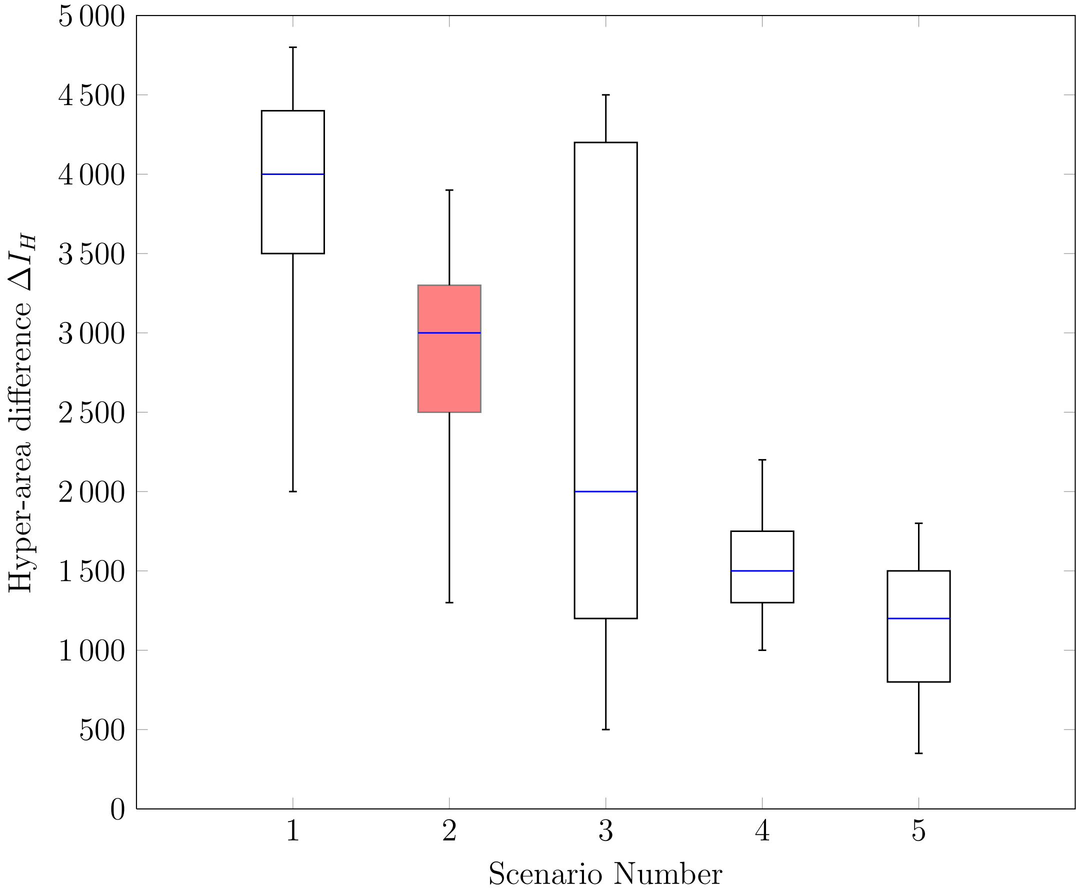


Figure 8.4: Hyper-area indicators for different scenarios (Illustrative).

In this illustrative example it can be seen that Scenario 5 is most likely to yield the lowest hyper-area difference. The mean hyper-area difference for Scenario 3 is lower than that of Scenario 2, but the variability in its results is very high. Therefore, if consistency in the hyper-area difference between replications is of importance then the researcher might prefer Scenario 2 rather than Scenario 3.

The quality indicators to be used for the performance evaluation of the agent-based approach to multi-objective optimisation were identified and described. The suggested approach to do the scenario comparison was also discussed. In the next section these quality indicators will be applied to the results from the inventory model.

8.4 Results from Inventory Model

The detail results from the base case scenario will now be presented. Several other scenarios were also run to determine the sensitivity of some of the parameters pertaining to the agents. The results from these scenarios are also analysed. The chapter ends off with a table summarising the results from all the scenarios.

8.4.1 Base Case

The base case scenario was run with all the default parameters used as described in Chapter 7. The approximate Pareto front obtained from one of the replications from the base case is shown in Figure 8.5.

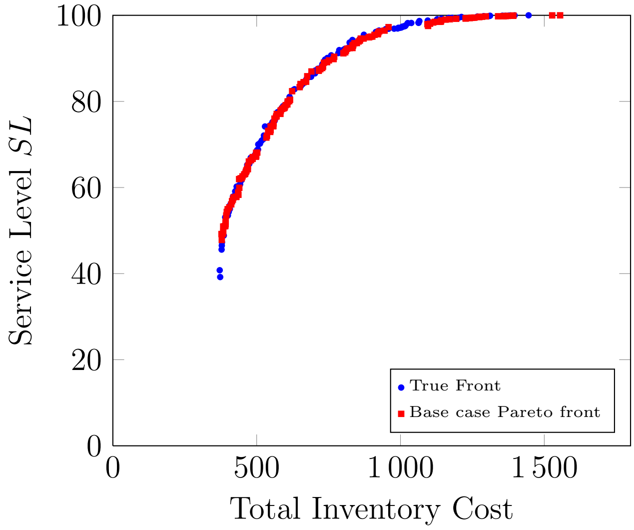


Figure 8.5: Base Case Pareto front.

The visual representation of the Pareto scenarios from the base case shows that the front produced by the agent-based approach is a good approximation of the true front. The focus now turns to the sensitivity of the model to the parameters that define the agents' behaviour. Attributes like the aggression and recollection ability of the agents are adjusted to determine if they have any effect on the performance of the approach. The effect of changes in the number of negotiation iterations is also investigated.

8.4.2 Agent Aggression Sensitivity

The aggression of the agents has a major influence on the negotiation process during the replenishment strategy reviews. The *aggression factor* τ controls how aggressively the inventory manager or sales manager wishes to change the current strategy. A default aggression factor, $\tau = 1$, was utilised in the base case which yielded a range of change ς as shown in Figure 7.3. The default values for ς can be simply multiplied with the new aggression factor to get ς for these scenarios. An example of the ς-plot for $\tau = 4$ is given in Figure 8.6, which now scaled the ς presented in Figure 7.3.

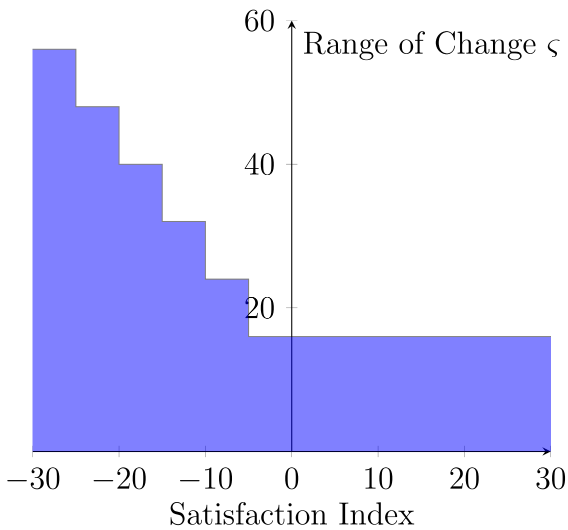


Figure 8.6: Range of Change for $\tau = 4$.

The sensitivity of the model to the agents' aggression was determined by running scenarios for $\tau = \{0.25, 1, 2, 4, 6\}$. The Pareto fronts obtained from a selected representative run for each of these scenarios are shown in Figure 8.7.

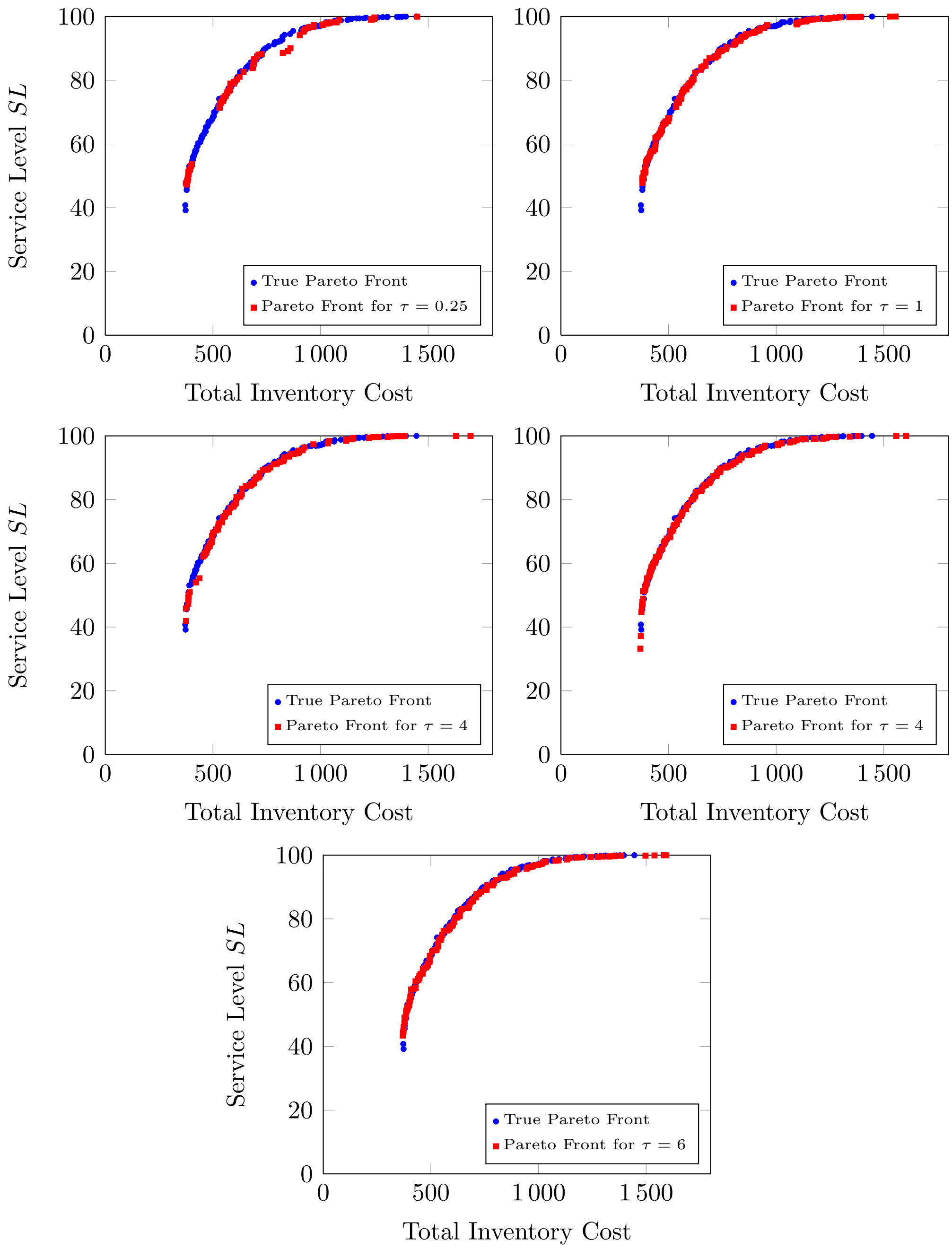


Figure 8.7: Approximated Pareto fronts for various levels of agent aggression.

The different aggression scenarios can be compared by looking at the box plot of the hyper-area differences as shown in Figure 8.8. It is clear that scenarios, where the agents have a higher aggression factor, yield a lower hyper-area difference. It is also very interesting to note that if $\tau \leq 4$, the agent-based approach is able to achieve better Pareto Front approximation sets. A likely explanation for this is that the lower aggression factors did not allow the agents to be persuasive enough in their negotiations with each other.

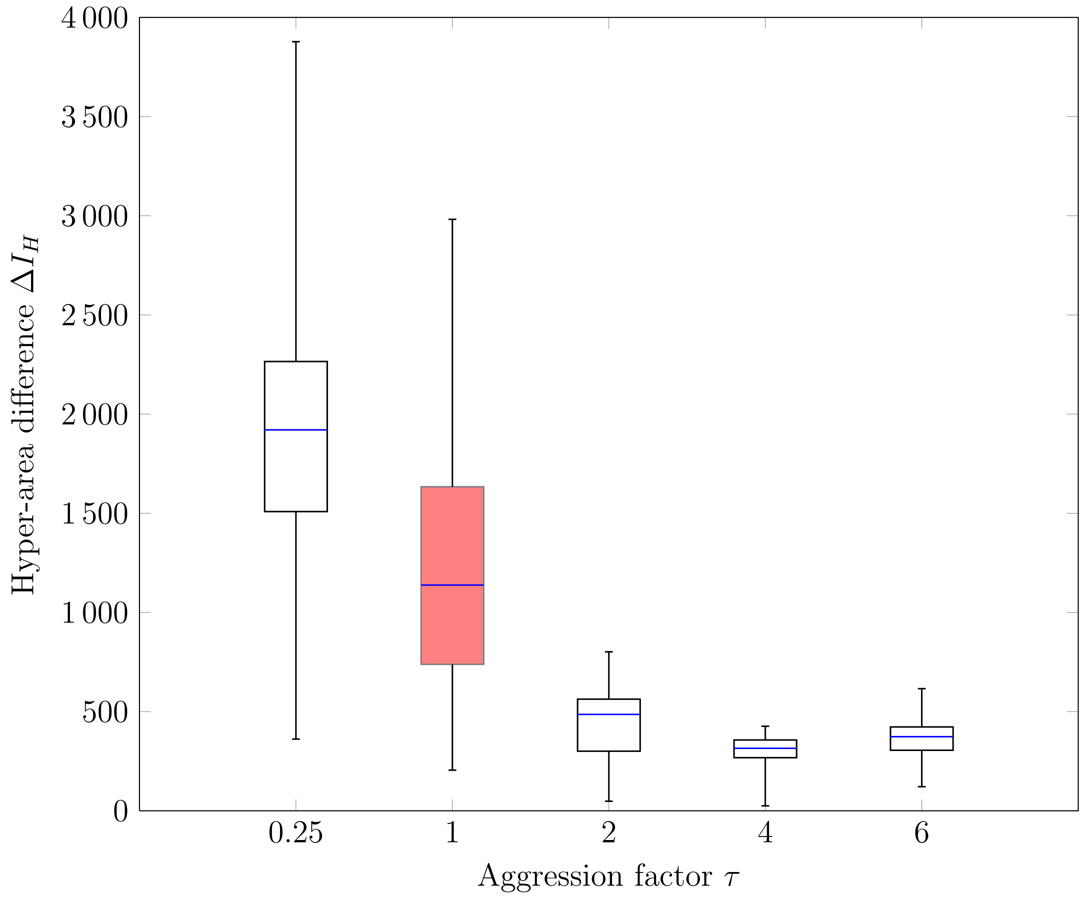


Figure 8.8: Hyper-area difference for aggression sensitivity.

8.4.3 Agent Recollection Ability Sensitivity

The negotiation process between the inventory manager and sales manager is driven by their satisfaction indexes, as described in Section 7.3.3.2. The agents' recollection ability ε is a measure of their ability to also take their satisfaction during the last few strategies into account when choosing a new strategy. Scenarios have been run for $\varepsilon = \{0, 2, 3, 5\}$.

The Pareto fronts obtained from a selected representative run for each of these scenarios are shown in Figure 8.9.

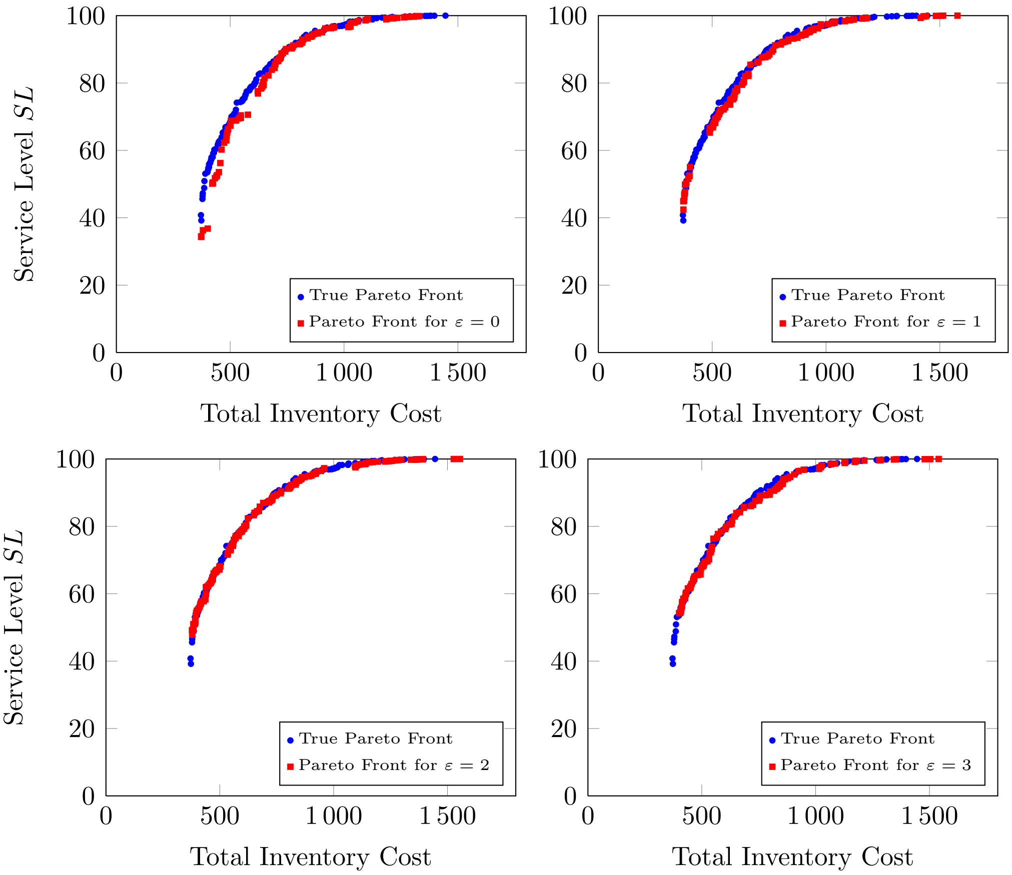


Figure 8.9: Approximated Pareto fronts for various levels of agent recollection ability.

In Figure 8.10 one can see that the model performs slightly better in the scenarios where $0 < \varepsilon < 1$. Therefore, the agents' ability to recall the outcome of the previous strategies does assist them in choosing more optimal strategies in the future. There is however no significant difference in the performance of the scenarios where $\varepsilon = \{1, 2, 3\}$.

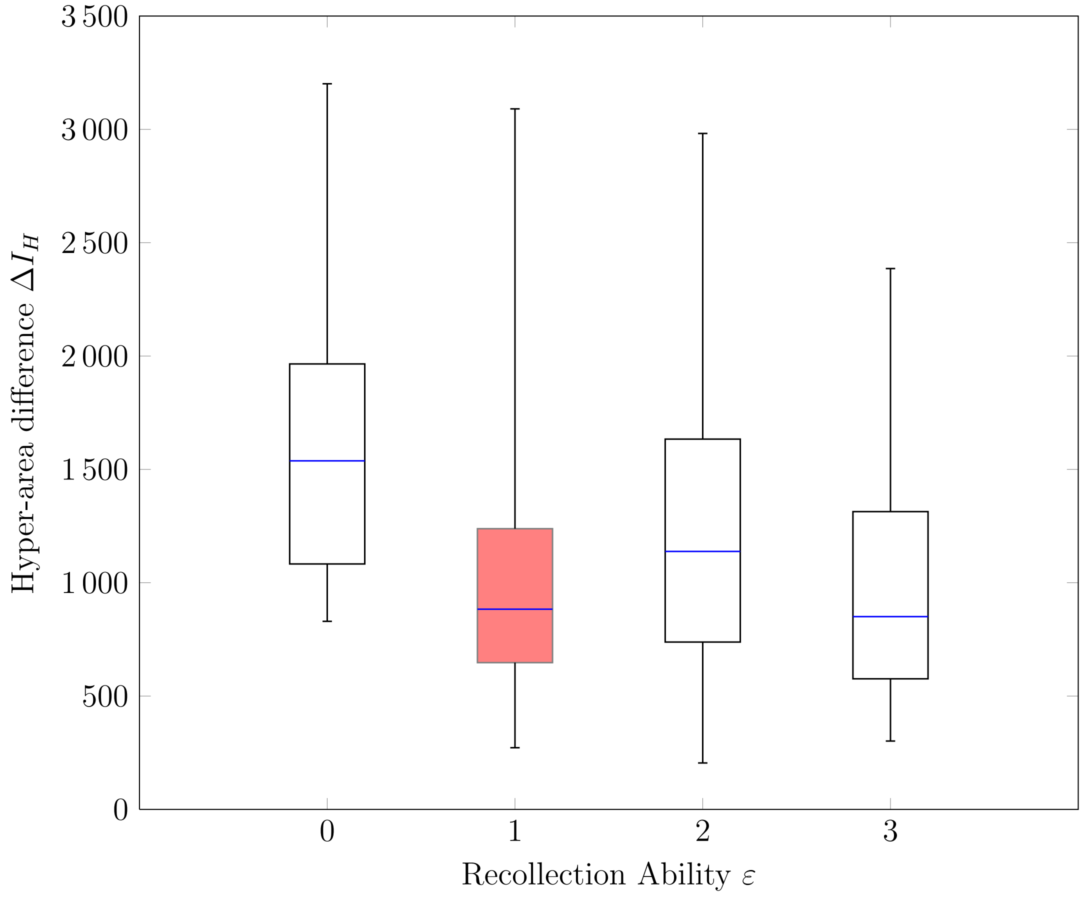


Figure 8.10: Hyper-area difference for recollection sensitivity.

8.4.4 Negotiation Iteration Sensitivity

The number of negotiation iterations n_{it}, as described in Section 7.3.3, and defined in Table 8.1, determines how many times the Pareto approximation set is repopulated. In the base case $n_{it} = 3$, which means that the agents will negotiate through three Pareto approximation sets before the model terminates. The sensitivity of the number of negotiation iterations was determined by running scenarios for $n_{it} = \{1, 2, 3, 4, 5\}$.

The Pareto fronts obtained from a selected representative run for each of these scenarios are shown in Figure 8.11.

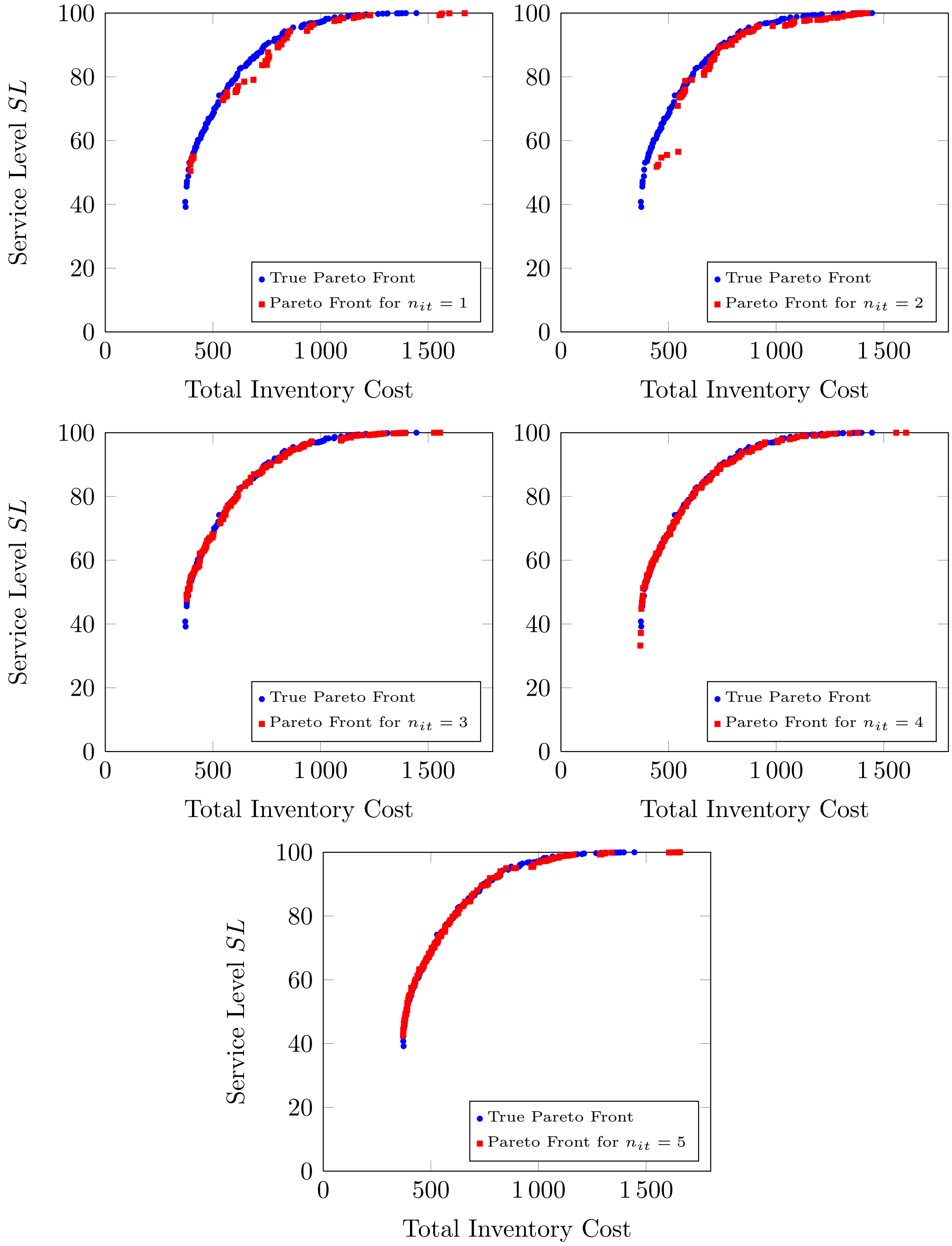


Figure 8.11: Approximated Pareto fronts for various levels of negotiation iteration sensitivity.

From Figure 8.12 it is clear that the number of negotiation iterations plays a big role in the performance of the approach. If $n_{it} = 1$ the agents do not have enough time to learn from previous strategies to improve their performance. More negotiation iterations are required in order to achieve a better PF approximation.

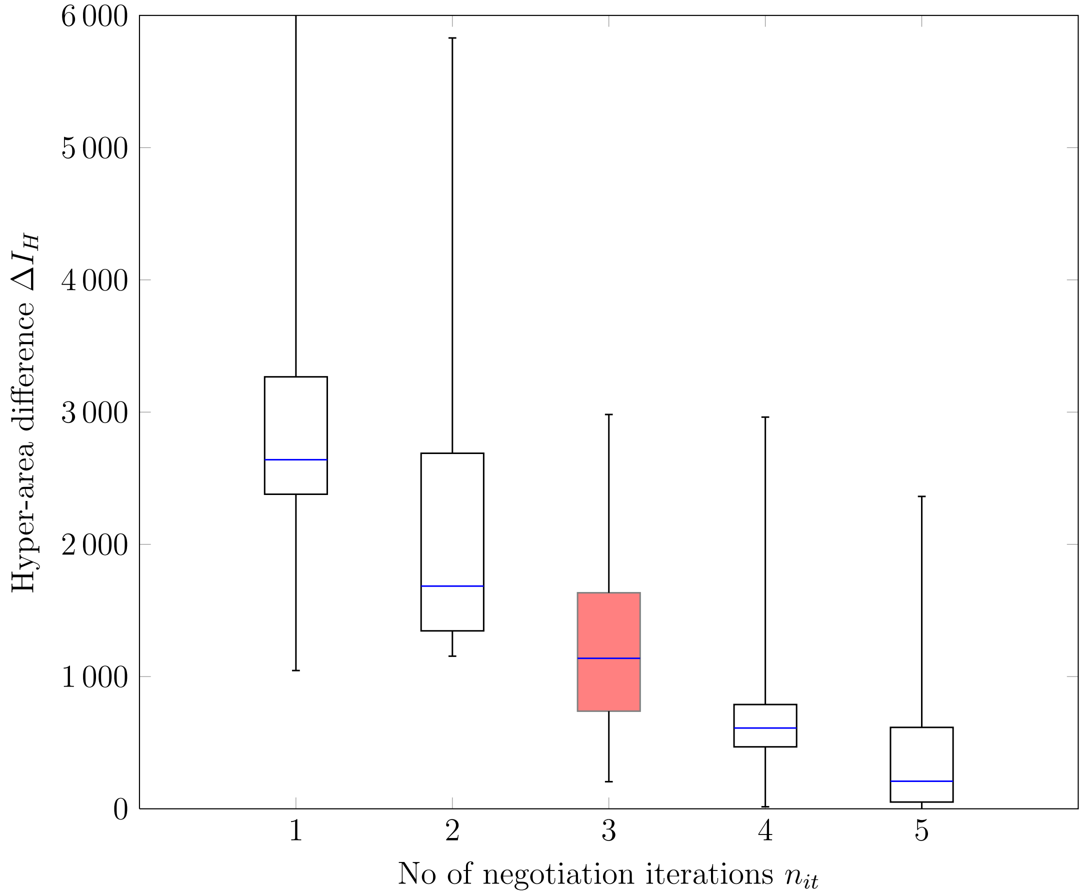


Figure 8.12: Hyper-area indicators for negotiation iteration sensitivity.

It must however be noted that as n_{it} increases, the number of strategies attempted during each run also increases (see Figure 8.13). A higher number of strategies attempted will naturally lead to better approximate Pareto fronts.

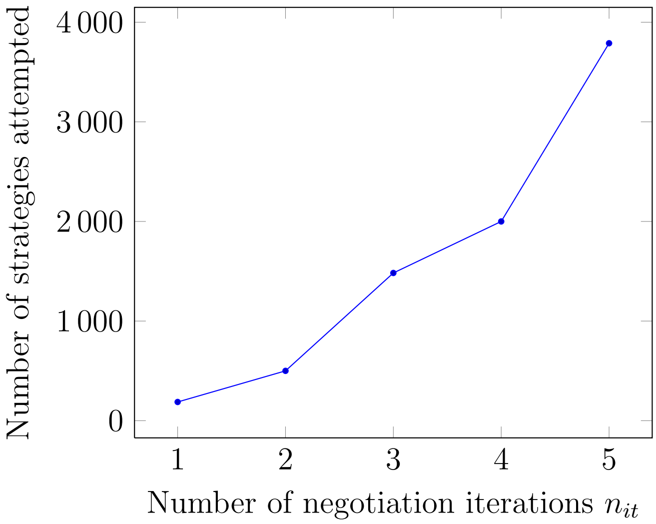


Figure 8.13: Average number of strategies attempted in negotiation iteration sensitivity scenarios.

8.4.5 Summary Results

A summary of the quality indicator values obtained from the simulation runs are shown in Table 8.2.

Scenario Description	Parameter Changed	No of Replications	ΔI_H	GD	SP	ME	CV	No of Strategies
Base Case	None	100	1 210.417	2.257	17.643	84.035	0.058	1456.4
Aggression sensitivity	$\tau = 0.25$	100	1 991.506	3.506	21.804	91.652	0.091	1335.5
Aggression sensitivity	$\tau = 2$	100	447.698	0.000	0.000	0.000	0.000	1501.5
Aggression sensitivity	$\tau = 4$	100	290.422	0.784	15.085	52.164	0.040	1663.5
Aggression sensitivity	$\tau = 6$	100	365.229	0.715	14.363	38.239	0.050	1606.5
Recollection sensitivity	$\varepsilon = 0$	100	1 618.748	2.233	17.198	80.340	0.065	1432
Recollection sensitivity	$\varepsilon = 1$	100	1 139.259	2.256	16.320	89.486	0.057	1436
Recollection sensitivity	$\varepsilon = 3$	100	994.557	1.959	16.716	82.962	0.047	1529
Iteration sensitivity	$n_{it} = 1$	100	2 937.392	6.603	27.633	105.158	0.179	187.5
Iteration sensitivity	$n_{it} = 2$	100	2 273.456	3.501	18.058	107.472	0.088	675
Iteration sensitivity	$n_{it} = 4$	100	788.274	1.359	14.821	64.563	0.040	2503
Iteration sensitivity	$n_{it} = 5$	100	426.737	1.082	13.591	72.596	0.020	3789

Table 8.2: Summary results from all scenarios.

8.5 Concluding Remarks: Chapter 8

The purpose of this chapter was to present the results and findings from the simulation runs assessing the various agent parameters. A few important concepts required to interpret the results were first explained in this chapter. A set of performance measures was identified and applied to evaluate the output of the runs. The base case scenario, in which all the model parameters were set to all its default values, indicated that the agents have the ability to find good strategies. The front obtained from the base case showed that the agent-based approach can create a good approximation to the known front.

Other scenarios were also run to determine the sensitivity of the approach to the parameters that define the agents' behaviour. The following parameters were investigated:

- Aggression factor τ of the agents – a measure of how aggressive they are during negotiations
- Recollection ability ε of the agents – a parameter determining how many previous strategies have an influence on the satisfaction of the agents
- Negotiation iteration sensitivity n_{it} – the number of times the Pareto approximation set is repopulated during each run

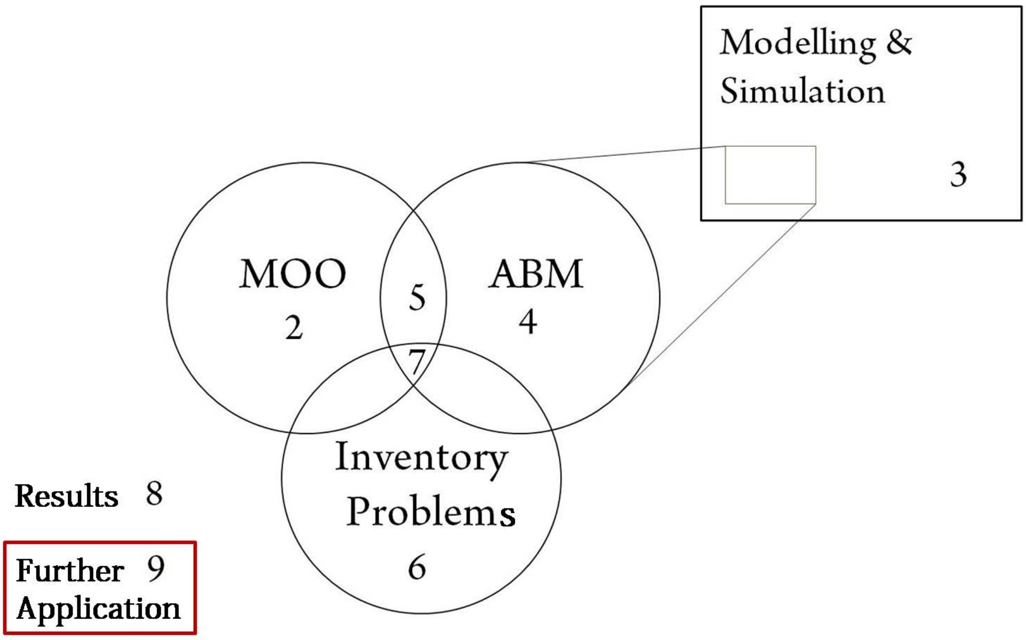
It was revealed that the aggression factor τ, which is an important feature of the negotiation process between the inventory manager and sales manager, has a significant influence on the performance of the model. Higher aggression factors enabled the agents to come up with better strategies more consistently. The default aggression factor applied in the base case did not give the agents enough persuasive capacity, which resulted in somewhat weak negotiation skills.

The recollection ability ε of the agents does assist them to find better strategies. The scenario where the agents only consider their current satisfaction resulted in a slightly worse performance than the other scenarios. Although it is recommended to introduce recollection ability to the agents, the length of the recollection ability does not have any significant impact on the outcome.

The negotiation iteration sensitivity n_{it} scenarios revealed that a higher number of negotiation iterations produces much better strategies. This indicates that the agents need to go through enough negotiations so that they can learn from the previous strategies attempted. It was highlighted though that more strategies are attempted in the scenarios with higher negotiation iterations.

Chapter 9

Further Application of ABM and MOO



The purpose of this study is to determine if an agent-based modelling approach can be used as a metaheuristic for multi-objective optimisation. The inventory problem was identified and described in Chapter 6 as a suitable subject area to which this approach could be applied. The technical design of the agent-based model developed for the inventory model was described in Chapter 7. The results and findings from the model in Chapter 8 proved that it is possible to apply an agent-based approach to optimise multiple objectives in an inventory problem.

However, applying the approach successfully to a single problem does not necessarily imply that it will work under different circumstances. In this chapter the approach will be applied to standard MOO test problems. The results from the model will be analysed using the hyper-area indicator described in Chapter 8 in order to determine the robustness of the approach. The performance of the approach will also be compared to the performance of a commercial optimiser for the same two test problems.

9.1 Formulation of the Test Problems

Coello Coello (2009) suggested several test problems that can be used to evaluate the performance of a MOO algorithm. The MOP3 maximisation function and the MOP6 minimisation function were selected as suitable test problems for this study. Both problems have two objective functions, disconnected and asymmetric regions in solution space, and complex Pareto front shapes. The true Pareto front is known for both problems.

The MOP3 maximisation function can be formulated with

$$\begin{aligned}
f_1(x,y) &= -[1+(A_1-B_1)^2+(A_2-B_2)^2] \\
f_2(x,y) &= -[(x+3)^2+(y+1)^2] \\
A_1 &= 0.5\sin 1-2\cos 1+\sin 2-1.5\cos 2 \\
A_2 &= 1.5\sin 1-\cos 1+2\sin 2-1.5\cos 2 \\
B_1 &= 0.5\sin x-2\cos x+\sin y-1.5\cos y \\
B_2 &= 1.5\sin x-\cos x+2\sin y-0.5\cos y
\end{aligned} \tag{9.1}$$

where $-\pi \leq x$ and $y \leq \pi$.

The MOP6 minimisation function can be formulated with

$$\begin{aligned} f_1(x,y) &= x \\ f_2(x,y) &= (1+10y)[1-(\frac{x}{1+10y})^2 - \frac{x}{1+10y}\sin(12\pi x)] \end{aligned} \tag{9.2}$$

where $0 \leq x, y \leq 1$.

9.2 Approach Followed for Solving the Test Problems

The same ABM MOO approach that was described in Chapter 7 was used to solve these test problems. An agent was assigned to each of the objective functions with the goal of optimising it. After each new strategy evaluation the agents had to negotiate in order to decide on the next strategy to be implemented. The outcomes of the previous evaluations affected the agents' satisfaction indexes and therefore influenced their negotiation style.

The Pareto approximation set was once again populated with all the acceptable solutions from the current strategy database and contained all the benchmark solutions.

9.3 Evaluating the Performance of the Test Problems

The quality of the solutions found by the agent-based MOO approach is compared to the performance of the MOO genetic algorithm (GA) of Matlab®. The Matlab® MOO GA is a commercial product based on the NSGA-II algorithm of Deb *et al.* (2000). The hyper-areas for 1000 replications of the model per test case is calculated using the ABM approach and the MOO GA algorithm. The hypothesis suggested by Bekker (2012) and described below is tested by making use of the standard two-sample t-test. The basic null hypothesis of the test states that the data in the two test sets are independent random samples from normal distributions with equal means and unequal, unknown variances. The alternative hypothesis states that the means from the ABM MOO approach is greater than the mean of the Matlab® MOO GA. A significance level of 5% is assumed and the one-sided right-tail hypothesis test formulated by:

$$\begin{aligned} H_0 &: m^i_{AI_H} \leq m^i_{MI_H} \\ H_1 &: m^i_{AI_H} > m^i_{MI_H} \end{aligned} \tag{9.3}$$

where $m^i_{AI_H}$ is the mean of the approximation sets produced by the ABM MOO approach of the i-th test problem, and $m^i_{MI_H}$ the mean of the Matlab® approximation sets.

The rejection of the null hypothesis in favour of the alternative hypothesis will show that the ABM MOO approach produced higher quality solutions than the algorithm of Matlab®.

The results from the test problems are presented next.

9.4 Test Problem Results

The Pareto front obtained from a replication of each of the test problems are shown in Figure 9.1 and 9.2.

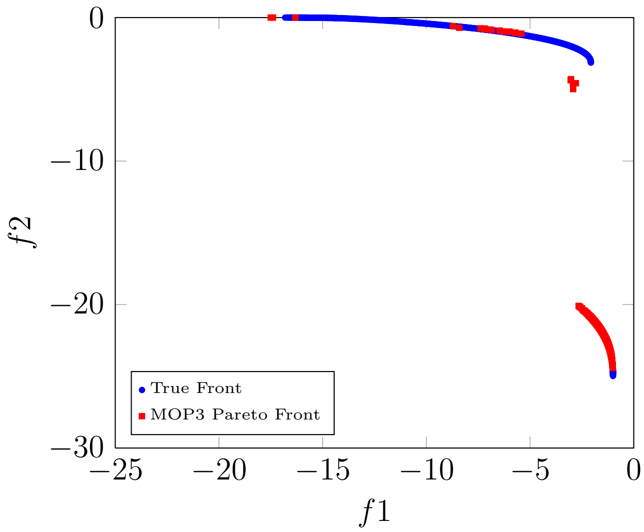


Figure 9.1: Pareto Front for MOP3 (Max).

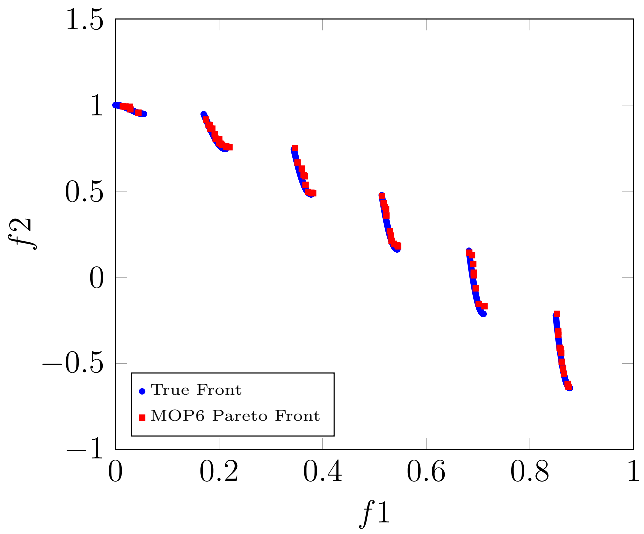


Figure 9.2: Pareto Front for MOP6 (Min).

It was found that the ABM MOO approach in general created larger hyper-areas than the Matlab® algorithm was able to achieve. The boxplots showing the ABM MOO and Matlab® hyper-areas are presented in Figure 9.3 and 9.4.

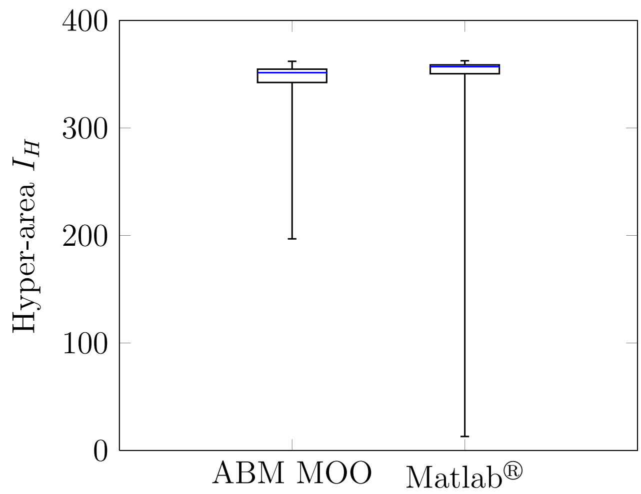


Figure 9.3: Box plot for the hyper-area comparison for the MOP3 test problem.

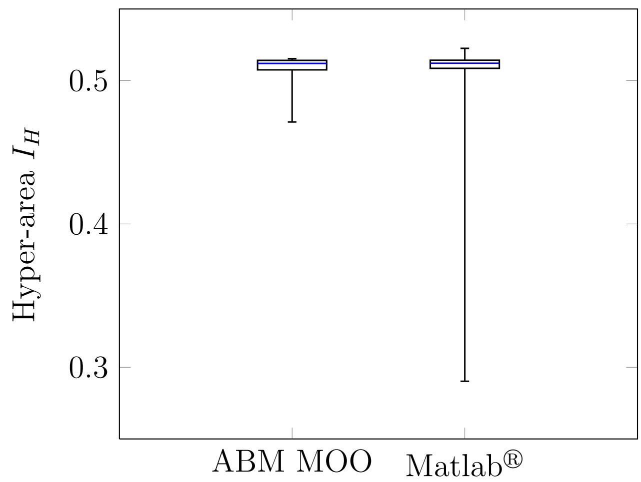


Figure 9.4: Box plot for the hyper-area comparison for the MOP6 test problem.

The results of the hypothesis tests for the MOP3 and MOP6 problems using the hyper-area indicator are shown in Table 9.1. The hypothesis is rejected for both problems. Therefore it is found that the ABM MOO approach is superior to the Matlab® algorithm for these problem instances.

Test Problem	***h*-value *p*-value**	**CI low**	**CI upper**	***t*-stat**	**Outcome**
MOP3	0.0032	1.9853	∞	2.7351	Reject
MOP6	0.0029	0	∞	2.7651	Reject

Table 9.1: Outcomes of the hypothesis tests for the test problems.

9.5 Concluding Remarks: Chapter 9

In this chapter it was proven that an agent-based approach to multi-objective optimisation can generate good quality solutions for two recognised MOO test problems. The solutions were of a higher quality than the solutions generated by a commercial optimiser from Matlab®. This proves that the ABM MOO approach has the potential to be effectively applied in a variety of different MOO problems.

The following chapter provides a conclusion for the study and suggests future research opportunities.

Chapter 10

Conclusions and Project Summary

This study investigates the feasibility of making use of an agent-based approach to multi-objective optimisation by applying it to the (s, S) inventory problem.

A literature study was presented to review the work that had been done in terms of agent-based modelling in multi-objective optimisation. Although the two research fields have been used together before, there were few instances in the literature where a pure agent-based approach was applied to MOO.

The characteristics of agents lend themselves to the modelling of the negotiation processes between conflicting parties. By considering multi-objective optimisation as a negotiation between two parties, each with their own objective, the potential therefore exists for agent-based modelling to do multi-objective optimisation.

The (s, S) inventory problem, which is well-known in the Industrial Engineering domain and formulated in Section 6.4, was identified as a suitable environment in which the approach could be evaluated. An agent-based model of the inventory problem was developed in Anylogic, with two agents responsible for the two conflicting objectives – minimising inventory cost and maximising service level.

The simulation model was run to determine if the agents have the capability to find good solutions.

Several performance measures were identified to analyse the performance of the approach. The difference in hyper-area was used as a measure in order to compare the performance of different scenarios.

The results and analysis revealed that:

- The human decision-making process can be effectively modelled by making use of the negotiation abilities of agents.
- Agent-based modelling can be used as a metaheuristic for multi-objective optimisation.
- Care should be taken to ensure that the agents are aggressive enough in their negotiations, otherwise they will not be able to achieve the desired results.
- Providing the agents with the ability to remember their satisfaction during the previous strategy's implementation allows them to find better solutions.

The agent-based approach to MOO was also successfully applied to a recognised MOO test problem and benchmarked against a commercial optimiser further proving the ability of the algorithm to generate meaningful solutions.

This study provided insight into how a multi-objective optimisation can be performed on a theoretical textbook problem by making use of agent-based modelling. Future research in this field could focus on further developing this metaheuristic for more practical applications. Possible application areas include complex supply chain systems, financial models, risk analysis and economics.

References

ALLAN, R. (2009). Survey of agent based modelling and simulation tools. *Engineering*, **501**, 5772. 26

ARROW, K. (1958). *Studies in the mathematical theory of inventory and production*. Stanford mathematical studies in the social sciences, Stanford University Press. 44

AXELROD, R. (1997). *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*. Princeton University Press, 1st edn. 19

BAKER, R.C. & URBAN, T.L. (1988). A deterministic inventory system with an inventory-level-dependent demand rate. *The Journal of the Operational Research Society*, **39**, pp. 823–831. 41

BANKS, J. (1999). Introduction to simulation. In *Proceedings of the 31st conference on Winter simulation: Simulation—a bridge to the future*, WSC '99, 7–13, ACM, New York, NY, USA. 11, 12

BEKKER, J. (2012). *Applying the cross-entropy method in multi-objective optimisation of dynamic, stochastic systems*. Ph.D. thesis, Stellenbosch University. 8, 79

BEKKER, J. & ALDRICH, C. (2011). The cross-entropy method in multi-objective optimisation: An assessment. *European Journal of Operational Research*, **211**, 112–121. 58

BERSINI, H. & VARELA, F. (1991). Hints for adaptive problem solving gleaned from immune networks. In *Proceedings of the 1st Workshop on Parallel Problem Solving from Nature*, PPSN I, 343–354, Springer-Verlag, London, UK. 9

Beyer, H.G. & Schwefel, H.P. (2002). Evolution strategies: A comprehensive introduction. *Natural Computing*, **1**, 3–52, 10.1023/A:1015059928466. 9

Bodt, M.A.D. & Graves, S.C. (1985). Continuous-review policies for a multi-echelon inventory problem with stochastic demand. *Management Science*, **31**, pp. 1286–1299. 41

Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America*, **99**, 7280–7287. 21

Bond, A.H. & Gasser, L. (1988). An analysis of problems and research in DAI. In A.H. Bond & L. Gasser, eds., *Readings in Distributed Artificial Intelligence*, 3–35, Morgan Kaufmann Publishers, San Mateo, CA. 23

Booth, T. (1967). *Sequential machines and automata theory*. Wiley. 20

Borshchev, A. & Filippov, A. (2004). From system dynamics and discrete event to practical agent based modeling: Reasons, techniques, tools. *Simulation*, 2529. ix, 13, 50

Box, G. & Draper, N. (1987). *Empirical model-building and response surfaces*. Wiley series in probability and mathematical statistics: Applied probability and statistics, Wiley. 11

Branke, J., Deb, K., Miettinen, K. & Slowinski, R., eds. (2008). *Multiobjective Optimization, Interactive and Evolutionary Approaches.*, vol. 5252 of *Lecture Notes in Computer Science*, Springer. 6

Cardon, A., Galinho, T. & Vacher, J.P. (2000). Genetic algorithms using multi-objectives in a multi-agent system. *Robotics and Autonomous Systems*, **33**, 179–190. 30

Casti, J. (1997). *Would-be worlds: how simulation is changing the frontiers of science*. J. Wiley. 18

Chapelle, J., Simonin, O. & Ferber, J. (2002). How situated agents can learn to cooperate by monitoring their neighbors' satisfaction. In *ECAI*, 68–72. 20

Chatfield, D.C., Hayya, J.C. & Harrison, T.P. (2007). A multi-formalism architecture for agent based, order-centric supply chain simulation. *Simulation Modelling Practice and Theory*, **15**, 153–174. 21, 23

Chen, D., Zhou, Z. & Hu, R. (2008). Research on the inventory scheduling model based on agent-oriented Petri net in supply chain. *Kybernetes – International Journal of Cybernetics Systems and Management Sciences*, **37**, 1234–1242. 25

Chen, F. (2000). Optimal policies for multi-echelon inventory problems with batch ordering. *Operations Research*, **48**, pp. 376–389. 41

Clark, A.J. & Scarf, H. (1960). Optimal policies for a multi-echelon inventory problem. *Management Science*, **6**, pp. 475–490. 41

Coello, C., Lamont, G. & Van Veldhuizen, D. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation Series, Springer. 59, 60

Coello Coello, C.A. (2009). Evolutionary multi-objective optimization: some current research trends and topics that remain to be explored. *Frontiers of Computer Science in China*, **3**, 18–30. 7, 8, 78

Coyle, J.J., Bardi, E.J. & Jr, C.J.L. (2002). *The Management of Business Logistics: A Supply Chain Perspective*. South-Western Publishing, 7th edn. 35, 36, 37, 39

De Santa-Eulalia, L.A., Frayret, J.M. & D'Amours, S. (2008). Essay on conceptual modeling, analysis and illustration of agent-based simulations for distributed supply chain planning. *INFOR Information Systems and Operational Research*, **46**, 97–116. 25

Deb, K. & Jain, S. (2002). Running performance metrics for evolutionary multi-objective optimization. Tech. Rep. 2002004, Indian Institute of Technology. 61

Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. (2000). A fast and elitist multi-objective genetic algorithm: NSGA-II. 30, 79

Dorigo, M. (1992). *Optimization, Learning and Natural Algorithms (in Italian)*. Ph.D. thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy. 8

Faratin, P., Sierra, C. & Jennings, N.R. (1998). Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, **24**, 159–182. 20

Forrester, J. (1999). *Industrial dynamics*. System dynamics series, Pegasus Communications. 41

Fox, M.S., Barbuceanu, M. & Teigen, R. (2000). Agent-oriented supply-chain management. *International Journal of Flexible Manufacturing Systems*, **12**, 165–188, 10.1023/A:1008195614074. 24

Garifullin, M., Borshchev, A. & Popkov, T. (2007). Using Anylogic and Agent-Based approach to model consumer market. In *EUROSIM 2007*. 21

Gilbert, N. & Bankes, S. (2002). Platforms and methods for agent-based modeling. *Proceedings of the National Academy of Sciences of the United States of America*, **99**, 7197–7198. 26

Gilbert, N. & Troitzsch, K.G. (1999). *Simulation for the social scientist*. Open University Press. 23

Giri, B. & Chaudhuri, K. (1998). Deterministic models of perishable inventory with stock-dependent demand rate and nonlinear holding cost. *European Journal of Operational Research*, **105**, 467–474. 41

Glover, F. & Laguna, M. (1997). *Tabu Search*. Kluwer Academic. 8

Graham, R.L. (1966). Bounds for certain multiprocessing anomalies. *Bell System Technical Journal*, **45**, 1563–1581. 30

Gupta, A., Whitman, L. & Agarwal, R.K. (2001). Supply chain agent decision aid system (scadas). In *WSC '01: Proceedings of the 33nd conference on Winter simulation*, 553–559, IEEE Computer Society, Washington, DC, USA. 19

Harris, F.W. (1913). How many parts to make at once. *The Magazine of Management*, **10**, 135–136. 37

Heath, B.L. & Hill, R.R. (2010). Some insights into the emergence of agent-based modelling. *Journal of Simulation*, **4**, 163–169. 22

Iglehart, D.L. (1963). Optimality of (s,S) policies in the infinite horizon dynamic inventory problem. *Management Science*, **9**, pp. 259–267. 36

Ingham, J. (1999). What is an agent? Tech. Rep. #6/99, Centre for Software Maintenance, University of Durhan, Durhan, London. 18

Iyer, A.V. & Schrage, L.E. (1992). Analysis of the deterministic (s, S) inventory problem. *Management Science*, **38**, pp. 1299–1313. 44

Julka, N., Srinivasan, R. & Karimi, I. (2002). Agent-based supply chain management: Framework. *Computers and Chemical Engineering*, **26**, 1755–1769. 24

Kelton, W.D., Sadowski, R.P. & Sadowski, D.A. (1998). *Simulation with Arena*. Boston, EUA : McGraw-Hill. 11, 12

Kendall, E., Malkoun, M. & Jiang, C. (1996). A methodology for developing agent based systems. In C. Zhang & D. Lukose, eds., *Distributed Artificial Intelligence Architecture and Modelling*, vol. 1087 of *Lecture Notes in Computer Science*, 85–99, Springer Berlin / Heidelberg. 20

Kennedy, J. & Eberhart, R. (2002). Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, 1942–1948. 8

KIMBROUGH, S.O., WU, D.J. & ZHONG, F. (2002). Computers play the beer game: can artificial agents manage supply chains? *Decision Support Systems*, **33**, 323–333. 43

KIRKPATRICK, S., GELATT, C.D. & VECCHI, M.P. (1983). Optimization by simulated annealing. *Science*, **220**, 671–680. 8

KNOWLES, J., THIELE, L. & ZITZLER, E. (2006). A tutorial on the performance assessment of stochastic multiobjective optimizers. 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland, revised version. 61

KNOWLES, J.D. & CORNE, D.W. (2000). Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, **8**, 149–172. 30

KONAK, A., COIT, D.W. & SMITH, A.E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, **91**, 992–1007. 7

KRISHNA, V. & RAMESH, V. (1998). Intelligent agents for negotiations in market games, part 1: Model. *IEEE Transactions on Power Systems*, **13**, 1103–1108. 20

LAW, A. & KELTON, W. (1991). *Simulation modeling and analysis*. McGraw-Hill series in Industrial Engineering and Management Science, McGraw-Hill. 14

MACAL, C. & NORTH, M. (2006). Tutorial on Agent-Based Modeling and Simulation Part 2: How to Model with Agents. In *Proceedings of the Winter Simulation Conference, 2006. WSC 06*, 73–83. 23, 26

MARIANO, C.E. & MORALES, E. (2000). A new distributed reinforcement learning algorithm for multiple objective optimization problems. In *In Memorias del Segundo Encuentro Nacional de Computaci on ENC99, paper No. 111*, 12–15. 32

Moscato, P. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Tech. rep., Caltech, Caltech Concurrent Computation Program, report 826. 9

Narzisi, G., Mysore, V. & Mishra, B.B. (2006). Multi-Objective Evolutionary Optimization of Agent-based Models: An Application to Emergency Response Planning. In B. Kovalerchuk, ed., *The IASTED International Conference on Computational Intelligence (CI 2006)*. 30

Pan, A., Leung, S., Moon, K. & Yeung, K. (2009). Optimal reorder decision-making in the agent based apparel supply chain. *Expert Systems with Applications*, **36**, 8571–8581. 25, 27

Papazoglou, M.P. (2001). Agent-oriented technology in support of e-business. *Communications of the Association for Computing Machinery*, **44**, 71–77. 18

Pratchett, T. (2007). *Making money: A novel of Discworld / Terry Pratchett*. Harper, New York, NY, 1st edn. 22

Raiffa, H. (1982). *The art and science of negotiation / Howard Raiffa*. Belknap Press of Harvard University Press, Cambridge, Mass. 20

Rogers, A. & von Tessin, P. (2004). Multi-objective calibration for agent-based models. In *Agent-Based Simulation 5*. 30

Sanchez, S.M. & Lucas, T.W. (2002). Exploring the world of agent-based simulations: simple models, complex analyses: exploring the world of agent-based simulations: simple models, complex analyses. In *Proceedings of the 34th conference on simulation: exploring new frontiers*, WSC '02, 116–126, Winter Simulation Conference. 20

Scarf, H. (1960). *The Optimality of (s,S) Policies for the Dynamic Inventory Problem*, 196–202. Stanford university press. 44

Schelling, T. (1978). *Micromotives and macrobehavior*. Fels lectures on public policy analysis, Norton, New York [u.a.], 1st edn. 23

Socha, K. & Kisiel-Dorohinicki, M. (2002). Agent-based evolutionary multiobjective optimisation. *Proceedings of the World on Congress on Computational Intelligence*, **1**, 109–114. 31

Storn, R. & Price, K. (1997). Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, **11**, 341–359, 10.1023/A:1008202821328. 8

Swaminathan, J.M., Smith, S.F. & Sadeh, N.M. (1997). Modeling supply chain dynamics: A multiagent approach. 24

Sycara, K. (1998). In-context information management through adaptive collaboration of intelligent agents. In *Intelligent Information Agents: Cooperative, Rational and Adaptive Information Gathering on the Internet*, 78–99, Springer Verlag. 19

Terano, T. & Naitoh, K. (2004). Agent-based modeling for competing firms: From balanced-scorecards to multi-objective strategies. *Hawaii International Conference on System Sciences*, **3**, 30090. 29

Tukey, J.W. (1977). *Exploratory Data Analysis*. Addison-Wesley. 61

Valluri, A. & Croson, D.C. (2005). Agent learning in supplier selection models. *Decision Support Systems*, **39**, 219–240, decision Theory and Game Theory in Agent Design. 25

Verdicchio, M. & Colombetti, M. (2002). Commitments for agent based supply chain management. *ACM SIGecom Exchanges*, **3**, 13–23. 25

Weise, T. (2008). Global optimization algorithms, theory and application. 8

Winston, W.L. (2004). *Introduction to Probability Models, Operations Research: Volume 2*. Thomson-Brookes/Cole, Toronto, Canada, 4th edn. 38, 40

Wong, D., Paciorek, N., Walsh, T., DiCelie, J., Young, M. & Peet, B. (1997). Concordia: An infrastructure for collaborating mobile agents. In

K. Rothermel & R. Popescu-Zeletin, eds., *Mobile Agents*, vol. 1219 of *Lecture Notes in Computer Science*, 86–97, Springer Berlin / Heidelberg. 19

WOOLDRIDGE, M. & JENNINGS, N.R. (1995). Intelligent agents: Theory and practice. *Knowledge Engineering Review*, **10**, 115–152. 18

YANG, X.S. (2008). *Nature-Inspired Metaheuristic Algorithms*. Luniver Press. 9