DOCTOR OF PHILOSOPHY

Integrated supply chain scheduling and inventory control in uncertain environments

Kalata, Magdalena Anna

Award date:
2021

Awarding institution:
Coventry University

Integrated supply chain scheduling and inventory control in uncertain environments

Magdalena Anna Kalata

Faculty of Engineering, Environment and Computing

Coventry University

A thesis submitted in partial fulfilment of the University's requirements for the Degree of Doctor of Philosophy.

December 2020

DECLARATION

This dissertation is the result of my own work and includes nothing, which is the outcome of work done in collaboration, except where specifically indicated in the text. It has not been previously submitted, in part or whole, to any university of institution for any degree, diploma, or other qualification.

Content removed on data protection grounds

Abstract

Scheduling and inventory control problems have been amongst research topics on designing and optimising Supply Chains (SCs) which have attracted a very high interest to both academics and practitioners for many decades. Without a doubt, the body of scientific literature and the real-world applications deliver a proof that these problems hold the interest not only of researchers and mathematicians, but also industry practitioners which deal with scheduling and inventory control decisions every day. Interest in these problems have resulted in many state-of-the-art mathematical models programmed to find optimal solutions and simulation frameworks allowing to observe parameters of complex environments in real time manner. However, these models very often do not consider uncertainties which are inherent in SCs.

The goal of this research is to create a new model supporting decisions for coordinated inventory and scheduling problems in a dynamic SC environment facing uncertainty in demand. A control scheme using fuzzy logic for modelling uncertainty was developed for a four echelon SC including Suppliers, Manufacturer, Distribution Centre and Customer. Fuzzy sets enable use of expert knowledge which allows representation of imprecise or vague data. The new method developed in this research proposes the decision support system which determines scheduling of orders by prioritising jobs to the resources available to the specific echelon with simultaneous determination of replenishment levels and order quantities of required raw materials. The objectives are to minimise the total holding cost of the inventory along SC and to minimise the delays of orders delivered to the customer. Simulation-optimisation approach was employed to test knowledge extraction capabilities of the proposed models, aiming to propose a robust control-schemes which is less sensitive to the changing demand. Four control-schemes were developed. Crisp dispatching rules (DRs) and two sets of fuzzy dispatching rules (FDRs) were used to provide inventory and scheduling control. First set of FDRs were delay-focused, so higher inventory levels were kept by echelons to quickly satisfy the demand, the second set were holding cost-focused FDRs where inventory levels were kept lower to minimise holding cost of additional stock. To

determine the optimal control the search process was guided by NSGAII and to increase the robustness of the model, a Monte Carlo simulation was conducted within NSGAII creating MCNSGAII control scheme. A benchmark scenario and a number of experiments with varying due dates, order sizes, processing times and order intensities were carried out. The results obtained are analysed and provide an insight into SC performances with uncertainty in demand and changing SCs parameters. Non-information sharing policy between echelons was employed and varying order intensity was simulated multiple times in various scenarios to test the proposed control schemes.

FDRs for both subproblems including inventory control and scheduling outperformed standard policies based on continuous review policy (CRP) and crisp DRs for scheduling. Uncertainty in both; demand for both subproblems was addressed by applying a multi-objective optimisation. NSGAII performed better than both manually determined FDRs leading to a decrease in the delay in delivering orders to customers by 66% in comparison to delay-focused FDRs, while keeping a very similar holding cost level.

Rule bases generated by MCNSGAII led to improvement of both objectives by capturing dynamics of changing demand offering robust solutions with a low standard deviation from the average objectives' values. A further decrease of the average holding cost by 8.2% and the average delay by 5.2% were also observed comparing to the standard NSGAII. The novel developed methodology displays robustness of solutions and success in making trade-offs between holding cost and delay offering an independent and flexible control for both scheduling and inventory control problems across multiple echelons.

ACKNOWLEDGEMENTS

First and foremost, I am deeply grateful to Prof. Dobrila Petrovic, my Director of Studies, for her knowledge, kindness, academic guidance, and motivation, during this project.

Also, I would like to thank my supervisor, Dr. Ammar Al-Bazi, for his insightful comments and assistance at every stage of the research project.

Many thanks to all Futures Institute colleagues for their support and unforgettable times.

Finally, I would like to express my gratitude for my husband, family and friends for their boundless dedication and support, especially during the final stage of the project.

Contents

LIST OF TABLES

List of Figures

LIST OF ALL ABBREVIATIONS AND ACRONYMS

SC	Supply chain
SCM	Supply Chain Management
SCE	Supply Chain Engineering
ESC	Emergency Supply Chains
SO	Simulation Optimisation
RO	Robust Optimisation
PRP	Periodic Review Policy
CRP	Continuous Replenishment Policy
VMI	Vendor Managed Inventory
EOQ	Economic Order Quantity
KPI	Key Performance Indicators
ILP	Integer Linear Programming
NLP	Non-Linear Programming
MILP	Mixed Integer Linear Programming
MINLP	Mixed Integer Non-Linear Programming
FIFS	First In First Served
EDD	Earliest Due Date
LTWR	Least Time Working Remaining
MTWR	Most Time Working Remaining
FDR_df	Fuzzy Dispatching Rule (delay focused)
FDR_hcf	Fuzzy Dispatching Rule (holding cost focused)
GA	Genetic Algorithm
NSGA-II	Non-dominated Sorting Genetic Algorithm
FIS	Fuzzy Interference System
UI	User Interface

1 INTRODUCTION

1.1 Background concepts of SC planning and inventory control

Supply chain (SC) management and control have become a strategic focus of leading manufacturing companies and has been recognised as one of the main manufacturing paradigms emerging in past few decades. SC planning and scheduling in particular, are often the focus of work of large manufacturers with its cooperating suppliers, and integrated distribution centres. However, majority of a research in this domain focuses on a single manufacturer optimisation or scheduling either upstream suppliers or downstream customers while manufacturer is used as the centre of the SC. In these cases, many assumptions are made and the processes happening inside of either upstream or downstream echelons are omitted. This leads to over-simplification of complicated and uncertain SCs. Excluding the dynamic aspect of SC's behaviour leads to developing a non-realistic and non-reactive models with poor representation of real-world problems.

Treating of SC uncertainty is a very broad subject (Paul, 2015). SC uncertainties might have many sources and they can be grouped depending on uncertainty origins. However, they are always a part of any functioning SC. An effective SC control-scheme should consider different types of uncertainty. Nevertheless, majority of the existing models do not treat uncertainty and available models for larger SC are very often deterministic (Sawik 2016b). Any disruption which occurs in a SC, may lead to an increase in cost and time of the whole production and delivery process. Very often, models optimising schedules do not consider any source of uncertainty (Hao et al. 2015) or are focused only on one type of it (Subramanian 2014). Hence, when a SC consisting of various types of echelon is considered and decisions have to be made on each level, the uncertainty might propagate through many or all echelons of the SC and it might cause unwanted bottlenecks, additional costs, and delays. Considering uncertainty in the model increases the model complexity (You and Grossmann 2008) but considering it can lead to an increase in solution robustness.

Coordination of a SC can be achieved if echelons work together by implementing integrated resource allocation, collaborative initiatives, and information sharing. An objective of the coordinated SC is to maximise profit for the entire SC. The coordination of supply, production, and distribution scheduling might be especially useful in complex SCs. Two types of information exist in SCs. The first type is forecasting and planning related information which focuses on future demand, seasonality, and prediction of orders. It is used to make tactical decisions, which include expansion of SC such as building additional storage or production facilities and other structural and design changes. This type of information, when shared, can lead to improvement of efficiency. The second type of information is related to the functioning of the SC. It includes information necessary for echelon to enable production, inventory of goods as well as be able to use location of echelons and transportation details to enable essential services between echelons. Fully accurate data can help in coordination of decisions and it often leads to lower costs and delays in a functioning SC. In practice acquiring of specific and full data is often complex as the data may be incomplete or insufficient quality and installing measures to collect it can be expensive (Hugos 2003: 40).

1.2 Motivation and objectives of this research

Overall SC scheduling and inventory control becomes a very complex and challenging problem, which aims to combine different objectives and constraints of SC planning. The motivation to this work is a deficit of research which considers scheduling of all parts of SC under most typical uncertainties with simultaneous inventory control for all echelons. In this work analysis of optimisation of echelons decisions are carried out for a general-structure SC. Decision-making process is presented for a complex system considering inventory control problem for multiple products and echelons and scheduling of multiple orders along SC's echelons.

To the best of the author's knowledge there is no published work on effect of various uncertainties in demand on dynamic coordination of scheduling, delivery dynamics and inventory control schemes. Creating a robust schedule which focuses on prevention of negative effects including strategies on how to handle uncertainties is very important and desirable solution in complicated and very often unpredictable SCs. At the same time robustness of a proposed control-scheme can be important performance measure. A robust schedule and inventory control throughout SC can prevent radical changes in performance after appearing of disruption and solution remaining close to the optimal. Objectives of this research are as follows:

1. To identify and model sources of uncertainty that characterise SC and to develop multi-objective control models that will include these uncertainties into the decision-making process. This includes decision support system with decisions and processes carried out on different echelons of SC for coordinated inventory control and scheduling at each echelon.
2. To develop a methodology using Fuzzy Dispatching Rules for simultaneously solving inventory control and scheduling problems across SC considering different types of echelons and transportation dynamics.
3. To test the proposed control schemes against various demand changes to observe behaviour of different types of control and provide sensitivity analysis of the proposed model. To deepen understanding on SC control decision in the face of uncertainty in the dynamic setting, the execution of various "what if" analyses to see how uncertain demand propagates through other echelons and how this affects the schedule and inventory control decisions.
4. To optimise a Fuzzy Dispatching Rules by minimising total holding cost and delays in the SC by using GA metaheuristic and knowledge extraction capabilities of Fuzzy Inference System.

5. To reduce the gap between theory and practice by considering uncertain demand of SCs by proposing the control-scheme which lead to robust performance despite changing demand.

1.3 Contribution to the knowledge

The following research proposes a dynamic fuzzy logic-based model to accommodate uncertainties across SC and a new simulation-based model for general-structure SC that enables control of dynamic schedule and inventory control for multiple echelons. Dispatching rules are used for orders prioritisation and Continuous Replenishment Policy is used for monitoring and changing inventory levels. Dispatching rules used for scheduling included priority sorted by the time of an order arrival, order due date or by the processing time required for production. A general-structure SC is considered. It consists of Suppliers, Manufacturer, Distribution Centre, and Customers. Each of Supplier producing different type of raw material which is then delivered to the Manufacturer echelon. Manufacturer echelon produces different types of final products and delivers them to the Distribution Centre. Distribution Centre is the only echelon in considered SC which do not produce any goods, but instead it is collecting orders from Customers and it has to schedule its deliveries. Each echelon has its own characteristic and different processes involved. To consider the uncertainty of the demand a Fuzzy Dispatching Rules (FDRs) are used for supporting scheduling and inventory control decisions. The FDRs uses expert knowledge to consider uncertainty. They are used to control inventory levels and schedule jobs to minimise holding cost and delays of orders.

Fuzzy scenarios which include different types of uncertainty are developed. Uncertainty such as customer demand, changing workload, uncertain processing time and varying due dates are incorporated in the proposed scenarios. Use of fuzzy logic and linguistic values of these parameters allow optimisation without partial or no historical data. Sensitivity analysis and various tests have been carried to introduce insights into SC optimisation problems. Impact of these changes on SC performance has been compared. Existing research has been mostly done in the area of inventory control of a single echelon and less often inventory control of multi-echelons. The scheduling of a single manufacturer, supplier, or distribution centre with different machine configurations on a shop floor, but there is hardly any scheduling model for multiple echelons. Usually just two echelons are considered such as supplier and manufacturer and manufacturer and distribution centre.

Due to the order size and complexity simulation optimisation encompassing metaheuristic has been used to solve this optimisation problem.

1.4 Structure of the thesis

Chapter 2 presents the literature review on problems in complex SCs, the gaps in knowledge are discussed and methods previously used for solving SCM problems are analysed. Chapter 3 delivers details of the selected methodologies relevant to this research including fuzzy theory for representing uncertainty and

Genetic Algorithm which is used for finding solutions to multi-objective problem. Chapter 4 consists of the formulation of the SC problem, followed by Chapter 5 with details of implementation of a new SC simulation framework which contains a SC structure with all necessary components such as machines, lorries, inventories etc. for both: scheduling and inventory problems. In Chapter 6 simple heuristics in form of dispatching rules are implemented throughout entire SC for scheduling problem and constant pre-set replenishment inventory are applied for the inventory control problem. The experiments on SC parameters such as due date, order sizes and processing times are conducted in order to observe behaviour of SC in the presence of demand uncertainty and no information sharing between echelons. Chapter 7 of this thesis includes development of fuzzy dispatching rules. One rule base is proposed for schedules of echelons and two rule bases are proposed for the inventory control problem. Adaptability to react to uncertain parameters is observed and decisions proposed by control scheme are further optimised by the Genetic Algorithms in Chapter 8. Comparison of all models developed is given in Chapter 9. A robustness of solutions is increased by nesting a Monte Carlo simulation inside the genetic algorithm. Chapter 10 is the last chapter of this work. It includes conclusions and possible directions for future work.

2 Literature Review

2.1 Introduction

2.1.1 Introduction of Supply Chain Management

A Supply Chain (SC) is a network composed of individual echelons (Wei, Krajewski 2000). Modern SC is usually a complex chain of facilities and it can include multiple echelons such as Suppliers, Manufacturers, Distribution Centres (DC) and Customers. The job of SC is to convert the raw resources which are produced and delivered by suppliers, throughout manufacturing process into the finished, ready to deliver final products. Supply Chain Management (SCM) and Supply Chain Engineering (SCE) are both concerning planning, designing, and operations in the SC. The main difference between these disciplines is type of approaches they consist of. SCM supposed to be mainly focused on traditional management and achieving an integrated approach, while SCE focus is on optimisation, mathematical modelling and implementation of solutions using software. In practice both terms SCM and SCE are used interchangeably. Both disciplines have the same goal of addressing and solving problems defined in different part of the chain. In this thesis 'SCM' will be used as an umbrella term that encompasses problems defined in this research, such as multi-echelon inventory control (Sarker 2014; Eruguz et al. 2016; Kok et al. 2018; Chinello et al. 2020), planning and scheduling of SC (Kreipl, Pinedo 2003; Sawik 2016), assembly and transportation scheduling problem (Baptiste et al. 2008; Pundoor and Chen 2009; Assarzadegan and Rasti-Barzoki 2016; Guo et al. 2018; Framinan et al. 2019) and integration between echelons (Yolmeh and Salehi 2015; Zahran et al. 2016). Full spectrum of SCM problems according to Lambert and Cooper, 2000 can be categorised into eight categories and it is introduced in Figure 2.1.

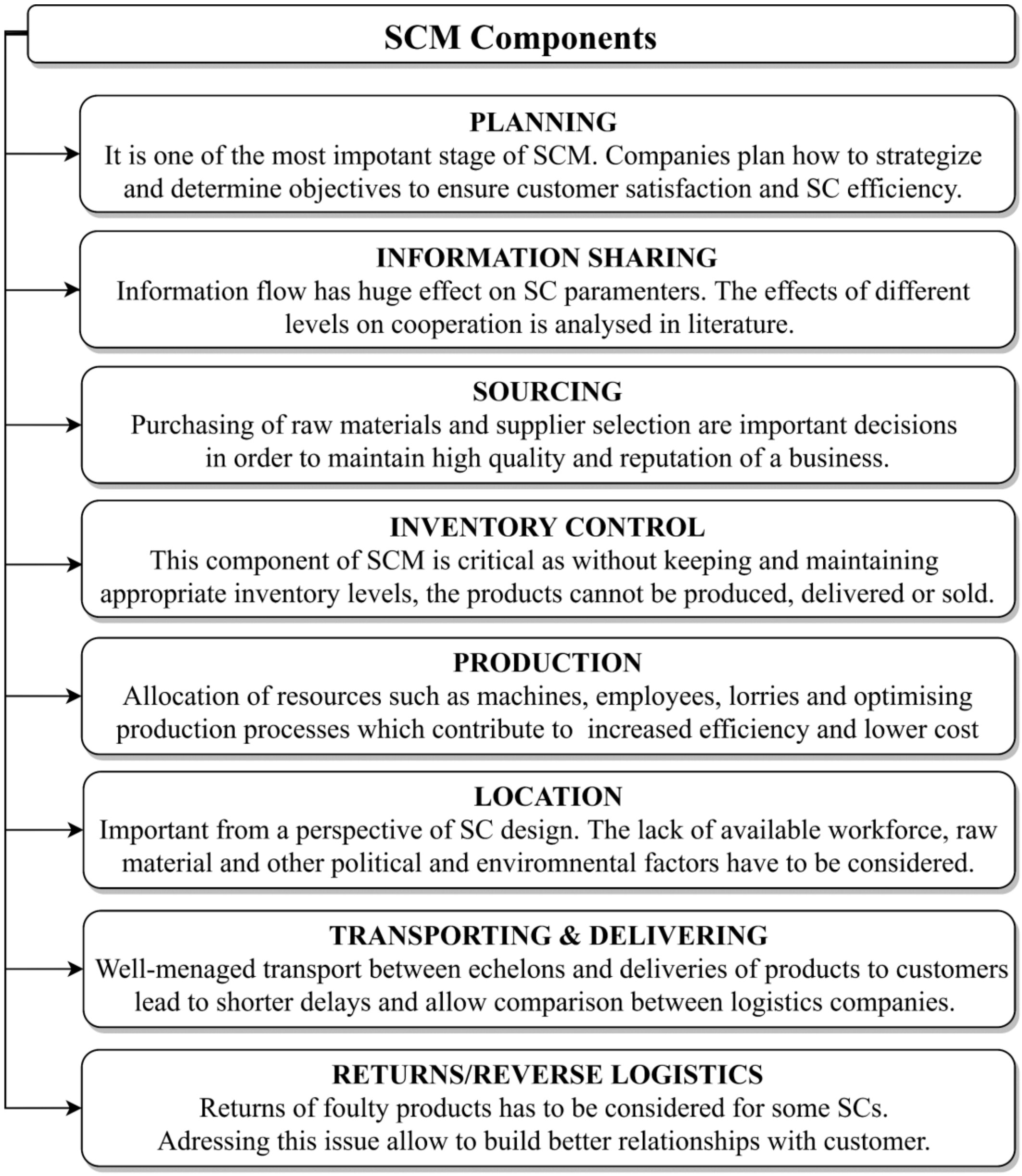

Figure 2.1 Categories of problems in Supply Chain Management

Research has been conducted on various type of SCs. It includes optimisation of Emergency Supply Chains (ESC) (Kaddoussi et al. 2013; Othman et al. 2017), optimisation of food SCs with strong time constraints and many more including medical, chemicals (You, Grossmann 2008), electronics (Li et al. 2008) and automotive industries. Each of the chain type is defined by its own characteristics. It is not a trivial task to model and optimise SC. Each category of SCM is presented in Figure 2.1. contains complex research questions which are researched over few decades by academics and practitioners and it offers outlook at the problems and frameworks for their optimisation. The gap found in the literature is lack of consideration more generic SC structures considering multiple categories of SCM problems. One of the biggest obstacles lay in SC's dynamic nature, complex interactions between echelons and many uncertainties appearing throughout the chain. Due to frequently changing environment, there is a strong need for new models and tools handling different types of uncertainty to be developed (Sawik 2015; Jain and Foley 2016). Improving SC performance is one of the main concerns of operational researchers since 1950s. Although the term '*Supply Chain'* did not appear in the literature before late 1980s, the research concerning more than one echelon was referred in the literature as the logistics, multi-echelon, or operational management problems (Hugos 2003: 2). It has been established by theory and practice, that by choosing right location, offering inventory control, optimisation of schedules as well as maintaining smooth flow of products between echelons can increase SC efficiency. Depending on objectives defined in the analysed problem, it also leads to customer's satisfaction increase and to decrease of undesired effects of the bottlenecks (Harjunkoski et al. 2009).

Bottlenecks tend to slow down a material flow between echelons and can be caused either by limited resources or unexpected events. In that case high or uncertain production demand on one echelon can affect the work of the whole chain.

2.1.2 Complexity of SCM problems

This research is focused on integration of inventory control and production and distribution scheduling in a multi-echelon SC. Presented literature review aims to analyse and summarise state of the art for these types of problems.

Due to the complexity and various interpretations of SCM problems, the presented review cannot be treated as exhaustive. As SCM covers a very broad subject, the research issues such as sales management, route planning, supplier selection, reverse logistics, single echelon scheduling and forecasting are not broadly discussed in this review. Several additional literature reviews papers summarising SCM are recommended to provide more in-depth perspective on this discipline. Interesting literature review by Croom et al. (2000) describe major issues researched in the field of SCM and propose an analytical framework for classification of these problems and analysis of used methodologies. To create this classification, they used a content-oriented criterion which allowed to analyse what kind of problem was considered, what kind of objectives were optimised and it classified the level of analysis into (i) single relationship between echelons, (ii) most complex chain level and (iii) network level. Second criterion was methodology-oriented and categorised current work based on two epistemological dimensions. First was divided into theoretical and empirical to introduce distinction between theoretical research focusing on introducing mathematical description of the problem and offers analytical solutions, while empirical research was described from perspective of practical use of tools and methodologies and observing behaviours of SC in practice. The second dimension was divided into prescriptive and descriptive which was understood as distinction between proposing new models and comparing already existing research.

The main distinction between planning and scheduling is in their objectives. Planning models are usually assessed by cost-oriented objective such as maximisation of profit, minimisation of shortage, transportation or holding cost. Planning can be used for designing of a SC or planning of long-term activities and setting inventory levels. Scheduling is concerned with tasks sequencing in shorter period and its objectives are focused on time. Majority of research for a scheduling problem focuses on minimisation of delays or completion times. Planning covering multiple-echelons does not require detailed information such as scheduling does, which can be defined as ‘short -term planning’ and includes sequencing production of one echelon at the time. Kreipl and Pinedo (2004) offered an analysis on theory and practice for planning and scheduling problems in SC for continuous and discrete manufacturing industries. They concluded that task of mid-term planning and short-term scheduling were not very often incorporated in one model. It was caused by different nature of objectives of these SCM problem categories. Planning stages included inventory control and were focused on profit maximisation or minimisation on various costs in SC, such as penalty, holding and transportation costs. Scheduling was focused on time target. It considered minimisation of objectives such as tardiness or earliness of the job. These objectives were also related to delays to the final customer. It could be understood as an important factor of customer satisfaction which was crucial in customer driven SCs. Authors pointed out that emphasis of SC planning was on set up time cost while transportation cost was very often omitted.

The description of categories used to classify papers relevant to this research can be found in Table 2.1. It consists of categories such as number of considered echelons, what kind of problem does it cover and what sort of uncertainties are defined in the chain.

Table 2.1 Categorisation criteria for SC planning and scheduling papers

Category	**Description**	
SC Component	Suppliers	Supplier is considered.
	Manufacturer	Manufacturer is considered.
	Inventory	Inventory for any echelon is considered.
	Distribution	Distribution Centre is considered.
	Customer	Customers/Customer demand is considered.
	Return	Returns of product is considered.
Scheduling	Orders	Prioritisation of orders.
	Machines	Detailed schedule of machines (single and parallel-machine scheduling) and for more complex problems (job-shop scheduling, flow-shop scheduling).
	Distribution	Scheduling of deliveries between echelons including customer are considered.
Inventory Control	Single echelon	Inventory is planned for only one echelon.
	Multi-echelon	Inventory is planned for more than one echelon.
Uncertainty	Supply	All uncertainties related to distribution and transportation between echelons, including natural disasters such as earthquake or tsunami but also men-made distributions such as fires, accidents etc.
	Production	Includes disturbance related to leading to production uncertainties such as broken equipment, problems with products quality or sick personnel.
	Demand	Uncertain or changing demand of already existing orders.
Cost	Inventory	Inventory cost is treated as part of the objective.
	Production	Production cost is treated as part of the objective.
	Transportation	Transportation cost is treated as part of the objective.
	Penalty	Penalty for late or early deliveries is included.

Control type	Decentralized	Decentralized control is often the case in large real word networks, where the individual companies are looking at optimising their own objective. Echelons are bounded only by contractual obligation between each other.
	Centralized	Centralized control requires only one decision maker. Single entity such a large manufacturer makes all the decisions for a benefit of whole SC by contrast to decentralized control, where each company makes their own, independent decisions to optimise their own objectives.
Solution Algorithm	Optimisation	Exact optimisation algorithms.
	Heuristic	Heuristics allowing solving a problem without guarantee of finding optimal solution. It includes approaches such as Ant Colony Optimisation algorithm (ACO), Genetic Algorithm (GA), Simulated Annealing algorithm (SA), Taboo-Search algorithm (TS).
	B&B	Branch and bound algorithm allowing total search of a solution space.
Modelling Approach	Simulation	Modelling type used for complex systems in which organisational dynamics and nonlinearity is taken into consideration. Especially useful when the size of the model is too large for analytical model to be practical (due to time required to obtain solution).
	Analytical	Mathematical models either static or dynamic (time dependent or time independent). Characterised by their way of representing the given system in mathematical formulation and possibility to find the optimal solution.
Inventory control	PRP*	Periodic Review Policy assumes that fixed time intervals for ordering stock are used and orders are placed in regular cycle.
	CRP*	Continuous Replenishment Policy does not assume fixed times for ordering. The time and quantity of order depends on orders placed by other echelons, usually customer.
	VMI*	Vendor managed inventory assumes that only one echelon, usually supplier is responsible for all decisions concerning inventory control.
	EOQ*	Economic Order Quantity determine the size of orders, which are optimal to satisfy demand. Demand in this type of models is usually considered to be constant.

Objective	Single	Single-objective optimisation aim to find an optimal solution for only one criterion. In scheduling: • minimisation of completion time, • minimisation of tardiness, • minimisation of earliness, • minimisation of cost are amongst frequently used objectives.
	Multi	Multi-objective optimisation has a goal of optimising two or more objectives simultaneously (Varhanan et al. 2012). It is especially useful in case where a trade-off between these objectives is apparent (e.g., satisfaction of customers can be increase but for exchange on profit, which may decrease).

First part of the subsequent section focuses on approaches used to address scheduling problem in SC. Next, SC's inventory control problems and additional work considering both problems at the same time is analysed in more detail. Aim of this review is to be thorough with work relevant to scheduling and inventory control problems and to provide a comprehensive analysis of proposed frameworks. It will provide an insight on aspects which has been omitted in reviewed papers. At the end of this chapter a discussion about research gaps will be provided.

Table 2.2 and Table 2.3 that demonstrates differences between papers for SC scheduling and SC inventory control problems, respectively, can be found at the end of two following subchapter. A comparison of the papers is based on categories described in .

2.2 Scheduling in Supply Chain

2.2.1 Different perspectives on SC scheduling problems

Scheduling is a decision-making process which determines when series of procedures, events and tasks should be executed to complete various jobs. It is constrained within given time frames and limited existing resources. Scheduling of the manufacturing process aims at generating the list of the sequential order of activities assigned duration of tasks defined for production and the start and finish times of each identified task (Shaw et al. 1992). Traditionally, planning is a mandatory step before scheduling because the outcome decisions from the planning stage are inputs necessary to schedule task sequence (Li et al. 2010). While planning can be described as an identifying process of all tasks necessary to complete the project, the general scheduling definition may be described as allocation of these tasks to finite resources over a production time interval (Zukui et al. 2008). Time of completing jobs mainly depends on how much resources are available. Some tasks defined in SC must be completed before the next can begin, while another can be done simultaneously. There are many techniques which are used to represent jobs relations and a schedule may optimise the objective of one or more performance measures. This performance measures, also known as performance criteria or key performance indicators (KPI) include metrics such as completion time (Huang et al., 2015, Gao et al., 2015), tardiness (Hassanzadeh et al. 2016; Tamannaei and Rasti-Barzoki 2019; Kim et al. 2020) earliness, cost (Behnamian et al. 2016; Guo et al. 2017), lateness of jobs, profit, risk and many more. In a real-world application objective will depend on goals of specific company in SC.

Scheduling can be interpreted differently for each part of SC. For the manufacturer, where industrial processes are performed, the schedule is often evaluated by its ability to optimise production time (Assarzadegan 2016). In the context of logistics and delivering, the resources of the schedule may be transportation vehicles like lorries, ships and planes and a job decision can be how to pack the orders in order to optimise schedule of production and deliveries by minimisation of transportation and tardiness penalty costs (Li et al. 2008). Scheduling in a SC is always a decision-making process which enables practices such as effective resources sharing and determination of priority, time, and sequence of echelons tasks processes. Scheduling should be carried out in such a way as to meet the due dates to minimise delays. It must be considered as one of the main paradigms of modern manufacturing companies with cooperating suppliers and DCs. Although, there is a clear increase in demand for optimisation of more complex systems, majority of research focuses on a single manufacturer optimisation. Many papers in the field of scheduling focuses on simpler structures of SCs (Temiz, Erol 2004; Cheng, Li 2010; Rokni, Fayek 2010; Lai, Wu 2011; Liao, Su 2017). Those structures can be defined such as single machine scheduling problem (Demirili, Cheng 2003; Li et al. 2010a; Cheng, Li 2010) or flow-shop problems, where jobs have to go through sequence of machines (McCahon, Lee 1990; Temiz, Erol 2004; Lai, Shu 2008; Lai, Wu 2011; Huang et al. 2012; Nakhaeinejad et al. 2013). Another large group of researchers focuses on scheduling of either upstream suppliers or downstream customers while manufacturer is used as the centre of the supply chain (Sawik 2014). In 80s and 90s hardly any paper considered schedule for whole SC (Wei, Krajewski 2000). Even in more recent papers, manufacturer is frequently used as the centre and most important echelon of the SC. It is scheduled separately to the rest of the chain or with a consideration of only upstream suppliers (Pundoor and Chen 2009).

Upstream schedule refers to planning supplying processes and the time and quantity of orders (Sawik 2014; Subramanian 2014; Chu 2015). The processes happening inside of supplier echelon are very often omitted, and supplier is treated only as a source of the raw material. The downstream scheduling is concerning manufacturer and DC or third-party logistics company or customers (Yeung 2011). That allows transportation scheduling to be considered simultaneously with production scheduling (Guo et al. 2017). In the next section a summary of multi-echelon SC scheduling will be discussed.

2.2.2 Multi-echelon scheduling

Scheduling of multi-echelon SC assume scheduling of tasks and processes for more than one echelon and include integration of two or more schedules across SC. As mentioned in the section above, there is plenty of research with scheduling optimisation problem for Manufacturer echelon only.

Some researchers took an interest in more than one problem and considered more of SCM problems simultaneously. However, it creates even more complex stochastic optimisation problems. Integration of production scheduling and routing problems was covered by Moon et al. (2017). They concluded that these types of problems were usually considered separately, but integration between them may lead to 5-20% increase of SC efficiency. Li et al. (2008) investigated a problem of synchronisation of schedules where number of jobs determined the problem size. The problem was divided into two decomposed problems: an air transportation allocation problem and a parallel machine scheduling problem with earliness penalties. The goal was to determine a schedule which ensured completion of the orders on time and minimised penalties between assembly and transportation. Techniques used to solve these two sub-problems were: ILP model for the air transportation problem which allocated orders to the existing air transportation capacities with minimum costs and MILP approach with SA algorithm was used to solve machine scheduling problem by diversifying neighbouring population for improved performance. Nikolopoulou and Ierapetritou (2012) took under investigation scheduling of production and distribution subject to inventory capacity levels. Their single objective model assumed minimisation of the total cost by concurrent optimisation of the production and transportation schedules. Their research identified strong interactions between decisions making in levels of planning and scheduling. Conclusion was that integration between those levels was necessary to create a globally optimal solution. In order to overcome computational complexity and provide a representation closer to real-world, they proposed a hybrid method by combining mathematical modelling with simulation. For the defined problem, they proposed simulation-based optimisation by applying a MILP model connected to agent-based reasoning. Simulation was used for capturing behaviours and interactions between 1-tier suppliers, manufacturers, and customers. For similar problem Hsu et al. (2016) used agent-based fuzzy constraint-directed negotiation model allowing various compromises and changes of initial decisions between agents until finding a collaborative, win-win strategy. Although mathematical models are widely used in SCM and optimisation and offer exact solutions, they are limited in terms of size, complexity and often must oversimplify the real-world problems. A simulation study has become more popular in the field of SCM due to ability to mimic stochastic and non-linear systems without oversimplifying the problem. Use of Simulation Optimisation (SO) bypasses oversimplification as contrary to the mathematical modelling as it does not assume that full algebraic description for such a complex SC problem is possible (Amaran et al. 2014). Longo (2011) described simulation as a notable approach which can outperform mathematical and stochastic models for complex SC structures, arguing that sum of benefits brought by this methodology was greater than by other methods in this field. Li et al. (2010) considered process planning and scheduling problem and proposed agent-based simulation to solve it. The developed method aimed to reduce scheduling conflicts and flowtime and increase adaptation of the model to uncertainties occurring in a flow-shop. Several agents representing tasks and resources were considered in this SC. Results of negotiation between agents proved effectiveness of the proposed approach. Schedule of distribution is very often a problem of emergency supply networks. Kaddoussi et al. (2013) considered distributed delivery scheduling problem, as part of the crisis management SC problem. Main task was to create a plan for delivery of first aid products such as food and clothes, in the case when a natural disaster occurred. The area of potential coverage was divided into smaller pieces of land and the multi-agent approach was executed. The lands creating separate units were treated as individual sub-problems and assigned a separate schedule, using an intelligent system, based on a framework for a distributed cooperation. A two-step delivery scheduling problem was defined: first, the local delivery schedules were built by assigning means of transports and routes to the whole SC; then, performance indicators were generated to evaluate the global performance of the covered areas, and to identify the assignments that need to be readjusted, to satisfy all connected areas. Othman et al. (2017) had also introduced a multi-agent-based scheduling system for an Emergency Supply Chain (ESC). The research question was how to plan delivering resources from supplying zones to areas damaged by disaster. Main goal of the proposed model was a quick response in the case of emergency, with the objective to optimally allocate limited resources such as a military units, clothes, food, and water. Tests were considering within two real-world scenarios: Mali and Japan crisis and were carried by using real-world data. Agents from the proposed simulation followed protocols between each other, automatically selected zone of emergency and provided a dynamical schedule subject to a characteristic of environment and a size of problem. Simulation is often the preferred methodology when many scenarios must be investigated. Simulation based Particle Swarm algorithm was proposed by Varhanan et al. (2012) for a multi-echelon, multi-product SC. The simulation-based approach enabled finding a solution obtaining the best trade-off between objectives of production and distribution scheduling problem.

Wang et al. (2015) investigated an operations scheduling problem for a multi-echelon SC with an objective to minimise sum of the shipping, processing, and penalty costs. Scheduling operations of a trans-shipment problem were introduced and solved as extended knapsack problem for a three echelon SC including:(1) heterogenous suppliers (2) capacitated processing centres (3) network of business customers. A subset of customers and suppliers was selected to be served with a given time and supplies, so penalty could be minimised among the other costs. Constraints such as capacity, flow balance and deadlines were considered. Serving all customers was not possible, so capacity of manufacturer became a limited resource. Non-linear penalty was applied for not achieving a given service level, by not serving enough demand from customers. The problem was solved via dynamic programming and applied to a business outsourcing production of semi-finished products with an aim to meet seasonal demand. A two-stage algorithm was developed. In phase one, a time window was determined to satisfy deadline constraint. The second phase selected which customer orders could be fulfilled with consideration of capacity constraints. A non-linear penalty was also applied for not serving big enough portion of demand of the customer network. Interesting recent study by Chen et al. (2019) proposed synchronized scheduling of production and distribution by offering a bi-level Simulated Annealing (SA) algorithm. The problem was divided into two selfish divisions, where each of them optimised its own objective and generated a schedule according to its objective. Minimisation make-span objective was conducted for each echelon separately and algorithm was developed to find a synchronized schedule between echelons with keeping their autonomy. Multi-echelon scheduling was evident in this research, but it very rarely considered more than two echelons.

Comparison between papers for SC scheduling problems can be found in Table 2.2.

Table 2.2 Comparison between papers closely related to multi-echelon SC scheduling problem

Scheduling Problem		Part of SC						Scheduling			Uncertainty			Cost				Control Type		Algorithm			Modelling Type		Objective	
Reference	*Year*	*Suppliers*	*Manufacturer*	*Inventory*	*Distribution*	*Customer*	*Return*	*Orders*	*Machines*	*Distribution*	*Supply*	*Production*	*Demand*	*Inventory*	*Production*	*Transportation*	*Penalty*	*Decentralized*	*Centralized*	*Optimisation*	*Heuristic*	*B&B*	*Simulation*	*Analytical*	*Single*	*Multi*
This research	*2020*	•	•	•	•	•		•		•			•	•					•		•		•			•
Han et al.	*2019*	•	•			•		•		•				•		•			•	•				•	•	
Guo et al	*2017*		•		•				•	•				•	•	•			•	•				•		•
Othman et al.	*2017*	•			•	•				•					•	•		•				•		•	•	
Hassanzadeh et al	*2016*		•		•				•					•		•			•		•			•		•
Hsu et al.	*2016*	•	•			•		•	•						•			•						•	•	
Yolmeh, Salehi	*2016*		•			•			•	•			•		•	•			•					•	•	
Cheng, Leung	*2015*		•		•				•						•	•					•			•	•	
Chu et al.	*2015*	•		•		•		•				•		•					•	•				•	•	•
Wang, Lei, Lee	*2015*	•	•			•		•							•	•	•			•				•	•	
Sawik	*2014*	•	•					•			•					•			•					•	•	
Kaddoussi et al.	*2013*				•	•				•						•			•			•	•		•	
Yeung et al.	*2011*	•	•	•				•						•				•					•		•	
Rokni, Fayek	*2010*		•						•			•							•				•	•		•
Li et al.	*2008*		•			•			•	•						•	•		•	•				•	•	
Torabi, Ghomi, Karimi	*2006*	•	•						•					•		•				•		•		•	•	
Wei, Krajewski	*2000*	•	•	•		•			•				•	•		•								•	•	

2.3 Inventory Control in Supply Chain

Inventory control is an important problem in SCM. Estimation of all inventories carrying costs in a SC are approximated to be between 20-60% (according to Baker 2007) and 25-55% of the total cost of company assets value (according to Zahran et al. 2016). An inventory control is a fundamental component of SCM. There is an extensive research considering single-echelon approaches (Costantino et al. 2016). Control of each echelon's inventory independently to the other parts of SC can lead to oversights, longer delays and low customer satisfaction (Klosterhalfen and Minner 2010).

Olugu & Wong (2009) proposed a SC performance evaluation and pointed out that SC are more customer-driven than ever before. It creates a difficulty regarding whether inventory decisions should be based on efficiency of SC (such as minimisation of inventory or other costs) or SC responsiveness which can be measured as ability to quickly satisfy customer demand by product availability (Longo 2011). The models for multi-echelon structures with uncertain demand brought researchers a very hard task involving modelling complex SC structures including relations between SC elements and information sharing. Different problems are defined for different echelons and multitude of problems requires complex solutions. In order to avoid non-linearity, the early research on this subject was focused on introducing exact solutions for oversimplified SCs consisting of only one echelon. Research on heuristics in SC optimisation between 1980s and 2000s has laid the groundwork for commercial software for inventory control. Kok et al. (2018) conducted an extensive research and identified gaps for stochastic multi-echelon inventory control problems. Simultaneous inventory control on all SC levels leads to creating multi-echelon optimisation approaches. Following section aims to deliver an outlook on papers concerning SC inventory control problems. First part includes discussion on different problems and used methodologies. Next integration of inventory control with other types of SCM problems is considered.

2.3.1 Outlook at SC inventory control problem

Eruguz (2016) proposed a comprehensive study of fifty papers which considered an inventory control problem of multi-echelon SC with unknown demand. Their analysis focused on three characteristics such as: (i) used methodologies, (ii) what assumptions were considered and (iii) what industrial applications of proposed solutions were. According to this classification, methodologies proposed in the literature included Mixed Integer Programming (MIP) approaches, heuristics approaches as well as optimal and dynamic approaches. The proposed classification of assumptions covered uncertainty modelling of unknown demand and stochastic lead times. Authors also considered different types of modelling demand and analysed effects it had on the safety stock policies. Extraordinary measures such as speeding up production by overtimes or express deliveries and outsourcing were also considered. The capacity constraint for different echelons was introduced in order to avoid surplus. These types of constraints are useful for SCs with sharing information structure. Schoenmeyr and Graves (2009) considered a ‘censored order policy’ which enabled holding cost minimisation by limiting orders to the upstream echelons when these echelons were incapable to fulfil more orders due to capacity restrictions. Another used classifier was the type of replenishment policies used, such as constant safety stock policies (Chen and Chen 2005), periodic-review policies (Petrovic et al. 2008) and continue replenishment policies (Gao et al. 2008).

Chen and Chen (2005) proposed research investigating decentralised and centralised inventory control policies in a two-echelon SC for deterministic demand. The aim of this work was to determine an optimal replenishment strategy for cost minimisation. Manufacturer due to products variety faced high set-up and transportation costs. The search algorithm was proposed for finding optimal replenishment policy. Results showed that centralised policy was always better than decentralised version by achieving lower costs. An iterative coordination procedure for selecting optimal inventory review policy periods was developed by Petrovic et al. (2008). The proposed model coordinated the distribution centres and the manufacturer to get satisfactory control of the SC. Authors proposed decomposition of the two-echelon SC consisting of one manufacturer and several distribution centres. The complex inventory control problem for SC was divided into smaller subproblems by modelling subproblems individually for each echelon. Then echelons determined their simplified optimisation tasks independently to each other. If for both echelons coinciding inventory review policy can be found, the satisfactory inventory control was obtained in a first step of this procedure. Alternatively, if founded solutions were different, fuzzy constraints related to the tolerance of objective function values were defined. A solution with the highest satisfaction degree became a final solution. In case that no such a solution exists a further adjustment of the tolerances and objective functions were possible. Nia et al. (2014) used VMI policy to find optimal order quantities for single supplier and single customer SC. Inventory cost was minimised with consideration of capacity, delivery, and order quantity constraints. Distribution recovery model for a single-echelon system was proposed by Paul (2015). Demand was known and constant, but different uncertainties such as production disturbances were examined. Proposed dynamic solution had the capability to propose revised plan after disruption occurs. Disruptions could appear as a single event or in sequence. It was handled by a mathematical model solved by GA and pattern search algorithms. Schaefer et al. (2015) proposed a bi-objective model concerning minimisation of cost and expected carbon emissions. A Pareto front which determined values for CRP policy was introduced.

Abdel-Aleem et al. (2016) implemented an Adaptive Neuro-Fuzzy Interference System (AN-FIS) for a production inventory problem. A production disruption such as machine breakdown were incorporated into simulation model for a single stage cement company. Inventory decision rules was also proposed by Costantino et al. (2016). The goal of this work was to mitigate a bullwhip effect appearing between echelons of seasonal demand SC. Impact of various inputs on bullwhip effect was analysed and smoothening replenishment rules were proposed and improved ordering patterns and inventory stability. Multi-echelon SC inventory planning was proposed by Dai et al. (2017) in order to minimize the sum of inventory costs. A retailer, several middlemen and production plant were parts of considered SC. Three different types of demand were considered. Proposed types of demand included ramp-type demand, reverse ramp-type demand, and trapezoidal-type demand. Computational experiments were solved using GA and

SA algorithms and sensitivity analysis was proposed to validate assumptions. A heuristic algorithm was developed by Puga and Tancrez (2017) to solve large SC location-inventory problem under uncertain demand. Non-linear mathematical formulation included location, allocation, and inventory decisions. A proposed model simplified to a linear model when certain parameters were fixed. These cases were solved by an iterative algorithm and proved to be able to find optimal solution even for a very complex SC. Adediran et al. (2019) used simulation optimisation approach for solving a complex flow-shop inventory replenishment problem. Agent-Based modelling and heuristic under three customer-imposed disruptions was considered. The novelty of the paper was a framework allowing gradual replenishment of stock with a customer satisfaction objective being maximised. Disruptions such as customer altering original demand (either in quantity or the deadline) and change of order sequence were taken into consideration. Simulation study on inventory optimisation was carried out by Chinello et al. (2020). Inventory control simulation studied a two-echelon toy manufacturer SC. Authors pointed out that majority of the existing research involved developing optimal policies while assessing an impact of such policies was often disregarded. The focus of the work was to identify and assess the main drivers used in inventory optimisation and proposing a framework to achieve it. A comprehensive literature review was followed by interviews with selected employees of the company used in their case study to further improve the proposed framework. Limitation of this paper was at its specific, descriptive case study approach which would not be easily transferable to other SC problems. Table 2.3 presented below provides a close comparison between multi-echelon inventory problems.

Table 2.3 Comparison between papers closely related to multi-echelon SC inventory control problem

Inventory Control Problem		Echelon		Uncertainty			Cost				Control Type		Algorithm			Modelling Type							Objective	
Reference	*Year*	*Single*	*Multi*	*Supply*	*Production*	*Demand*	*Inventory*	*Production*	*Transportation*	*Penalty*	*Decentralized*	*Centralized*	*Optimisation*	*Heuristic*	*B&B*	*Simulation*	*Analytical*	*PRP*	*CRP*	*Coordinated*	*VMI*	*EOQ*	*Single*	*Multi*
This research	*2020*		•			•	•				•			•		•			•	•				•
Chinello et al.	*2020*	•				•	•					•	•			•								•
Adediran & Al-Bazi	*2018*		•		•	•	•	•			•			•		•							•	
Hemmati et al.	*2017*		•			•	•	•	•		•		•								•		•	
Dai et al.	*2017*		•		•		•				•			•			•						•	
Puga and Tancrez	*2017*		•			•	•	•	•		•			•			•			•			•	
Costantino et al.	*2016*		•				•				•					•		•					•	
Abdel-Aleem et al.	*2016*		•				•	•				•					•				•		•	
Chen, Chen	*2015*		•			•			•	•	•	•		•			•						•	
Liu et al.	*2015*		•			•	•		•		•		•	•					•				•	
Paul	*2015*	•			•		•	•				•		•			•			•			•	
Schaefer et al.	*2015*	•			•	•			•		•						•		•					•
Subramanian et al.	*2014*		•				•					•					•						•	
Nia et al.	*2014*		•				•	•			•	•	•				•				•	•	•	
You, Grossmann	*2008*		•			•	•	•		•	•						•							•
Petrovic et al.	*2008*		•			•	•	•	•			•					•	•					•	

2.4 Treating of Uncertainty

SC which consists of several up to tens of echelons such as Suppliers, Manufacturers, Distribution Centres and Customers that often incorporate uncertainties. The SCs ambiguous, stochastic and fuzzy parameters such as prices of products and deliveries, reliability of machines and other resources as available trucks, orders made by customer and times of production and deliveries are often fluctuating and cannot be described in a crisp, deterministic way Pistikopoulos (1995). Treating of SC uncertainty is a very broad subject (Paul 2015) and this chapter does not provide exhaustive comparison of all available models. Supply chain disruption might have many sources and it can be grouped depending on uncertainty origins. Effective scheduling and inventory control should both consider different types of uncertainty. Nevertheless, majority of existing models does not treat uncertainty and available models for larger SC are very often deterministic (Sawik 2016b). Any disruption which occurs in a SC, may lead to an increase in cost and time of the whole production and delivery process. Very often, models optimising schedules do not consider any source of uncertainty (Hao et al. 2015) or are focused only on one type of it (Subramanian 2014). Hence, when large and complicated SC is considered, uncertainty might propagate through many or all echelons of a network and it might cause unwanted bottlenecks, additional costs, and delays. Considering uncertainty in the model increases the model complexity (You and Grossmann 2008). There are many techniques used for description of uncertainty and three distinctive methods are presented in Table 2.4.

Table 2.4 Methods of describing uncertain parameters in scheduling

Description of uncertainty	**Definition**
Probability description	The most common way to describe uncertainty. Used in cases where there is enough knowledge about uncertainty behaviour. Probability description is associated with an event thanks to which the pattern of uncertainty can be found, it is either consistent or random.
Bounded form description	Used in cases where there is insufficient information about the uncertainty to create probability description. The knowledge is enough to broadly describe error bounds of the uncertainty and bounds include all possible values of these uncertain parameters.
Fuzzy description	Might be used in both cases, when there is enough knowledge about parameter such as historical data and in case when enough data is not accessible. There are three substantial advantages of this description: - In comparison to probability description, they do not need complex integration schemes when the continuous probabilistic models are proposed - In case of discrete probabilistic models, they do not need as many scenarios as probability description - It is the most natural way to describe any information given in linguistic values. It can easily translate linguistic values to numbers.

The uncertainty is modelled differently for the stochastic and fuzzy optimisation methods. Fuzzy programming takes into consideration uncertain constraints and objectives and those are introduced as fuzzy sets or fuzzy relations. Some violation of these parameters is allowed in fuzzy optimisation. Membership functions of belonging to the fuzzy set can be introduced. Bounding the objective functions by upper and lower bounds can lead to the improvement of decision making. Balasubramanian and Grossmann (2003) applied a non-probabilistic approach to the analysis of processing time uncertainty for new product development process and flow-shop scheduling problem. Good estimates of the uncertain parameters have been obtained by using proposed discretisation. The proposed models were solved with reasonable computation time. A fuzzy multi-objective model in the presence of uncertain due date of jobs was implemented for a single machine scheduling problem by Duenas and Petrovic (2008). Uncertain values of the system were modelled by using satisfaction degrees. The model proposed was multi-objective and attempted to minimise maximum and average tardiness of tasks, by combining searching and GA algorithms. Model was validated on real-world data of pottery manufacturer. In study of Nia et al. (2014) uncertainties in both: demand and shortages were considered and handled by an ant colony algorithm. The objective of implemented method was to minimise the total cost inside the supply chain.

The most common source of uncertainty in SC is unknown demand (Salem and Haouari 2017). It includes unknown demand as well as changes in already placed orders. Other uncertainties occurring in SC are located on supply part of the chain. It includes tragical consequences of eruptions, earthquakes, and other natural and manmade disasters such as fires and accidents (Childerhouse amd Towill 2004). Another type are uncertainties can be defined for production phase. It includes disturbances such as machine breakdowns, unknown processing times, staff availability and an uncertain quality of products. Uncertainties in SCs do not refer only to disturbances during production and transportation. Common problem for SC is lack of available historical data which also often must be considered in models. Uncertainty in a single echelon which is the most common case can be handled locally and within limits of this echelon.

Jia and Ierapetritou (2006) introduced a multi-objective robust optimisation model for a scheduling problem. The aim of this research was to handle uncertainty. The expected performance, model robustness and solution robustness Normal Boundary Intersection technique were utilised to solve this problem and the Pareto optimal solutions with a trade-off between objectives was proposed. Mulvey et al. (1995) developed the concept of Robust Optimisation (RO) to provide a trade-off between finding optimal and robust solution. According to the proposed definition, a solution is robust if for different scenarios it stays close to the optimal value. A robust schedule can prevent radical changes in performance after appearing of disruption and solution remains close to the optimal Creating a robust schedule which focuses on prevention of negative effects and include policies how to handle disruptions is very important and desirable solution in complicated and very often unpredictable SCs. At the same time robustness of a schedule can be important performance measure and can be used as validation tool to assess solution performance. Sawik (2014) proposed mixed integer linear programming (MILP) optimisation for coordinated scheduling and supplier selection problems considering various types of disruptions in customer driven SC. Supply disturbances such as earthquakes were considered. Suppliers delivering raw materials to manufacturers, and both: single and multi-sourcing of raw material scenarios were examined. A proposed model allocated supplier to an order subject to two objectives. First was minimisation of risk based on ranking of suppliers and second was minimisation of cost.

Above literature is focused on scheduling and inventory control in SC and a review concerning uncertainty in SC described in this subchapter cannot be treated as extensive. A comprehensive review on uncertainty for Supply Chain Network Design are covered by Tordecilla et al. (2020). Further information and review on representation and methods used to model uncertainty in a closed-loop SCs are introduced by Peng et al. (2020). The conclusion from this research suggests that large gap exists in modelling methods. Authors conclude that exact solutions proposed by solver are rarely suitable for real-world problems and simulation modelling can provide more realistic thus applicable models as well as they can provide a new insight which cannot be achieved by linear programming methods.

2.5 Integration and Information Sharing

Integration of decision making between echelons may effectively mitigate the risks occurring in the SC (Ye and Wang 2013; Sawik 2016a). It has been established by Yu et al. (2001) that information sharing and coordination between different echelons create a win-win strategy for all the members in two-echelon SC. There is a body of research focusing purely on the impact of information sharing between echelons (Harjunkoski et al. 2009; Costantino et al. 2015). Mitigation of bullwhip effect, improved inventory management and minimised costs are amongst primary advantages of SC coordination (Shaban et al. 2019). According to Cachon (1999) coordination and information sharing can decrease the total cost up to 35%. The tightening-up relationships between echelons of SC resulted in creation of practises such as Vendor-Managed Inventory (VMI) where only one echelon, usually manufacturer is responsible for all decisions concerning inventory control or quick-response. To maximise a profit of SC Hemmati et al. (2017) proposed a new VMI agreement with consignment stock (VMI-CS) policy in which a manufacturer uses supplier's inventory. The proposed model guaranteed a higher profit in case of coordinated strategy.

Coordination requires that SC echelons have the will and capabilities to apply required mechanisms. While manufacturing process involves multiple suppliers and multiple tiers it is important to notice that uncertainty in SC can propagate and amplify from one tier to another. To combat this problem Wei and Krajewski (2000) investigated the cost implications for different levels of coordination between a manufacturer and multi-tier suppliers. Their stochastic model sought to integrate purchasing and scheduling decisions while minimising the total cost. Authors indicated that the integration of critical path was more cost effective than the tier-1 approach. Exception was the case when the suppliers were moderately flexible, and the cumulative delivery and lead-time was longer than the maximum lead-time of the tier-1 supplier. Tier-1 approach is an intermediate integration technique in which only flexibility of first tire of echelons is taken into consideration while any other echelons are ignored. A critical path of SC is a sum of leads times path between top and bottom tiers. This paper compared different schedule integration approaches and provided cost implications for three different levels of SC integration. It is focusing on integration through schedule sharing. The authors considered how costly each level of integration is and they tried to find all advantages of improved forecast. First studied integration level was Myopic where the top tier member considered only its own internal flexibility when purchasing products. The proposed policies do not include other SC's echelons. The second level of integration is Intermediate level, where the top tier took into consideration the flexibility of entire tier-1 instead of single echelon. The last type is the Total integration, where all the members flexibility capabilities are taken in consideration upon the formulation of the solution policies. Sahin et al. (2005) analyse five levels of coordination which enabled cost reduction by over 47% when the proposed system was fully integrated and shared all information. Discussion on cost of information sharing indicated that coordination of SC requires higher set-up and equipment costs. Harjunkoski et al. (2009) worked on finding answer to questions when and which information should be exchanged amongst echelons of SC. The most important question was to discover; which information companies were less happy to share? Authors underlined that to find a globally optimal solution, very high degree of transparency must be implemented between targeted echelons of SC. It has been identified that solutions such as modelling approaches of MILP and MINLP are the most common ones. Their investigation concluded that building integrated supply chain is still on very early phase, and successful models should be a balance between total transparency and decentralized optimisation. That suggests that echelons of SC should cooperate through common goals instead of just exchange the data.

A three-echelon chemical SC with demand uncertainty was considered by You and Grossmann (2008). A trade-off between minimisation of expected lead time objective and maximisation of the profit was propped in order to create a responsive SC. The authors proposed a large-scale Mixed Integer Non-Linear Programming (MINLP) model to solve this design, planning, and SC inventory control problems. Constraints determining a network structure and scheduling were considered. A heuristic subproblem was proposed to simplify the problem and obtain a near-optimal solution for design and scheduling subproblems and to reduce the solution space. In the next step MINLP model was used to solve scheduling and inventory control decisions. Subramanian (2014) considered scheduling and inventory control problems for a customer driven SC. The manufacturer produced two products from raw materials delivered by supplier. A deterministic MILP model aimed to fulfil two objectives which were minimisation of risk and maximisation of profit. Experiments were carried for very simple and short supply chains consisting of only two echelons - one manufacturer and one supplier, up to SC with eight echelons. Liu et al. (2015) proposed integration of facility location, inventory control, and vehicle routes scheduling problems for an online e-commerce SC. Optimal order size and order times are determined by a hybrid algorithm based on a pseudo-parallel GA and an SA algorithms. Another integration problem between supplier and manufacturer was studied by Zahran et al. (2016). Their research focused on incorporating the Consignment Stock (CS) policy for a three-level SC. First, a supplier that make semi-finished product from raw material and then it is delivered to the vendor, where they were changed into the final product. These final products were shipped to the customer. In general integration was achieved by adopting a consignment agreement between echelons which ensured better management and services levels. In proposed model, CS requires a downstream echelon to pay for items after they are withdrawn from inventory oppose to policy when downstream echelon pays after receiving ordered stock. The proposed research showed that enhanced collaboration between echelons requires information sharing on product flow. Four different scenarios were considered: (1) Holding stock agreement between the supplier and the manufacturer and between the manufacturer and the customer were considered. (2)

There was no holding stock agreement between the supplier and the manufacturer, but it was one between the manufacturer and the customer. (3) There was a holding stock agreement between the supplier and the manufacturer, but not between the vendor and the customer. (4) There was no holding stock agreement between the supplier and the manufacturer, or between the vendor and the customer. The inventory related costs for proposed SC were estimated to be between 25 and 55% of the SC's total cost. Coordinating orders and shipments among echelons in a SC substantially reduced SC costs and increased the profitability of all echelons.

2.6 Discussion and Conclusions

This research goal is to create a framework for general multi-echelon, multi-product, dynamic SC with integrated inventory control and scheduling problems with consideration of uncertainty. Published literature in the area of multi-echelon SC scheduling and SC inventory control was introduced and described in this chapter. Furthermore, a limited research on both problems and integration of problems considered simultaneously has also been discussed. It can be noted in the reviewed literature that mathematical modelling is an established framework for this type of optimisation problems. Many sources in the presented literature suggest that simplifications incorporated in analytical modelling can lead to developing a non-realistic and non-reactive model with poor representation of real-world problems.

Another main common simplification can be found in many available models which are defining supply chains problems as deterministic. In the literature the dynamic aspects of production scheduling and inventory control have been frequently neglected. In the process of planning and scheduling, many companies must address multiple objectives at the same time which also is very often omitted. The models concerning scheduling has assumed that processing times were fixed, crisp values, while in real world cases, these times were very often imprecise or ambiguous. That emerges that model which consider simultaneous schedule of all SC units with uncertainties defined in demand and processes still needs to be developed. At the same time optimal inventory policies are very often offered under very stylized assumptions. Determining optimal policy structures for this problem is computationally intensive even for small SC structures. It is caused by complex non-linearities of the cost functions, namely total holding cost and time of delay. The research following-up mathematical programming investigated the optimal or heuristic setting of parameters for different simple policy structures.

Interesting observation regarding information sharing were discovered. Businesses integrating various stages of production and control decisions between echelons is called vertical integration. Vertical integration of SC is possible when SC of a company is entirely owned by it. That type of information sharing is possible for a specific group of giants and corporations which very often use their own software to deal with specific SC problems. In a situation, where echelons are independent parts of SC and they do consider cooperation with another autonomous echelons, information sharing can be harmful for the company (Costantino et al. 2015). At the same time a multi-echelon inventory control problems review paper by Kok et al. (2018) which compared 394 papers explicitly emphasize that hardly any paper assume that information is not shared in considered SCs.

Review of relevant papers shows a few gaps in the literature. Although scheduling and inventory control problems for multi echelon SC are crucial for SCM they are not very often considered simultaneously. Practical relevance for SCM with uncertain parameters and a high complexity of integrated problems calls for a more general structure framework allowing extraction of information from orders which can benefit the SC for more than one objective.

3 METHODOLOGY

3.1 Introduction

This chapter aims to familiarise the reader with concepts of Operational Research applied in inventory control and scheduling problems. Additionally, a fuzzy systems and evolutionary computational intelligence approaches applied in this research are also discussed. The following subchapters provide a description of basic concepts and background of methods used for modelling robust algorithms to solve inventory control and scheduling problems across multi-echelon SC. The methods and procedures used in this study are discussed in-depth in the following subchapters and conclusions can be found at the end of this chapter. First, basic concepts of sequencing dispatching rules are presented amongst other scheduling algorithms solutions. The next subchapter is focusing on multi-objective Genetic Algorithm (GA) description and its applications. The aim of subchapter about GA is to explain common concepts for evolutionary algorithms and elements necessary for designing an algorithm solution. After this subchapter, the idea behind fuzzy logic is introduced. The description focuses on the difference between fuzzy and binary logic, representation of fuzzy numbers and applications of fuzzy logic in SC scheduling and control. Fuzzy Inference Systems description is also discussed in this subchapter including explanation of how fuzzy rules can be implemented. The following subchapter is focusing on SC's modelling, major differences between analytical and simulation models.

3.2 Solution algorithms and Dispatching Rules

3.2.1 Algorithms classification

An algorithm is an effective procedure which consists of a well-defined sequence of instructions, which can find a solution for many problems in the field of operational research, mathematics, finance, computer science and many more. Such a solution algorithm allows to solve a problem of a specific class (Horowitz, Sahni 1978). Operational Research algorithms can be divided into three major groups which can be found in Figure 3.1 and categorisation description can be found below.

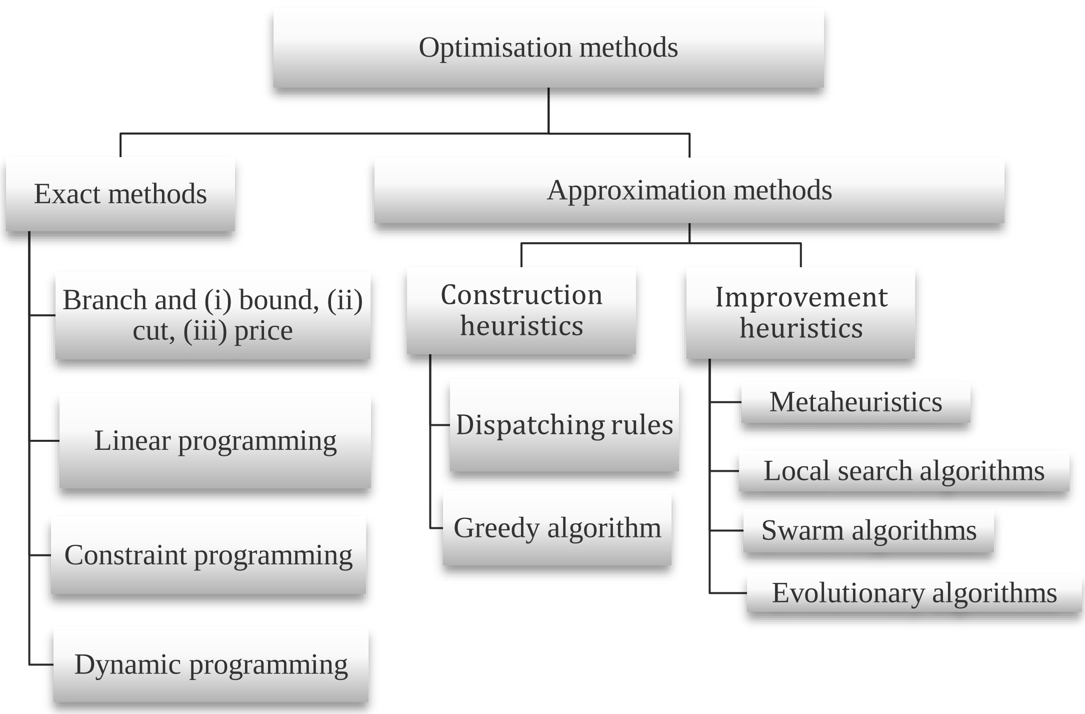

Figure 3.1 Classical optimisation methods and categorisation of algorithms solutions

The first group of algorithms is classified as *exact algorithms*. This type does guarantee an optimal solution and it includes methods such as constraint and dynamic programming as well as Branch and Bound algorithm, which can also be referred to as a full search algorithm. The main weakness of this type of algorithms is limited use in case of larger instances when the time required for finding optimal solution exceeds polynomial time and solution for this problem become infeasible. This type of methods includes Linear, Constraint and Dynamic Programming. Use of exact algorithms is a computationally expensive task.

Second group contains *approximation algorithms.* Solutions proposed by this type of algorithms may not be close to optimum at all. It is known that use of this type of algorithms sometimes leads to finding bad or not rational solutions so that solutions must be evaluated. One of the common practices is evaluation by empirical study. The main advantage of this type of solution is its ability to process larger cases in a shorter time. Heuristic algorithms can be further divided into (i) Improvement and (ii) Construction heuristics. Improvement heuristics include (1) Local search algorithms such as Tabu Search and Simulated Annealing, (2) Swarm algorithms such as Ant Colony and Particle Swarm algorithms and (3) Evolutionary algorithms such as GA. These improvement algorithms start with a base solution and then algorithms aim to improve it in a defined number of iterations.

A group of Constructive heuristics do not aim to improve originally proposed initial solutions determined by algorithms. It starts from an 'empty' solution and opposite to Improvement heuristics it creates a new solution step by step. This group includes algorithms such as Dispatching rules or Greedy algorithm. As this type of heuristics does not aim to find optimal solution, algorithms do not backtrack their decisions. It can lead to decisions which are locally optimal on each stage, but as much information is omitted and only possibilities of the next step are considered, in many cases it cannot produce a globally optimal solution. The main advantage of the construction heuristics is their quickness and flexibility to create models which can be optimal for specific problems for specific criteria.

In metaheuristics there are two conflicting criteria which must be considered. On one part of the spectrum there is diversification, which focuses on broader exploration of the solution space. Random search algorithms are good example of algorithms which are focused on diversification as those are searching for a solution in many random places in the solution space. Diversification of an algorithm is opposite to intensification. Intensification of metaheuristic focuses on exploring the best solutions in more promising regions of a solution space. Algorithms which focus more on this aspect such as local search algorithms first select promising regions and then aim to find the best solution in this neighbourhood. Hence, having both criteria considered, a metaheuristic can deliver a good solution for optimisation problems, especially in case of limited computational capacity.

In conclusion, the approximation algorithms are faster than exact algorithms but produced solutions cannot guarantee optimality. They use so called provable quality and provable run time which consist of information on how far the proposed solution is from the optimal solution (Hochbaum 1997). There is a trade-off between running time and quality of solutions and approximation algorithms can be used in cases where optimal solution is not essential.

3.2.2 Dispatching rules

Dispatching rules (DR) for scheduling belong to the stepwise deciding greedy heuristics group of algorithms. They are classified as part of the construction heuristics group and can be used without a pre-existing schedule, where only one job at a time is added. They are fast and simple to implement as they can be computed in polynomial time. DRs have been used for solving NP-complete scheduling problems and have been extensively used in manufacturing sector (Pickardt et al. 2013). DR can also be useful for data mining. Learning capabilities of algorithms can lead to generation of new DR which learn directly from data.

A single-resource scheduling problem with an aim to improve manufacturer performance with consideration of local disturbances was researched by Kaban et al. (2012). Dispatching rules were extensively implemented and a comparison between rules was discussed. Exactly 44 dispatching rules were presented and compared to provide the final DR ranking. For validation of used rules, a large flow-shop of automotive industry with 10 tasks carried out on 14 machines was used. Due dates were not specified, machines had breakdowns every 3 months and transportation time between tasks was assumed to be deterministic. Each implemented rule had a different effect on the final score. The base model was using First in First served rule (FIFS). This research determined, which rule was effective for 5 important criteria such as:

- average number of tasks in the system,
- average completion time,
- queue waiting time for jobs,

- total waiting time for separate parts and
- average waiting time for products for different work centres**.**

Determining criteria of a given problem is an important task in optimisation planning. Hence, the performance of DR is highly dependent on a chosen criterion. Appropriate DR can deliver an optimal solution for some criteria. DRs are used to prioritise jobs and can find a good solution in real-time by creating a queue based on the selected performance measure. In case of a scheduling problem, when the necessary machine is freed up, a job with the highest priority is handled. DRs can consider many scheduling performances measures such as:

- Tardiness which measures delay of various SC operations and may include issues which are more complex to model such as loss of trust being direct result of delay and loses incurred due to delay such as fines paid to the customer.

$$T_j = \max\left(0, d_j - C_j\right) = \max(0, L_j)$$

where:

T_j − tardiness of the job j
d_j − deadline for the job j
C_j − completion time (makespan) of the job j
L_j − lateness of the job j

- Lateness measuring delay similarly to the Tardiness. The difference between tardiness and lateness is the fact that tardiness cannot be negative.

$$L_j = C_j - d_j$$

- Earliness measuring the time of the job j delivered before the deadline.

$$E_j = \max(0, C_j - d_j)$$

DR specify a sequence in which jobs should be carried on a given resource on a manufacturing floor. Rules can be static and do not change over time, or dynamic which are time dependent. The most common DRs are: Shortest Processing Time (SPT), which will sequence a job with shortest time first and will minimise the flowtime; FIFS rule which prioritises orders due to the time of its arrival; Earliest Due Date (EDD) which focuses on the due date of the order; this rule will sequence the jobs with earlier deadlines first and will optimise the schedule by minimising of tardiness. Crisp DRs focus on one input, which is used to assign priorities. Initial DRs, used for driving control for a scheduling subproblem are selected after considering which input parameters they observe. The parameters closely related to the uncertain demand such as time of the order arrival, due date and the size of incoming orders are selected. Description of selected DRs can be found in Chapter 6.

A great amount of research for deterministic single-resource scheduling problem was carried using crisp dispatching rules. However, many papers in recent years also considered uncertainty (Schuster Puga and Tancrez 2017; Petrovic and Kalata 2019; Lima et al. 2021). Using fuzzy DRs is possible to define to accommodate some types of uncertainty. In this case, if the rule has a high value the criteria, the priority of job is high, and if the value is low, the priority is low. It can be especially useful when historical data is unavailable or incomplete. An explanation of how fuzzy rules can be created and how are those implemented in SC can be found in the next subchapter.

3.3 Fuzzy reasoning

Uncertainty in SCs is unavoidable due to incomplete or inconsistent information presented by human experts or databases. Some uncertainty is also difficult to be measured directly. The SCs imprecise parameters such as production times, number of orders placed by customers, prices of products, holding costs and availability of other resources as lorries and times of deliveries are often fluctuating and cannot be described in a crisp, deterministic way. SC disruptions might have many sources and they can be grouped based on uncertainty origins. In the proposed research, addressed uncertainties will be introduced in a form of imprecise numerical values which can be represented by fuzzy sets, introduced in the following subchapter.

A fuzzy decision-making approach is used in this research for better understanding and handling of the uncertain nature of a general-structure SC. Models including this type of parameters representation are known as fuzzy programming or fuzzy logic-based models. This subchapter aims to familiarise the reader with a concept and benefits of fuzzy thinking. It includes an explanation on how fuzzy logic can be incorporated into SC modelling, which is especially useful when historical data is not available or when various data or parameters can be described only in linguistic form.

Fuzzy logic or fuzzy set theory was first proposed by Professor Lofti Zadeh in 1965 (Zadeh 1965). Although it was not widely used at its beginning, fuzzy logic gained acceptance of the technical community in late 1980s after being incorporated in Japanese house appliances controllers and cars designs and proved to be applicable to many problems (Negnevitsky 2002). First, fuzzy sets and differences between Boolean and fuzzy logic are introduced in this subchapter. Next fuzzy sets properties are explained. Inference and rule-based expert systems are described at the end of this subchapter.

3.3.1 Difference between Boolean and Fuzzy logic

Fuzzy logic allows parameters to be described in a more realistic way by introducing concept of a *linguistic variables* such as 'late' or 'more' and provide a representation which might be introduced as 'partially true' or 'quite false'. It is an extension of the traditional, Boolean logic, which permits only exact reasoning and describe variables with the binary true or false values. Fuzzy logic is useful to represent imperfect or incomplete data. Depending on where uncertainty is defined, different fuzzy models exist to represent it. Baykasoglu and Göçken (2008) introduced 15 different models available for each type of fuzziness source.

The main difference between the stochastic and fuzzy optimisation modelling is the way of presenting uncertain or unknown parameters. Classical logic allows variables to be introduced as part of a set, but in these cases a value of the variable belongs to the set with crisp borders and only to the one set at the time. Fuzzy programming allows representation of uncertain parameters in *fuzzy sets*. Fuzzy constraints, variables and objectives can be defined by their membership functions to a given fuzzy set. It means that there is a certain *membership degree* μ of belonging to the given fuzzy set. Differences between these approaches can be found in Figure 3.2 and Figure 3.3. As can be seen in Figure 3.2 in Classical logic representation, when the order x arrives, its membership degree is either equal to zero or one for one set at the time. Fuzzy sets presented in Figure 3.3 can be defined as a set with crisp boundaries in which membership degree μ is equal to any value between zero and one and depends on a shape of a given fuzzy set. Linguistic variables represented by fuzzy sets such as 'Early', 'Timely' and 'Late' give more information as the same time of arrival will be interpreted differently.

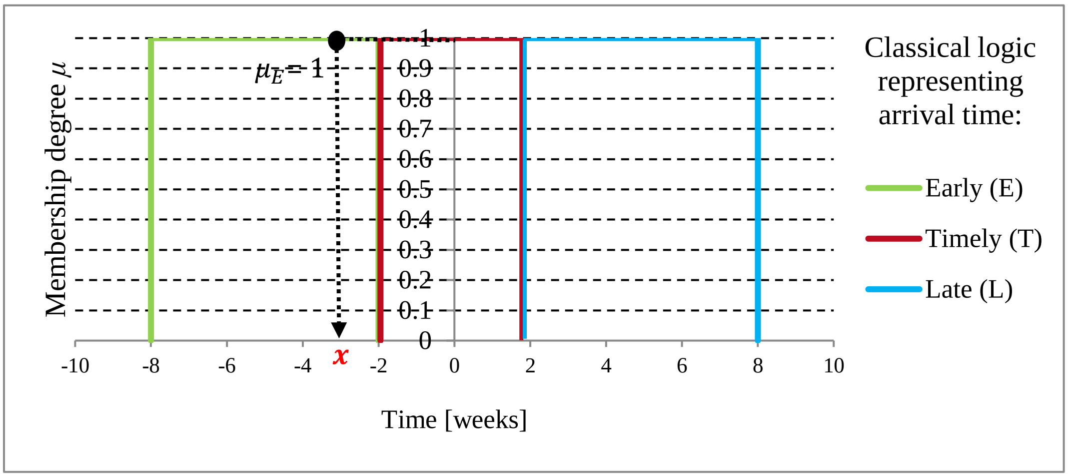

Figure 3.2 Example of three crisp sets: Early, Timely and Late regarding the arrival time of ordered goods

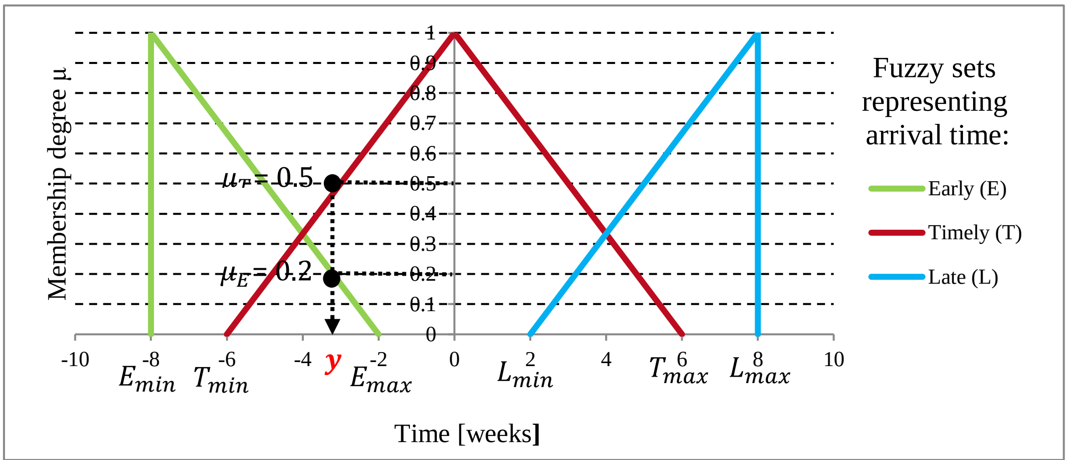

Figure 3.3 Example of three Fuzzy sets: Early, Timely and Late regarding the arrival time of ordered goods

In the presented example regarding the arrival time of order, when order x arrives 3 weeks before deadline, which is defined by time 0, in a Classical logic it belongs to the 'Early' set with membership degree $\mu_E = 1$ and in a Boolean reasoning penalty will always be high. The same time of arrival of order y in Fuzzy logic representation belongs to two sets with membership degrees equal to $\mu_E= 0.2$ and $\mu_T= 0.5$ for 'Early' and 'Timely' sets respectively. In the case of Fuzzy logic, the penalty will be between high and medium according to the rules and it depends on the membership degree to a given set. Table 3.1 shows the difference between characteristic function $f_E(x)$ of arrival time universe of discourse denoted as U for orders arrival denoted by x for Classical reasoning for the set E (Early) and fuzzy reasoning. Arrival of order y in the given universe of discourse in fuzzy logic has a membership degree μ for each set it belongs to.

Table 3.1 Difference between Classical and Fuzzy reasoning

Classical logic	$\boldsymbol{f_E(x): \rightarrow 0, 1}$ where: $$\boldsymbol{f_E(x) = \begin{cases} 1, if\ x \in E \\ 0, if\ x \notin E \end{cases}}$$ For any arrival **x** characteristic function $\mathbf{f_E(x) = 1}$ when **x** belongs to set **E** and $\mathbf{f_E(x) = 0}$ when **x** do not belong to set **E**
Fuzzy logic	$\mu_E(y): \rightarrow 0,1$ where: $$\mu_E(y) = \begin{cases} 1, if\ y\ arrive\ exactly\ 8\ weeks\ or\ more\ before\ deadline \\ 0, if\ y\ \notin E \end{cases}$$ $$0 \leq \mu_E(y) \leq 1, if\ y\ arrive\ 2-8\ weeks\ before\ deadline$$

As it can be seen in these figures, in the case of fuzzy logic sets of linguistic values may represent different shapes of fuzzy numbers and values of the sets can overlap. It is important to correctly transfer values to the linguistic values and to carefully define overlaps between these values. The clear advantage of fuzzy programming is that it gives the opportunity to the decision maker to evaluate parameters in a way more like human reasoning and transform them into fuzzy logic. It is also a good way to represent uncertainty, in addition to multi-level decision-making when negotiation and finding a satisfactory solution for many levels of a SC is necessary.

3.3.2 Rule-based systems and Fuzzy Inference Systems

Expert systems rely on experts' common sense, knowledge and experience while aim to solve a problem. Fuzzy reasoning based on a multi-valued logic has abilities to represent this experience in a set of mathematical expressions with values between 0 and 1 which gradually translate from 'completely false' to 'completely true' in a linguistic representation. As fuzzy logic is in line with natural human reasoning and how human describe quantitative values as; temperature as cold, warm and hot; time as early, timely, late or very late; height of a person as short, average and tall, using mathematical description of such a value can be useful for modelling of systems with uncertainty. Expert knowledge, historical data and uncertainty can be captured and represented by fuzzy rules. Fuzzy systems use IF-THEN rules to incorporate variables defined by words rather than numbers. Use of less rules to control systems allowed fuzzy rule bases to be faster than other expert systems (Cox 1999). Fuzzy rules consist of two parts: (1) IF part, which is called an *antecedent* of the rule and (2) THEN part, which is called a *consequent* of the rule. When the antecedent part of the rule has a value which membership degree is higher than 0, it will fire the consequent part of the rule to degree determined by the antecedent part. For example, in a system in which the input is a *fuzzy time of arrival* and output is a *fuzzy penalty* to be paid by a company, fuzzy rules can be used to create an expert system such as:

- **RULE 1:** $\boldsymbol{If}$ time of arrival is $Very\ early$ $\boldsymbol{Then}$ penalty is $High$
- **RULE 2:** $\boldsymbol{If}$ time of arrival is $Early$ $\boldsymbol{Then}$ penalty is $Medium$

Both parts of the rules may have multiple components such that there can be more than one antecedent and consequent.

Fuzzy Inference System (FIS) also known as Fuzzy Experts or Fuzzy Rule-based System is using fuzzy sets theory to define steps of reasoning process of converting vague and incomplete input information into a crisp output. Use of a linguistic value enables better understanding of system behaviour and provide flexibility of control. One of the first proposed FIS is Mamdani-type inference methodology (Mamdani 1975). Mamdani introduced a fuzzy expert system which was used to control steam engine and boiler combination. Inputs and outputs of the system were gathered from experts which in this case were experienced operators of an equipment. The rule-based system offered in this methodology is capable of mapping inputs to outputs with linguistic expressions and it is a simple way to include logical reasoning to inputs that are hard to relate precisely with outputs. The process requires execution of four base steps which are presented in Figure 3.4 and described below.

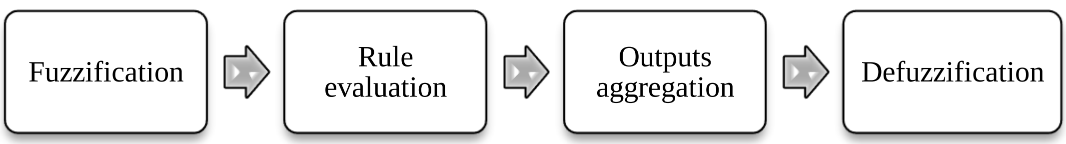

Figure 3.4 Mamdani-style FIS structure

Fuzzification is a process of converting crisp values into fuzzy into a form of fuzzy sets. These sets may be represented in many possible shapes such as triangular, trapezoidal or a Gaussian fuzzy number. Different shapes are appropriate for different types of parameters. A crisp input always must belong to the universe of discourse, for example, time of arrival from Figure 3.3 hast to belong between $E_{min} \leq y \leq L_{max}$. That guarantees that membership degree of belonging to a fuzzy set can be established. When input values are transferred into fuzzy values, the rule evaluation step can take place. *Rule evaluation* is responsible for taking fuzzified inputs with their membership degrees and evaluation of the antecedents. As mentioned above, both antecedent and consequent of the rules may have multiple parts. Antecedents consists of two logic operators, namely 'AND' and 'OR'. For example, rule can take a form of:

- **RULE:** $\boldsymbol{If}$ time of arrival is $Early$ $\boldsymbol{and}$ holding cost is $High$

  $\boldsymbol{Then}$ penalty is $High$

Operators differ and a consequent of the rule depends on the selected operators' definitions of the proposed system. The rule evaluation step aims to determine a value of the antecedent so that the *firing strength* of the rule can be established. The firing strength for Mamdani inference is related to the value of an input and the corresponding membership functions of the fuzzy set. Figure 3.5 represents a three fuzzy penalty values. When input is equal to 300, membership functions of *low* is fired with a strength 0.5 and membership function *medium* is fired with strength of 0.2. Then, depending on the rule operator in the antecedent part of the rule, either '*AND'* or '*OR'* rule operator is applied, which mathematically represents either an intersection or conjunction of two fuzzy sets respectively.

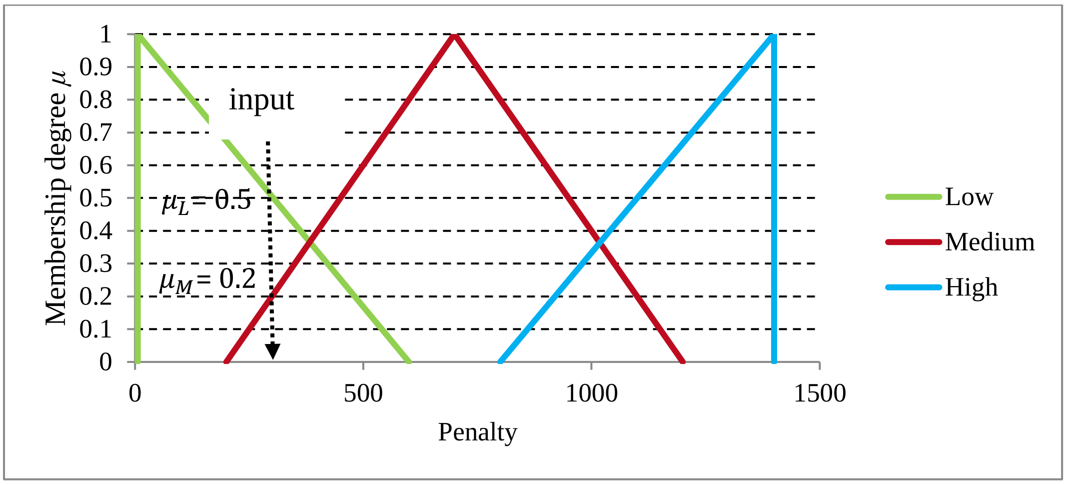

Figure 3.5 Firing strength of a rule in Mamdani-style inference system

This value is usually the minimum or maximum value of membership degrees belonging to fuzzy set in the antecedent part of the rules. The membership degree of the consequent of the rule cannot be higher than membership degree of the antecedent of the rule. *Aggregation of the rules* is the process of consolidation of consequents parts into one fuzzy set. The final step in FIS is a *Defuzzification* process which enables a transfer of the resulting fuzzy value into a crisp output. Centre of Gravity (COG) or weighted average methods can be used to obtain crisp outputs. Triangular fuzzy number are used to represent values of due date and slack due as they enable intuitive, computationally simple way of representing information in fuzzy environment (Zhang et al. 2012). According to Van Laarhoven and Pedrycz (1983) definition of triangular number x can be defined by a triplet x_1, x_2 and x_3 where membership function $\mu(x)$ is

$$\mu(x) = \begin{cases} 0, & x < x_1, \\ \dfrac{x - x_1}{x_2 - x_1}, & x_1 \leq x \leq x_2, \\ \dfrac{x - x_3}{x_2 - x_3}, & x_2 \leq x \leq x_3, \\ 0, & x_3 < x, \end{cases}$$

where $0 \leq x_1 \leq x_2 \leq x_3$ and x_1 stand for lower band and x_3 for the upper band of triangular number x. Fuzzy operator *AND* is used to obtain a single representation value of the antecedent part of a rule which can be later used to set priority. Operator AND enabling intersection operation between two fuzzy sets such that minimum value of membership is used as $\mu_{SHORT \cap SMALL}(x) = min[\mu_{SHORT}(x), \mu_{SMALL}(x)]$.

3.4 Multi objective GA

3.4.1 Evolutionary algorithms and introduction to GA

GA, based on principles of genetics, belongs to the Evolutionary Computing class of metaheuristics as it does enable use of the natural evolutions' concepts such as mutation, succession, and natural selection. There was a long way between nineteenth century works of J. Mendel and C. Darwin on the theory of evolution and 1970s, when biological research inspired engineers and mathematicians to create first evolutionary algorithm. A first use of GA and representation of potential solution in the form of artificial 'chromosomes' were proposed in work of Holland (1975).

A natural intelligence of selecting best genes is a product of evolution in a biological systems and computational models created by humans which aims to follow the same 'survival of the fittest' strategy. GA can be used to solve complex optimisation problems in many science fields, and it is gaining more popularity in recent years as problems increase in both, size, and complexity. Moreover, GA offers flexibility to model non-linearities and those are mostly used in cases, where exact algorithms are not able to find the optimal solution or when problem cannot be formulated in a mathematical notation due to complexity.

GAs represents an iterative process which aims to improve group of solutions it can find in the subsequent generations. After the optimisation problem is specified and encoded, the steps of GA metaheuristic are as presented in Figure 3.6.

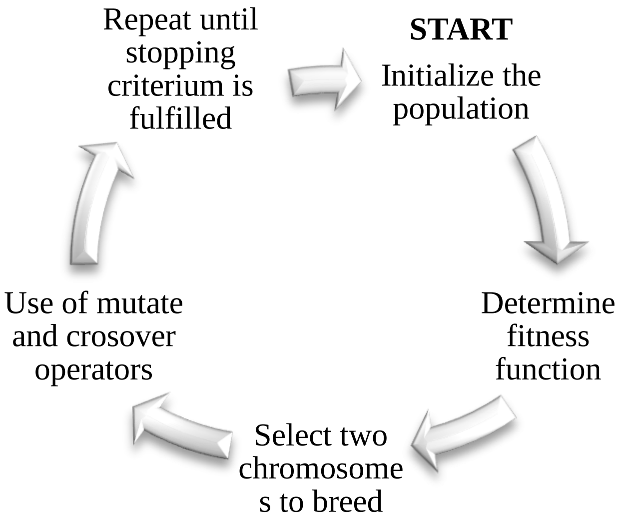

Figure 3.6 Steps of GA

The following steps consist of processes as observed in the evolution (Talibi 2009). The steps necessary to create this type of metaheuristic include:

- *Representation stage.* Chromosomes of the population must be encoded which (as in the nature) is individual for different problems. The chromosome represents a string of decision variables and it is set with a fixed length, where the length is equal to the number of decisions. Each decision is represented in a form of a single gene of a chromosome.
- *Initialisation of population*. Size of population which represents a group of solutions must be established. The initial population is usually randomly

selected for a given problem. In these cases, an initial solution may not be representative. Hence, initial solutions may benefit to be built by other algorithms such as simple heuristics. The initial population of solutions can represent diverse possibilities. Natural selection strategies which allow to produce offspring solutions will be able to be performed on initialised population.

- *Objective function selection.* This step of creating a GA is common to all metaheuristics. A fitness allows evaluation of the proposed solutions and is necessary for the selection process. Evaluated solutions can be ranked and scored based on this value which represent an objective of the algorithm and can represent multiple criteria.
- *Strategy of selection*. At this stage algorithm matches two chromosomes which will become parents for the two offspring solutions for the next generation. To preserve "survival of the fittest" strategy only the best solutions can breed. Different types of selection methods exist in the literature aiming to select the most suited parents, that includes:
    a. *Fitness proportionate selection* which can be used to rank individuals in relation to the population and this value represent probability of selection the single chromosome.
    b. *Rank-based selection* is second method concerning *relative fitness* of solutions. From a group of individuals, a chromosome with the highest fitness in the group is selected as the first parent. The same is repeated to select the second individual. This way of selection gives a better chance to solutions with lower ranks to be included.

Increase in diversity of the genetic material may lead to better solutions.

- *Reproduction strategy*. In this stage of the algorithm crossover operators as well as mutation probability must be established.

    a. *Crossover* is an operator used in GA to increase a genetic variation. Crossover is a recombination of genes between two chromosomes. It randomly selects the place of crossover and perform recombination procedure as presented in Figure 3.7. It is a necessary element to evolution and can lead to finding superior genes.

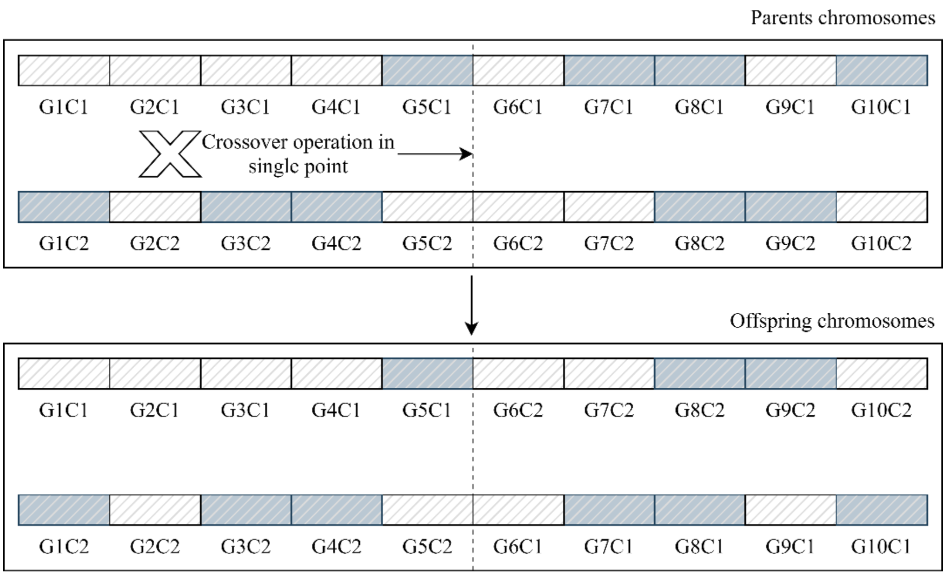

Figure 3.7 Crossover operation between two parents' chromosomes with recombination of genes

Single point crossover is one of the most popularly used operator and it is generalisation of n-point crossover. It selects a single point in chromosomes after which apply changes into chromosome structure. As can be seen in Figure 3.7 single point is applied between gene 5 and 6 in a two chromosome parents which each contains 10 genes. First five genes G1C1-G5C1 and G1C2-G5C2 remain unchanged in produced offspring chromosome while the second part of chromosome G6C1-G10C1 and G6C2-G10C2 are interchanged. It is different from uniform crossover (Figure 3.8), which does not divide chromosome into segments and each parent contributes equally to creation of offspring. In this case each gene is treated separately and will create genes which are very different from their parents in comparison to single point crossover.

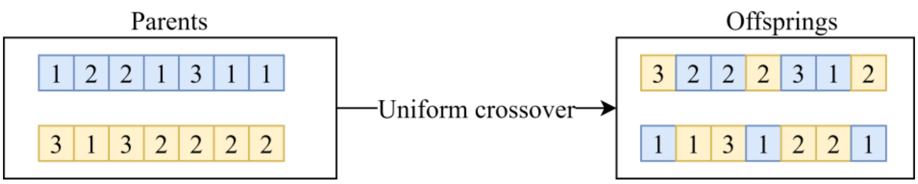

Figure 3.8 Uniform crossover

b. *Mutation* in a chromosome aims to prevent an algorithm solution to be trapped in a local minimum. As in the biological world a change in gene can improve fitness of an individual gene, but it also can lead to bad results. Representation of mutation can be found in Figure 3.9. Mutation is usually a very small change in a chromosome affecting one gene at the time. Mutation operator typically used range can stand for probability of occurrence of the mutation. According to Croydon (2001) the mutation factor should fit into range between 0.001 and 0.2 for each individual.

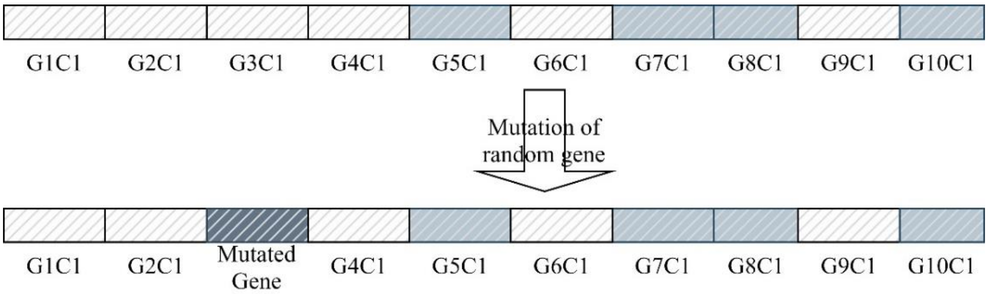

Figure 3.9 Mutation of the gene in a chromosome in GA

- *Replacement strategy stage*. This stage uses a fitness function in order to compare old and new solutions and decide which of them should be replaced.
- *Selection of stopping criteria*. The stopping criteria aim to prevent stagnation of solutions. Static and adaptive procedures are available. The static procedure includes a priori knowledge on the number of required iterations and can be set as a constant value. Adaptive procedures for selecting a stopping criterion can be based on some statistical value which does not improve or falls below the selected threshold.

Solutions proposed by the GA present well the trade-off between objectives. There are several optimisation techniques allowing selecting only one solution for multi-objective problems. Two of these techniques are described below. The (1) ideal and (2) preference-based optimisation procedures. Procedures can be seen in Figure 3.10 and Figure 3.11 respectively.

(1) - To find the ideal multi-objective solution to a problem, a two-step procedure is proposed, where:

- Step 1: Multiple trade-off solutions with a broad range of objectives values must be found,
- Step 2: One solution is selected based on higher-level information or subject expert knowledge

Figure 3.10 represents ideal solutions proposed by NSGAII, where red points represent all solutions from rank 1 and blue points stand for all other ranks.

Ideal multi-objective optimisation procedure does not require changing a problem domain into single objective but at the same time require an expert knowledge or additional information about SC strategy to make a final decision.

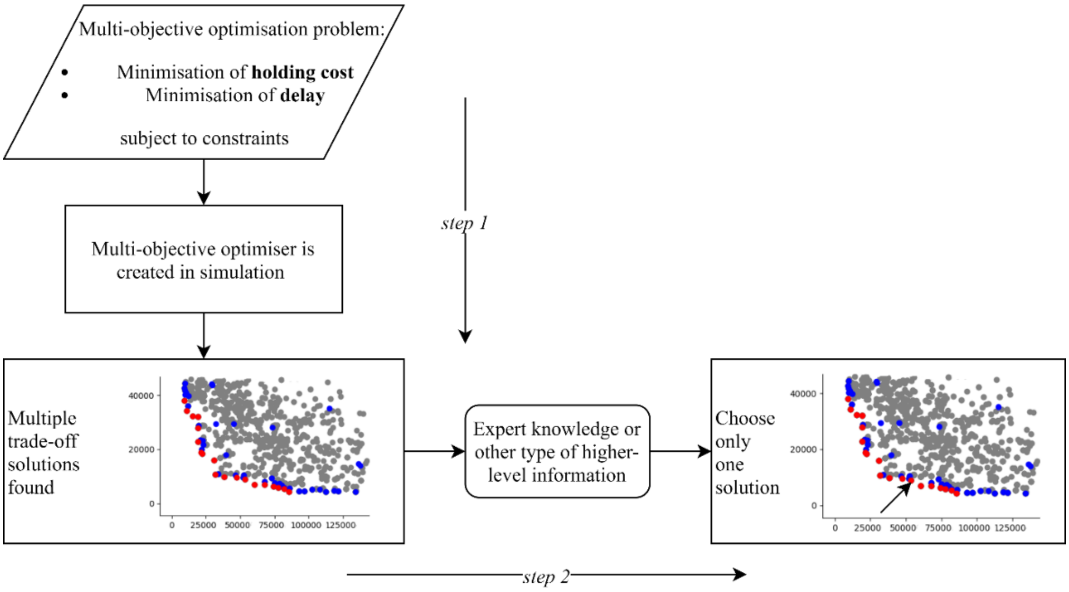

Figure 3.10 Ideal multi-objective optimisation procedure

(2) The higher-level information in Supply Chains usually are partial, subjective, experience-driven and might be non-technical. This information can be used to select only one solution according to the decision maker expertise. If higher-level information is not available, the weight of objectives which corresponds to preference factor, can be used to create a composite objective function. This method is transferring the problem from multi to the single-objective domain. This technique is based on the decision maker preference and can conduct experiments with different objectives ratios. The preference-based method is more subjective than ideal optimisation procedure as it depends on subjectively determined preference vectors during the first step. Ideal optimisation procedure uses higher-level knowledge to select one solution from many solutions available in Pareto

Front, while preference-based solution allows obtaining one solution. In the second case, any change of the preference vector should lead to different trade-off solution proposed by the algorithm.

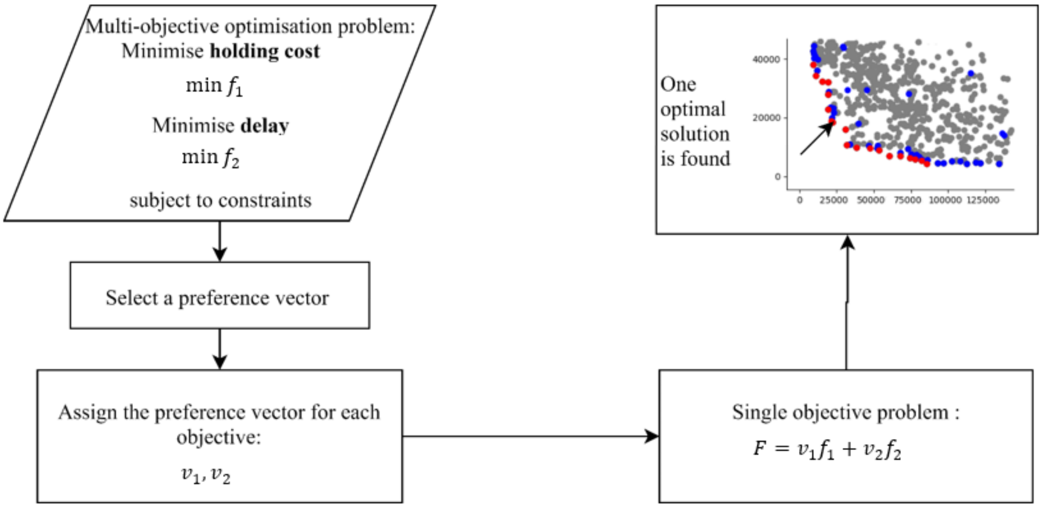

Figure 3.11 Preference-based optimisation procedure

3.4.2 Multi-objective NSGAII

Optimisation problems generally involve minimisation or maximisation of a specific objective function. This objective function is individual for each problem. Multi Objective Problems (MOP) and Multi objective Genetic Algorithm (MOGA) consider at least two objectives. The task of MOP is to minimise a function $F(x)$.

$$\min F(x) = \begin{cases} \min(f_1(x), f_2(x), \dots f_n(x) \\ x \in D \end{cases}$$

Where x represent a *feasible solution* and number of objectives $n \geq 2$, D represent *decision space* in a feasible region. Quite often two or more objectives of a considered problem are in conflict e.g., if the aim is to increase production and thus profit, it might conflict with an objective aiming to minimise carbon emission or other environmental criteria. Multi-objective problems require simultaneous optimisation of more than one objective, which usually do not have one global solution and some type of trade-off is considered.

Multiple solutions might be proposed by the algorithm to these type of problems as different objectives can have different weights. In most cases which problem is transferred from multi-objective into a single objective as in Figure 3.11. It can lead to losing multi-objective nature of considered problem. The other way is to present solutions using pareto optimality concept (Horn et al. 1994). Pareto front which consists of solutions where no individual cannot be improved without negatively affecting another objective i.e., improvement of one of the objectives leads to worsening other objective. In this approach the final choice relies on the decision maker. Without additional subjective preference information all Pareto solutions are considered equally good. Non-Dominated Sorting Genetic Algorithm II (NSGAII) proposed by Deb (2002) use the concept of Pareto optimality and dominance for its search process. It allows to treat objectives separately. *Dominance ranking* is used in this research identify how many solutions in population are not dominated. Dominance concept refers to the fitness assignment procedure, which determines when one solution is dominated by other solution. This happens if at least one of the fitness functions can be improved without worsening other considered fitness functions. When the solution is not dominated by other solutions (i.e., solutions cannot be optimised without negatively affecting objectives), those are assigned rank 1, then solutions with this rank are removed from the population. In next iteration of NSGAII a rank 2 is assigned to the solutions which are not dominated by any other solutions and again removed from the population. The procedure is repeated until the entire population is ranked.

For better understanding how the dominance ranking is determined, a Figure 3.12 placed below presents a population of six solutions for multi-objective optimisation problem with two fitness functions. In a case where it is assumed that fitness function representing output values are hc (which standing for holding cost) and d (which standing for the delay) should be minimised, a $ProposedSolution1(hc_1|d_1)\ dominates\ ProposedSolution2(hc_2|d_2)$ under two certain conditions:

- $(hc_1 \leq hc_2\ and\ d_1 \leq d_2)$ both fitness function values of one solution are smaller or equal to another solution and
- $(hc_1 < hc_2\ or\ d_1 < d_2)$ at least one of the fitness functions is smaller

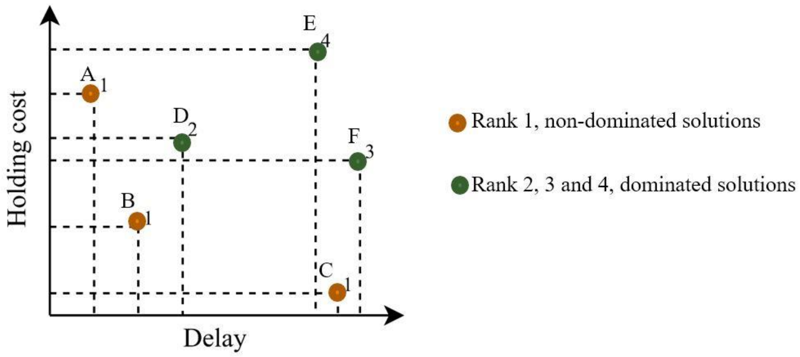

Figure 3.12 Example of multi objective solution with holding cost and delay objectives with three solutions of rank 1 and one solution of rank 2,3 and 4

Solutions of NSGAII can be seen in Figure 3.12 and letters *A*-*F* are used to differentiate them. Any solution with rank 1 dominates all solution with rank 2. Rank equal to 1 means that a proposed solution is not dominated by any other solution. Solution B has a higher rank than solution D as $hc_B \leq hc_D\ and\ d_A \leq d_D$ and solution C has a higher rank than solution F as $hc_C < hc_F$ and no other solution dominates solution C. The dominance rank is used to score solutions.

3.5 Supply Chain Modelling

3.5.1 Simulation and analytical approaches to SC modelling

Over the years, with increasing size and globalization of a SC, scheduling and inventory control have become very complex and challenging problems. As stated in the literature review chapter, SC optimisation aimed to combine different objectives and constraints for SC inventory planning and scheduling. When optimising large data set, heuristics are widely used to reduce the size of the problem. Decomposition of the problem into very small and simple sub-problems enables faster decision making and finding potentially optimal solutions.

Analytical approaches for solving large SCM problems can be categorised into: (1) *planning-based* and (2) *demand-driven* approaches. These are respectively mathematical programming and simulation approaches. Well established mathematical theory exists for planning-based methods category. In this type of approach, real-world problems are usually introduced as a centralised system with multiple simplifications of complex infrastructure and connections between echelons. Methods such as LP, Mixed-Integer LP (MILP), deterministic and stochastic mathematical programming lead to creating steady and reliable models. These types of methods guarantee optimal solution while information is very often shared globally across the chain.

Demand-driven approaches including Simulation Optimisation, Agent-based models, and Discrete Event Simulation (DES) provide more flexibility and capture better the dynamic nature of SCs. These approaches provide realistic representation of a problem, but they also require additional development of efficient optimisation strategies. There are several benefits which occurs from simulation. This approach enables structured approach for data collection and sensitivity analysis, what is important for multi-site and cluster evaluations, so results can be compared for various scenarios. Simulation is amongst methods to solve this type of SC problems since simulation-mathematical approaches returning more realistic representation of the SC system.

Available simulation software's such as ARENA, AnyLogic and SIMIO are good for their purposes. However, they are limited to what they have been programmed to achieve and there is much lower flexibility than in framework proposed in the next chapter. Available software allows observation of behaviour of any given SC, but they are restricted in terms of modelling some of the problems and untypical solutions, representing uncertainty and often they operate as a black box simulation. The complexity of SC system is associated with interconnections between echelons and integration approaches as presented in Figure 3.13.

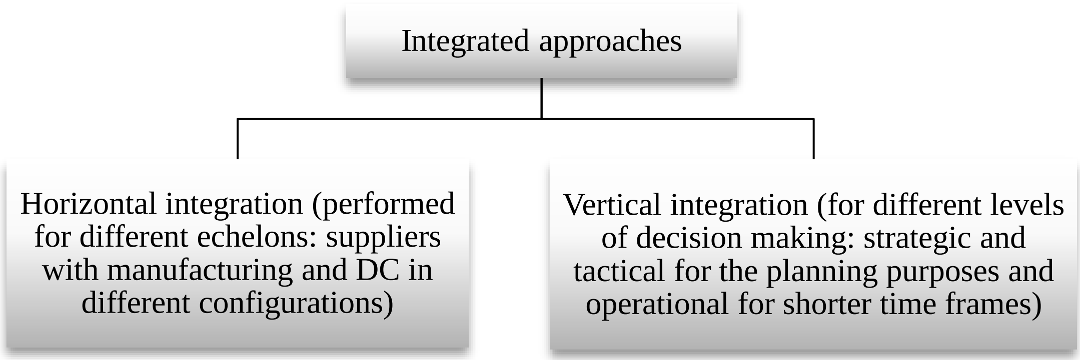

Figure 3.13 SC integration approaches

As a model flexibility is important, developing of a new, problem-focused software is preferable. The decision about creating a new simulator enables easy changes for the problem structure, observation of changes for different, customised

KPIs, access to all parameters changing over time and offers possibility of further development, which could be limited by other available software.

Proposed simulator supports a novel simulation model in a form of a scenario for two dynamic subproblems and it can deliver control-scheme solutions for multiple echelons.

3.5.2 Inventory control modelling for SC

Holding cost of inventory can become the most expensive cost of SC. As such, an inventory control which balance between demand and supply, plays a crucial role in a well-managed SC. Inventory control exists in many forms in SC echelons and can be modelled and executed in several different ways.

Among inventories there are purchased raw materials inventory and inventory of finished and semi-finished products. Several inventories may exist in one tier of echelons. Inventory control can take into consideration several objectives (Franzelle 2001) and among those most often used are: *(i)* Predicting and improvement of the forecast of orders of products, *(ii)* Reduction of delivery time on one or more echelons of considered SC, *(iii)* Minimisation of various costs such as ordering, penalty or holding cost, *(iv)* Improvement of visibility of kept inventory. Inventory serves as a buffer between production and distribution providing additional stock which can guard robustness and flexibility of the SC in case of prices fluctuation, changing demand or late deliveries. Inventory control decisions variables usually include two parameters, which are *when* and *how much* to order (Hugos 2003).

One of the subproblems defined in this research is an inventory control problem. There are several methods of replenishing products. One of the often-applied methods is to order products in real time equal to the sold amount as can be seen in Figure 3.14. Continuous Replenishment Program (*CRP*) is a concept of inventory replenishment that can reduce the orders loss, inventory holding cost and stock level, and entire cost of the SC. CRP for the considered SC requires two decisions: (i) how high stock level of products or elements should be and (ii) orders quantities when products or elements will drop below previously selected level. CRP uses the ordering point method (presented in Figure 3.15) in which products quantities are calculated for order replacement. Hence, when the inventory level drops into predetermined level, replenishment orders will be equal to a difference between the required level and the available stock.

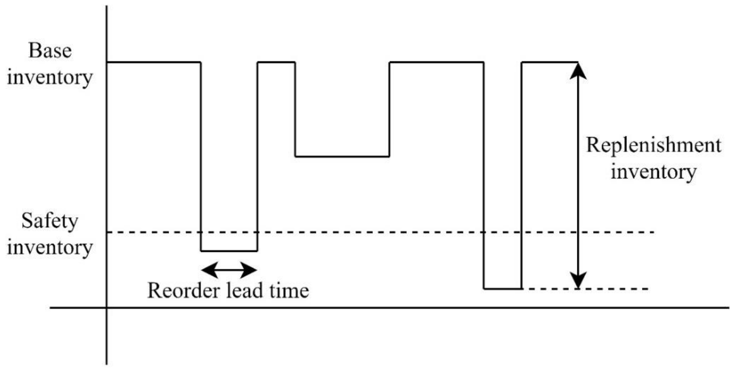

Figure 3.14 Basic inventory method (CRP)

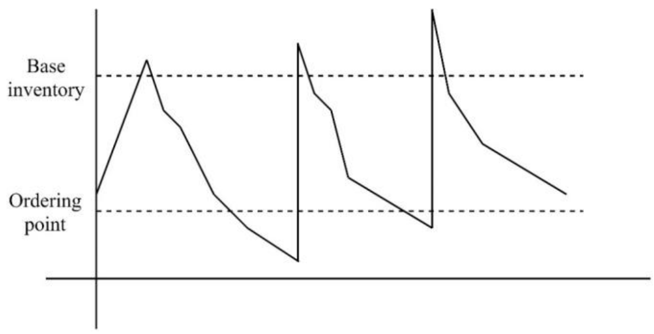

Figure 3.15 Ordering point method

When the constant demand rates can be identified in the model Economic Order Quantity (EOQ) method is often used. For a deterministic cases EOQ guarantee minimal ordering and holding costs by determining order quantities within given time horizon.

3.6 Discussion and Conclusions

Simulation enables introducing many uncertainties which in comparison to static representation in mathematical programming are dynamic in real-world plants. Simulation of defined problem gives opportunities for observation long-term decision effects, observation of behaviour of a SC over time and allow identification of potential issues and bottlenecks. In case of obtaining results, it allows analysist to make hypothesis about behaviour and implement various scenarios to validate it.

Simulation facilitates finding alternative solutions and better understanding of a system and can help with a risk mitigation. Modelling of the considered SC includes creating a simulation framework in Python, use of fuzzy logic and FIS for representing the uncertainties found in the SC and optimisation with algorithms including both; construction and improvement heuristics, to address a complex and multi-layered problem. Modelling of complex, dynamic system in more intuitive way can lead to creating intelligent control system for SC environment. Fuzzy sets allow mathematical representation of human reasoning and in this study, they are used to describe SC's parameters uncertainty. Representing incomplete or unknown data using expert knowledge is one of main advantages of fuzzy logic. This can lead to significant improvement of control and it guarantees a quick response to quickly changing input parameters, which is crucial in complex SCs. Another advantage includes flexible nature of FIS and its ability to deal with nonlinearities of the model.

Use of simple DRs for scheduling and a fixed CRP for inventory control allow testing of SC under various parameters changes. Although use of these methods introducing control-scheme for multiple echelons they are unable to provide optimal solution. NSGAII was selected to solve the problem as it is known for providing diverse solutions for multi-objective problems, which is essential for defined problem.

4 PROBLEM DESCRIPTION

4.1 Introduction

Research carried out focuses on scheduling and inventory control for a four-layer, multi-product SC. Proposed decentralised decision-making allows echelons to make independent decisions and addresses a reality of limited information sharing between SC participants. One of the goals of this research is to propose a methodology which allows for simultaneous scheduling and inventory control of SC echelons, including Supplier, Manufacturer, Distribution Centre and Customers. In this chapter, a SC problem statement will be given to guide a new SC simulation model design which incorporates multiple problems, multiple echelons and multiple parts and elements produced in the SC with consideration of uncertainty of demand and no information sharing policy between echelons. This chapter aim is to introduce notation used throughout the work and will be followed by a description of implemented Simulator framework in Chapter 5. The description of decision-making and SC performance under basic control-scheme can be found in Chapter 6.

This chapter is organised as follows. Subchapter 4.2 introduces a problem description, notation used throughout the work and overview of SC's echelons. Finally, model formulation including description of SC behaviour, decision variables, assumptions, and performance indicators of the considered SC of the model are given in subchapter 4.3.

4.2 Description of the SC problem

4.2.1 Formal problem description

Integrated scheduling and inventory control model and simulation framework for a general structure SC is considered. General SC definition was determined based on Sawik (2014) SC structure which consists of Suppliers, Manufacturer, Distribution Centres and Customers with internal operations and transportation channels between each other. In general structure SCs it is only possible to process products, which have been processed by lower tier echelon and have been delivered to the next echelon. The problem in this study considers scheduling and planning for multi-echelon, multi-product and multi-element SC showed in Figure 4.1, considered SC problems in a presence of uncertainty in demand are inspired by the complexity of a real-world SC. Provided model enable introduction of uncertain parameters using fuzzy logic and allows dynamic nature of incoming orders to be considered. The following subchapter aims at providing an overview of considered SC. Detailed descriptions of SC processed, connections between echelons, information about product flow are described below. Formally, the problem can be defined as follows: given a SC structure, find a control scheme allowing the schedule of production and inventory replenishments that satisfies delivery of all orders with minimum holding cost and delay.

Orders in this Supply chain:

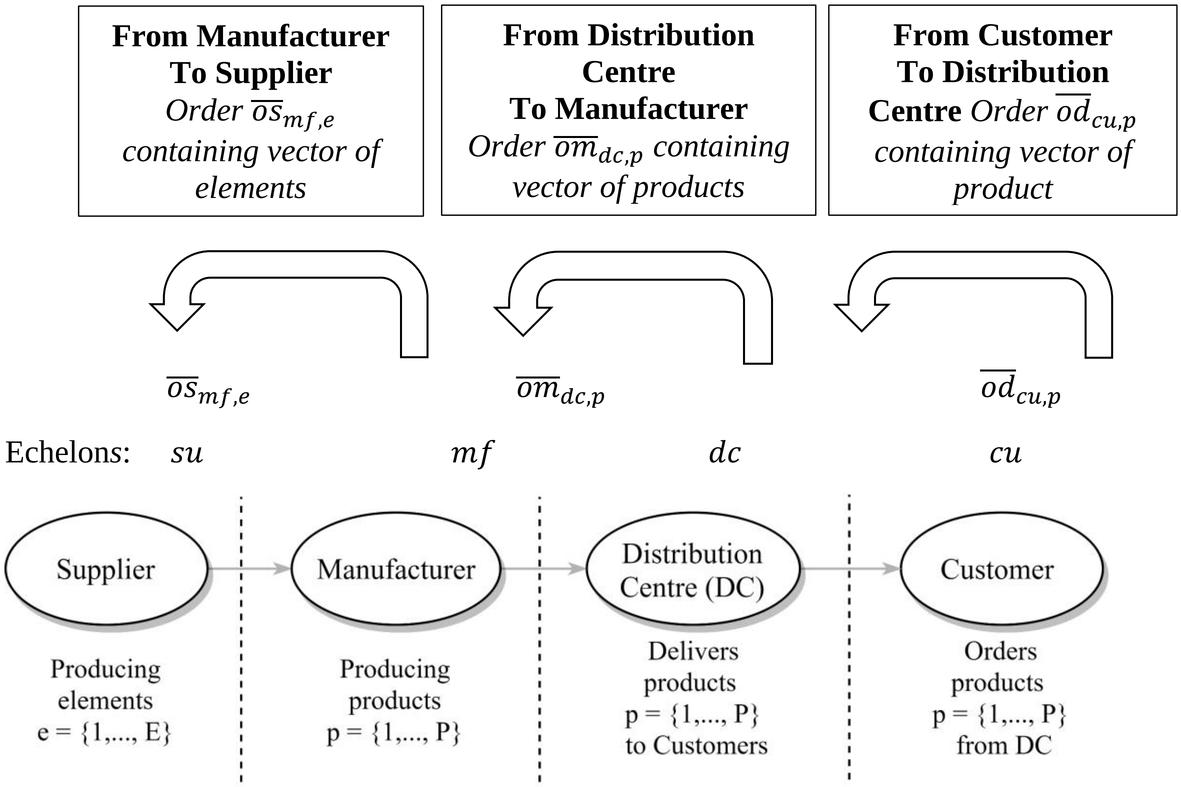

Figure 4.1 Structure of the considered SC with orders description

4.2.2 Notation

The following notation is used in a considered SC problem. Parameters used for specific echelons i.e. a Customer, Distribution Centre, Manufacturer and a Supplier echelons are introduced in subchapters 4.2.2.1, 4.2.2.2, 4.2.2.3 and 4.2.2.4 respectively. Indices:

- $cu = 1,\ldots,$ CU – Index of Customer
- $dc = 1, \ldots, DC -$ Index of Distribution centre
- $su = 1, \ldots, SU$ – Index of Supplier
- $ms = 1, \ldots, MS$ – Parallel machines index for the Supplier echelon
- $m = 1, \ldots, M$ – Machines index for the Manufacturer echelon
- $p = 1, \ldots, P$ – Index of product
- $\bar{B}_p -$ Matrix of bill of material of all elements e for product p

- $p_e - \begin{cases} 1\ if\ \ \bar{B}_p\ contains\ an\ element\ e \\ \qquad 0\ if\ otherwise \end{cases}$
- $e = 1,\ldots,$ E – Index of elements
- $do - 1,\ldots,DO -$ Index of orders placed by Customer cu to Distribution Centre dc
- $mo - 1,\ldots,MO -$ Index of orders placed by Distribution Centre dc to Manufacturer mf
- $so - 1,\ldots,SO -$ Index of orders placed by Manufacturer mf to Supplier su
- $t -$ Discrete time in the simulation
- $\overline{od}_{cu} -$ Order placed by Customer cu
- $\overline{om} -$ Order placed by Distribution Centre dc
- $\overline{os} -$ Order placed by Manufacturer mf

Following sub-chapters consists of a description of each of the considered echelon. The goal of the proposed research is a simultaneous making of scheduling decisions with inventory control decision along SC. These decisions are different for different echelons as there are different types of products and different tasks for different echelons. The explanation of processes happening in the echelon in the form of flow charts, inputs and outputs description are also given.

4.2.2.1 The Customer echelon

The Customer is a basic echelon that does not own any resources and does not model any internal processes. Its only task is to make orders to the assigned Distribution Centre. The $\overline{od}$ is a list of pre-defined random orders. Each order placed specifies the time at which order is to be made, order due date and its contents. The orders list is generated in the following way. First the time between orders is generated according to the exponential distribution with a given parameter λ. If that time exceeds the specified final time, the order generation is finished.

Otherwise, the product quantities for the order are chosen according to a discrete uniform distribution over the specified range. The range for each product is a set separately. Next, the order due date is determined in the following way. The *base time* $bt_{cu,od}$ necessary for order completion is calculated as follows:

$$bt_{cu,od} = \sum_{p} \overline{ptm_p} \times \overline{od}_{cu,p}$$

where the sum index product includes all products in the order. The $bt_{cu,od}$ is then extended by a random increase of time to provide enough time for order processing and required time $rt_{cu,od}$ for production is calculated.

$$rt_{cu,od} = bt_{cu,od} \times (1 + \frac{boost}{100})$$

The order due date is subsequently calculated as follows:

$$dod_{cu,od} = ot_{cu,od} + rt_{cu,od} + tt_{cu}$$

Finally, the due date is rounded up to full hours. The order is added to the list. The above steps are repeated for all the orders.

4.2.2.2 Distribution Centre echelon

The Distribution Centre is a high-level echelon. Distribution Centre is an echelon responsible for collecting orders and delivering finished products to Customers. It keeps an inventory of products, so they can be delivered to the Customers by using available set of lorries. After Distribution Centre receives an order from Customer, the activities in the entire SC start. Customer can send more than one order with different due date which consists of demand on one of the products or multiple products. Each lorry has a fixed capacity in terms of volume. Each lorry can only deliver a single order at the time, but big orders may be distributed among multiple lorries as split orders. If not all of the products are available at the scheduled time, a delay is reported. Once all required products are ready to be delivered to the Customer, the lorry loading is started. If the lorry does not complete its journey before it is scheduled for another trip, that trip becomes delayed. If the completed order does not fit entirely in the fixed cargo space of the lorry has to be split into multiple lorries. In that case the Customer only records the delivery of the completed order when the last part of that order is unloaded. All parameters for this echelon are described in Table 4.1.

Table 4.1 Distribution Centre parameters description

Parameter	**Description**
$bt_{cu,od}$	The base time necessary for order completion
$\overline{od}_{cu}$	Order placed to the distribution centre by the customer cu which contains an array of required products p
$dod_{cu,od}$	Due date of the order $\overline{od}_{cu,p}$ of customer cu
$Dod_{cu,od}$	Actual time of the order $\overline{od}_{cu,p}$
iid_p	Initial inventory level of product p in the distribution centre inventory
id'_p	Maximum inventory level of product p in the distribution centre inventory
hd_p	Unit holding cost of product p in the distribution centre inventory
$ot_{cu,od}$	Order time of order od placed by Customer
v_p	The volume of the product p
lor	Number of the available lorries
st_{lor}	Space of the lorry lor
tt_{cu}	Transportation time between the Distribution Centre and Customer
ym_p	The reorder point level of stock. When the level drops below this point order $\overline{om}$ (defined in Table 4.2)must be placed.
$Z_{od,lor}$	Quantity of order $\overline{od}_{cu}$ allocated to the lorry lor

The decision of the Distribution Centre seeks to determine inventory level and quantity of products to be ordered from the Manufacturer and to schedule a limited number of lorries to ensure that Customers' orders are delivered. The problem in this echelon consists of two subproblems:

- Inventory control subproblem which seeks to determine how much products should be kept in inventory considering that the Distribution Centre must rent the inventory space, so keeping too much inventory generates high holding cost hd_p and products can become obsolete. The time of products hold in the inventory should be minimised. CRP is used for inventory control. Decisions to be make are to determine *reorder point,* which is a level of the inventory at which new order should be placed, such that if level of a product falls beyond this point *order quantity* $\overline{om}$ should be placed. Planned inventory levels from the first subproblem should be balanced with allocation of lorries and their capacities

- Scheduling and allocation subproblem which seeks to determine the allocation and schedule of available lorries, where capacity and number of lorries are limited resources. This task considers allocation of collected orders $\overline{od}_{cu}$ to available lorries capacities in such a way as to consider unused space of a lorry. It is possible that in the case of larger orders more than one lorry must be used. A decision to be make are a schedule of orders to be send to the Customers by prioritising orders. The decision-making applied in this echelon is presented in Chapter 6.

4.2.2.3 Manufacturer echelon

The Manufacturer is a middle-tier echelon. It maintains an inventory of raw materials provided by the Supplier and produces finished products on a set of independent machines. Table 4.2 presents parameters used in this echelon. The decision to be made are machine schedule and delivery schedule. It also makes orders which are delivered by the Supplier to replenish inventory. The machine schedule determines when each machine will start production of a given product required for fulfilling a given order. The delivery schedule determines when any given order can be sent to the Distribution Centre.

Table 4.2 Manufacturer parameters description

Parameter	Description
$\bar{B}_p$	Matrix of bill of material of all elements e for all products p in form of the matrix $\bar{B}_p = [B]_{E \times P} = \begin{bmatrix} e_{1,1} & \cdots & e_{E,1} \\ \vdots & \vdots & \vdots \\ e_{1,P} & \cdots & e_{E,P} \end{bmatrix}$
$dom_{mf,dc}$	Due date of order $\overline{om}$ delivered by manufacturer to distribution centre
$\overline{ptm_p}$	Processing time of product p on Manufacturer's floor in form of vector $\overline{ptm_p} = [ptm_1, \dots, ptm_P]$
iim_e	Initial inventory level of element e in the manufacturer inventory
im'_e	Maximum inventory of element e in the manufacturer inventory to the distribution centre
IM_p	Ready to ship final products p level waiting in the manufacturer inventory
hm_e	Unit holding cost of element e in the manufacture inventory
HM_p	Unit holding cost of keeping final product p in the manufacturer inventory
$k_{e,su}$	Quantity of element e delivered by supplier su
$\beta_{su,e}$	Information about elements e produced by supplier su in a form of array $\beta_{su,e}$ = [${}^{0}/_{1_1}$, … , ${}^{0}/_{1_E}$]
ys_e	Reorder point level, when the level of stock drops below this point an order $\overline{os}$ must be placed

Manufacturer echelon is responsible for collecting and scheduling incoming orders $\overline{om}_{dc,p}$ received from its distribution centres dc. After receiving an order from a Distribution Centre, the Manufacturer must check if there is enough inventory of products p, IM_p, which could satisfy demand and be shipped to the Distribution Centre. In the case of insufficient number of products, the production of product p must be scheduled. There are several subproblems considered at this echelon.

- The first subproblem is to replenish the inventory by placing orders $\overline{os}$ to Suppliers. Each Supplier delivers different type of elements and only one Supplier is available for each type of elements. Manufacturer must make two decisions regarding ordering of elements. The first is to determine order quantity $\overline{os}$ of element e to be ordered from the Supplier, the second is reorder point ys_e.
- The second subproblem is to schedule a production of orders and allocate tasks to available resources i.e., parallel machines on the Manufacturer floor. Manufacturer's machines are limited resources. A parallel machine scheduling problem requires scheduling of n jobs on m machines. Each job operations must be performed. The job n can start being processed on machine m only when machine m is free. Each operation done on the Manufacturer floor on product p has known processing time $\overline{ptm}_{m,p}$.

When the order is finished, it is packed and shipped to the Distribution Centre instantly. To schedule orders in a form of a Gantt Chart a decision-making procedure must be defined for this echelon. The further explanation can be found in Chapter 6.

4.2.2.4 Supplier echelon

The Supplier is a lowest tier echelon. It does not consume resources from another echelon and provides raw materials (elements). The elements are produced by a set of independent machines. Each machine can only produce elements in batches of a fixed size. Each machine can produce any of the elements offered by the Supplier, however, changing from one element type to the another incurs a setup time. Each product produced by the Manufacturer and delivered by the Distribution Centre consists of elements which are produced only by Supplier echelons. Bill of material $\bar{B}_p$ defined in Table 5.6 contain information about which elements are necessary for specific product. Table 4.3 presents parameters used in this echelon.

Table 4.3 Supplier parameters description

Supplier	
Parameter	Description
$b_{e,su}$	Minimum batch of element e which can be ordered from supplier su
$et_{e,su}$	Unit production time of element e from supplier su
$dos_{su,os}$	Due date of the order $\overline{os}_{mf,e}$ delivered by supplier su to a manufacturer
hs_e	Unit holding cost of element e in the supplier su inventory
sut_{su}	Set-up time of machine ms in supplier su echelon
tt_{mf}	Transportation time between the Supplier and Manufacturer

In the proposed SC, the Supplier is responsible for delivering elements to the Manufacturer. Elements are required by the Manufacturer to produce products which are later delivered to Distribution Centre and to Customers. The Manufacturer places the order $\overline{os}$ which can contain multiple elements. Each

Supplier produces only certain types of elements, so problem of supplier selection is not considered in this research. The Supplier is producing elements continuously on parallel machines, where each machine must be allocated to produce required elements. The machines on the Supplier floor may be identical or nonidentical. In this study it is assumed that machines are identical. There are several machines on the Supplier production floor, and one subproblem is considered for this echelon.

- Scheduling of identical parallel machines. The Supplier produces element e. There are n jobs to be processed on m identical machines which run in parallel. Each job must be processed by one of the machines. A set up time sut_m must be considered when the machine must change between production of different elements.

It is assumed that when the order $\overline{os}$ is finished, it is packed and shipped to the Manufacturer instantly. The whole order $\overline{os}$ must be delivered to the Manufacturer at once (there is no splitting of orders). The Supplier's echelon is notified when: a new order arrives, the execution of a task on a given machine is delayed, the delivery of the completed order is delayed or when the level of the finished elements storage is changed. This choice of events effectively allows to implement both *make-to-order* and *make-to-stock* policies.

Both machine schedule and delivery schedule are generated and explained in Chapter 6. The machine schedule determines when each machine will start producing a given element and for how long. The delivery schedule determines when any given order can be sent to the Manufacturer echelon.

4.3 Model formulation

The SC contains four echelons and several types of orders, which are sent from higher tiers to lower tiers echelons. Initiation of order driven SC is order $\overline{od}_{cu}$ received by a Distribution Centre and placed by the Customer. Customer's order $\overline{od}_{cu}$ has a specified due date $dod_{cu,od}$ and contains information about quantity of all ordered products. This due date is considered for the entire order which cannot be split and must contain all ordered products before they can be delivered to Customer cu. The Distribution Centre either has enough stock of all ordered products p to satisfy Customer demand fully or it does not. In the case when there is not enough stock, Distribution Centre must place order $\overline{om}$ for products p which will be produced and delivered by the Manufacturer. Order $\overline{om}$ must be sufficient to fulfil an inventory of the distribution centre to the level specified in the inventory policy and by that satisfy demand for the Customers' orders $\overline{od}_{cu}$.

The Manufacturer echelon produces all types of products p and this echelon is described by three main parameters. The first parameter refers to inventory of element e, im_e. The second parameter describes a flow shop where products p are produced on each of the machines m. To produce the product on the Manufacturer floor, elements specified in a bill of material $\bar{B}_p$ must go through production on one of parallel machines to be assembled into a final product p. The product must spend a certain time on a Manufacturer machine. Production of product p on machine m have assigned processing time $\overline{ptm}_p$. The third parameter describes the Manufacturer is an inventory IM_p of product p before it can be packed in orders and delivered to the Distribution Centre. This inventory contains only products which are assembled for a specific order $\overline{om}$. The Manufacturer does not keep any additional stock in this inventory.

After the Manufacturer receives order $\overline{om}$ from Distribution Centre dc, it checks bill of material $\bar{B}_p$ of each product p, which is used to calculate how much elements e must be used to produce enough of product p to satisfy Distribution Centre demand. Manufacturer can use elements from inventory im_e or order elements e from Supplier su and start production after delivery of the ordered elements e. In the case of ordering elements, variable $\beta_{su,e}$ gives information to Manufacturer mf about availability of element e at Supplier su. If Supplier delivers the ordered element, it is assumed that all the ordered elements will be delivered in good quality and in the same quantity as ordered amount. Each Supplier su delivers different type of elements so the Manufacturer sends the order to the Supplier who is producing the required elements.

4.3.1 Key Performance Indicators

Two key performance indicators (KPIs) are considered: the total holding cost and the delay of delivering orders to the Customer.

Holding cost KPI

$$H_{dc} + H_{mf}$$

Total delay KPI

$$\sum_{od} D_{od}$$

The task of proposed metaheuristic is to find optimal fuzzy dispatching rules for integrated control of scheduling and inventory control problems for all SC's levels. Inputs of the SC are highly uncertain (Salem and Haouari 2017).

4.3.2 Assumptions of the proposed model

Assumptions of the proposed model are listed below.

- Each product p is independent and requires going through the Manufacturer machines m.
- Each product p can have different processing time.
- Each machine m and ms can process one product at the time.
- Times of production, delivery between echelons and set-up times are known.
- Uncertainties in demand are taken into consideration. It includes time of arrival and varying size of incoming orders.
- Elements can be produced on any of the machines in the Supplier's production floor. A change between different elements requires set up time.

4.3.3 Decision variables

To summarise, the decision made by each echelon are listed below.

- $\overline{om}$ – Orders for Manufacturer placed by distribution centre dc which contains p products in the form of the vector $\overline{om} = [om_1, \ldots, om_P]$
- ym_p – The reorder point level of stock. When the level drops below this point order $\overline{om}$ has to be placed

- $\overline{os}$ – Orders for Supplier su placed by the Manufacturer mf which contains e elements in the form of the vector $\overline{os} = [os_1, \dots, os_E]$
- ys_e – Reorder point level, when the level of stock drops below this point an order $\overline{os}$ must be placed
- $\overline{od}_{\mathrm{cu}}$ – Orders for Manufacturer placed by the Distribution Centre dc which contains products list in the form of the vector $od = [od_1, \dots, od_P]$
- yd_p – Reorder point level, when the level of stock drops below this point an order $\overline{od}$ must be placed
- $Z_{od,lor}$ – Quantity of order $\overline{od}_{cu,p}$ allocated to the lorry lor

4.3.4 Variables used for performance measures

Variables used in this model are listed below. The delay can be calculated for cases where actual time of delivery was later than due date $dod_{cu,od}$ specified by the Customer.

- D_{od} – Delay of the order $\overline{od}_{cu}$

$$D_{od} = \begin{cases} Dod_{cu,od} - dod_{cu,od} \ \boldsymbol{if} \ Dod_{cu,od} - dod_{cu,od} \geq 0 \\ 0 \ \boldsymbol{otherwise} \end{cases}$$

- $id_{p,t}$ – Inventory level of product p in the Distribution Centre inventory at the time t
- $td_{p,n}$ – Time of a n^{th} change of stock level of the product p for the Distribution Centre dc.
- $\tau_{p,n}$ – Timespan between consecutive changes of stock level of products p

$$\tau_{p,n} = td_{p,n+1} - td_{p,n}$$

- H_{dc} – Total holding cost of keeping inventory by Distribution Centre

$$H_{dc} = \sum_{p}\sum_{n} \tau_{p,n} \times hd_p \times id_{p,td_{pn}}$$

- $im_{e,t}$ – Inventory level of element e in the Manufacturer inventory at the time t.
- $tm_{e,n}$ – Time of a n^{th} change of stock level of the element e for the Manufacturer.
- $\tau_{e,n}$ – Timespan between consecutive changes of stock level of elements.

$$\tau_{e,n} = tm_{e,n+1} - tm_{e,n}$$

- H_{mf} – Total holding cost of keeping inventory by Manufacturer.

$$H_{mf} = \sum_{e}\sum_{n} \tau_{e,n} \times hm_e \times im_{e,t}$$

5 SIMULATION FRAMEWORK

5.1 Introduction

Dynamic nature of the considered problem requires a high flexibility to determine decisions regarding inventory control, scheduling and planning of different processes which depend on echelon characteristics. Many general-purpose simulators as Arena, Simul8 or AnyLogic lack the flexibility in specifying control policies and scheduling algorithms, optimisation, and stochastic modelling functionality. Not many available programs allow modelling of fuzzy numbers or logic of controllers. Creation of a new software provides better extensibility and adjustability in considering the identified SC problem.

Moreover, developing a new simulation software delivers additional advantages: 1) a possibility of modelling all relevant outputs and KPIs important for the SC, 2) the proposed software aims to avoid black boxes as some of the existing simulators do. 3) It allows observing all steps in SC processes and proposed heuristics and metaheuristics. The proposed simulator provides a high flexibility, where various SC's structures and functionality can be analysed. They can be further extended to accommodate investigation of additional SCM problems occurring in uncertain environments. In this thesis it was decided to design and implement a custom simulation environment. The decision was made to give the author the maximum flexibility when modelling and implementing various decision-making policies and algorithms inventory control and scheduling in SCs. This chapter will cover the design and implementation of a developed simulator. When designing the simulation, the following objectives are considered:

- ability to implement a discrete-event simulation,
- ability to simulate SCs with arbitrary structures,
- enabling an easy development and implementation of decision-making,
- ability to produce various visual and data outputs/KPIs/other metrics,
- support of non-interactive invocation which does not require graphic interaction with the user, enabling usage in optimisation routines and automation.

The Python programming language was chosen for implementation of the simulator. This decision was made due to Python's ease of use and a wealth of available packages and modules that can be used.

Python also offers *Symmetric multiprocessing* (SMP) capabilities, which enable multiple concurrent simulations on multi-core processors. This is important for speeding up optimisation of various experiments carried out in this research. The simulation environment was based on *SimPy*, which is a process-based discrete event simulation (DES) framework developed using standard Python. It offers various *primitives* such as events, processes, shared resources etc. Therefore, it was a good starting point for developing a DES in Python.

5.2 Simulation process description

Each simulation begins with initialisation of the environment, including *SimPy's* environment. Each echelon includes a decision-making component. It is created based on the echelon type and decision-making parameters specified for it in the *scenario* (Subchapter 5.3). At this stage, the simulation enters the simulation loop. Each time a single event is processed. An event examples are incoming order, finish of the production, delivery of supply to the inventory etc. The loop is terminated when one of the following occurs:

- There are no more events to be processed.
- All orders issued by the customer have been delivered.
- A specified maximum simulation time elapsed.

The last condition is used only when simulation is conducted as a part of optimisation, to cap the simulation time. After the simulation loop is complete, the simulation output such as reports, and objective values are generated.

One of the design goals of the simulation environment was separation of *decision-making* from the *echelon's processes*. The echelon process includes a functionality of the echelon beyond decision making. During the simulation, implemented *decision maker* is informed by the echelon about new orders, inventory level changes, delays etc. In turn the *decision maker* will affect its echelon by issuing orders to replenish inventory and/or schedule echelon 's resources. Such separation of concerns simplifies echelon modelling and makes it easy to change and develop decision making strategies independently from the simulation process.

The decision maker specified for each echelon passes decisions to the echelon using *machine schedule* and *delivery schedule* entities. Schedule is an entity that can capture the assignment of arbitrary tasks with given start date and duration to a set of resources. The *machine schedule* determines when each machine will start manufacturing a given product required for fulfilling a given order. The *delivery schedule* determines when any given order can be sent. The *decision maker* also makes orders from the Supplier to replenish the Manufacturer inventory. The Distribution Centre *decision maker* can make orders to replenish the inventory and plans the delivery by deciding the *delivery schedule*. Contrary to other echelons, the Decision Maker's delivery schedule does not contain only start of the task. Rather it considers the time needed to load a lorry, reach the customer, unload the payload and return to the Distribution Centre.

5.3 Simulation Scenario

To perform a simulation, the information about all echelons, their connectivity and the selected decision-making strategies were needed. The collection of these information is called a *Scenario*. The *Scenario* holds the following information:

- Scenario name.
- Start date, which is an absolute date used as a start point of the simulation which in simulator is introduced as a relative date, measured as the days, hours and minutes since the beginning of the simulation.
- Definitions of all echelon's parameters.
- Decision-making strategies for each echelon.
- Connectivity between echelons, which includes strategy on how produced orders are delivered between supplier and manufacturer or distribution centre and customer.
- List of all products and product-specific parameters.

Data structure implemented in the simulator supports reading and writing to *XML* (*eXtensible Markup Language*) format, making it human-readable. That allows scenarios to be opened and saved in the XML format in any browser and do not require any additional software. All observed outputs of the simulator were modelled with the help of *plotly* visualisation tool supported by Python. It allowed to present various outputs as showed on Figure 5.2, Figure 5.3, Figure 5.4 and Figure 5.5. A simple editor with a graphical user interface was also developed to make it easier to create, change and inspect the scenarios. The main window of the proposed simulator can be seen in Figure 5.1. As it can be seen in Figure 5.1, echelons can be added as presented by the red highlight and connected in various ways in a section highlighted by a green colour. The options used for editing scenario are available through the tabs different for each echelon. It is highlighted in a figure with a blue colour.

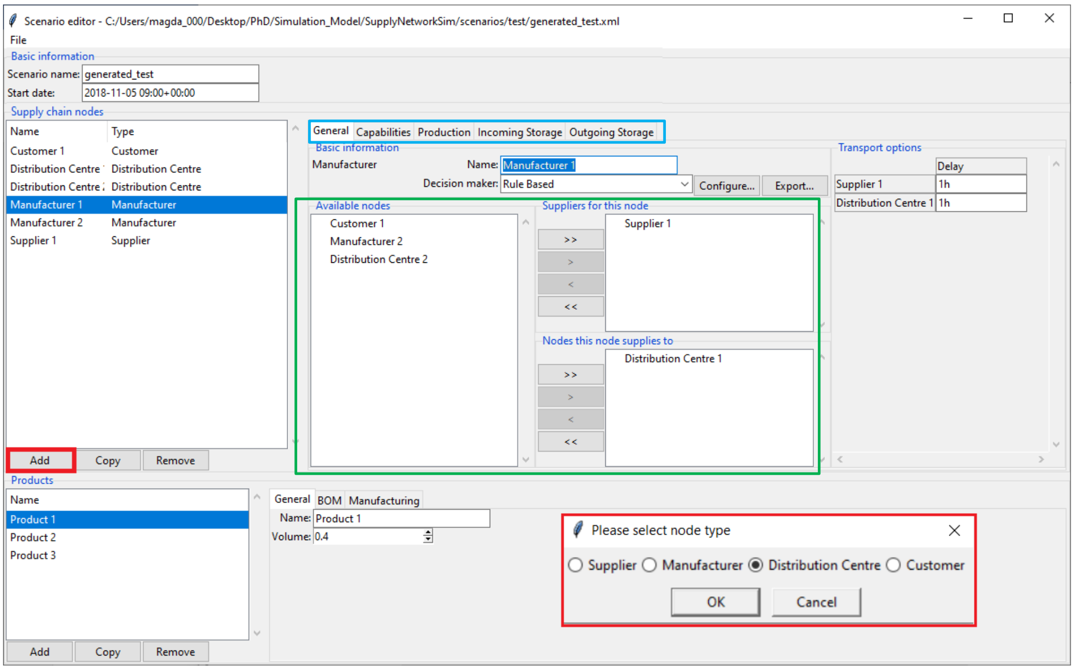

Figure 5.1 User interface of the *Scenario* in the implemented Simulator

Figure 5.2 presents an example of output for the Distribution Centre echelon which schedules deliveries of orders to the Customer.

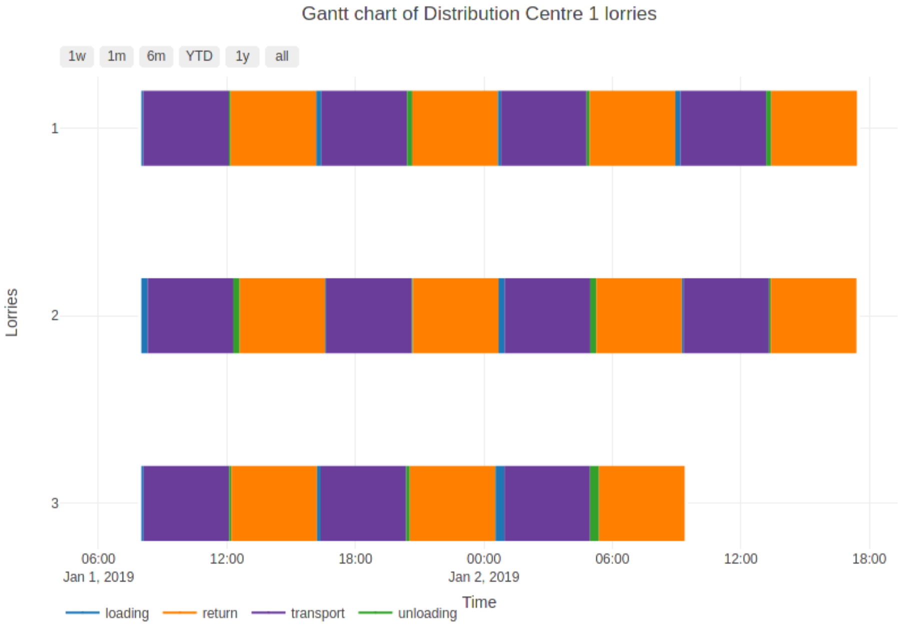

Figure 5.2 Example Gantt Chart of the Distribution Centre lorries

Figure 5.3 introduce an inventory stock of the Manufacturer changing over simulation run time.

Figure 5.3 Example inventory levels for the Manufacturer echelon

Figure 5.4 presents a Gantt Chart generated for the Manufacturer echelon. As can be seen, an actual production may differ from the planned schedule as inventory necessary for the production may not be available, causing the delay.

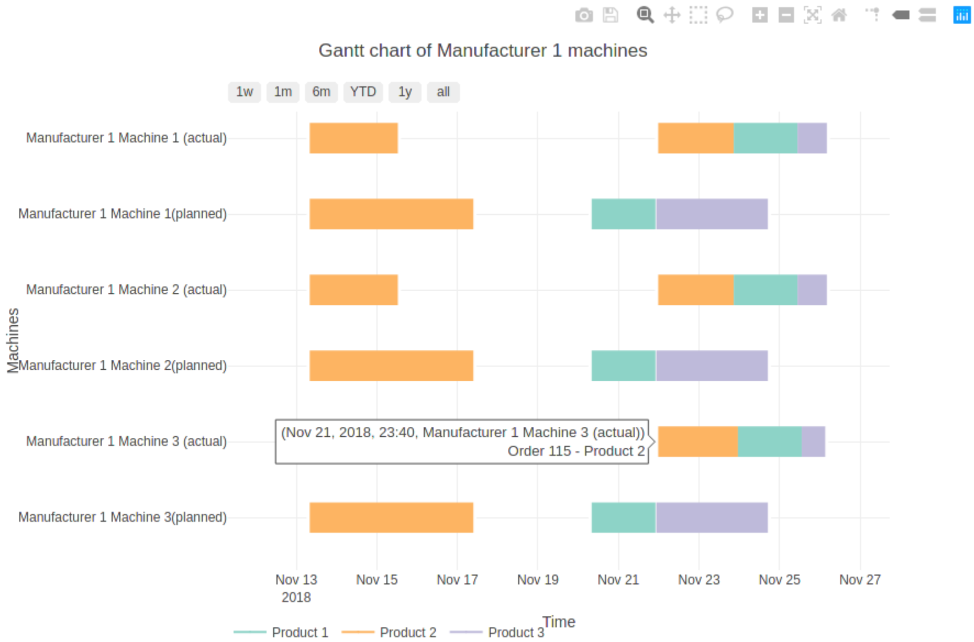

Figure 5.4 Example Gantt Chart of the Manufacturer echelon

Figure 5.5 presents an example Gantt Chart for the Supplier echelon. As can be seen it includes set up time of machines.

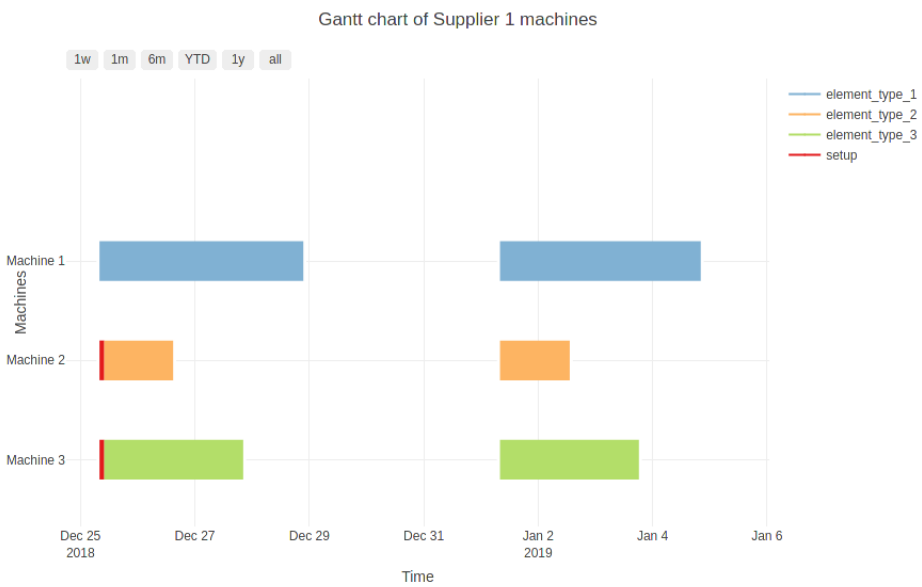

Figure 5.5 Example Gantt Chart of the Supplier echelon

5.4 Benchmark scenario

Planning period 16 weeks is considered. Orders are collected in 1-week intervals. Each echelon has different characteristics as described in Chapter 4 and consequently it is simulated in a different way. Below, a benchmark scenario is defined.

5.4.1 Input to the simulation

Tables presented below contain input parameters values. Table 5.1 contains basic information about number of echelons in the considered SC.

Table 5.1 Input of the simulation supply chain (number of customers, manufacturers, distribution centres and suppliers)

Parameter	$\boldsymbol{cu}$	$\boldsymbol{mf}$	$\boldsymbol{dc}$	$\boldsymbol{su}$
Input value	1	1	1	2

Table 5.2, Table 5.3 and Table 5.4 introduce values of input parameters for the Supplier, the Manufacturer and the Distribution Centre echelons, respectively.

Table 5.2 Input for Suppliers echelons

Parameter	***Input value***	
β_{su}	$\beta_1 = [1_1, 1_2, 1_3, 0_4, 0_5]$,	$\beta_2 = [0_1, 0_2, 0_3, 1_4, 1_5]$
$b_{e,su}$	$b_{e,1} = [10_{1,1}, 10_{2,1}, 10_{3,1}]$	$b_{e,2} = [10_{4,2}, 10_{5,2}]$
$et_{e,su}$	$et_{e,1} = [3_{1,1}, 1_{2,1}, 2_{3,1}]$	$et_{e,2} = [1.5_{4,2}, 1.5_{4,2}]$

$sut_{ms,su}$	$sut_{ms,su_1} = \begin{cases} 10\ minutes\ (element\ 1\ \rightarrow element\ 2) \\ 60\ minutes\ (element\ 1\ \rightarrow element\ 3) \\ 15\ minutes\ (element\ 2\ \rightarrow element\ 1) \\ 20\ minutes\ (element\ 2\ \rightarrow element\ 3) \\ 50\ minutes\ (element\ 3\ \rightarrow element\ 1) \\ 30\ minutes\ (element\ 3\ \rightarrow element\ 2) \end{cases}$ and $sut_{ms,su_2} = \begin{cases} 20\ minutes\ (element\ 4\ \rightarrow element\ 5) \\ 30\ minutes\ (element\ 5\ \rightarrow element\ 4) \end{cases}$

Table 5.3 Input for the Manufacturer echelon

Parameter	***Input value***
$\bar{B}_{e,p}$	$\bar{B}_{e\times p} = \begin{bmatrix} 10 & 15 & 0 & 0 & 16 \\ 8 & 24 & 16 & 12 & 0 \\ 10 & 0 & 0 & 10 & 15 \end{bmatrix}$
$\overline{ptm}_p$	$\overline{ptm}_p = [1_1, 2.5_2, 1.75_3]$
iim_e	$iim_e = [500_1, 500_2, 500_3, 500_4, 500_5]$
im'_e	$im'_e = [10000_1, 10000_2, 10000_3, 10000_4, 10000_5]$
hm_e	$hm_e = [2_1, 0.5_2, 1_3, 1_4, 6_5]$

Table 5.4 Input for the Distribution Centres echelons

Parameter	***Input value***
v_p	$v_p = [0.2_1, 0.5_2, 0.2_3]$
st_{lor}	$st_{lor} = [100_1, 100_2, 100_3, 100_4, 100_5]$
iid_p	$iid_p = [200_1, 200_2, 200_3]$
hd_p	$hd_p = [7_1, 10_2, 7_3]$
lor	$lor = 3$
st_{lor}	$st_{lor} = [100_1, 100_2, 100_3]$

Pre-set replenishment levels for the Manufacturer and the Distribution Centre can be found in Table 5.5 and Table 5.6 respectively. The levels of replenishment were determined empirically for the benchmark scenario. Orders placed by the Customer to the Distribution Centre are presented in Table 5.7.

Table 5.5 Crisp CRP inventory levels for the Manufacturer echelon

Elements	Reorder Point	Order size
Element 1	1300	400
Element 2	1000	300
Element 3	2000	300
Element 4	1000	400
Element 5	1000	400

Table 5.6 Crisp CRP inventory levels for the Distribution Centre echelon

Products	Reorder Point	Order size
Product 1	100	100
Product 2	100	100
Product 3	100	100

Table 5.7 Benchmark scenario orders

Benchmark scenario											
Order id	*Product 1*	*Product 2*	*Product 3*	*Order date*	*Due date*	*Order id*	*Product 1*	*Product 2*	*Product 3*	*Order date*	*Due date*
1	4	25	15	06/11/18 10:00	11/11/18 01:00	**57**	11	1	11	22/12/18 01:00	23/12/18 19:00
2	16	25	12	06/11/18 21:00	11/11/18 20:00	**58**	10	14	19	22/12/18 02:00	26/12/18 00:00
3	23	4	23	09/11/18 12:00	13/11/18 05:00	**59**	1	10	17	22/12/18 05:00	25/12/18 02:00
4	19	4	9	10/11/18 03:00	12/11/18 11:00	**60**	22	19	4	22/12/18 21:00	26/12/18 17:00
5	11	25	10	10/11/18 19:00	15/11/18 08:00	**61**	1	6	25	24/12/18 06:00	27/12/18 07:00
6	5	14	16	11/11/18 23:00	15/11/18 10:00	**62**	16	0	1	25/12/18 19:00	26/12/18 20:00
7	18	10	22	13/11/18 02:00	17/11/18 04:00	**63**	7	14	23	26/12/18 23:00	31/12/18 02:00
8	19	4	9	17/11/18 05:00	19/11/18 13:00	**64**	1	2	14	28/12/18 04:00	29/12/18 20:00
9	16	9	8	17/11/18 18:00	20/11/18 11:00	**65**	25	25	1	28/12/18 05:00	01/01/19 16:00

10	25	18	6	18/11/18 00:00	22/11/18 01:00	**66**	4	17	5	28/12/18 12:00	31/12/18 08:00
11	0	15	23	18/11/18 15:00	22/11/18 13:00	**67**	19	6	3	29/12/18 05:00	31/12/18 07:00
12	24	0	24	21/11/18 03:00	24/11/18 11:00	**68**	14	17	4	29/12/18 20:00	02/01/19 02:00
13	25	25	12	21/11/18 04:00	26/11/18 13:00	**69**	1	0	8	29/12/18 22:00	30/12/18 20:00
14	0	2	16	22/11/18 19:00	24/11/18 13:00	**70**	4	10	20	30/12/18 01:00	02/01/19 07:00
15	14	17	8	23/11/18 00:00	26/11/18 14:00	**71**	12	2	10	30/12/18 23:00	01/01/19 19:00
16	24	6	13	23/11/18 09:00	26/11/18 13:00	**72**	0	1	12	31/12/18 23:00	02/01/19 07:00
17	6	0	3	24/11/18 16:00	25/11/18 09:00	**73**	18	21	10	01/01/19 09:00	05/01/19 19:00
18	13	23	16	24/11/18 19:00	29/11/18 17:00	**74**	1	20	9	01/01/19 18:00	05/01/19 03:00
19	24	4	17	25/11/18 03:00	28/11/18 09:00	**75**	17	0	17	02/01/19 07:00	04/01/19 17:00
20	18	1	20	26/11/18 12:00	29/11/18 08:00	**76**	0	25	13	02/01/19 11:00	06/01/19 18:00
21	4	6	22	29/11/18 05:00	02/12/18 04:00	**77**	0	25	3	04/01/19 00:00	07/01/19 10:00
22	24	21	6	29/11/18 16:00	04/12/18 01:00	**78**	8	11	25	04/01/19 09:00	08/01/19 09:00
23	19	21	15	30/11/18 10:00	05/12/18 07:00	**79**	25	8	9	05/01/19 02:00	08/01/19 04:00
24	22	1	24	30/11/18 17:00	04/12/18 02:00	**80**	1	19	16	06/01/19 04:00	10/01/19 00:00
25	7	16	2	01/12/18 02:00	03/12/18 17:00	**81**	16	6	3	06/01/19 09:00	08/01/19 07:00
26	9	0	15	01/12/18 04:00	03/12/18 01:00	**82**	8	20	9	06/01/19 13:00	10/01/19 06:00
27	25	16	17	02/12/18 06:00	06/12/18 23:00	**83**	15	20	14	06/01/19 15:00	11/01/19 02:00
28	14	19	8	02/12/18 11:00	06/12/18 06:00	**84**	25	21	5	08/01/19 15:00	12/01/19 23:00
29	10	10	12	02/12/18 17:00	05/12/18 14:00	**85**	14	18	18	08/01/19 22:00	13/01/19 11:00
30	23	0	6	03/12/18 01:00	04/12/18 20:00	**86**	0	15	8	09/01/19 15:00	12/01/19 07:00
31	1	5	5	03/12/18 06:00	04/12/18 12:00	**87**	24	24	21	10/01/19 20:00	16/01/19 19:00
32	13	14	6	05/12/18 01:00	08/12/18 01:00	**88**	1	14	14	12/01/19 06:00	15/01/19 08:00
33	19	9	6	06/12/18 18:00	09/12/18 10:00	**89**	10	21	1	12/01/19 07:00	15/01/19 13:00
34	5	8	0	07/12/18 01:00	08/12/18 10:00	**90**	13	18	10	12/01/19 18:00	16/01/19 13:00
35	19	22	15	07/12/18 13:00	12/12/18 13:00	**91**	1	3	6	13/01/19 13:00	14/01/19 15:00
36	25	12	4	07/12/18 22:00	11/12/18 02:00	**92**	5	9	20	14/01/19 02:00	17/01/19 06:00
37	8	15	22	08/12/18 08:00	12/12/18 13:00	**93**	8	4	22	14/01/19 06:00	17/01/19 03:00
38	10	19	2	08/12/18 20:00	11/12/18 23:00	**94**	19	7	14	14/01/19 22:00	18/01/19 01:00
39	2	24	3	09/12/18 06:00	12/12/18 16:00	**95**	6	7	25	16/01/19 18:00	20/01/19 04:00
40	12	24	11	09/12/18 18:00	14/12/18 07:00	**96**	14	23	20	16/01/19 23:00	22/01/19 06:00
41	13	13	9	11/12/18 08:00	14/12/18 11:00	**97**	12	10	12	17/01/19 02:00	20/01/19 01:00
42	7	20	11	11/12/18 17:00	15/12/18 13:00	**98**	5	4	23	18/01/19 06:00	21/01/19 02:00
43	25	1	17	11/12/18 20:00	14/12/18 18:00	**99**	20	10	6	19/01/19 03:00	21/01/19 23:00
44	20	3	17	11/12/18 23:00	14/12/18 21:00	**100**	21	6	6	20/01/19 23:00	23/01/19 09:00
45	21	24	20	12/12/18 00:00	17/12/18 18:00	**101**	1	10	15	22/01/19 02:00	24/01/19 19:00
46	8	8	12	13/12/18 02:00	15/12/18 15:00	**102**	14	11	20	22/01/19 08:00	26/01/19 04:00
47	19	8	15	13/12/18 09:00	16/12/18 17:00	**103**	8	7	4	22/01/19 11:00	24/01/19 05:00
48	14	21	14	13/12/18 16:00	18/12/18 05:00	**104**	4	17	14	22/01/19 14:00	26/01/19 04:00
49	9	13	15	14/12/18 05:00	17/12/18 15:00	**105**	11	16	20	24/01/19 04:00	28/01/19 11:00
50	15	12	10	15/12/18 04:00	18/12/18 08:00	**106**	12	24	22	24/01/19 19:00	30/01/19 07:00
51	14	22	24	16/12/18 14:00	22/12/18 02:00	**107**	0	2	10	25/01/19 14:00	26/01/19 20:00
52	25	14	12	17/12/18 11:00	21/12/18 13:00	**108**	24	19	21	27/01/19 07:00	01/02/19 16:00
53	9	3	4	18/12/18 02:00	19/12/18 10:00	**109**	6	21	14	30/01/19 06:00	03/02/19 10:00
54	17	6	20	18/12/18 10:00	21/12/18 20:00	**110**	7	24	17	31/01/19 03:00	04/02/19 23:00
55	16	19	14	20/12/18 05:00	24/12/18 15:00	**111**	17	22	3	01/02/19 00:00	04/02/19 21:00
56	25	18	25	21/12/18 13:00	27/12/18 04:00	**112**	13	24	5	02/02/19 20:00	06/02/19 23:00

6 Scheduling and Inventory Control Model

6.1 Introduction

A simulation model for dynamic multi-stage SC scheduling and inventory control is introduced in this chapter. The goal is to observe how decisions regarding inventory stock and order prioritisation for scheduling affect defined KPIs. A control-scheme for inventory of elements and products and scheduling of orders is implemented, and effects of implemented decisions on SC performance are observed. Real world SC faces demand uncertainty which influences processes carried out in all echelons. All production and distribution processes depend on customer demand and order deadline. Poor scheduling and inventory control decisions can cause a long delay, shortages of a raw material required for production and too high holding costs. Therefore, using a CRP maintaining a preferred inventory policy and DRs assigning priority of orders for all echelons is analysed.

A simulation framework described in the previous chapter allows selection and configuration of the procedure which is used for scheduling of incoming tasks and continuous replenishment of inventory. Inventory control is spread throughout the SC and includes decisions on ordering and stocking of raw material for

Manufacturer and products delivery for the Distribution Centre. Scheduling of production and deliveries takes place at all three echelons including Supplier Manufacturer and Distribution Centre echelons.

Decision making for each echelon is assessed based on KPIs, which depend on all echelon’s decisions. Each echelon has its own structure, parameters, and individual decision-making algorithms. The model of *echelon processes* such as how products and elements are processed, how they are transported and range of system capabilities are separated in model design from the *decision making* inside echelons. The separation of concerns is achieved by building a modular simulation system. While the former was introduced in Problem Statement and Simulation Chapters the latter is covered in this chapter. The decision-making procedure for both; scheduling and inventory control subproblems is explained in pseudo-codes throughout Subchapter 6.2. The effects of shorter and longer due dates of orders, different sizes of orders and different product processing times are observed and analysed in Subchapter 6.3. The proposed general-structure SC is considered with unknown demand. Selected dispatching rules with a fixed CRP are compared for simultaneous scheduling and inventory control of all echelons. One of the main uncertainties for SC is unknown demand with changing number of orders and different due dates. Subchapter 6.4 contain comparison between proposed DRs and conclusions.

6.2 Decision making

6.2.1 Scheduling decision making

Scheduling of an order received by supplier requires a free slot in a schedule. Free slots are defined as time intervals between end of one task and beginning of next scheduled task on machine. Supplier works using a *make-to-order* policy and time of delivery is set into the closest to the ideal send date. Ideal date is the difference between due date of an order and travel time of a lorry delivering order to the Manufacturer. Ideal send date is calculated for each incoming order such that:

$$ideal\ send\ date_{os} = dos_{su,os} - tt_{mf}$$

Ideal send date includes transportation time of orders calculated by each echelon individually. Figure 6.1 presents Gantt Chart as used in the Simulation framework in which preferable schedule window for incoming orders is localised between the time defined as *current time* and *ideal send date.* The model design considers dynamic nature of scheduling processes and prevent the scheduling of any tasks before the current time. Proposed scheduling algorithm following constraints, do not split orders and aim to send an order as close to ideal date as possible.

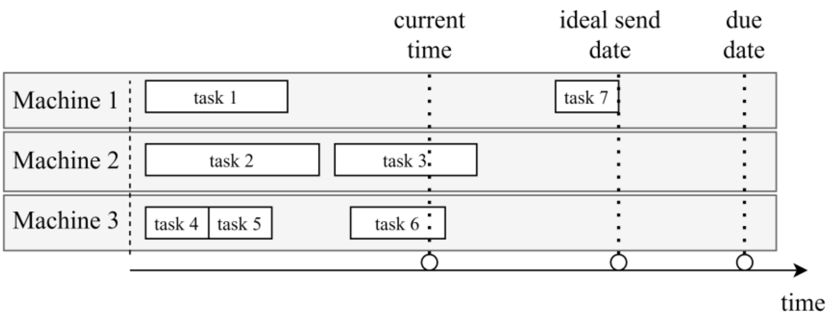

Figure 6.1 Window of schedule between two points in simulation

If that is not possible, algorithm try to find slot which is closest to *current time* which causes production to be finished before ideal send date causing earlier delivery. In a case when there are no available time slots between *current time* and *ideal send date* scheduling window; the delay will occur. Then a delivery of an order is set to the date closest to *ideal send date* but after the preferable scheduling window. Priority in which orders are scheduled is determined by crisp Dispatching rules described in subchapter 6.2.2.

Each echelon in the proposed SC incorporates decision making strategies represented in pseudocodes below. Supplier's algorithms are presented in Figure 6.2 Figure 6.3 and Figure 6.4.

Decision-making procedures for Supplier's scheduling production of orders placed by the Manufacturer is explained in *Algorithm 1*. *Algorithm 1* is used to schedule delivery of the produced order. Delivery time is allocated and procedure of sending an order to the Manufacturer is implemented. This algorithm is used to set a sending date for an order either it was produced on time represented by *ideal sent date* variable or when it is ready to send but delayed represented by *already* variable.

Algorithm 1 Supplier's order scheduling algorithm

```
procedure SCHEDULE(echelon, order)
    ▷ Schedule element production
    ▷ echelon.excesses denotes a vector of excess elements from the last scheduling
    ▷ allready denotes a time when all elements for the order are produced
    allready ← SCHEDULEPRODUCTION(order, echelon.machineschedule, echelon.excesses)
    ▷ Schedule order delivery
    triptime ← TRIPTIME(eschelon, order.customer)
    ▷ Order should be sent so it arrives on time
    ▷ But not before all elements are produced
    sentdate ← max (order.duedate - triptime, allready)
    ADDTASK(echelon.shipmentschedule, sentdate, order)
```

Figure 6.2 Supplier's order scheduling algorithm

Supplier's production scheduling algorithm is presented in Figure 6.3. Schedule of production requires allocation of all tasks to available resources and initialisation of the production. *Algorithm 2* calculate how many elements must be produced for Manufacturer's order. Supplier production scheduling algorithms records work of the supplier machines and processing of orders. It includes calculation of the *excesses*, which represent quantity of stored elements remaining from batches of elements which were produced previously, but not ordered by the Manufacturer. The *excess* of elements is equal to zero at the beginning of the simulation, but any elements remaining after the batch production is completed can be used for next orders.

Algorithm 2 Supplier's production scheduling algorithm

```
function SCHEDULEPRODUCTION(order, schedule, excesses)
    ▷ Initialise time of production completion
    allready ← currenttime
    ▷ For each element in the order
    for element ∈ order do
        quantity ← order.quantity(element)
        ▷ Establish how many elements have to be produced given any
        ▷ excesses
        ▷ quantity denotes total quantity of element in order
        ▷ missing denotes quantity that has to actually be produced
        missing ← max (quantity − excesses(element), 0)
        ▷ Deduct quantity taken from the excesses
        excesses(element) ← excesses(element) − (quantity − missing)
        ▷ Do we need to produce that element?
        if missing > 0 then
            machine ← FIRSTFREEMACHINE(schedule, element)
            ▷ Establish the necessary quantity in multiples of a batch
            realquantity ← ⌈missing / minimumbatch(element)⌉ ∗ minimumbatch(element)
            ▷ Update excesses
            excesses(element) ← excesses(element) + (realquantity − missing)
            ▷ Get the last assignment of the machine
            task ← LASTASSIGNMENT(schedule, machine)
            if task then
                ▷ Record the element previously manufactured on
                ▷ the machine
                prevelement ← task.element
                ▷ New task will start after last assignment
                start ← max (currenttime, task.end)
            else
                ▷ Machine was idle since the start of simulation
                prevelement ← None
                ▷ New task will start now
                start ← currenttime
            ▷ Calculate production time for given quantity of the element
            ▷ (including setup time)
            prodtime ← PRODUCTIONTIME(machine, element, realquantity, prevelement)
            ▷ Assign task to the machine
            ADDTASK(schedule, machine, start, prodtime, element)
            ▷ Keep track of production completion for that order
            allready ← max (allready, start + prodtime)
    ▷ Return time when production should be completed
    return allready
```

Figure 6.3 Supplier's production scheduling algorithm

A Supplier's machine selecting algorithm is presented in Figure 6.4. A free machine is assigned to production of an element. The algorithm considers set-up times of machines and elements excess. Supplier's machines require a set-up time between production of different type of elements. *Algorithm 3* selects machine, which will be used for order production. The machine which does not require the set-up time for production of considered element and it is free is used to prevent additional delays.

Algorithm 3 Supplier's machine selection algorithm

```
function FirstFreeMachine(schedule, element)
    ▷ Initialise
    ▷ firstfree denotes the machine that will be free first (so far)
    ▷ firstfinished denotes the time the machine firstfree will be free
    firstfree ← None
    firstfinished ← −1
    ▷ For each machine
    for machine ∈ schedule.machines do
        ▷ Find machine's last assignment
        task ← LastAssignment(schedule, machine)
        if task then
            ▷ Machine will be free after last assignment
            finished ← task.end
            ▷ Prefer machine that does not require setup
            if task.element ≠ element then
                ▷ Setup required - add setup time
                finished ← finished + setuptime(task.element, element)
        else
            ▷ Machine is idle - it is free now
            finished ← currenttime
        ▷ If it is the first machine
        ▷ or this machine is available earlier than best machine so far
        if firstfree = None ∨ firstfinished > finished then
            ▷ This machine becomes current best
            firstfree ← machine
            firstfinished ← finished
    ▷ Return best machine
    return firstfree
```

Figure 6.4 Supplier's machine selection algorithm

Suppliers and manufacturer have their own lorries which are not scheduled, just sent when the finished order is ready to send.

Manufacturer algorithms are presented in Figure 6.5, Figure 6.6 and Figure 6.7. *Algorithm 4* is used to schedule delivery of orders to the Distribution Centre echelon. *Algorithm 5* is used to schedule production of products on free machines by selecting best available slots. The slot selection procedure is introduced by *Algorithm 6.*

Algorithm 4 requires all products to be produced before sending to the Distribution Centre and constraints are implemented to guarantee the order is delivered in one delivery.

Algorithm 4 Manufacturer's order scheduling algorithm

```
procedure SCHEDULE(echelon, order)
    ▷ Schedule product manufacturing
    triptime ← TRIPTIME(eschelon, order.customer)
    allready ← SCHEDULEPRODUCTION(order, echelon.machineschedule, triptime)
    ▷ Schedule order delivery
    ▷ Order should be send so it arrives on time
    ▷ But not before all elements are produced
    sentdate ← max (order.duedate - triptime, allready)
    ADDTASK(echelon.shipmentschedule, sentdate, order)
```

Figure 6.5 Manufacturer's order scheduling algorithm

Decisions can be made for all future orders, but they cannot be changed for orders which are currently produced i.e., task 3 and task 6 on Figure 1.1 has to be finished before machine 2 and 3 can be considered idle. *Algorithm 5* which is a main algorithm for the Manufacturer scheduling which favour the slots for incoming orders as close as possible to the ideal send date.

Algorithm 5 Manufacturer's production scheduling algorithm

```
function SchedulePRODUCTION(order, schedule, triptime)
    if order.duedate ≠ None then
        deadline ← order.duedate
    else
        deadline ← currenttime
    allready ← −1
    idealsd ← max (deadline − triptime, currenttime)
    for product ∈ order do
        remaining ← order.quantity(product)
        while remaining > 0 do
            remaining ← remaining − 1
            bestmachine ← None
            beststart ← None
            bestend ← None
            bestdelay ← ∞
            for machine ∈ schedule.machines do
                prodtime ← ProductionTime(machine, product)
                (start, end, delay) ← FindSlot(schedule, machine, prodtime, idealsd)
                if IsBetter(delay, bestdelay) then
                    bestmachine ← machine
                    beststart ← start
                    bestend ← end
                    bestdelay ← delay
            ▷ Schedule manufacturing of a single product on the selected machine
            AddTask(schedule, bestmachine, beststart, bestend, product, order)
            allready ← max (allready, bestend)
    return allready
```

Figure 6.6 Manufacturer's production scheduling algorithm

Algorithm 6 localises a free slot in machines and selects the time slot closest to ideal send date which represents date with the lowest possible delay. Possible slots are compared and the one with preferable scheduling window is selected and described by variables *best start*, *best end* and *best delay*.

Algorithm 6 Manufacturer's slot selection algorithm

```
function FindSlot(schedule, machine, prodtime, idealsd)
    ▷ Try to find before send date (but close to it)
    ▷ If that fails try after send date
    ▷ Get all free slots in the schedule from now onward
    slots ← FreeSlots(schedule, machine, currenttime)
    ▷ For each free slot we attempt to place a production task in such
  way that it is finished close to idealsd
    ▷ We also calculate a delay associated with that task and compare it
  with the best found so far
    ▷ The best task so far is described by start (beststart), end (bestend)
  and delay (bestdelay)
    beststart ← None
    bestend ← None
    bestdelay ← None
    for slot ∈ slots do
        slotstart ← max (slot.start, currenttime)
        if slotstart < idealsd then
            idealend ← min (slot.end, idealsd)
            start ← max (idealend − prodtime, slotstart)
            end ← start + prodtime
            if end ≤ slot.end then
                delay ← end − idealsd
                if bestdelay = None∨ IsBetter(delay, bestdelay) then
                    beststart ← start
                    bestend ← end
                    bestdelay ← delay
        else
            start ← slotstart
            end ← start + prodtime
            if end ≤ slot.end then
                delay ← end − idealsd
                if bestdelay = None∨ IsBetter(delay, bestdelay) then
                    beststart ← start
                    bestend ← end
                    bestdelay ← delay
    return (beststart, bestend, bestdelay)
```

Figure 6.7 Manufacturer's slot selection algorithm

Algorithm 7 compares possible production dates by comparison of the available slots and the send date and which is used in *Algorithm 6*.

Algorithm 7 Delay comparison algorithm

```
function IsBetter(newdelay, delay)
    early ← delay < 0
    newearly ← newdelay < 0
    if newdelay = 0 then
        ▷ On time is best, unless we already have one on time
        return delay ≠ 0
    else if early then
        if newearly then
            ▷ Both are early, favour less early
            return newdelay > delay
        else
            ▷ Favour early over delayed
            return False
    else if newearly then
        ▷ New is early but current is not
        ▷ Favour early over delayed
        return delay ≠ 0
    else
        ▷ Both are delayed or current is on time
        ▷ Favour less delay
        return newdelay < delay
```

Figure 6.8 Delay comparison algorithm used by the Manufacturer and the Distribution Centre

Distribution Centre echelon algorithms are presented in Figure 6.9, Figure 6.10, Figure 6.11.

Distribution Centre does not consider production of any element or product. The task of this echelon is to load, deliver and unload orders placed by the Customer. It prioritises orders according to selected method and schedule delivery of order by utilising available lorries.

Algorithm 8 presents a similar strategy as used in scheduling algorithms for Supplier and Manufacturer echelons. This algorithm tries to schedule orders possible close to the ideal send date. One lorry must handle one order but one order my not fit in into one lorry. Processes of loading, delivery time and unloading of order are implemented. Algorithms 9 and 10 are used for lorry selection and packing and unpacking of products ordered by the Customer.

Algorithm 8 Distribution Centre's lorry scheduling algorithm

```
procedure SCHEDULELORRIES(echelon, order)
    schedule ← echelon.lorriesschedule
    ▷ Time a lorry needs to reach the customer
    triptime ← TRIPTIME(eschelon, order.customer)
    ▷ The goods that are still to be assigned a lorry and a time slot
    remaining ← order.quantitites
    deadline ← order.duedate
    while ¬empty(remaining) do
        lorry ← SELECTLORRY(schedule, triptime, remaining, deadline)
        (payload, rest) ← LOADLORRY(lorry, remaining)
        time2deliver ← triptime+ LOADANDUNLOADTIME(lorry, payload)
        totaltime ← time2deliver + triptime
        (prevtask, nexttask) ← NEARESTASSIGNMENTS(schedule, lorry, triptime, remaining, deadline)
        remaining ← rest
        desiredstart ← deadline − time2deliver
        if prevtask ≠ None then
            start ← max (currenttime, prevtask.end)
        else
            start ← currenttime
        if nexttask ≠ None then
            end ← nexttask.start
        else
            end ← ∞
        start ← max (start, desiredstart)
        desiredend ← start + totaltime
        if desiredend > end then
            start ← end − totaltime
            end ← start + totaltime
        else
            end ← desiredend
        ADDTASK(schedule, lorry, start, end, payload)
```

Figure 6.9 Distribution Centre's lorries scheduling algorithm

Algorithm 9 selects lorry which is available for the given delivery and it prioritises the lorries which are idle. The lorry with the lowest delay is preferred.

Algorithm 9 Distribution Centre's lorry selection algorithm

```
function SelectLorry(schedule, triptime, goods, deadline)
    ▷ bestlorry denotes the lorry with lowest delay
    ▷ bestdelay denotes the actual delay associated with bestlorry
    bestlorry ← None
    bestdelay ← ∞
    for lorry ∈ schedule.lorries do
        timefree ← currenttime
        timeend ← ∞
        (prevtask, nexttask) ← NearestAssignments(schedule, lorry, triptime, goods, deadline)
        if prevtask ≠ None then
            timefree ← max (timefree, prevtask.end)
        if nexttask ≠ None then
            timeend ← nexttask.start
        delay ← LorryDelay(triptime, lorry, goods, timefree, timeend, deadline)
        if IsBetter(delay, bestdelay) then
            bestlorry ← lorry
            bestdelay ← delay
    return bestlorry
```

Figure 6.10 Distribution Centre's lorry selection algorithm

Algorithm 10 determines order parts which will fit to the lorry. It uses a greedy algorithm (explained in subchapter 3.2.1) to pack the possible highest number of products with consideration of capacity constraint. Algorithms 11-13 are help in a lorry selection. Algorithm uses its comparison to use the best possible lorry.

Algorithm 10 Distribution Centre's lorry loading algorithm

```
function LoadLorry(lorry, goods)
    remaining ← goods
    payload ← ∅
    sparecapacity ← lorry.capacity
    for product ∈ goods do
        ▷ Check if the whole quantity of product will fit into remaining space of the lorry
        volume ← product.volume ∗ goods.quantity(product)
        if volume ≤ sparecapacity then
            payload(product) ← goods.quantity(product)
            sparecapacity ← sparecapacity − volume
            remaining(product) ← 0
        else
            amount ← ⌊sparecapacity / product.volume⌋
            volume ← product.volume ∗ amount
            payload(product) ← amount
            sparecapacity ← sparecapacity − volume
            remaining(product) ← goods.quantity(product) − amount
    return (payload, remaining)
```

Figure 6.11 Distribution Centre's lorry loading algorithm

6.2.2 Dispatching rules for the scheduling problem

Dispatching rules (DR) are sequencing algorithms used widely in manufacturing for scheduling problems, known for their capabilities for producing good solutions in a real-time. Main disadvantage of DR is that they very often cannot deliver optimal solution and they are not as effective for some KPIs as for others (Holthaus, Rajendan 1997) . Each rule has its advantage over a specific performance measure and based on Kaban (2012), DRs have a significant advantage in facilitating scheduling problems within dynamic context as their low computational complexity allows its use in an online manner.

Scheduling and allocation subproblem seek to determine the schedule of available lorries for the Distribution Centre and machines schedule for Manufacturer and Supplier. Scheduling is performed periodically. Arriving orders are not processed immediately. Instead, the orders received within predetermined time interval are collected, and the scheduling happens only at the beginning of the following time interval, specified by the day and the time of the day. All the collected orders are then scheduled and the cycle repeats. DRs are used to schedule the orders individually. The actual order scheduling is performed by an echelon-specific routine as explained in Problem statement and pseudocodes above. Four selected DRs are presented in Table 6.1 and their description is given below.

Table 6.1 DRs used for scheduling in the considered SC

Crisp Rule		Input parameter	Output parameter
FIFS	First In First Served	Time of arrival	Priority of order
EDD	Earliest Due Date	Due Date	
MTWR	Most Total Work Remaining	Processing Time	
LTWR	Least Total Work Remaining	Processing time	

The first consider DR is FIFS (First In First Served) rule, which prioritises orders based on their arrival time, where the earlier arrived order has the higher priority. FIFS is the most common DR and naturally occurs in manufacturing floor which awaits incoming orders. This rule simply prioritises customer orders by the date of its arrival and this priority depend only on this parameter.

The second selected DR is an EDD (Earliest Due Date) rule, which prioritise orders based on their due date. This rule is proven to perform well for the delay performance indicator as orders which has the shortest due date are scheduled first. Due date is also one of the most important order parameters related to uncertainty of demand as uncertain demand can be reflected by unknown due dates and unknown number of incoming orders.

The next two rules are related to the processing time parameter which is linked to SC production uncertainty such as breaking of machine, different level of skills of operators which in real world can either prolong or reduce processing time etc. MWTR (Most Working Time Remaining) rule prioritise orders based on how much time will be used for their production and the rule prioritises orders with the longest time of production. It requires calculation of a *workingtime* which is a production time required to produce an entire order. It is calculated separately for orders consisting of elements, wt_{os} and orders consisting of products, wt_{om} for the manufacturer and wt_{od} for the distribution centre and it is calculated as follows:

$$wt_{os} = \sum_{e} et_{e,su} \times \overline{os}_e$$

$$wt_{om} = \sum_{p} \overline{ptm}_{\mathrm{p}} \times \overline{om}_{\mathrm{p}}$$

$$wt_{od} = \sum_{p} \overline{ptm}_{\mathrm{p}} \times \overline{od}_{cu,p}$$

LWTR (Least Working Time Remaining) DR prioritises orders on the very similar basis as MTWR rule. The production time required for the whole order is used to determine priority and orders with the shortest working remaining are prioritised.

6.2.3 Inventory Control decision making

The second problem is an inventory control problem, where two decisions are made, namely (1) to determine the replenishment product inventory level (2) how much products should be ordered if stock will drop below this pre-set replenishment level. For each echelon and product in the echelon's inventory a CRP is proposed as can be seen in Figure 6.12.

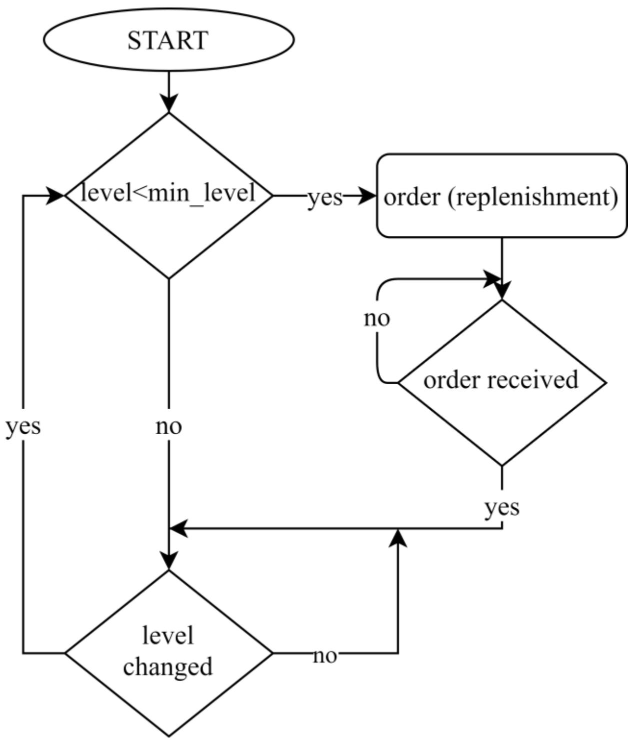

Figure 6.12 CRP for each element or product in the echelon's inventory

6.3 Results

The benchmark scenario presented in subchapter 5.6 is used for following simulation experiments. Two KPIs: total holding cost and delay of orders delivered to customers, are used to evaluate the selected DRs and analyse how different parameters changes KPIs. Additionally, this research focuses on problem when information sharing is kept to a minimum. Customer demand is not shared amongst SC's echelons, there is no cooperation between echelons which translate into uncertain demand in both, uncertain quantity, and due dates of incoming orders.

6.3.1 Different due dates

The values of KPIs for different changes in orders' due date are analysed in this subchapter. In this experiment shorter and longer orders due dates effects are observed. Due dates are shortened by decreasing the due date of all orders by 50% or 25% (e.g. for 50% it changes due date from 2 weeks to 1 week) or changed into longer due dates by extending it by 25% or 50%. Table 6.2 and Figure 6.13 present a holding cost values for changed and benchmark due dates and Table 6.3 and Figure 6.14 present delay values for schedules proposed by different DRs.

Table 6.2 Holding cost of DRs for different due dates

		Holding cost for different due dates (£)				
		shorter by 50%	shorter by 25%	benchmark	longer by 25%	longer by 50%
Dispatching rules	FIFS	150327	150031	152188	153411	156738
	EDD	150151	151142	151480	153366	156841
	MTWR	189002	191439	181415	186206	198934
	LTWR	184785	188629	188990	200519	197363

As can be seen in Figure 6.13 the lowest holding cost is achieved in benchmark scenario while using the EDD followed closely by FIFS. Both DRs which focused on the *processing time* input parameter to prioritise orders i.e., MTWR and LTWR; present inferior results for this KPI. When the due date is shortened by 50% all rules except MTWR result in lower cost in comparison to benchmark due dates. When due dates of orders are increased it gives MTWR the opportunity to produce large orders at the beginning of the time interval resulting in lower delay as there is more time available. Therefore, the delay of smaller orders with lower priority decreases in delays as well. MTWR prioritise larger orders which as can be seen in Table 6.3 also resulted in the worst performance for the delay KPI when due dates were shortened. As seen in this experiment shorter due dates further worsen the delay. When MTWR is applied, as larger orders are allocated less time to be produced and larger orders occupy machines for longer time. That intensify the delays of orders with lower priority and the smaller orders with shorter due dates incur long wait before the production.

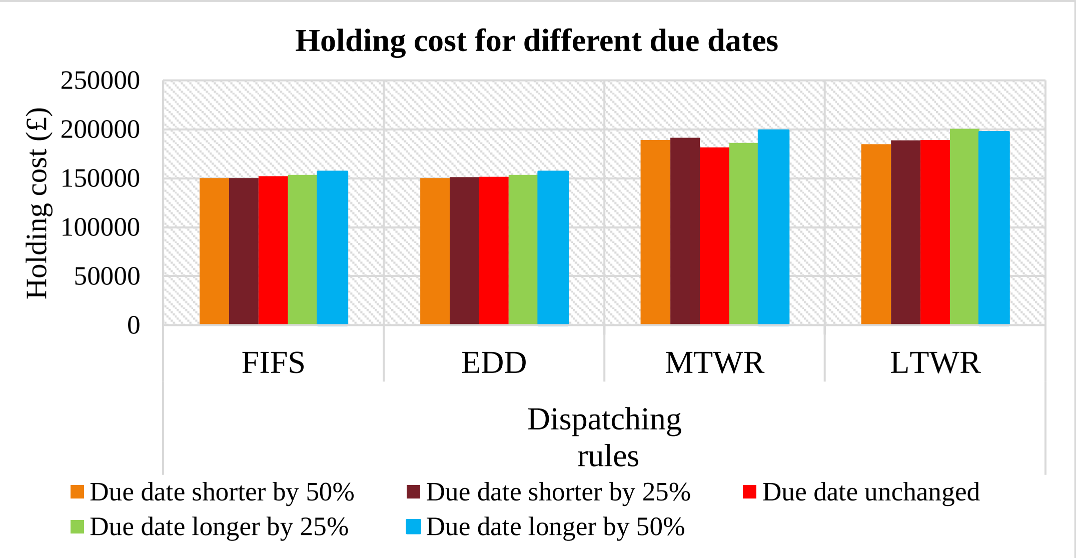

Figure 6.13 Holding cost of DRs for different due dates

LTWR is also the most consistent rule for delay performance indicator as it eliminates delays in smaller orders. The delay decreases for all four rules when the due dates of orders is extended. It behaves similar for FIFS and EDD rule and performs the worst for MTWR DR.

Table 6.3 Delays of DRs with changing due dates

		Delays for different due dates (hours)				
		shorter by 50%	shorter by 25%	benchmark	longer by 25%	longer by 50%
Dispatching rules	FIFS	7798	7645	7418	7044	6790
	EDD	7707	7609	7473	7338	7185
	MTWR	9259	9178	8509	8141	7947
	LTWR	7070	7039	7042	7177	7040

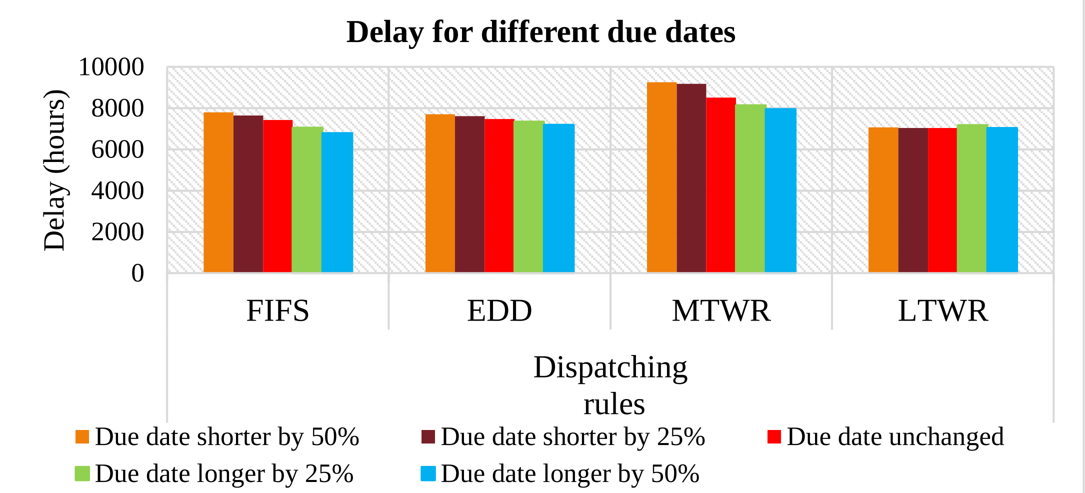

Figure 6.14 Delays of DRs with changing due dates

6.3.2 Different order size

Experiments carried out to examine the impact of a different order sizes are examined in the following subchapter. The orders' processing times for both products and elements are decreased or increased by changing the number of elements and products in an order. Decreasing number of products by half does also cut a production time by 50%. Table 6.4 and Figure 6.15 present holding cost KPI for four compared DRs.

In the case of a lower workload, the value of holding cost decreases for all DRs. In the case when order size is decreased by 50%, the processing time on manufacturer and supplier floor decreases, MTWR rule attain the lowest holding cost followed by LTWR rule. When the processing time is decreased into smaller orders, the delay is the lowest for all DRs as presented in Table 6.5. In this case LTWR and MTWR rules perform best for holding cost KPI when required order size is substantially decreased. The situation changes substantially when the required order size is increased.

Table 6.4 Holding cost for DRs s with different order size

		Holding cost for different order size (£)				
		shorter by 50%	shorter by 25%	benchmark	longer by 25%	longer by 50%
Dispatching rules	FIFS	125685	124005	152188	191390	238688
	EDD	125760	124051	151480	197486	231858
	MTWR	122954	139338	181415	225129	297334
	LTWR	124751	139172	188990	268219	340376

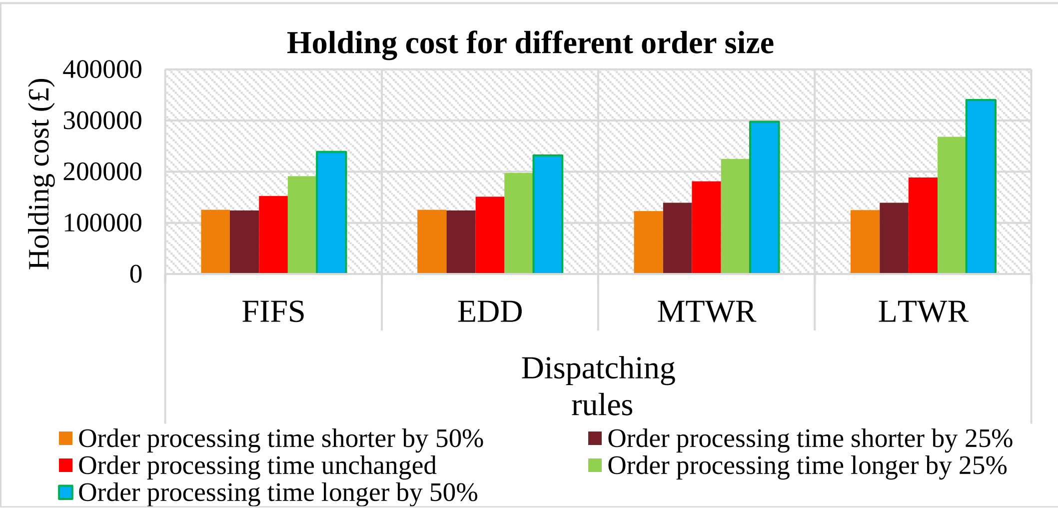

Figure 6.15 Holding cost for DRs with different order size

Table 6.5 and Figure 6.16 presents values for the delay KPI. In a case of the longest processing time a strong trade-off is observed for LTWR rule. In a case of the smallest order when less bottlenecks can be found in a SC all rules return similar values of both KPIs. It shows that if there is enough stock of raw materials and production times of orders are low each DR perform equally good.

When echelon of SC is overloaded with orders and the orders, which are unknown at the beginning of the planning period are taking longer time to be manufactured, the delay KPI is similar for all the rules. The shortest delay is recorder using LTWR rule but in a trade-off with holding cost KPI which is considerably higher than for all other rules. It suggests that an arrival time and due date focused DR provide better solutions when both KPIs are considered.

Table 6.5 Delay for DRs with different order size

		Delays for different order size (hours)				
		shorter by 50%	shorter by 25%	benchmark	longer by 25%	longer by 50%
Dispatching rules	FIFS	129	2303	7418	13288	19564
	EDD	129	2255	7473	13443	19863
	MTWR	130	3128	8509	14255	21479
	LTWR	130	2333	7042	12328	18880

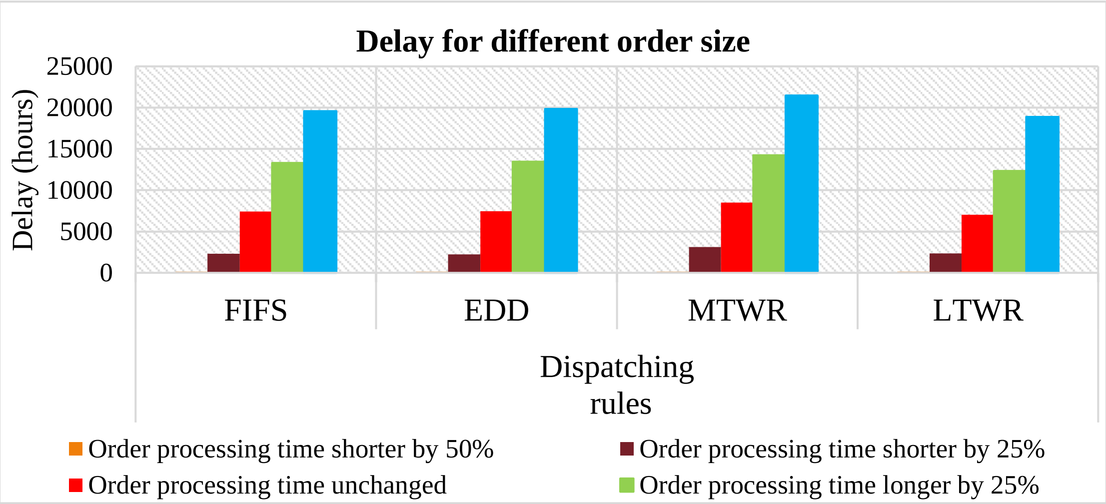

Figure 6.16 Delay for DRs with different order size

6.3.3 Different product processing time

In the previous experiment different order processing time were examined and number of parts and elements in orders were either increased or decreased to change the processing time parameter. The following experiment keeps the number of elements and products unchanged, but the processing time required for production of element or product or time required for packing and delivering orders are changed in a similar way as in previous experiments.

When the processing time of product or element is increased, no more raw material is required by an echelon, and a schedule for the same quantity of products and elements is required. Inventory levels are dropping slower when the production time is longer. LTWR DR returns the worst values of holding cost. It is interesting to notice that shorter processing time leads to the highest holding cost for LTWR rule. It can be explained by the fact, that smaller orders are being processed first and unused inventory which awaits bigger orders with higher priority generates a high cost. EDD performs the best for all cases followed closely by FIFS rule.

Table 6.6 Holding cost for DRs with different product processing time

		Holding cost for different product processing time (£)				
		shorter by 50%	shorter by 25%	benchmark	longer by 25%	longer by 50%
Dispatching rules	FIFS	145697	148926	152188	154853	158511
	EDD	145646	148369	151480	154091	156916
	MTWR	171090	175883	181415	180687	184018
	LTWR	214183	217133	188990	194630	199070

MTWR rule behaves in the similar way as FIFS and EDD rules, where shorter production processing time led to lower holding cost.

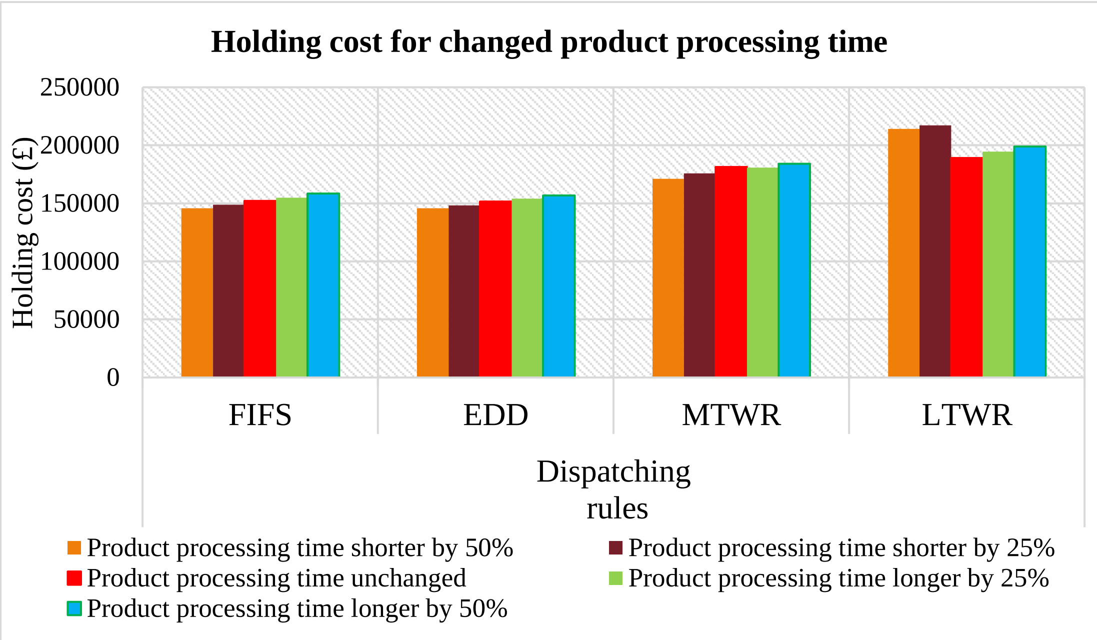

Figure 6.17 Holding cost for DRs with different product processing time

Table 6.7 Delay for DRs with different product processing time

		Delay for different product processing time (hours)				
		shorter by 50%	shorter by 25%	benchmark	longer by 25%	longer by 50%
Dispatching rules	FIFS	7358	7366	7418	7392	7391
	EDD	7453	7449	7473	7473	7421
	MTWR	8362	8467	8509	8441	8427
	LTWR	7135	7145	7042	7003	6995

Delay KPIs does not present drastic changes in values after product processing time is changed. MTWR rule, which returned reasonably good schedule in respect of holding cost KPI, presents the longest delay for orders delivered to the customers while LTWR which returned the highest holding cost presents the lowest delay. The clear trade-off between these KPIs can be observed for these rules.

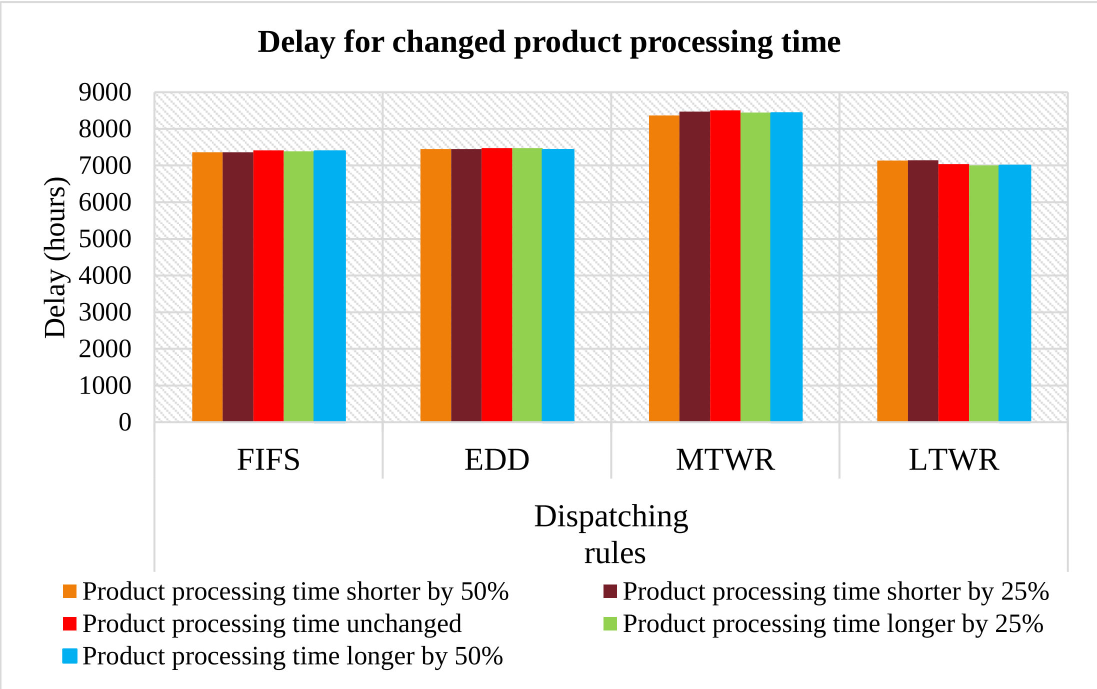

Figure 6.18 Delay for DRs with different product processing time

6.4 Conclusions

This chapter consists of details of decision making implemented in this SC followed by experiments section which aim is to compare how different DRs and changes of uncertain parameters affect defined KPIs.

Unknown demand increases complexity of this problem as it cannot be planned at the beginning of the planning period while incoming orders need a schedule. Uncertainty in demand can cause bottlenecks in any echelon. The crisp DR presented in this chapter do not consider uncertain value of parameters and each of them is focused only on one parameter. It can be beneficial to base a schedule on more input data. Introduction of control which can also accommodate uncertainty could improve KPIs of SCs. A Fuzzy Dispatching Rules are proposed in the next chapter to advance decision-making processes.

Using CRP in the current form can favour some of the echelons facing flow of raw material on a pre-set replenishment level but it requires to be manually selected for different scenarios. One of the possible solutions to improve inventory control could be resignation from the standard CRP in favour of fuzzy inventory control. Instead of ordering only when element or product stock drops below certain crisp number, several other factors could be taken into consideration before replenishing inventory.

Although by selecting a DR one can observe an increase and decrease in both cost and delay, their purpose is not optimisation of the observed parameters. Crisp DRs are very quick but present a lack of the adaptability to the changing and random demand. Use of the simulation tool allowed analysis and observation of Supply Chain behaviours which led to better understanding of a network and comparison of DRs commonly used in practice. Above experiments led to insights for creating a new fuzzy rule base which will be described in the following Chapter.

7 DEVELOPMENT OF FUZZY DISPATCHING RULES

7.1 Introduction

Uncertainties have a significant impact on behaviour and decision-making in the SC. The simulation model proposed in the previous chapter allowed a comparison between crisp DRs, observation of SC behavior and better understaing of a problem domain. Decisions as determining priority of orders, reorder points and order quantities have a significant impact on both observed KPIs; holding cost and delay, which are crucial parameters to be considered by the SC.

One of the main disadvantages of selected DRs are their long delays and lack of adaptivity to the changing demand. DRs consider only one input parameter which also must be crisp, which additionally prevent the SC from reacting to the uncertain demand. The decisions made by one echelon depend on one or more independent parameters which affect inputs to other echelons which causes parameters to be uncertain. Multitude of uncertainties in SC leads to non-linear type of relationship between echelons which increase complexity of the model. Crisp DRs are used to rank priorities of orders, which as a control-scheme includes advantages like speed, straightforward logic, possible optimal solution in some cases and easiness of implementation on real production floor. Although finding a solution which minimise holding cost and delay might be addressed by using many optimisation techniques, many of them are not suitable to consider uncertainty of parameters.

It is a challenging problem to find an optimal reorder point and order quantity for multiple echelons under uncertain demand. Fuzzy interference system (FIS) including Fuzzy Dispatching Rules (FDR) are developed to create a control-scheme considering changing demand. Use of fuzzy logic for representing uncertain parameters can have a positive impact on creating schedule and inventory policy and could preserve advantages of crisp DRs by delivering solutions in reasonable time. Linguistic values allow representation of expert knowledge of uncertain parameters in a form of a fuzzy sets. To create rule bases for both considered subproblems several inputs are taken into consideration. For the inventory subproblem three inputs including unit holding cost of element or product, order processing time which depends on the order size and number of incoming orders which represents an echelon workload are considered. The second and third parameter are linked to uncertain demand. For the scheduling subproblem slack and due dates are considered. FDRs proposed by FIS include the same output parameters as used in the crisp DRs presented in the previous Chapter, which are reorder point and order quantity for inventory control and priority of orders for the scheduling problem. The following Subchapter 7.2 introduces the development of FDRs for both subproblems. Subchapter 7.3 presents analysis of results for changes in demand for the rule base for scheduling problem and two rule bases for inventory subproblem, which consists of holding cost-focused and delay-focused FISs. Improvement of KPI is observed after the fuzzy control-scheme is used, which is discussed in Subchapter 7.4.

7.2 Development of FDRs

An iterative approach is used to develop and improve the FDRs, which is reached by analysis of initial solutions and use of fuzzy logic for representing uncertain values and reviewing new solutions. The FDRs development proposed for the considered problems was conducted in two phases which are presented in Figure 7.1. Phase 1 included implementation of fixed CRP for inventory control subproblem and crisp DRs with supporting algorithms for scheduling manufacturing and distribution echelons as described in Chapter 6.

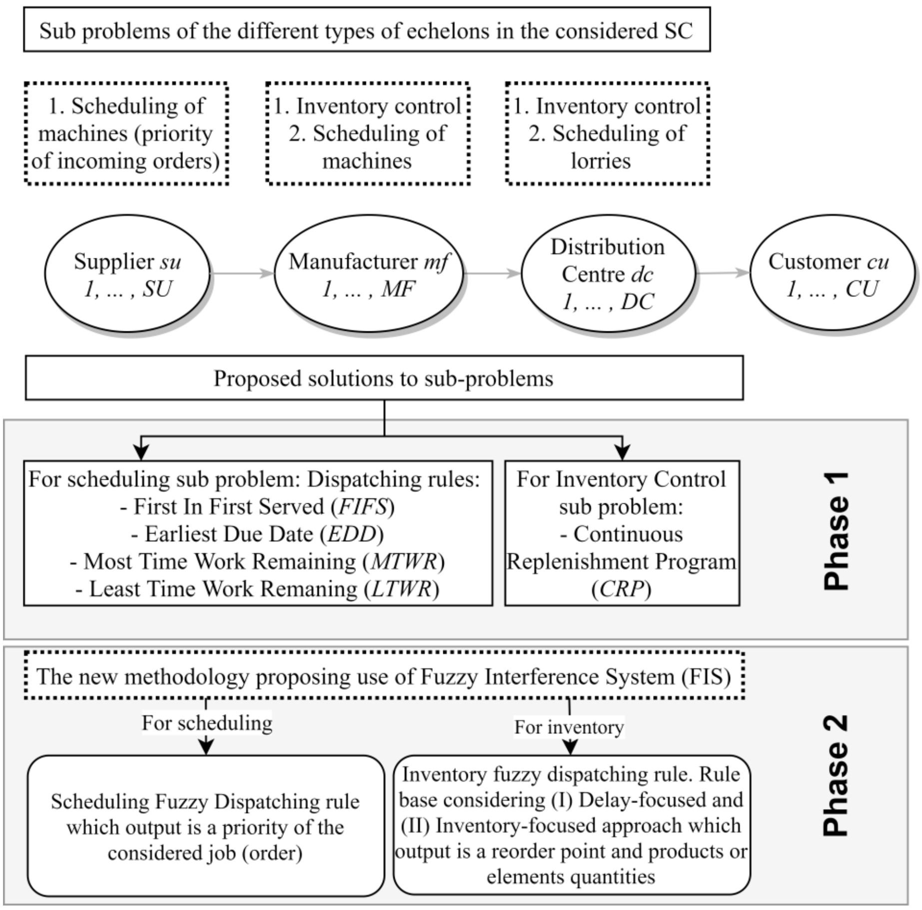

Figure 7.1 Methodology proposed for the developed control-scheme

Phase 2 includes use of Mamdani-style FIS to incorporate FDRs for the proposed control-scheme. Fuzzy sets are used to describe uncertain input values which are defined separately for the scheduling subproblem in Subchapter 7.2.1and inventory control subproblem in Subchapter 7.2.2. Benefits of fuzzy logic, explanation of fuzzy reasoning and fuzzy representation of uncertain parameters has been described in Subchapter 7.3. This chapter aim is to explain how FDRs are created for inventory control and scheduling subproblems. Two types of rule bases are proposed, one for prioritising of orders and one for the inventory control subproblem. An effective SC control-scheme is proposed to manage all echelons. The SC experiments are conducted to observe KPIs which are affected by all echelons' decisions.

7.2.1 Scheduling subproblem

In the previous chapter, the scheduling decision-making used a crisp value of input for determining priority of orders. In this subchapter a fuzzy logic is used for representation of the uncertain parameters to set priority of orders.

Crisp rules such as FIFS, EDD, MTWR and LTWR focuses only on evaluating one input information for ranking priority. The first difference for establishing a new priority by using the FDRs is the dependence on more than one input parameters. Scheduling FDRs consider two inputs, namely due date and slack. Introduction of these two values as antecedent of the rule aims to capture variety of incoming demand by applying fuzzy sets to represent its uncertainty. The demand of considered SC is characterised by the uncertainty in quantity of elements or products in an order and unknown due dates of incoming orders. The considered uncertainties mimic the lack of information faced by SC echelons when information about market demand is not shared with any higher-tiers echelons from the distribution centre. The only information shared between echelons are orders placed by the echelon directly in higher tier echelon. Representation of uncertain parameters in fuzzy logic allows using expert knowledge to determine input and output parameters in the form of linguistic variables. It enables easier gathering of required knowledge, especially for the complex systems with lacking data.

As the MTWR DR was underperforming in experiments conducted in Chapter 6, the processing time (workload) is not used as determining factor for priority of orders in the proposed FDR. However, as the size of order is uncertain, subchapter 6.3.2 shows that the order size changes can lead to a much higher holding cost and long delays in all DRs. Prioritising orders just by their size in the case of LTWR lead to the lowest delays but highest holding cost among other rules and it underperforms in comparison to EDD rule which focuses on due date. Therefore, slack of an order is used as an input parameter instead of the order size. Slack of an order is a difference in time between order deadline and production time which still allows to take size of an order into consideration. Output of the proposed FIS is a priority of orders. Fuzzy antecedents and their effect on priority change is presented in Table 7.1. Firstly, the order with a shorter due date is prioritised. Secondly, to consider the uncertain size of the order, the shorter the slack of an order the higher the priority.

Table 7.1 Relation between input and output for the scheduling problem

Input parameter	**Input change**	**Output parameter**	**Output change should be:**
Due date	Shorter ↓	Priority	Higher ↑
Slack		Priority	Higher ↑

The objective of the FIS is to determine orders priority. Nine rules are proposed for the fuzzy scheduling. This FIS determines fuzzified values of input parameters including due date and slack time and output priority for each echelon. Table 7.2 can be found below. It presents a fuzzy value of two antecedents of rule and one consequent e.g., Rule 1 considers orders with short due date and small slack, which if not prioritised risking a higher delay, hence the priority is very high.

Table 7.2 FDRs for the scheduling problem

	Inputs		Output
Rule	Due Date	Slack	Priority
1	Short	Small	Very High
2	Short	Medium	High
3	Short	Large	Medium
4	Medium	Small	High
5	Medium	Medium	Medium
6	Medium	Large	Low
7	Long	Small	Very High
8	Long	Medium	Medium
9	Long	Large	Very Low

Figure 7.2, Figure 7.3 and Figure 7.4 represent fuzzy sets used to describe uncertain input parameters of order due date and slack and fuzzy representation of priority. All membership functions have been determined empirically for the benchmark scenario. To determine the range of all fuzzy inputs the simulation was first run with crisp DR. During that simulation, the range of each input parameter was measured. Some of those ranges were widened as appropriate and applied to the benchmark scenario. The ranges were then verified in simulation using FDR rules and updated if necessary. The unit of slack and due date is days.

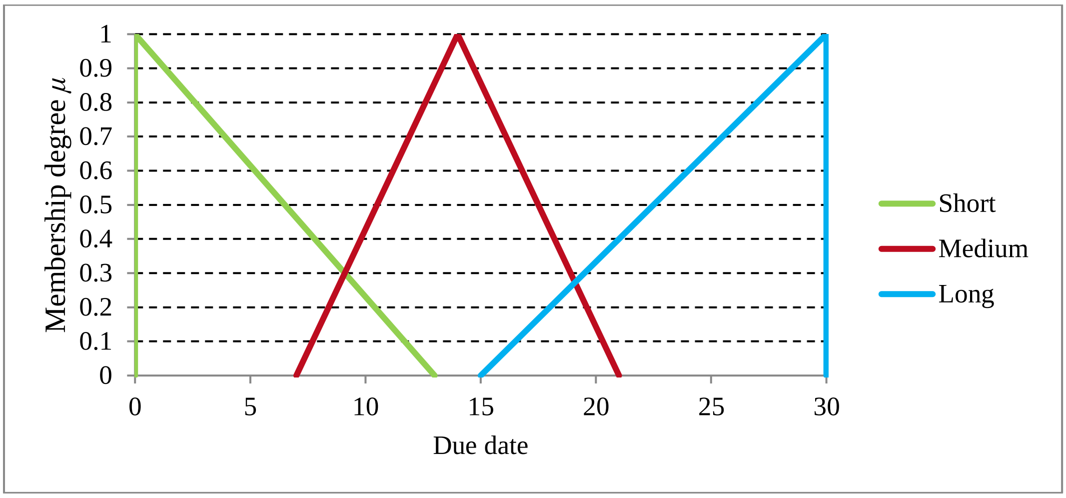

Figure 7.2 Fuzzy representation of the due date

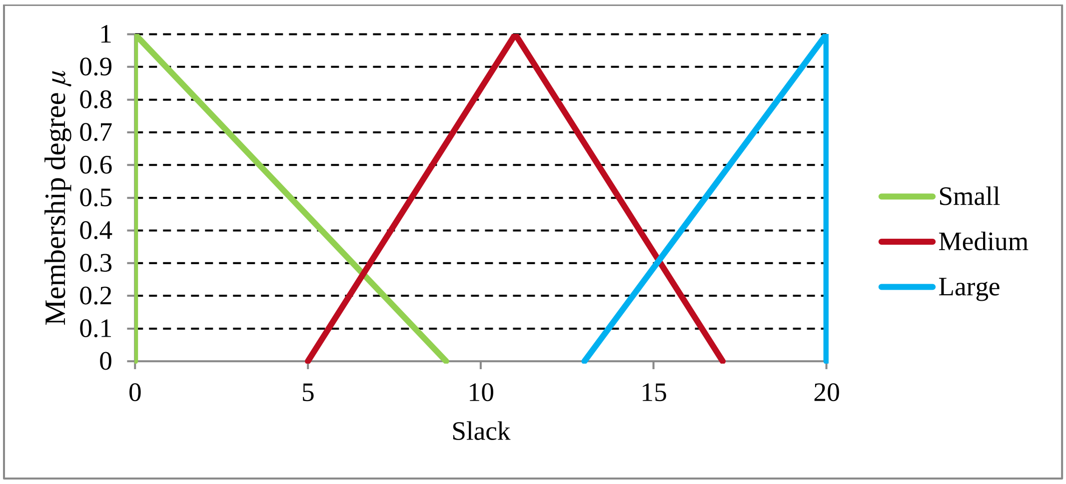

Figure 7.3 Fuzzy representation of the slack value

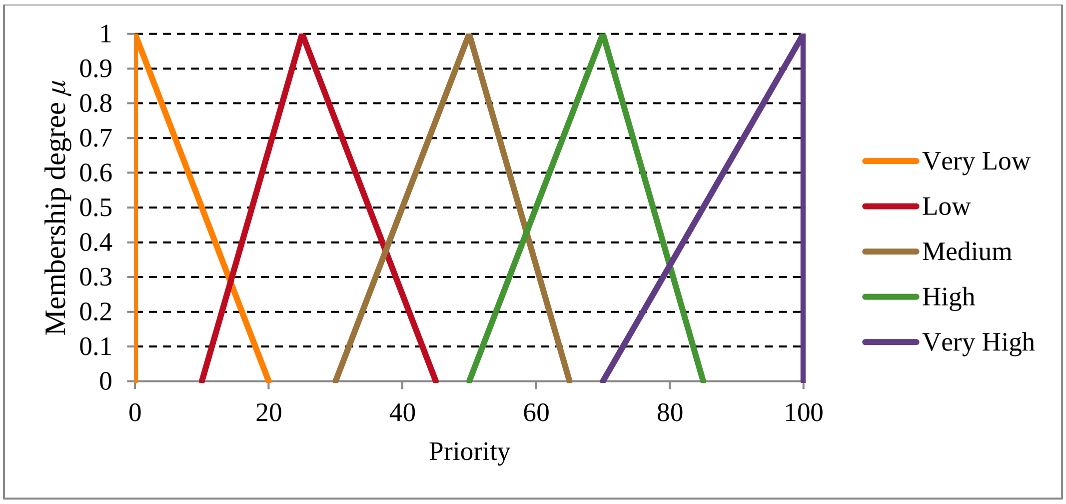

Figure 7.4 Fuzzy representation of the priority

7.2.2 Inventory subproblem

In the previous Chapter 6, a fixed CRP was used which monitors the level of inventory and when the minimum level was reached an order quantity was made. Fuzzy logic uses the same process for the control of the inventory, but adversely to the fixed CRP the output values depend on changing demand. CRP is implemented for two echelons in the proposed SC structure i.e., the Manufacturer and Distribution Centre echelons. Supplier does not hold inventory. Use of the standard crisp CRP led to the lack of raw material which does increase delays.

The main purpose of the proposed fuzzy CRP is to improve a product flow according to the consumer demand. The advantage of the fuzzy CRP is its ability to maintain continuous delivery of products and elements. Two decision variables, the same as for the crisp CRP are used as an output of the proposed FDRs: (1) reorder point and (2) quantity of ordered products and elements. In the proposed SC, the Manufacturer orders, stores and processes elements and the Distribution Centre orders, stores and schedules deliveries of products. To create a rule base for the inventory control problem several crisp inputs were taken into consideration, namely: unit holding cost, order processing time and number of orders to be processed. A relationship between inputs and outputs must be determined. These relationships are introduced in Table 7.3. Each of input affects outputs of the reorder point and order quantities. Holding cost is a crisp parameter and it is not considered uncertain, but it does affect decisions made by echelon. When the holding cost is high, the reorder point should be lower as keeping too much of costly elements and products leads to higher total holding cost.

Table 7.3 Relation between input and output for inventory problem

Input parameter	**Input change**	**Output parameter**	**Output change should be:**
Holding cost	Increasing ➚	reorder point	Lower ↓
Holding cost		order quantity	Higher ↑
Order processing time		reorder point	Higher ↑
Order processing time		order quantity	Higher ↑
Workload (no. of orders)		reorder point	Higher ↑
Workload (no. of orders)		order quantity	Higher ↑

The second input parameter is order processing time related to the varying size of the order. This value is uncertain as there is no certain knowledge about incoming orders available to any echelon. When orders requiring more processing time, the reorder point for inventory and order quantity should be higher, in order to enable flow of orders and lowering delay to the customer. Finally, the last antecedent part for FDRs rules includes changing number of incoming orders. Workload is used to increase both output parameters.

The representation of fuzzy values for the inventory problem can be found in Figure 7.5 for holding cost, Figure 7.6 for order processing time, Figure 7.7 for workload and for outputs representation, reorder point is defined for Figure 7.8 and order quantity in Figure 7.9.

The rules for inventory control include three inputs and two outputs and are summarised in Table 7.4. The unit of processing time is hours. The holding cost is expressed in £/week. The reorder level has more fuzzy values so it can be specified by the rule base at finer granularity.

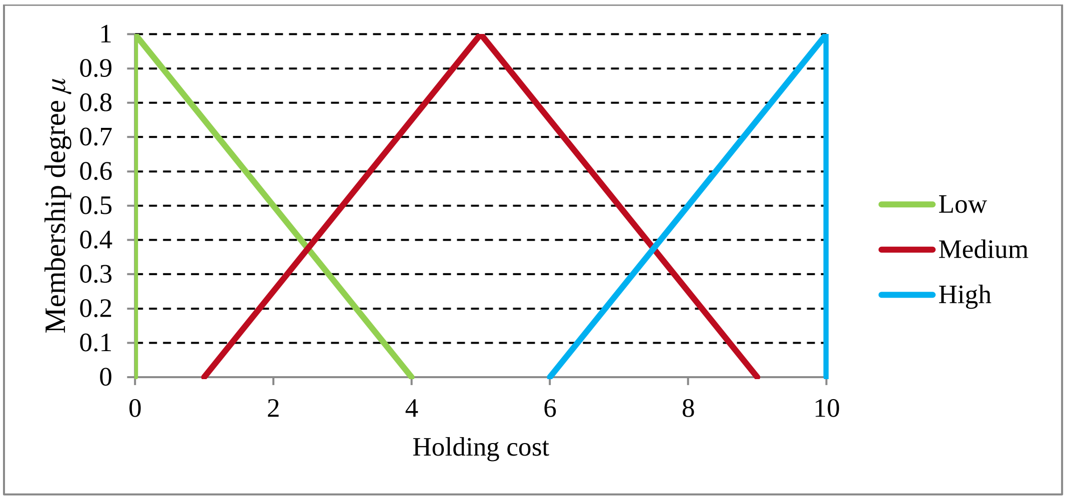

Figure 7.5 Fuzzy representation of the holding cost

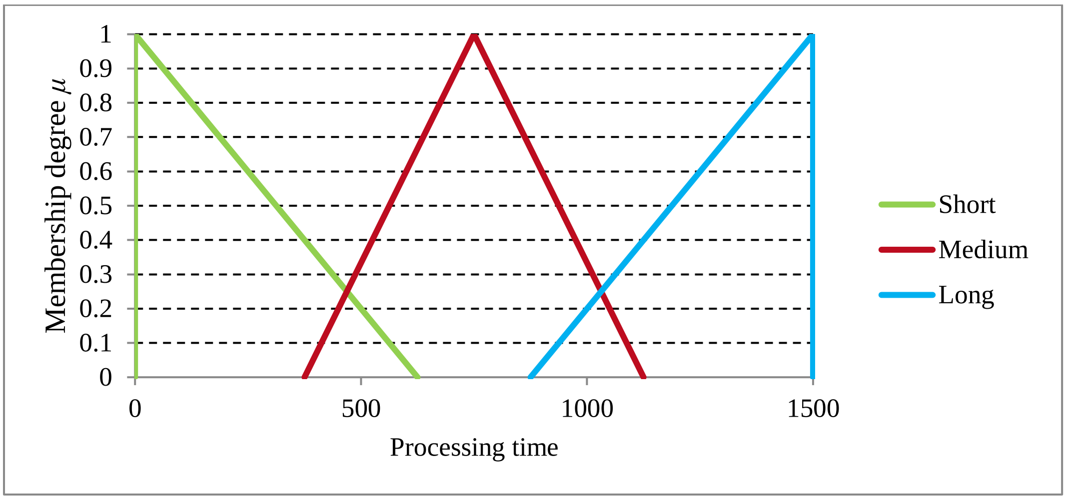

Figure 7.6 Fuzzy representation of the processing time

Figure 7.7 Fuzzy representation of the number of orders

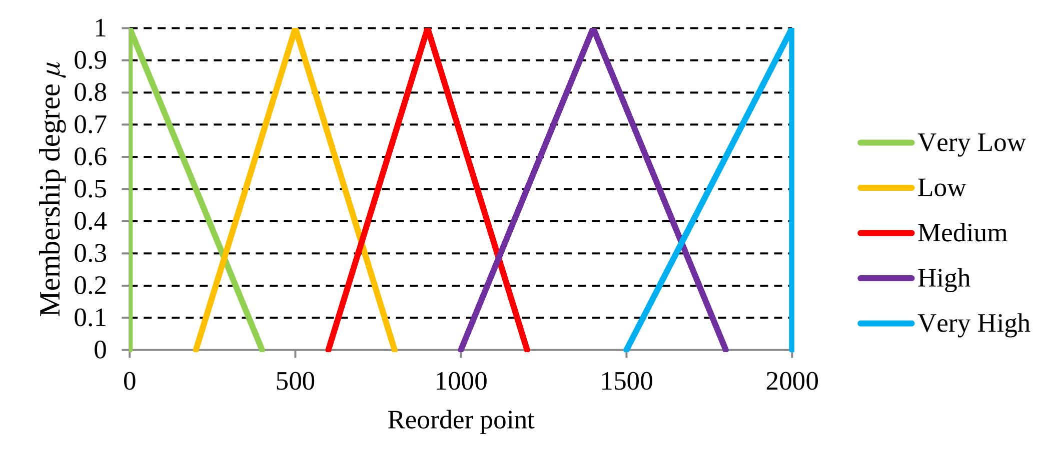

Figure 7.8 Fuzzy representation of the reorder point

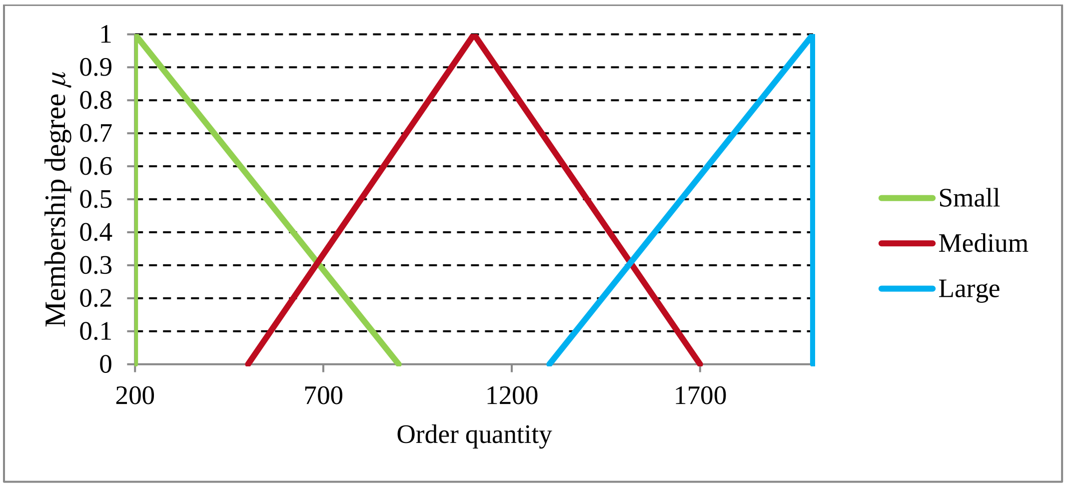

Figure 7.9 Fuzzy representation of the order quantity

Proposed rules can be found in Table 7.4. The values are set empirically, and two rule bases are proposed.

- **A fuzzy Delay-focused CRP** which aims to decrease delays of orders delivered to the customer. Delay-focused rules have a much lower delay risk tolerance. Hence use of this set of rules should ensure a much smaller delays but can increase a holding cost considerably.

- **A fuzzy Holding cost-focused CRP** which aims to decrease the holding cost for all echelons in the SC. Inventory-focused rules will have higher delay risk tolerance and will be willing to keep less in stock even if that can cause some delays. This set of rules should aim to keep the holding cost lower for the price of longer delays.

Table 7.4 FDRs proposed for inventory control subproblem

				Outputs			
	Inputs			**Delay-focused**		**Inventory-focused**	
id	Holding cost	Processing time	Number of orders	Reorder point	Order quantity	Reorder point	Order quantity
1	Low	Short	Small	Low	Small	Very Low	Small
2	Low	Short	Medium	Medium	Medium	Very Low	Medium
3	Low	Short	Large	High	Large	Low	Medium
4	Low	Medium	Small	Low	Medium	Very Low	Small
5	Low	Medium	Medium	High	Large	Low	Small
6	Low	Medium	Large	Very High	Large	Medium	Medium
7	Low	Long	Small	Medium	Medium	Very Low	Small
8	Low	Long	Medium	Very High	Large	Medium	Medium
9	Low	Long	Large	Very High	Large	Medium	Medium
10	Medium	Short	Small	Low	Small	Very Low	Small
11	Medium	Short	Medium	Medium	Medium	Very Low	Small
12	Medium	Short	Large	High	Large	Low	Medium
13	Medium	Medium	Small	Medium	Medium	Very Low	Small
14	Medium	Medium	Medium	High	Large	Low	Small
15	Medium	Medium	Large	High	Large	Low	Small
16	Medium	Long	Small	Medium	Medium	Very Low	Small
17	Medium	Long	Medium	High	Large	Low'	Medium
18	Medium	Long	Large	Very High	Large	Medium	Medium
19	High	Short	Small	Medium	Small	Very Low	Small
20	High	Short	Medium	High	Medium	Low	Small
21	High	Short	Large	Very High	Large	Medium	Medium
22	High	Medium	Small	Medium	Medium	Very Low	Small
23	High	Medium	Medium	Very High	Large	Medium	Small
24	High	Medium	Large	Very High	Large	Medium	Medium
25	High	Long	Small	Medium	Medium	Very Low	Small
26	High	Long	Medium	Very High	Large	Medium	Medium
27	High	Long	Large	Very High	Large	High	Medium

Defuzzification uses Centre of Gravity method.

7.3 Results

The following experiment aim to explore the effects of changing parameters including due date, order size and the processing time on the performance of proposed FDRs. FDRs should be more flexible against changing Customer demand as they consider uncertainty of inputs in the form of FIS. Proposed experiment contains: 112 orders from Customer to a Distribution Centre from a benchmark scenario and the performance of FDRs is measured by the KPIs introduced in Chapter 6.

The values of KPIs are analysed in this subchapter with consideration of different due dates in Subchapter 7.3.1, different order sizes in Subchapter 7.3.2and different product processing time in Subchapter 7.3.3.

7.3.1 FDRs for changing due dates

Experimental results confirmed that delay-focused FDR systematically achieves lower delays than holding cost-focused FDR when the due date id changed. The opposite is true for the holding cost. As can be observed in Figure 7.10 where the due date increases the holding cost rises for both FDRs albeit at a different rate.

Table 7.5 Comparison of two FDRs holding cost for different due dates

		FDRs holding cost for different due dates (£)				
		shorter by 50%	shorter by 25%	benchmark	longer by 25%	longer by 50%
FDR	Delay-focused	199844	200353	229346	237885	254723

	Holding cost-focused	163968	165950	171263	180204	186841

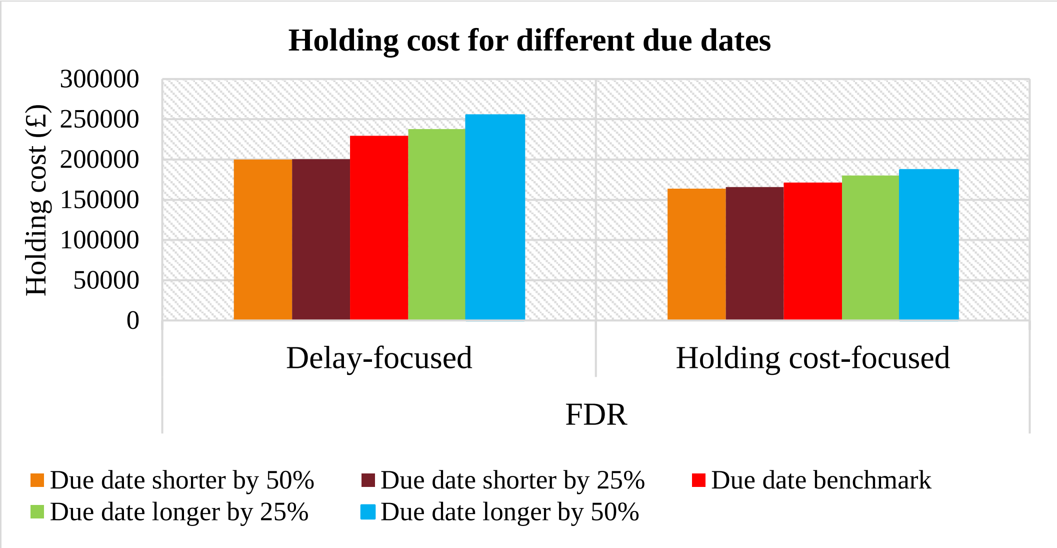

Figure 7.10 Comparison of two FDRs holding cost for different due dates

Increase in holding cost is expected to raise with increasing of the due dates as the inventory is kept for longer time before it can be processed by Manufacturer or sent by the Distribution Centre to Customers. Use of delay-focused FDR result in keeping similar delays regardless of changes in due dates. As this FDR goal is to avoid delays it keeps higher inventory levels throughout the simulation which cause by higher holding cost. It is interesting to notice that for a benchmark scenario the holding cost achieved by applying holding cost-focused FDR is only 25% lower from delay-focused FDR while the delay of delay-focused FDR achieves delay reduction of 63%. This can be explained by the fact that excessive delay leads to holding cost being incurred in the stock which is unused, and it is kept in inventory while waiting for all products or elements required to fully satisfy the order.

Table 7.6 Comparison of two FDRs delay for different due dates

		FDRs delays for different due dates (hours)				
		shorter by 50%	shorter by 25%	benchmark	longer by 25%	longer by 50%
FDR	Delay-focused	1789	1699	1929	1948	1864
	Holding cost-focused	5393	5302	5252	5053	4960

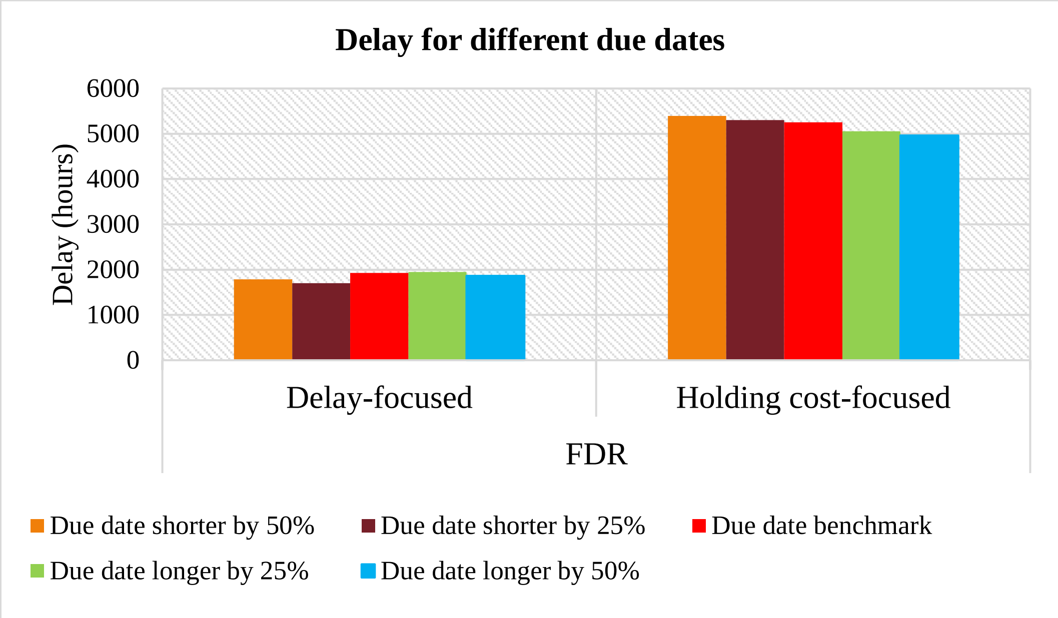

Figure 7.11 Comparison of two FDRs delay for different due dates

7.3.2 FDRs for changing order size

Increase in order of orders causes demand for larger quantities of raw material. The due date and rate of production does not change therefore keeping a sufficient inventory becomes crucial as order size increases. Figure 7.12 and Table 7.7 present holding cost for different order sizes and Figure 7.13 and Table 7.8 present comparison of the delay KPI when the two FDRs are applied.

Table 7.7 Comparison of FDRs holding cost for different order size

		FDRs holding cost for different order size (£)				
		smaller by 50%	smaller by 25%	benchmark	larger by 25%	larger by 50%
FDR	Delay-focused	202025	158899	229346	264274	265541
	Holding cost-focused	132196	159760	171263	205226	251005

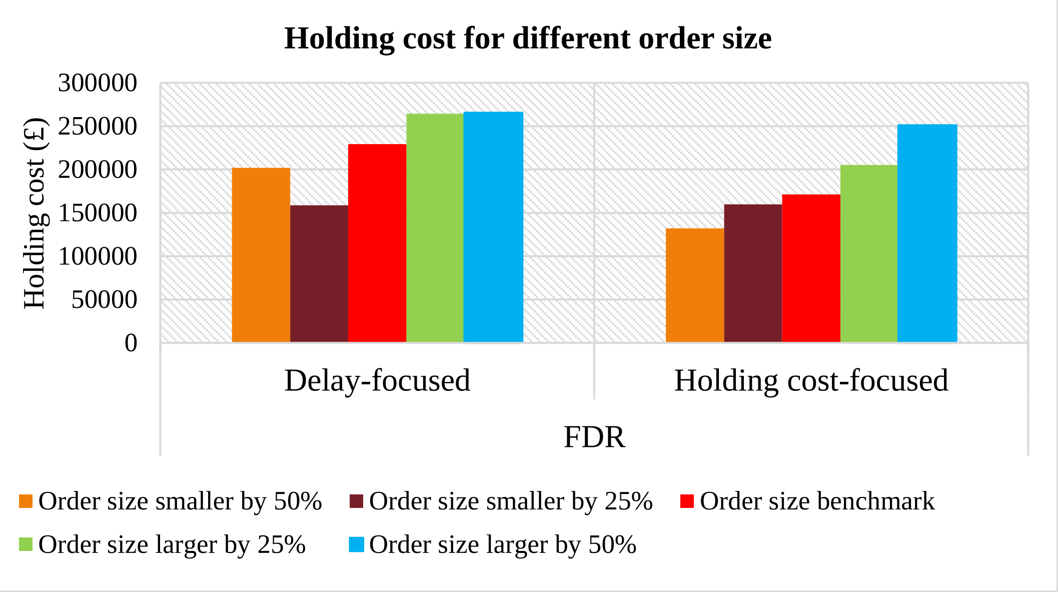

Figure 7.12 Comparison of FDRs holding cost for different order size

Both KPIs were increased in line with the order size for holding cost-focused FDR. Increasing processing time of orders affect inventory levels, thus the holding cost goes up. However, since this FDR attempt to keep the holding cost low it is not sufficient to satisfy increasing demand and leads to escalation of delay, as presented in Figure 7.13 and discussed later.

On the other hand, the delay-focused FDR keeps the holding cost higher by ordering more and thereby succeeding at containing the delay. This leads to keeping much more inventory than needed when the order size is smaller.

Interestingly, both FDRs reach similar holding costs while order size is increased by 50% and when it is reduced by 25% despite the disparity in the delay. This can be explained by different holding cost structure in different echelons. Delay-focused FDR keeps sufficient levels of products for the Distribution Centre throughout entire simulated time horizon, while holding cost-focused FDRs struggle to keep sufficient levels for Product 1 and 3. Long delays increase the time required for delivery of all orders to the Customer by over two months. The holding cost for that extra time contributes to the total cost for holding cost-focused FDR thus erasing any savings made by keeping lower levels of stock.

Table 7.8 Comparison of FDRs delay for different order size

		FDRs delays for different order size (hours)				
		smaller by 50%	smaller by 25%	benchmark	larger by 25%	larger by 50%
FDR	Delay-focused	129	138	1929	3398	5438
	Holding cost-focused	185	2407	5252	8390	12167

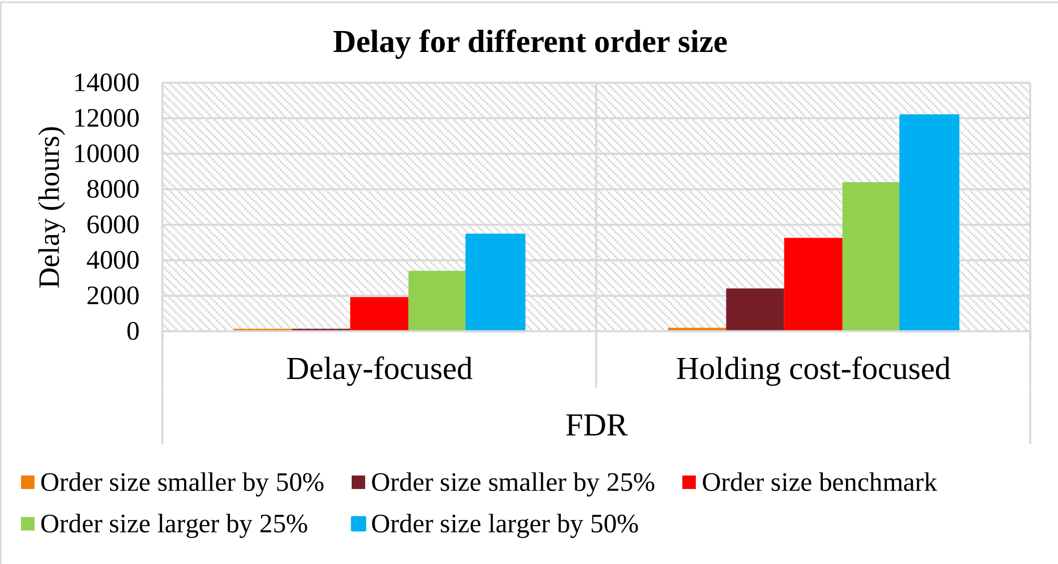

Figure 7.13 Comparison of FDRs delay for different order size

7.3.3 FDRs for changing product processing time

In this experiment product processing time is changed. When production of the product is about to be started, the necessary quantities of raw materials are collected from the inventory. Then, after the processing time passes, the finished product is obtained, and production of another product can be started. Since only the processing time is changed, that means that when product processing time is shortened, the rate of inventory consumption increases. The opposite is also true when product processing time is extended, the rate of inventory consumption decreases. It is important to note that rate of production of raw materials and their lead times are not affected. If processing time is shortened, the inventory stock level in the Manufacturer echelon may no longer be high enough to sustain manufacturing process. Accordingly, when processing time is increased, the inventory level in the Manufacturer echelon may be higher than necessary.

The effect of shortening processing time on KPIs for FDRs has been mixed. The decrease of processing time by 25% lead to lower delays and lower holding cost for both FDRs, compared to benchmark scenario. This can be explained by the fact that higher rate of inventory consumption will mean that stock will be effectively kept for shorter time, reducing the holding cost. If the inventory level is not sufficient and the production task is delayed, then once replenishments arrive, the production can be finished faster due to shorter processing time. That may result in lower delays. On the other hand, when processing time was shortened by 50%, the holding cost was lower compared to the benchmark scenario only for delay-focused FDR, but not as low as when it was only shortened by 25%.

Similarly, the total delay was only lower compared to the benchmark scenario only for delay-focused FDR, but again not as low as when it was only shortened by 25%. In case of holding cost-focused FDR both KPIs were higher compared to the benchmark scenario. Those results suggest that since delay-focused FDR tends to keep more stock, it was able to keep sufficient stock most of the time, despite increases in the consumption rate. In turn, the higher consumption rate means less stock remains, reducing the holding cost. However, since KPI values were not as good as when processing time was shortened by 25%, the inventory stock was at times insufficient, leading to the escalation of delay and thus rise of the holding cost as has been explained above. Since holding cost-focused FDR tends to keep lower stock levels, it suffered frequent stock shortages which contributed to both delay and holding cost of remaining stock.

In case of delay-focused FDR, the increase of processing time resulted in lower KPIs values compared to the benchmark scenario. Extending production time has an effect of lowering consumption rate as noted above. It will also mean that since order due dates are not changed, the production will be scheduled to start faster. Of course, this is not always possible and a large increase in production time will contribute to delays. Both lower consumption rate and earlier start of production helps with inventory replenishments, since the inventory may reach reorder point earlier, and longer processing time allows for longer lead-times of ordered replenishments. However, if any delay is incurred, it will most likely be more severe as manufacturing products takes more time. This will counteract the benefits, explaining why KPI levels were better when processing time was shortened. The holding cost-focused FDR achieved lower delays compared to the benchmark scenario when processing times were increased as can be seen in Table 7.10 and in Figure 7.15. However, the total holding cost increased compared to the benchmark scenario as can be seen in Table 7.9 and Figure 7.14. This can be explained by lower rate of inventory consumption leading to more stock being kept for longer, thus contributing to holding cost. In terms of the delays, this FDR could have benefited from the same effects the processing time has as have been explained above for the delay-focused FDR.

Table 7.9 Comparison of FDRs holding cost for different product processing time

		FDRs holding cost for product processing time (£)				
		shorter by 50%	shorter by 25%	benchmark	longer by 25%	longer by 50%
FDR	Delay-focused	204884	177248	229346	202864	209781
	Holding cost-focused	200619	161776	171263	185149	185713

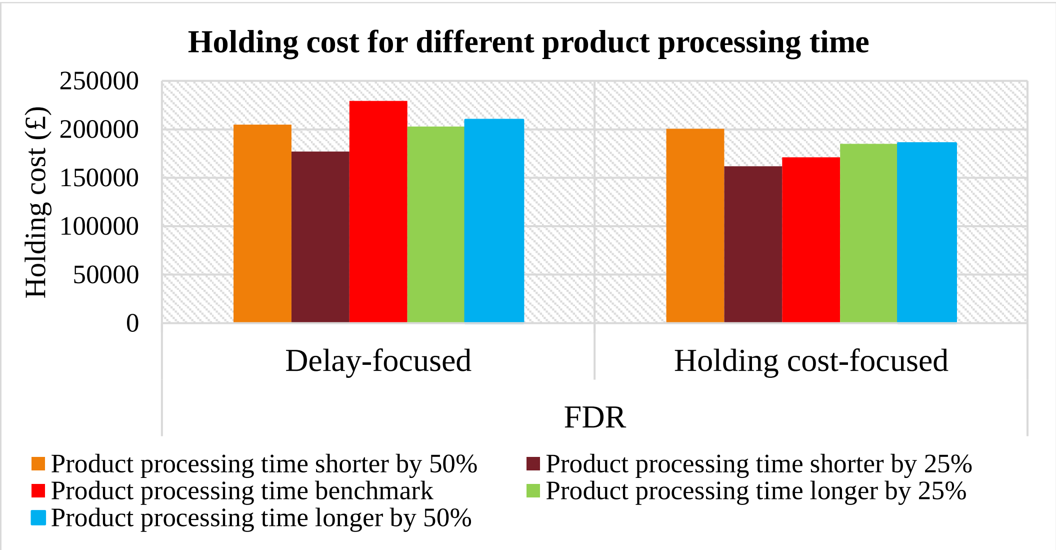

Figure 7.14 Comparison of FDRs holding cost for different product processing time

Table 7.10 Comparison of FDRs delay for different product processing time

		FDRs delays for different product processing time (hours)				
		shorter by 50%	shorter by 25%	benchmark	longer by 25%	longer by 50%
FDR	Delay-focused	1618	1342	1929	1698	1759
	Holding cost-focused	5706	4979	5252	4800	4775

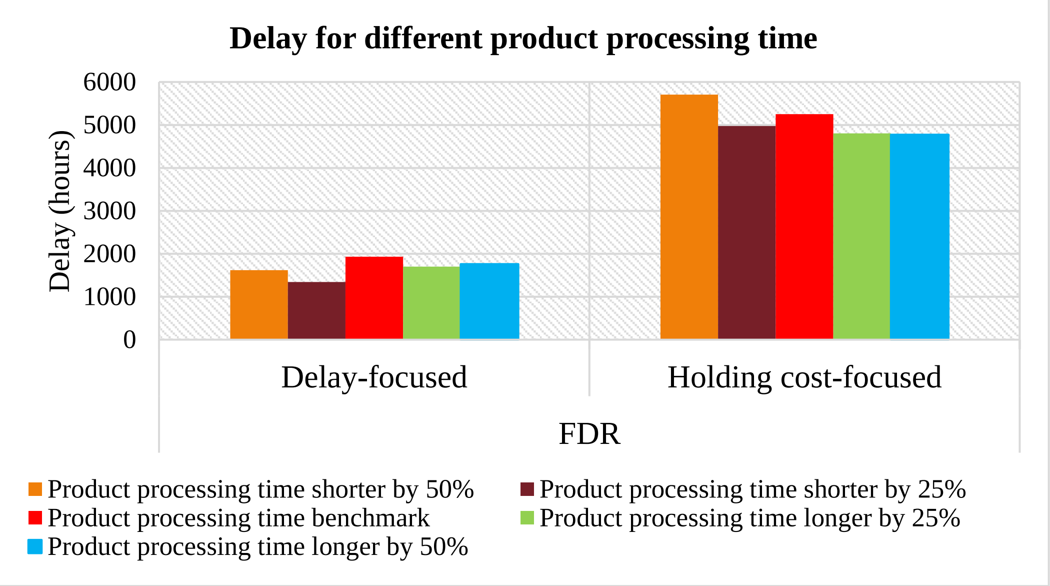

Figure 7.15 Comparison of FDRs delay for different product processing time

7.4 Conclusions

If the inventory levels are not adequate the delay becomes substantial and it starts to contribute to the total holding cost by extending the time required for completing of all orders. For example, the control-scheme may opt for lower reorder point and/or lower order quantity values, to keep holding cost KPI low. This will most likely result in increasing the delay KPI, as there is a trade-off between

KPIs. If reorder point and/or order quantity is kept too low too often by the control scheme, the inventory will be often depleted, and significant delays will occur. However, it likely that at least some of the inventories in the SC will still have some stock while production is delayed. For that stock, the additional holding cost will be incurred, compared to the situation where that delay did not happen because the stock was sufficient. Thus, any gains in terms of lower holding cost achieved by keeping inventories at lower levels may be lost when substantial delay is incurred. This results in both KPIs being worsened, indicating that the trade-off between KPIs has a limit.

The proposed FIS allows insight to be obtained to all parameters in any moment of a simulation. Two types of fuzzy rule bases were proposed as control-schemes for considered SC with consideration of inputs which are related with SC uncertainty. It is worth noticing that all input and output fuzzy numbers used in the proposed rule bases were defined manually and there is a further opportunity to improve performance on both Fuzzy CRP controllers holding cost and delay focused. It might be obtained by automatization of selection of membership functions, changing fuzzy outputs and testing different overlaps between defined inputs and outputs

Factors affecting values selected for the FDRs are highly dependent on SC structure, its experts' knowledge (as inventory managements, staff maintaining the machines, echelons capacity and so on). It is then assumed that Expert-based fuzzy model can be improved according to available experts' knowledge and proposed FDRs are flexible to be used for different echelon structures.

8 NSGAII FOR SC INVENTORY CONTROL AND SCHEDULING PROBLEMS

8.1 Introduction

To determine improved parameters for simultaneous inventory control and scheduling of orders across SC, a new Fuzzy Dispatching Rules considering uncertainties of SC parameters were proposed in Chapter 7. The following chapter aim is to propose a multi-objective optimisation of the proposed FDRs by improving decision variables determined in the previous chapter.

One of the challenges of this optimisation is to find FDRs which lead to the minimised values of holding cost and delay of metaheuristic's fitness functions selected for decision-making in a proposed control-scheme. NSGAII is selected to solve this dynamic problem. A simulation framework introduced in Chapter 5 is used to evaluate proposed control and conduct a comparison between NSGAII generated results with Crisp and FDRs proposed in the previous chapters. The following subchapter includes representation of rules in chromosome, decision, fitness functions and reproduction parameters (Subchapter 8.2). The results section includes input parameters and analysis of FDRs proposed by NSGAII algorithm. Comparison of different intensities of orders is considered in Subchapter 8.3.2.

Next an increase of solutions robustness is considered by creating a new robustness metric and use of Monte Carlo Simulation. This enables generation of more robust FDRs which lead to good SC KPIs in various scenarios in addition to the benchmark scenario. The robustness is measured by standard deviation and average values of holding cost and delay after small changes are applied to the input data, such as different order time, increased and decreased quantities of products and cancelation of orders (Subchapter 8.4). Discussion and conclusions are described in Subchapter 8.5.

8.2 Parameters of NSGAII

Dominance-based MOGA is proposed to find a solution for a multi-objective inventory and scheduling control problem across the SC echelons. A dominance concept allows creating a Pareto front of solutions which represents a trade-off between high delays of the orders and a cost of keeping an inventory.

8.2.1 Representation of NSGAII

8.2.1.1 Encoding of Chromosome and Fitness function

Previously used KPIs including inventory holding cost and delay of the customers’ orders will be used as objectives to NSGAII algorithm. Advantage of NSGAII in solving MOP is in its design which allows finding Pareto front of solutions and allow decision maker to assess a trade-off between multiple objectives. The role of the decision maker is to specify additional information to select a preferable solution. A fitness function is not changed into a single objective. Instead, a dominance-based approach is used to guide the search process.

Proposed GA is using a chromosome in a form of 27 pairs of decisions for inventory control problem and 9 decisions for scheduling problem, one for each FDR, which together defines solutions. Chromosome for the GA for the inventory control subproblem requires 54 genes as each FDR requires two genes to represent reorder point and order quantity decisions. Each gene of this chromosome can have a fuzzy parameter value. For the inventory control a gene can take three possible values for reorder point and five for order quantity values.

- Three possible values for the reorder point stands for *low*, *medium* or *high* (1, 2, 3)
- Five possible values for the order quantity stand for *very low*, *low*, *medium*, *high* and *very high* (1,2, 3, 4, 5)

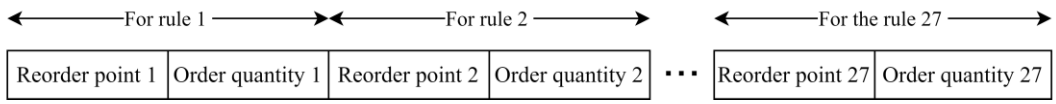

Figure 8.1 Chromosome’s solutions representation for inventory subproblem

A chromosome for the GA for the scheduling subproblem is proposed below.

- Each requires 9 decisions and for the priority FDRs. Chromosome with solution can take five possible values for each gene which stands for *very low, low, medium, high* and *very high* priority.

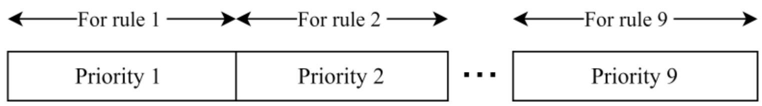

Figure 8.2 Chromosome’s solutions representation for scheduling subproblem

A GA population size includes 100 individuals with two chromosomes to represent possible solution space. Termination criteria used in this work includes a fixed number of 60 iterations, but the improvement was not seen after 40 iterations. Following results are presented for solutions obtained in 40^{th} iteration.

8.2.2 Reproduction Operators

Reproduction phases include determination of *fitness functions* used by the algorithm, a *crossover* responsible for inheriting characteristics from two parents, and *mutation* which represents a random change of individual solution.

8.2.2.1 Fitness functions

Two fitness functions are considered by the GA. Both previously used KPIs are used for this, where first fitness function is a total holding cost, second is the delay of delivering orders to the final Customer.

Minimise the holding cost:

$$\min H_{dc} + H_{mf}$$

Minimise the delay:

$$\min \sum_{od} D_{od}$$

8.2.2.2 Mutation

Mutation can be defined as a flip operator which does change the gene in a chromosome with a consideration of its validity in a search space. A mutation should be a minimal change in gene as defined in Methodology Chapter. The effect of mutation *locality* is its ability to search solution space. High locality leads to more thorough search of solution space as opposite to the cases where locality of mutation operator is weak.

With $27 \times 2 = 54$ decision variables in the proposed problem, probability of the mutation is selected to be $P_{Mutation} = \frac{1}{54}$ for inventory subproblem. There is $\frac{1}{9}$ probability, that one of the values for scheduling problem will be changed into a random fuzzy value during each iteration. Mutation rate fits into range $0.001 \leq P_{Mutation} \leq 0.2$ that was suggested in Methodology Subchapter 3.4.1.

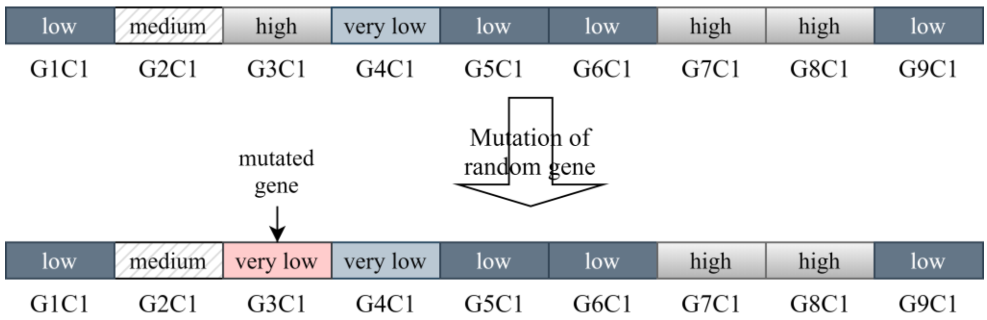

Figure 8.3 Mutation operator used for this problem on priority example

8.2.2.3 Crossover

Crossing two best rules in a uniform way is selected for this problem where two parents are randomly selected to create two offspring. Figure 8.4 presents a crossover of two chromosomes for the scheduling subproblem. Each gene takes one out of five values representing priority. Genes 1, 3, 7 and 8 are randomly selected and switched creating two new offspring rules.

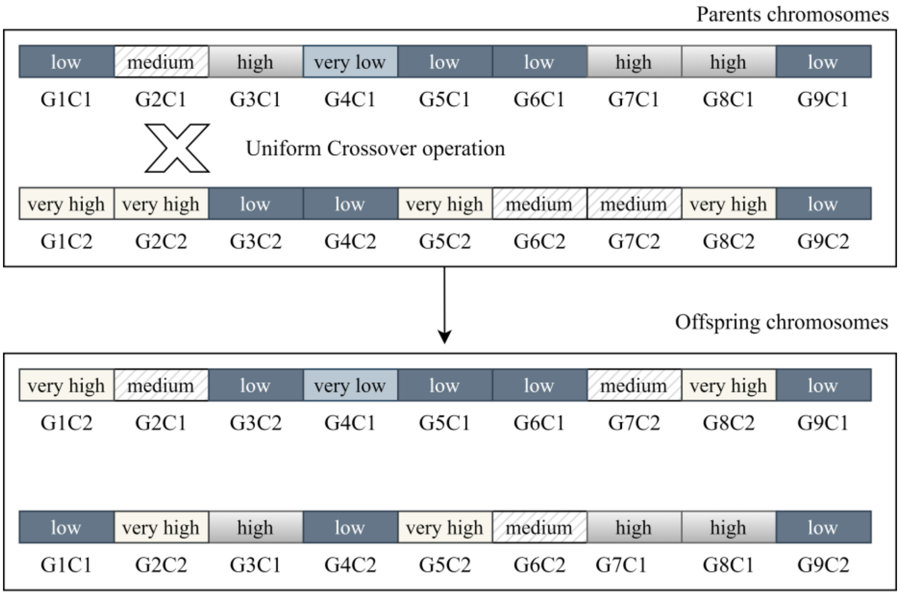

Figure 8.4 Crossover operator used for this problem on priority example

8.3 Results for FDRs proposed by the NSGAII

8.3.1 Different intensity of orders

So far only a single scenario was used to guide optimisation and evaluate the performance of DRs with pre-set replenishment levels and FDRs for both; scheduling and inventory control subproblems. There is a risk of the rule bases generated by NSGAII overfit to the used benchmark scenario. To assess performance of FDRs generated by GA and the robustness of all proposed control-schemes, experiments with different intensity of incoming orders are conducted. Benchmark scenario of medium intensity is analysed, and two additional scenarios are introduced. The scenarios are derived from the benchmark one and differ in intensity of the uncertain customer orders. The low intensity scenario represents a scenario of lower demand, while the high intensity scenario represents a higher demand. Orders coming from the Customers are unrelated and therefore data representing incoming orders from the Customer to the Distribution Centre echelon has been modelled by a Poisson process. The Poisson Process models a sequence of independent random events, where number of such events in a fixed time interval is given by random variable with a Poisson distribution. The process is stochastic. Each orders list generated by the Poisson Process will be different in each sequence of orders which are placed by the Customer. The process is discrete as the number of incoming orders per given time interval must be an integer number. The time intervals are independent between each other, so that events in subsequent time intervals are independent from each other.

The Poisson Distribution gives the probability of observing ***n*** incoming orders ***o*** in each time interval and the average number of events per time. The probability that ***n*** orders will happen in the given time interval can be described by a Probability Mass Function (PMF):

$$P(n\ orders\ \ for\ interval) = e^{-\frac{orders}{time}\times time\ period} \times \frac{\left(\frac{orders}{time} \times time\ period\right)^{n}}{n!}$$

Where $\frac{orders}{time} \times time\ period$ can be simplified into parameter λ, so that, probability $P = e^{-\lambda} \times \frac{\lambda^{n}}{n!}$, where:

- e = 2.71828 Euler's number
- λ is the expected number of events in the time interval
- n is an integer number of events

A Poisson process with a rate (λt) is used to model a sequence of orders placed from Customers to the Distribution Centre. PMF of n orders in time interval t of one week can be described as $P(n) = \frac{(\lambda t)^n \times e^{-(\lambda t)}}{n!}$.

For simulation purposes it is more convenient to work with the time between events in a Poisson process rather than number of events. Therefore, exponential distribution was used to generate incoming orders of different intensity levels. Three different λ (the total number of events per week) values are considered:

- λ = 5, corresponding to *Low* intensity
- λ = 10, corresponding to *Medium* intensity
- λ = 15, corresponding to *High* intensity

Orders generated for these experiments are presented in Figure 8.5 and Table 8.1.

Table 8.1 Orders generated by Poisson Process for different intensities

Week	Orders for λ = 5	Orders for λ = 10	Orders for λ = 15
1	8	6	13
2	4	5	10
3	7	8	13
4	8	10	12
5	7	11	11
6	8	11	6
7	4	9	14
8	2	11	17
9	3	12	10
10	6	8	9
11	5	9	8
12	6	8	15
13	3	4	12

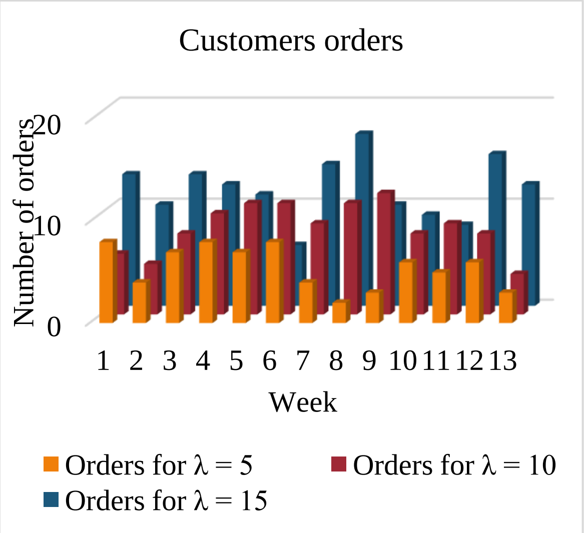

Figure 8.5 Orders generated by Poisson Process for different intensities

Several GA solutions from the rank 1 were selected to be compared. Although GA produced rule bases which led to the lowest values of objectives as seen in Figure 8.8, Figure 8.7 and Figure 8.9 . As can be seen, FDRs selected by NSGAII outperform all previously analysed methods. It is interesting to notice that different rule bases proposed by NSGAII lead to the same SC performance. Each rule base consists of 27 rules and these similarities can be explained by *different levels of rules activation*. For example, let's consider *FDR 1* and *FDR 5* for the four different rule bases of rank 1 generated by NSGAII. The IDs of rule bases are: 1131, 1234, 1246 and 1275 and use any of them lead to exactly same objectives values of holding cost and delay regardless significantly different *FDR 5* as presented in Figure 8.6.

In a given scenario, *FDR 1* was activated, whereas *FDR 5* was not; the scenario, did not include an instance of *FDR 5*, where **holding cost** was *low*, **processing time** was *medium,* and **number of orders** was *medium.* That allowed NSGAII to propose FDR*1* which considers similar reorder point and the same value of order quantity, but *FDR 5* has significantly different output values for reorder point without changing the values of KPIs.

ID:	1131		1234		1246		1275	
	eorder point	Order quantity	Reorder point	Order quantity	Reorder point	Order quantity	Reorder point	Order quantity
Rule1	Very Low	Medium	Low	Medium	Low	Medium	Very Low	Medium
Rule5	Very Low	Small	Very Low	Small	Very High	Small	Very High	Small

Figure 8.6 *FDR 1* and *5* for four different rule bases generated by the NSGAII for inventory control subproblem

The NSGAII solutions with ID:1131 and ID:1246 for a scenario which does include *FDR 5* instance lead to different results. Therefore, it may be concluded that rules that are generated depend on scenario use. These differences can be observed on how different GAs behave under different intensities (Figure 8.7, Figure 8.8, Figure 8.9). It will be beneficial to conduct a Monte-Carlo analysis, i.e., generate multiple random scenarios for each order intensity to ensure that the set of rules proposed is the most appropriate. Results from each random scenario can then be used to calculate statistical parameters of the results such as mean value and variance in order. These statistical values are then used to evaluate the robustness of proposed solutions and the stability of the trade-off between the SC KPIs.

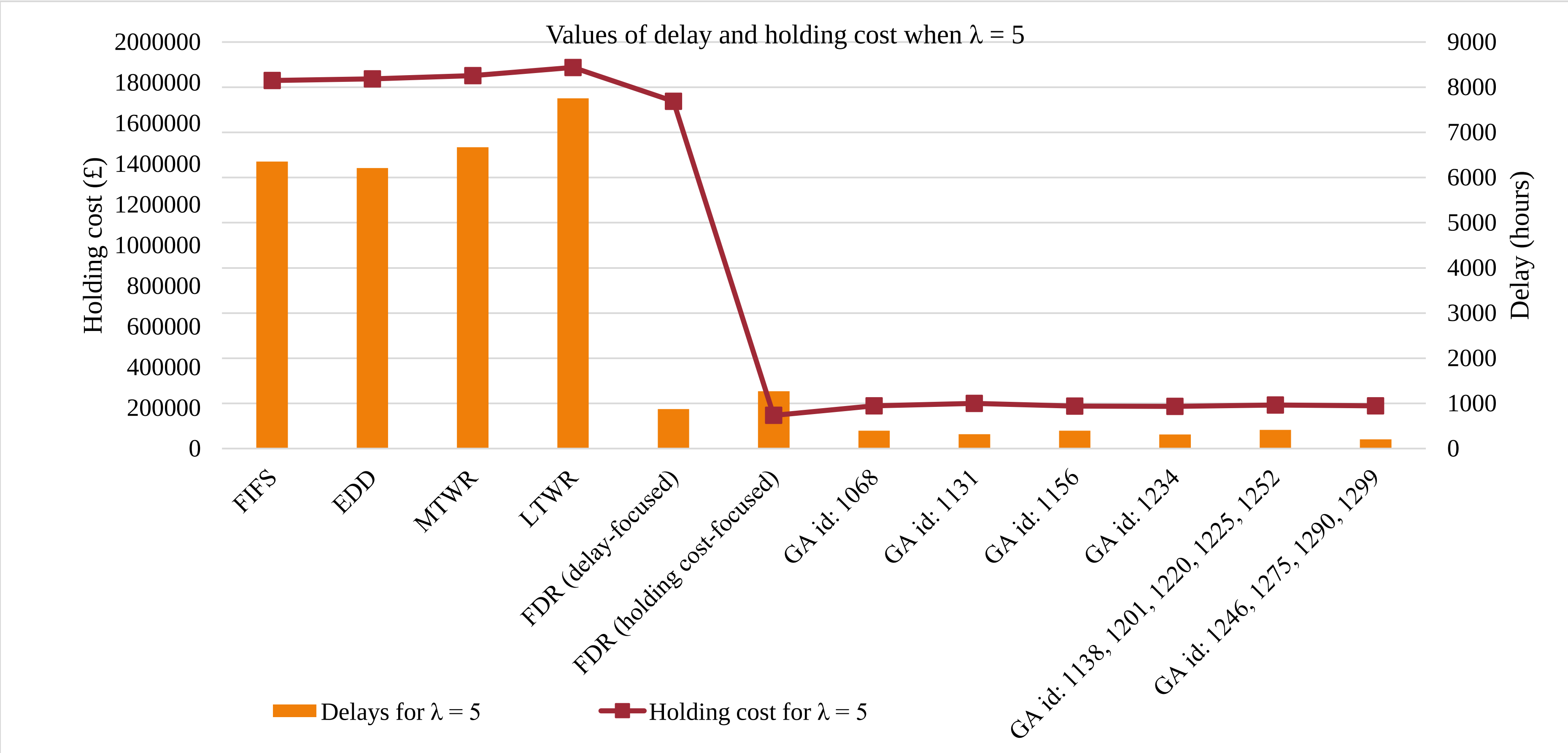

Figure 8.7 KPIs for low intensity orders

Different solution obtained via GA had similar performance values in medium intensity scenario, despite considerable differences in the rule bases. In the case of lower and higher intensity scenarios, the differences among the performance values were more pronounced. Radically different rules were not activated very often in a medium intensity scenario, but their activation levels increased in other scenarios. Monte Carlo Simulation can eliminate problem of not activated rules as experiment can be conducted on many scenarios.

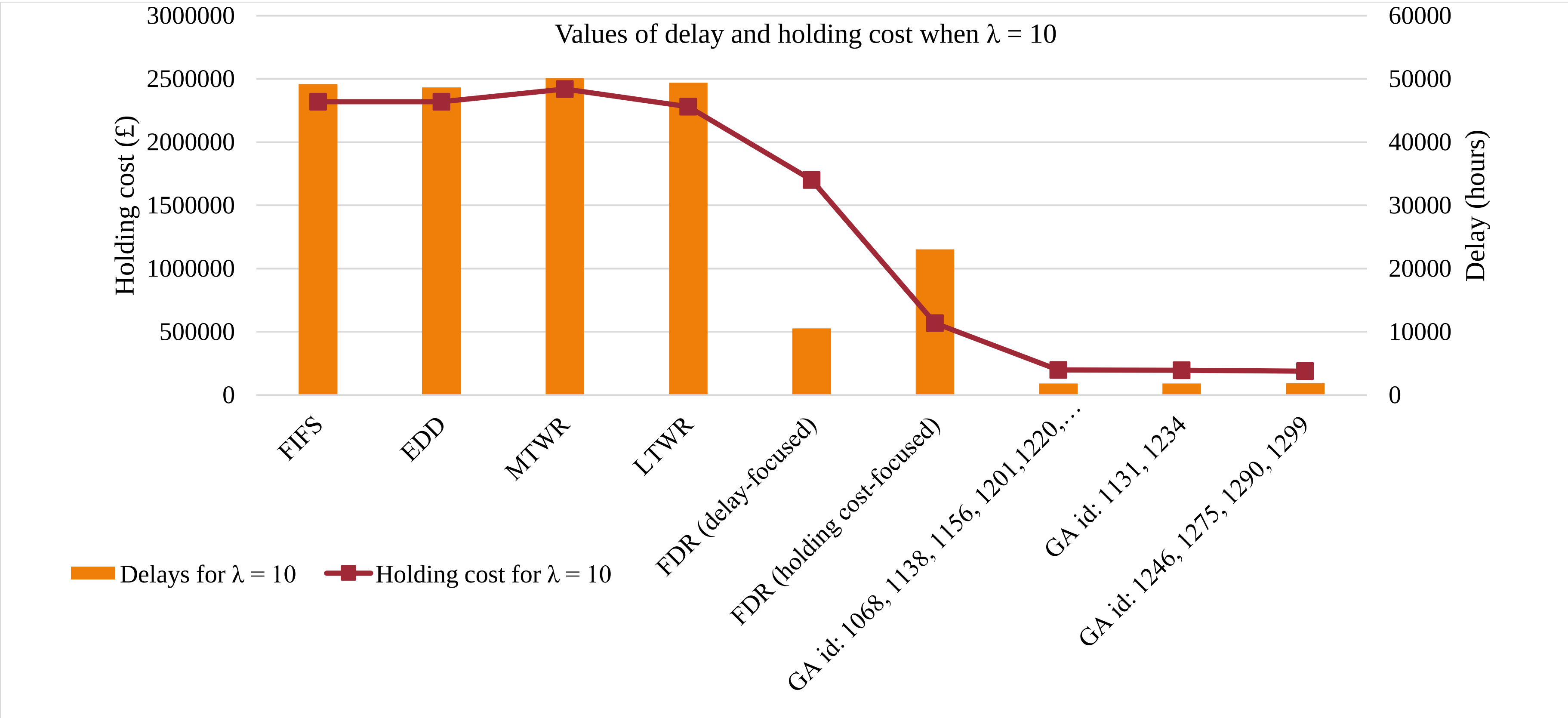

Figure 8.8 KPIs for medium intensity orders

When the same solutions were applied in high intensity scenarios, the GA-based solutions provided superior objective values. The objective values of individual GA solutions are even more diverse. Those solutions provided even more trade-off between objectives than for low and medium intensities.

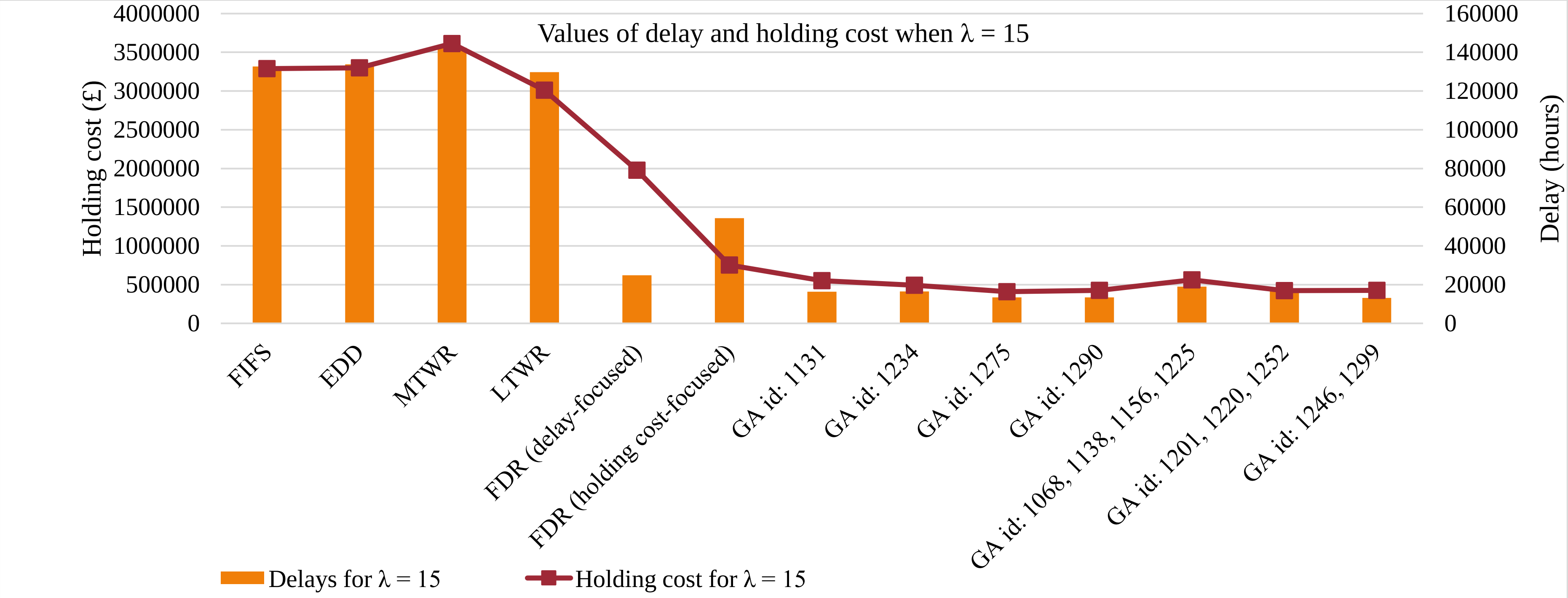

Figure 8.9 Values of KPIs for high intensity orders

8.4 Monte Carlo simulation within NSGAII

The goal of applying the NSGAII is not only to find a good solution for one specific scenario as described in the section above, but to propose solution which can perform well for different sequence of orders of the same intensity as incoming orders are uncertain. To assess the robustness of FDRs introduced for scheduling and inventory control problems, the performance of these FDRs must be measured on multiple scenarios. The values such as number of orders, order sizes and order deadlines are different for each scenario. Robust FIS design should allow to deal with the input uncertainty to achieve robust performance i.e., the use of the FDRs proposed by the NSGAII should lead to the equally good solutions even in a face of these uncertainties.

There is no commonly accepted definition of the solution robustness. In the context of metaheuristics, robustness can be understood as a solution ability to remain close to the optimal solution despite changes in the system input and its ability to perform on variety of instances (Mulvey et al. 1995). In the stochastic metaheuristics' robustness can also be measured by observing standard deviation and average values of solutions linked to its performance.

To implement robust control that support finding *comparably* good solutions for similar scenario cases, a Monte Carlo Simulation is proposed. The flowchart of Monte Carlo is presented in Figure 8.4.

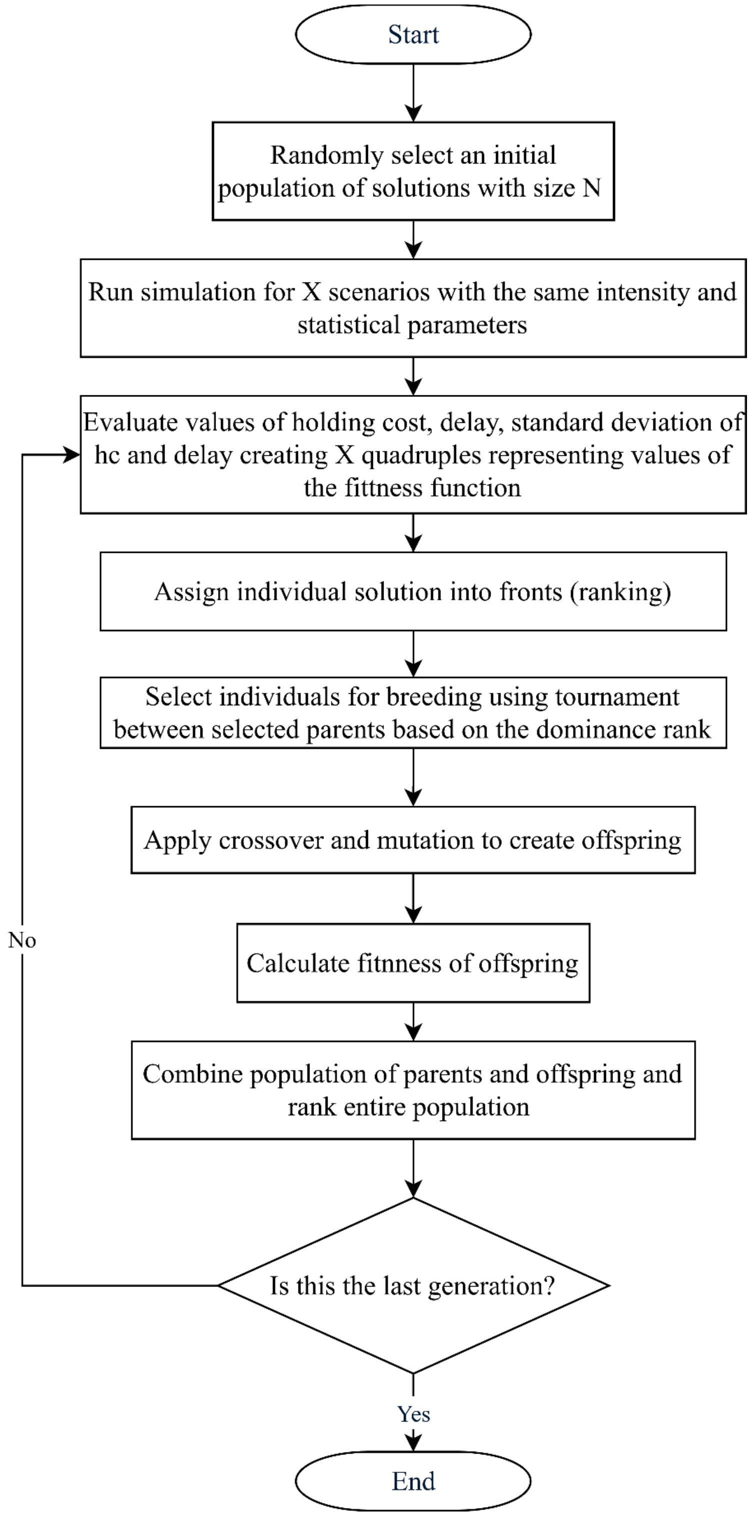

Figure 8.4.GA with Monte Carlo flowchart

The proposed metric will be simply called a *normalisation* as it refers to the creation of shifted and scaled version of uncertain parameters such as change of the order's size for a smaller or larger, deleting of some parts of orders or changing the order date. Intention for calculating normalised values of holding cost and delay allow the comparison of different datasets in a way that eliminates the effects of changing input parameters. The list of small, medium, and large changes can be found in Table 8.2. This table includes different types of changes applied randomly to the demand.

Normalisation factor is different for each echelon. It will be different for different scenario as quantity of orders and time necessary to keep those will be different. Based on the normalised value of holding cost and delay, the standard deviation and average values are calculated from multiple Monte Carlo trials. Both, average values, and standard deviation for these two KPIs, becomes new fitness functions of the Monte Carlo NSGAII (MCNSGAII). The goal of MCNSGAII is to generate rule bases invariant to the small input changes thus to the uncertain input. Normalised KPIs are calculated as follows.

8.4.1.1 Holding cost

To normalise a holding cost KPI following data must be known:

- BOM of order and B_p of product
- Unit holding cost (per time/per unit) of products (for the Distribution Centre) and elements for the Manufacturer and Suppliers.
- Time given for production = Due date – Order date

In order to normalise uncertainties of input such as different quantity of ordered products, a normalisation factor is proposed for each echelon:

- For Distribution Centre: $\sum_{order}\underbrace{\sum_{product}\overbrace{HC_p \times Time}^{\text{Cost of keeping product for time of production}} \times Quantity_{product}}_{\text{All products kept per time required for production}}$

- For the Manufacturer and the Supplier (per element)

$$\sum_{order}\sum_{product}\sum_{element} HC_e \times Time \times Quantity_{(order,product)}$$
$$\times BOM(P,E)$$

Normalised holding cost value $= \frac{hc\ of\ Su}{normalisation\ factor\ for\ Su} + \frac{hc\ of\ mf}{normalisation\ factor\ for\ mf} + \frac{hc\ of\ dc}{normalisation\ factor\ for\ dc}$

8.4.1.2 Delay

To normalise a delay KPI, delay and processing time must be known:

- Delay = Delivery date – Due date
- Processing time = Due date – Order date

The delay is expressed as fraction of the processing time and averaged among the delayed orders.

Normalised value of the delay = $\frac{\sum_{\boldsymbol{delayed\ orders}}(\frac{\boldsymbol{delay}}{\boldsymbol{processing\ time}})}{\boldsymbol{number\ of\ orders\ which\ were\ delayed}}$

Table 8.2 List of changes used validation of proposed metric

Change	Name of scenario	Number of changed order	Description of a random change
Small	0_1_1_1_order1_prod1_bigger	Order 1	Product 1 increased from 4 to 20
	0_1_1_2_order1_prod2	Order 1	Product 2 increased from 25 to 40
	0_1_1_3_order2_prod1	Order 2	Product 1 increased from 16 to 20
	0_1_2_1_order1_prod2_smaller	Order 1	Product 2 decreased from 25 to 15
	0_1_2_2_order1_prod3	Order 1	Product 3 decreased from 15 to 10
	0_1_2_3_order2_prod1	Order 2	Product 1 decreased from 16 to 15
	0_1_3_1_order1_earlier	Order 1	Order 1 deadline is decreased from 5d16h to 4d16h
	0_1_3_2_order111	Order 111	Order 111 deadline is decreased from 91 to 90 days
	0_1_3_3_order112	Order 112	Order 112 deadline is decreased from 93 to 92 days
	0_1_4_1_order1_later	Order 1	Order 1 deadline is increased from 5d16h to 6g16h
	0_1_4_2_order111	Order 111	Order 111 deadline is increased from 91 to 93 days
	0_1_4_3_order112	Order 112	Order 112 deadline is increased from 93 to 94 days
	0_1_5_1_cancel_p1_p_2order_1	Order 1	Cancel the order of product 1 from order 1
	0_1_5_2_ cancel_p2_order_111	Order 111	Cancel the order of product 2 from order 111
	0_1_5_3_ cancel_p3_order_112	Order 112	Cancel the order of product 3 from order 112
Medium	0_2_1_1_order1_prod1_2_bigger	Order 1	Product 1 increased from 4 to 20, Product 2 increased from 25 to 125
	0_2_1_2_order2_prod1_2	Order 2	Product 1 increased from 16 to 60 Product 2 increased from 25 to 250
	0_2_1_3_order112_prod1_2	Order 1 Order 112	O1: Product 1 increased from 4 to 20 O112: Product 2 increased from 24 to 60
	0_2_2_1_order1_prod1_2_smaller	Order 1	Product 1 decreased from 4 to 1, Product 2 decreased from 25 to 5
	0_2_2_2_order2_prod1_2	Order 2	Product 1 decreased from 16 to 6 Product 2 decreased from 25 to 10
	0_2_2_3_order2_prod1	Order 1 Order 112	O1: Product 1 decreased from 4 to 1 O112: Product 2 decreased from 24 to 4
	0_2_3_1_order1_earlier	Order 1	Order 1 deadline is decreased from 5d16h to 3d16h
	0_2_3_2_order111	Order 111	Order 111 deadline is decreased from 91 to 89 days
	0_2_3_3_order112	Order 112	Order 112 deadline is decreased from 93 to 91 days
	0_2_4_1_order1_later	Order 1	Order 1 deadline is increased from 5d16h to 9d16h
	0_2_4_2_order111	Order 111	Order 111 deadline is increased from 91 to 95 days
	0_2_4_3_order112	Order 112	Order 112 deadline is increased from 93 to 100 days
	0_2_5_1_cancel_p1_order_1	Order 1	Cancel: Product 1 and 2 from order 1
	0_2_5_2_ cancel_p2_order_111	Order 1 Order 111	Cancel: Product 1 from order 1 Product 2 from order 111
	0_2_5_3_ cancel_p3_order_112	Order 2 Order 112	Cancel: Product 1 from order 2 Product 3 from order 112
Larg e	0_3_1_1_order1_prod123_bigger	Order 1	Product 1 increased from 4 to 14, Product 2 increased from 25 to 35 Product 3 increased from 15 to 25

	0_3_1_2_order2_prod123	Order 2	Product 1 increased from 16 to 160 Product 2 increased from 25 to 250 Product 3 increased from 12 to 120
	0_3_1_3_order1112_prod123	Order 1 Order 112	O1: Product 1 increased from 4 to 14, Product 2 increased from 25 to 35 Product 3 increased from 15 to 25 O112: Product 1 increased from 13 to 23, Product 2 increased from 24 to 34 Product 3 increased from 15 to 25
	0_3_2_1_order1_prod123_smaller	Order 1	Product 1 decreased from 4 to 2 Product 2 decreased from 25 to 12 Product 3 decreased from 15 to 7
	0_3_2_2_order2_prod123	Order 2	Product 1 decreased from 16 to 1 Product 2 decreased from 25 to 1 Product 3 decreased from 12 to 1
	0_3_2_3_order1112_prod123	Order 1 Order 112	O1: Product 1 decreased from 4 to 2, Product 2 decreased from 25 to 15 Product 3 decreased from 15 to 5 O112: Product 1 decreased from 13 to 3, Product 2 decreased from 24 to 14 Product 3 decreased from 5 to 1
	0_3_3_1_order1_earlier	Order 1	Order 1 deadline is decreased from 5d16h to 1d16h
	0_3_3_2_order111	Order 111	Order 111 deadline is decreased from 91 to 87 days
	0_3_3_3_order112	Order 112	Order 112 deadline is decreased from 93 to 90 days
	0_3_4_1_order1_later	Order 1	Order 1 deadline is increased from 5d16h to 20d16h
	0_3_4_2_order111	Order 111	Order 111 deadline is increased from 91 to 120 days
	0_3_4_3_order112	Order 112	Order 112 deadline is increased from 93 to 200 days
	0_3_5_1_cancel_p1_order_1	Order 1	Cancel the order 1
	0_3_5_2_ cancel_p2_order_111	Order 1, 111	Cancel order 1 and order 111
	0_3_5_3_ cancel_p3_order_112	Order 111, 112	Cancel Order 111 and 112

8.4.2 Robustness metric of Crisp DRs, FDRs, NSGAII and MCNSGAII

The aim of a robustness experiment is to assess the normalisation metric. For this purpose, three sets of scenarios were manually created. Each scenario is derived from the benchmark scenario. Moreover, in each set, scenarios include small, medium, or large changes. Additionally, 10 scenarios generated via Monte Carlo was included as a fourth set. In the experiment, four crisp DR, two FDRs, six rule bases proposed by NSGAII and six rule bases proposed by MCNSGAII were simulated on each of those 10 scenarios. In each set standard deviation and average values are calculated for both KPIs for each model and can be found in Table 8.3 and Table 8.4.

As expected, the standard deviation of both KPIs; the delay and the holding cost increases with larger changes. Crisp DRs exhibit no adaptability, therefore values of standard deviation for those are the highest. FDRs are characterised by lower values of standard deviation while FDRs proposed by NSGAII are even lower. This is observation is expected as FDRs can adapt to changing demand conditions. The average values of normalised KPIs tend to increase proportionally to how much deviation from the benchmark scenario given set has. Relative differences in average normalised KPIs between crisp DRs, FDRs and FDRs proposed by NSGAII seem to resemble those of raw (not normalised) KPIs.

Table 8.3 Standard deviation and average values of the holding cost

Control-scheme	Small changes		Medium changes		Large changes		MC generated	
	SD	Average	SD	Average	SD	Average	SD	Average
FIFS	1.05	13.52	1.23	13.85	1.27	13.39	8.57	19.74
EDD	1.38	12.83	1.51	13.29	1.20	12.61	9.31	20.08
MTWR	0.87	15.80	1.15	15.92	1.68	15.80	11.10	21.06
LTWR	0.70	13.87	1.49	14.63	1.84	14.40	8.73	22.16
FDR delay-focused	1.14	11.22	1.28	10.77	1.93	10.37	4.74	14.53
FDR holding cost-focused	0.27	8.50	0.58	8.70	1.51	8.85	4.28	14.09
GA1_rank1	0.40	4.70	0.65	4.88	0.60	4.94	1.63	6.86
GA2_rank1	0.94	5.05	1.01	5.21	1.26	5.43	2.62	6.90

Table 8.4 Standard deviation and average values of the delay

Control-scheme	Small changes		Medium changes		Large changes		MC generated	
	SD	Average	SD	Average	SD	Average	SD	Average
FIFS	1.93	152.12	5.07	153.69	19.34	155.18	47.62	191.58
EDD	2.75	150.63	4.83	151.94	17.33	152.46	44.94	196.14
MTWR	1.15	154.08	4.43	155.09	17.11	157.12	42.48	197.56
LTWR	3.08	152.11	7.09	154.73	21.16	157.33	47.55	201.94
FDR delay-focused	1.05	23.84	1.70	23.92	4.96	24.46	11.79	33.35
FDR holding cost-focused	0.53	28.26	1.34	28.44	4.45	29.22	12.22	40.19
GA1_rank1	0.47	7.39	0.88	7.70	2.27	8.04	6.07	14.60
GA2_rank1	0.82	7.50	1.02	7.83	2.68	8.58	4.93	13.44

8.5 Discussion and Conclusions

Presented solutions of Fuzzy NSGAII considerably improved performance of the FIS by decreasing inventory holding cost and delays. Previously considered models and ideas presented in Chapter 1 are likely to produce better results for deterministic SC planning and scheduling problems. However, dynamic problems considering uncertainty of the SC's parameters as in this problem allows to acknowledge and accommodate unavoidable SC's disturbances.

The goal of this chapter was to optimise a proposed FDRs and improve efficiency of FDRs introduced in the previous chapter. NSGAII algorithm has been developed to deal with multi-objectivity of this problem. Chromosomes representing decisions variables selected for simultaneous, dynamic control for two crucial SCM problems consisting of scheduling processes of production and delivery, and inventory control of different multiple elements and products of various echelons. To prevent overfitting of NSGAII-generated FDRs into one benchmark scenario, an additional metric which increase the robustness is proposed. Small changes in uncertain demand led to big differences in the observed KPIs, thus normalised values of the holding cost and delay were determined in this chapter and used as an additional fitness function to new MCNSGAII. Use of Monte Carlo Simulation allowed determining FDRs which can perform better for scenarios with similar intensity. Full comparison of all proposed control schemes can be found in Chapter 9.

9 COMPARISON OF CONTROL SCHEMES

9.1 Introduction

FDRs considering uncertainty of demand were proposed for simultaneous scheduling and inventory control of multi echelon SC. The multi-objective optimisation of holding cost and delay fitness function was developed for a benchmark scenario and FDRs were generated by NSGAII for a new control-scheme. Monte Carlo simulation was used for improving robustness of decision variables determined by NSGAII, so that it can perform well for similar scenarios with uncertain demand.

One of the challenges of this optimisation is to answer the question, how to incorporate an *extraction of knowledge* approach into the considered models. The aim of MCNSGAII is to process rule bases generated by NSGAII in multiple runs. The average values and standard deviation of the delay and the holding cost KPIs are measured for multiple scenarios.

The FDRs which consistently perform well for scenarios with similar statistical values, are promoted by the MCNSGAII into next iteration. This led to creating new FDRs, which uses runs of Monte Carlo simulation to extract the knowledge from the uncertain echelon inputs. Therefore, a control-scheme, which offer good solutions customised by fuzzy representation of inputs should perform consistently better in face of uncertain demand challenged by changing intensity of orders.

The Subchapter 9.1.1 includes comparison between control-schemes proposed by Crisp DRs, FDRs and rule-bases proposed by NSGAII before and after Monte-Carlo simulation. The effect on changing intensity of incoming orders is analysed in Subchapter 9.2.2. Conclusions to the results, implications of proposed control scheme and future research possibilities can be found in the last Chapter 10.

9.1 Comparison results

9.1.1 Comparison of solutions for the benchmark scenario

The benchmark scenario serves as a starting point for comparing all control-schemes models developed for the scheduling and inventory control problems. Models can be grouped into three main categories, namely crisp DRs, FDRs and FDRs proposed by two metaheuristics: NSGAII and MCNSGAII.

Use of crisp DRs with pre-set replenishment levels for control is characterised by keeping low holding cost of raw materials amongst echelons, but it also leads to long delays of orders delivered to the customer. Crisp DRs do not apply any form of adaptation to uncertain demand as they rely on simple priority sorting. The main advantage is easiness of implementation in real-world problems. To address the problem of uncertain demand a FDRs were proposed as the second control-scheme. Two types of fuzzy rule bases were created for scheduling and inventory subproblems. The delay-focused and inventory-focused FDRs introduced in Chapter 7, allowed consideration of the uncertain input parameters represented by fuzzy sets to a decision-making. Fuzzification of information such as due date of orders, number of incoming orders or workload led to performance improvement. The considerable decrease of the delay KPI was observed, which demonstrated that FDRs have ability to the adaptation to the uncertain demand.

FDRs determined for the problem, were not optimal thus NSGAII allowing multi-objective optimisation was selected to determine optimal set of FDRs. As expected, NSGAII performed better than manually determined FDRs by extracting knowledge from data and proposing a control-scheme leading to further decrease of the delay KPI by 66% in comparison to delay-focused FDRs, while keeping very similar holding cost level. As explained in Chapter 8, use of NSGAII for the specific benchmark scenario may lead to overfitting its decisions into the specific case. MCSGAII was proposed to deal with this issue. The general comparison of proposed control-schemes for the benchmark scenario will be analysed throughout subchapters 9.1.1.1, 9.1.1.2 and 9.1.1.3. Performance and average values of KPIs for changing intensity generated by the Monte Carlo simulation and therefore analysis of the robustness of considered control-schemes is presented in subchapter 9.1.2 .

9.1.1.1 Different due date of orders

Comparison of proposed models under different due dates is presented for the holding cost KPI in Table 9.1 and Figure 9.1 and for the delay it is presented in Table 9.2 and Figure 9.2. Six rule bases are randomly selected from the pareto front for NSGAII and MCNSGAII. The best and the worst solution are highlighted in green and red colour, respectively for each scenario.

Table 9.1 Holding cost for changing due dates of orders

		Holding cost for different due dates of orders (£)				
		shorter by 50%	shorter by 25%	benchmark	longer by 25%	longer by 50%
Crisp DRs	*FIFS*	150327	150031	152188	153411	156738
	EDD	150151	151142	151480	153366	156841
	MTWR	189002	191439	181415	186206	198934
	LTWR	184785	188629	188990	200519	197363
FDRs	*Delay-focused*	199844	200353	229346	237885	254723
	Holding cost-focused	163968	165950	171263	180204	186841
NSGAII	*NSGAII_1*	151486	152310	153576	165472	171853
	NSGAII_2	151486	152310	153576	165472	171853
	NSGAII_3	150792	153016	149722	181662	153748
	NSGAII_4	150756	152905	148989	165687	167883
	NSGAII_5	150756	152905	148989	170353	167601
	NSGAII_6	154130	149100	150122	159414	166406
Robust NSGAII	*MC_NSGAII_1*	197526	180757	185401	193630	186286
	MC_NSGAII_2	187539	199991	200869	192049	196764
	MC_NSGAII_3	162596	168116	176118	200436	175388
	MC_NSGAII_4	179579	181139	177738	178066	190530
	MC_NSGAII_5	187840	163979	166035	180132	164029
	MC_NSGAII_6	161731	163979	163780	155616	165733

Table 9.2 Delay for changing due dates of orders

		Delays for different due dates of orders (hours)				
		shorter by 50%	shorter by 25%	benchmark	longer by 25%	longer by 50%
Crisp DRs	*FIFS*	7798	7645	7418	7044	6790
	EDD	7707	7609	7473	7338	7185
	MTWR	9259	9178	8509	8141	7947
	LTWR	7070	7039	7042	7177	7040
FDRs	*Delay-focused*	1789	1699	1929	1948	1864
	Holding cost-focused	5393	5302	5252	5053	4960
NSGAII	*NSGAII_1*	827	755	663	684	663
	NSGAII_2	827	755	663	684	663
	NSGAII_3	985	909	748	1049	535
	NSGAII_4	980	904	756	816	636
	NSGAII_5	980	904	756	819	636
	NSGAII_6	901	780	673	596	675
Robust NSGAII	*MC_NSGAII_1*	967	631	570	582	365
	MC_NSGAII_2	1761	1740	1575	1537	1388
	MC_NSGAII_3	873	863	812	1044	865
	MC_NSGAII_4	1211	1126	994	893	781
	MC_NSGAII_5	1801	1377	1389	1137	1044
	MC_NSGAII_6	1517	1458	1170	1017	987

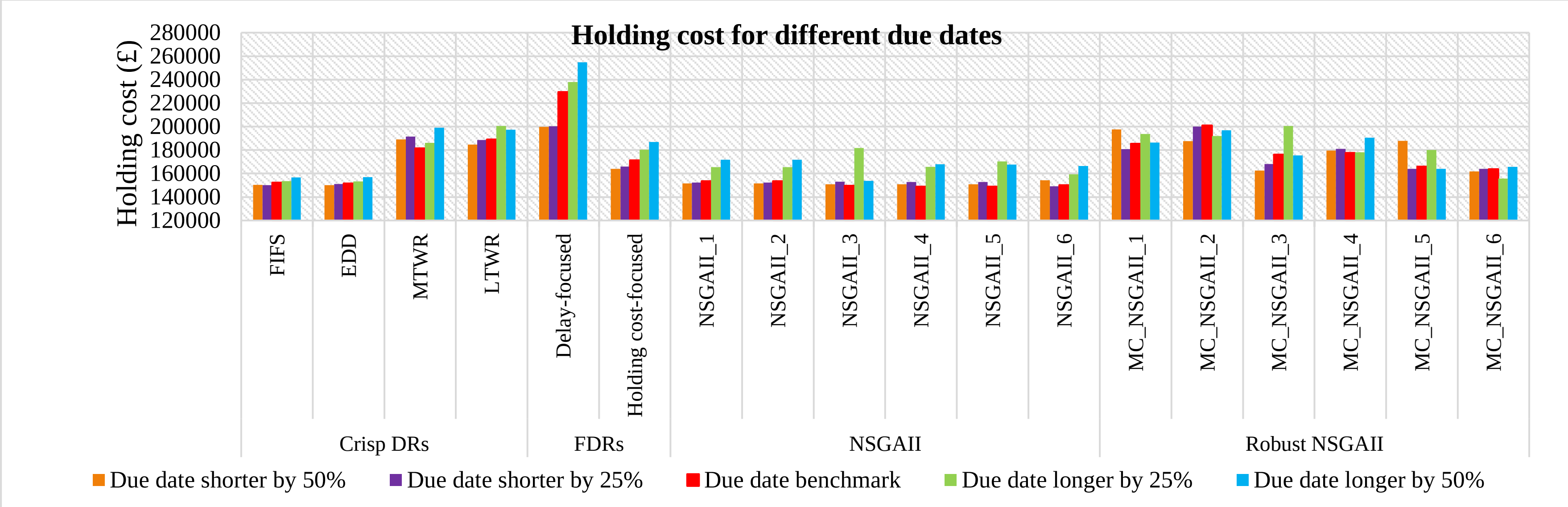

Figure 9.1 Holding cost for changing due dates of orders

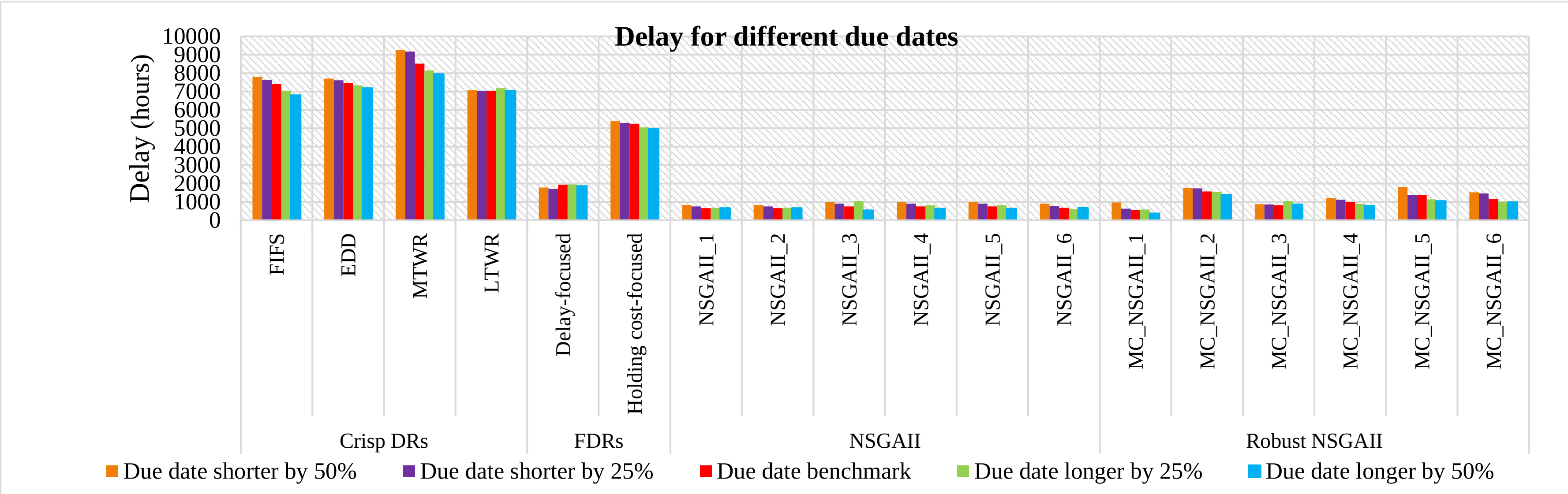

Figure 9.2 Delay for changing due dates of orders

One can observe that among the crisp DRs, values of KPIs are highest when MTWR DR is used. LTWR rule leads to the lowest holding cost and EDD has the lowest delay amongst crisp DRs. FDRs both outperformed crisp DRs by considerable decrease of the delay KPI. There is a clear trade-off between the two observed SC KPIs for this solution. As expected, use of the delay-focused FDRs led to decrease of the total delay in benchmark scenario, decreasing by the 73% while increasing the holding cost only by 34% compared to EDD rule.

The best solutions are achieved by using FDRs proposed by the GA. Each solution from the rank 1 outperforms all other crisp DRs and FDRs. This experiment showed that FDRs which consider uncertain parameters may lead to improvement in both KPIs. There is also a wide range of trade-off in importance of KPIs which can be a helpful insight for inventory managers. It is important to note that FDRs proposed by NSGAII achieved lower KPIs than those proposed by MCNSGAII. This is expected, as changing due dates does not affect other properties of the scenario, such as order sizes, time at which orders arrive etc. making those scenarios very close to the benchmark scenario. Since NSGAII optimisation is guided only by the performance on the benchmark scenario itself, it is expected that FDRs proposed by it will generally achieve optimal performance on that scenario and its derivatives that were considered in this experiment.

9.1.1.2 Different order sizes

In this experiment the order sizes were decreased or increased from the base level in the benchmark scenario, while orders remained otherwise unchanged. That means that in each scenario the SC had effectively more or less time to deliver the orders. This is quite evident in its effect on the delay KPI among all proposed control-schemes. This KPI kept increasing with increasing order sizes in all cases. As expected for inflexible crisp DRs only the smallest order sizes could be delivered with low delay. With each increase in the order size the delay increased in relatively large steps. On the other hand, both delay-focused FDR and almost all FDRs proposed by NSGAII achieved low delays for both scenarios with order sizes lower than in benchmark scenario. Even with increase of order size beyond that in the benchmark scenario those FDRs achieved similar levels of delay. It is possible that in the scenario with order sizes increased by 50% achieving order deliveries close to the due dates that were adequate for the benchmark scenario, was simply impossible and thus performance in terms of delay KPIs became less differentiated.

Observation of the holding-cost KPI paints a bit more nuanced picture. All FDRs except the holding cost-focused one exhibited excessive holding cost on scenarios with lower order sizes, especially when compared to the crisp DRs.. On the remaining scenarios the holding cost KPI remained on similar level as the best crisp DRs for holding cost-focused FDR and FDRs proposed by NSGAII.

The remaining FDRs achieved slightly higher levels of holding cost, but generally no higher than the worst performing crisp DR. It important to note that a holding cost is expected to raise with increased order sizes, especially when due dates remain unchanged, as larger quantities of finished products and raw materials need to be provided in the same timeframe. This implies higher stock levels since production capabilities remain unchanged.

Table 9.3 Holding cost for changing the size of orders

		Holding cost for different order sizes (£)				
		shorter by 50%	shorter by 25%	benchmark	longer by 25%	longer by 50%
Dispatching rules	*FIFS*	125685	124005	152188	191390	238688
	EDD	125760	124051	151480	197486	231858
	MTWR	122954	139338	181415	225129	297334
	LTWR	124751	139172	188990	268219	340376
FDRs	*Delay-focused*	202025	158899	229346	264274	265541
	Holding cost-focused	132196	159760	171263	205226	251005
NSGAII	*NSGAII_1*	196815	156584	153576	203246	240652
	NSGAII_2	196815	156584	153576	203719	252023
	NSGAII_3	189824	154083	149722	209993	255393
	NSGAII_4	189824	151246	148989	209240	351341
	NSGAII_5	189824	155697	148989	208597	260184
	NSGAII_6	173868	144686	150122	206816	262486
Robust NSGAII	*MC_NSGAII_1*	244950	188201	185401	177535	271835
	MC_NSGAII_2	214389	171722	200869	249232	355645
	MC_NSGAII_3	236991	183054	176118	276866	311362
	MC_NSGAII_4	228743	178759	177738	228337	315003
	MC_NSGAII_5	210065	161030	166035	216988	270774
	MC_NSGAII_6	208015	153993	163780	194229	241653

Table 9.4 Delays for changing the size of orders

		Delays for different order sizes (hours)				
		shorter by 50%	shorter by 25%	benchmark	longer by 25%	longer by 50%
Dispatching rules	*FIFS*	129	2303	7418	13288	19564
	EDD	129	2255	7473	13443	19863
	MTWR	130	3128	8509	14255	21479
	LTWR	130	2333	7042	12328	18880
FDRs	*Delay-focused*	129	138	1929	3398	5438
	Holding cost-focused	185	2407	5252	8390	12167
NSGAII	*NSGAII_1*	130	131	663	2805	4800
	NSGAII_2	130	131	663	2805	5063
	NSGAII_3	130	158	748	2929	6162
	NSGAII_4	130	150	756	2982	6168
	NSGAII_5	130	131	756	3024	5719
	NSGAII_6	130	519	673	3023	5262
Robust NSGAII	*MC_NSGAII_1*	130	131	570	1857	4375
	MC_NSGAII_2	131	132	1575	3486	6532
	MC_NSGAII_3	133	134	812	3049	5400
	MC_NSGAII_4	130	131	994	2963	6087
	MC_NSGAII_5	133	134	1389	2808	5165
	MC_NSGAII_6	130	132	1170	3448	5875

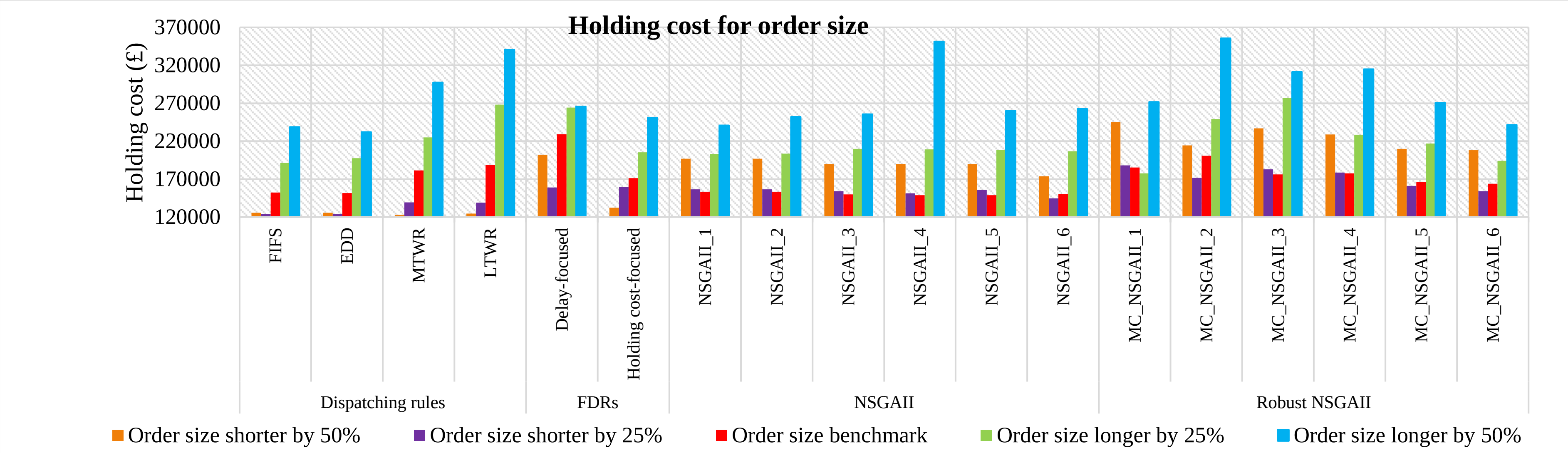

Figure 9.3 Holding cost for the changing size of orders

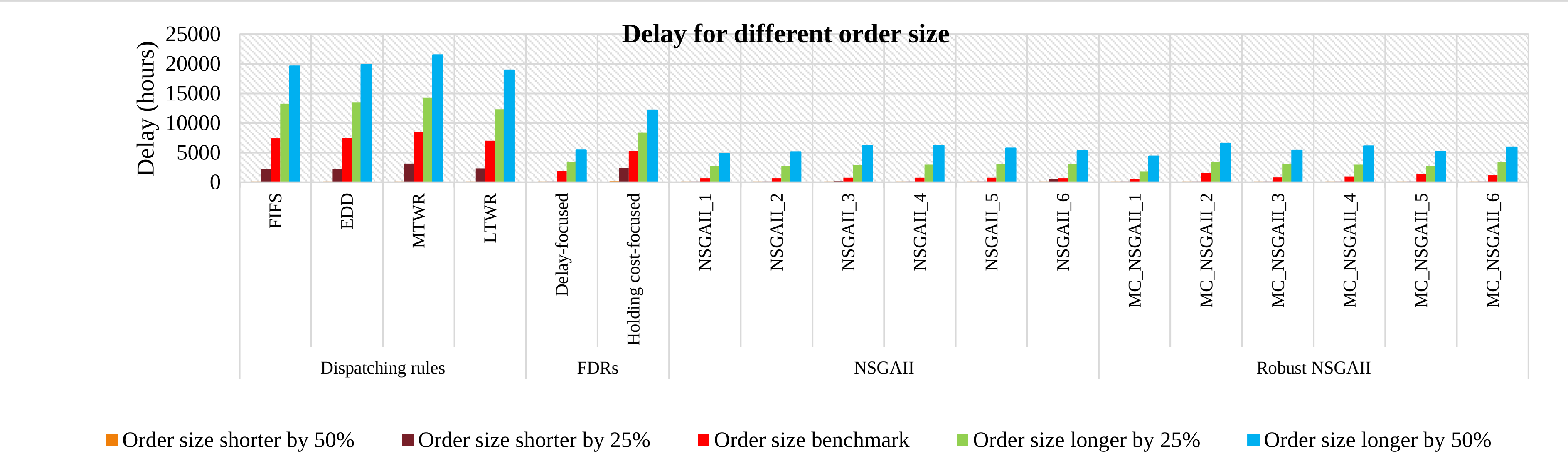

Figure 9.4 Delay for the changing size of orders

9.1.1.3 Different product processing time

In this experiment the processing time of products was proportionally changed from the base level in the benchmark scenario. Crisp DRs generally achieved similar performance in terms of KPIs, irrespective of changing processing time. This is in line with expectations, as the change between scenarios does not influence the actual decision making for those DRs.

All FDRs improved delay KPI in all cases, with lowest gains in case of the holding cost-focused FDR. The performance of all FDRs proposed by GA was less even than that of the crisp DRs and fixed FDRs in terms of both KPIs. The starkest example is higher delays for most NSGAII FDRs for scenarios with smaller processing time. This may have been caused by overfitting to the benchmark scenario. The FDRs proposed by MCNSGAII achieved more uniform delay levels, suggesting that the proposed robustness-increasing measures had the desired impact. It is important to stress that the performance was not as even as in case of crisp DRs, reinforcing the need for emphasis on the robustness of the proposed control-schemes.

The holding cost KPI increased significantly with increased processing time for most GA proposed FDRs. This is expected, as lower rate of production means that the distribution centre may need to keep more stock to be able to deliver orders on time. For echelons lower in the chain the lower rate of production means that the inventory is being depleted at a lower rate. Since inventory is replenished in batches, it means that more items will be held for longer, driving the holding cost up.

Table 9.5 Holding cost for different product processing time

		Holding cost for different product processing time (£)				
		shorter by 50%	shorter by 25%	benchmark	longer by 25%	longer by 50%
Dispatching rules	*FIFS*	145697	148926	152188	154853	158511
	EDD	145646	148369	151480	154091	156916
	MTWR	171090	175883	181415	180687	184018
	LTWR	214183	217133	188990	194630	199070
FDRs	*Delay-focused*	204884	177248	229346	202864	209781
	Holding cost-focused	200619	161776	171263	185149	185713
NSGAII	*NSGAII_1*	145390	148339	153576	213346	192770
	NSGAII_2	145390	148339	153576	186354	199797
	NSGAII_3	217902	165594	149722	184343	216634
	NSGAII_4	217902	168759	148989	177786	214594
	NSGAII_5	217902	168759	148989	174916	215662
	NSGAII_6	199022	182551	150122	204796	219136
Robust NSGAII	*MC_NSGAII_1*	162265	161518	185401	205249	222402
	MC_NSGAII_2	162948	230757	200869	192230	223049
	MC_NSGAII_3	162949	161503	176118	201034	217651
	MC_NSGAII_4	162290	161983	177738	202772	207707
	MC_NSGAII_5	163942	193579	166035	192034	197847
	MC_NSGAII_6	196411	153135	163780	166305	217650

Table 9.6 Delay for different product processing time

		Delays for different product processing time (hours)				
		shorter by 50%	shorter by 25%	benchmark	longer by 25%	longer by 50%
Dispatching rules	*FIFS*	7358	7366	7418	7392	7391
	EDD	7453	7449	7473	7473	7421
	MTWR	8362	8467	8509	8441	8427
	LTWR	7135	7145	7042	7003	6995
FDRs	*Delay-focused*	1618	1342	1929	1698	1759
	Holding cost-focused	5706	4979	5252	4800	4775
NSGAII	*NSGAII_1*	688	704	663	1624	2153
	NSGAII_2	688	704	663	1316	2257
	NSGAII_3	3324	2031	748	796	1900
	NSGAII_4	3324	2031	756	840	2079
	NSGAII_5	3324	2031	756	817	2087
	NSGAII_6	2940	2580	673	1757	2631
Robust NSGAII	*MC_NSGAII_1*	1025	933	570	1268	2165
	MC_NSGAII_2	1199	1796	1575	1591	2243
	MC_NSGAII_3	1054	1044	812	1487	2097
	MC_NSGAII_4	1025	935	994	1780	2342
	MC_NSGAII_5	1308	1520	1389	1624	2061
	MC_NSGAII_6	2451	1741	1170	1170	1588

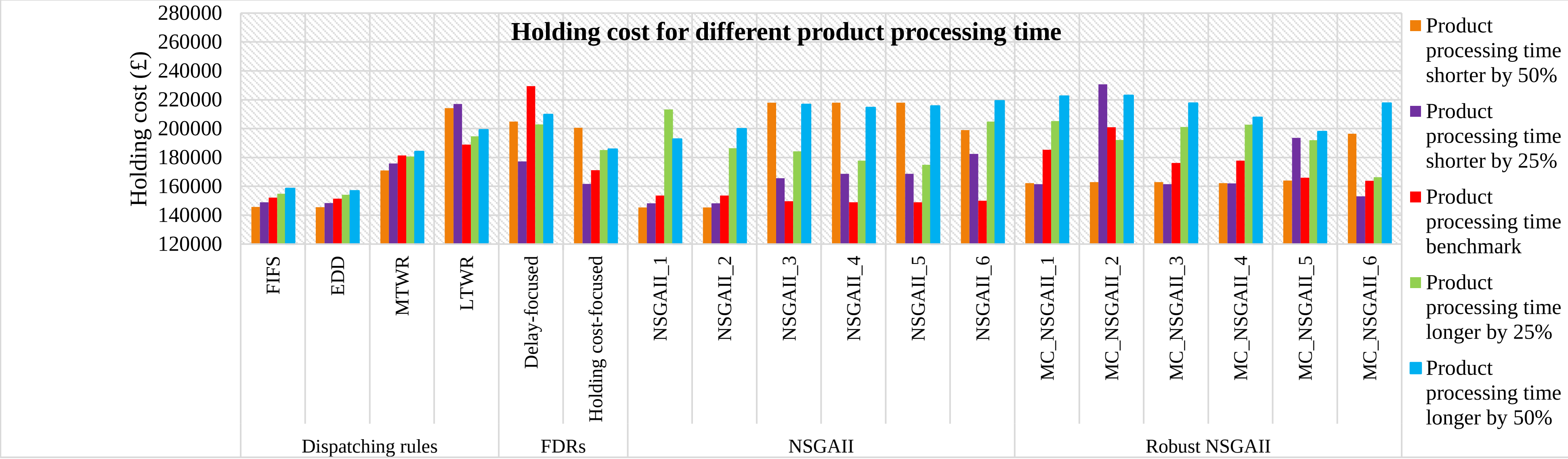

Figure 9.5 Holding cost for different product processing time

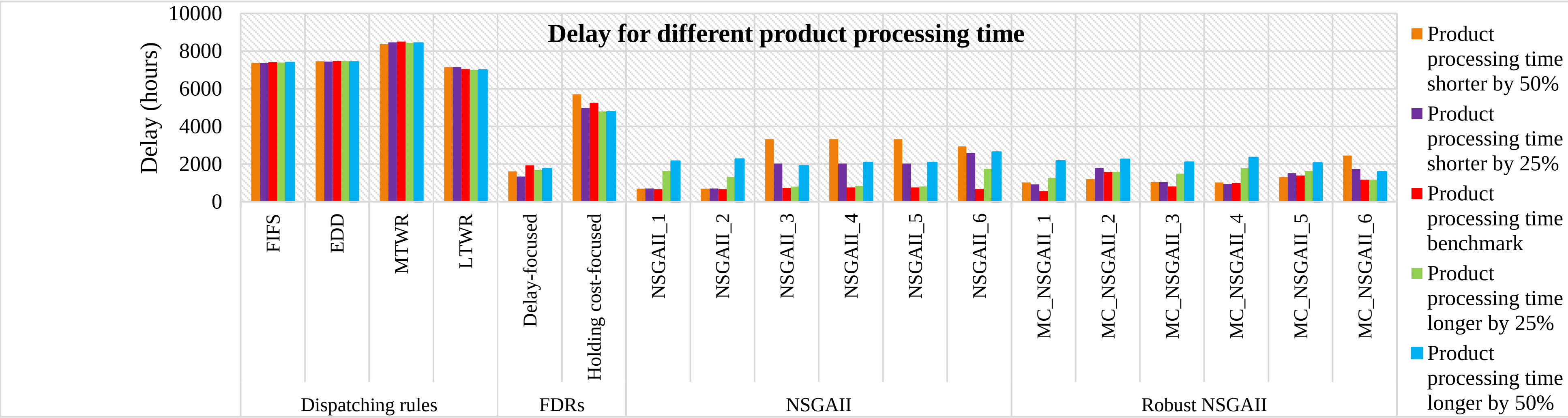

Figure 9.6 Delay for different product processing time

9.1.2 Changing intensity of orders

Previous experiments focused on performance in single instances of each scenario. To understand the effect of uncertainty in the demand better, this experiment involved Monte-Carlo simulations. In this setup orders were generated randomly at desired intensity for each trial. The values of both KPIs from 20 trials were then accumulated and mean values and their respective standard deviation was presented in Table 9.7. The scenario in each trial involved variable intensity for the first time. In these scenarios orders intensities were changed during the simulation. Namely the scenario was composed of four parts. In each part orders were generated with different intensity. The scenario starts with medium intensity ($\lambda = 10$), which is equivalent to the benchmark scenario. In the next part orders were generated at low intensity (λ=5). Third part was composed of orders generated again at the medium intensity. The last part involved rising to high intensity (λ=15). Such a scenario represents change of intensity in a sequence ‘M-H-M-L’ and aim to serve as example of a seasonal variation in demand. The medium-intensity is the base (expected) level of demand, while low-intensity and high-intensity parts represent deviation from that level. It is important to note that only the medium intensity was used by MCNSGAII and the scenario in this experiment is about four times as long as the benchmark scenario.

In this experiment crisp DRs achieved the highest average levels of both holding cost and delay KPIs. The FDRs proposed by GA achieved lowest average holding cost values, even lower than holding cost-focused FDR. The average delay KPI for those FDRs was similar to that of delay-focused FDR or slightly higher, but significantly lower than that of holding cost-focused FDR. In terms of robustness, represented by the standard deviations of KPIs, the FDRs proposed by MCNSGAII achieved the best results in both KPIs in each rule base.

On the other hand, the majority of the FDRs proposed by NSGAII achieved higher values of standard deviation, especially for the holding cost KPI. This could be caused by overfitting to the benchmark scenario which had medium intensity of incoming orders. The similar robustness was achieved by the manually defined FDRs, since robustness was not directly considered when designing rule sets. Lastly crisp DRs were characterised by even larger standard deviation, corresponding to even worse robustness of the proposed solutions.

Table 9.7 Holding cost and delay comparison for 20 scenarios generated by Monte Carlo simulation

		Holding cost		Delay	
		Average	Std.Dev	Average	Std.Dev
DRs	*FIFS*	1145019	279659	337322	35479
	EDD	1157204	275948	336876	35522
	MTWR	1271415	321753	344324	36959
	LTWR	1807599	251553	331373	31273
FDRs	*Delay-focused*	1130660	140582	65753	11123
	Holding cost-focused	1011147	125336	169114	18082
NSGAII	*NSGAII_1*	958917	164561	62649	9659
	NSGAII_2	949089	180370	62541	9684
	NSGAII_3	877315	68271	87364	16015
	NSGAII_4	957343	54303	89682	12265
	NSGAII_5	944946	172165	76865	12590
	NSGAII_6	988346	163919	75226	10873
Robust NSGAII	*MC_NSGAII_1*	964164	48720	70340	8340
	MC_NSGAII_2	946788	61551	63748	10149
	MC_NSGAII_3	965808	56618	64948	9840
	MC_NSGAII_4	805391	46441	60927	8365
	MC_NSGAII_5	823958	54179	64631	7955
	MC_NSGAII_6	955676	52972	59300	10449

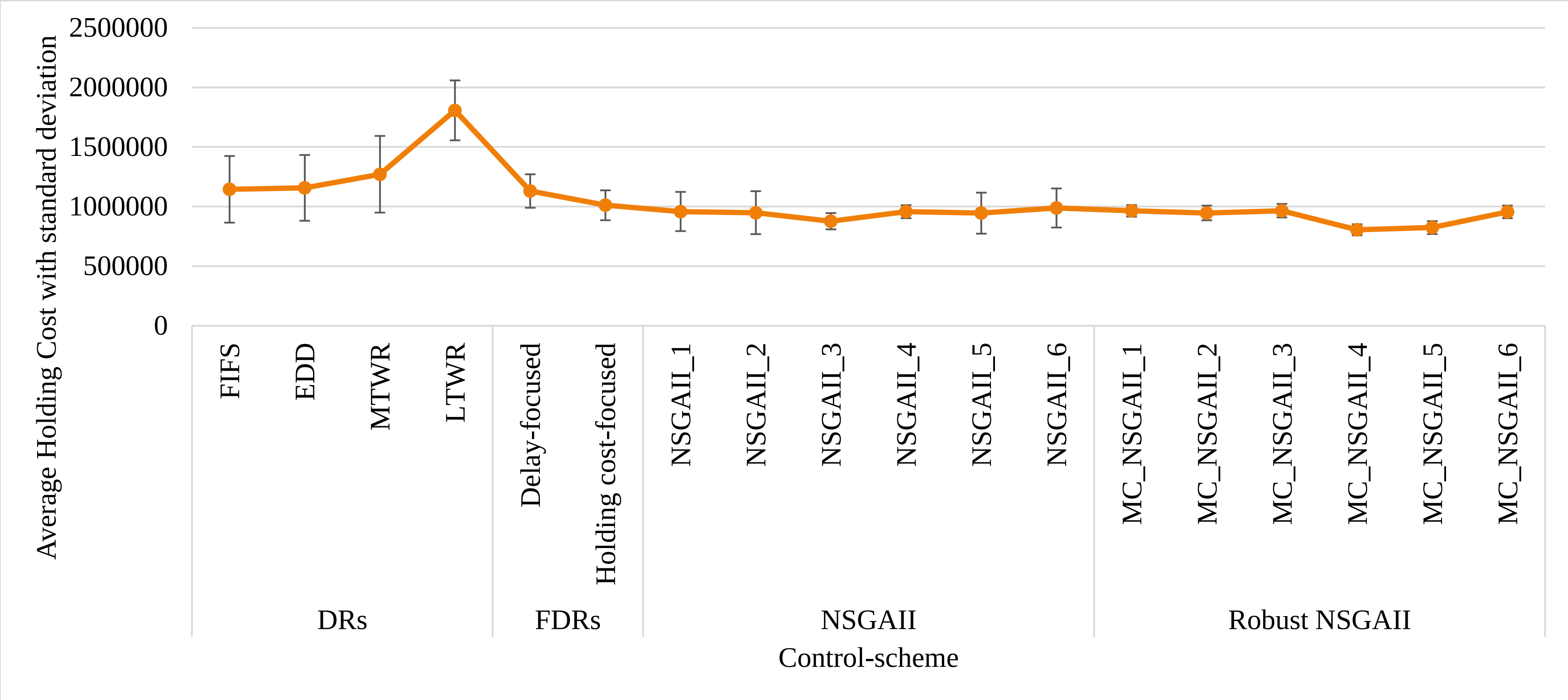

Figure 9.7 Performance of holding cost for different control-schemes with standard deviation for 20 scenarios

Additional experiments considering different sequences of intensity such as: ‘M-H-L-H’, ‘M-L-H-M’ and ‘L-H-M-L’ of orders were conducted. The lowest KPIs average and standard deviation of fitness functions is always achieved by the rule base proposed by the MCNSGAII.

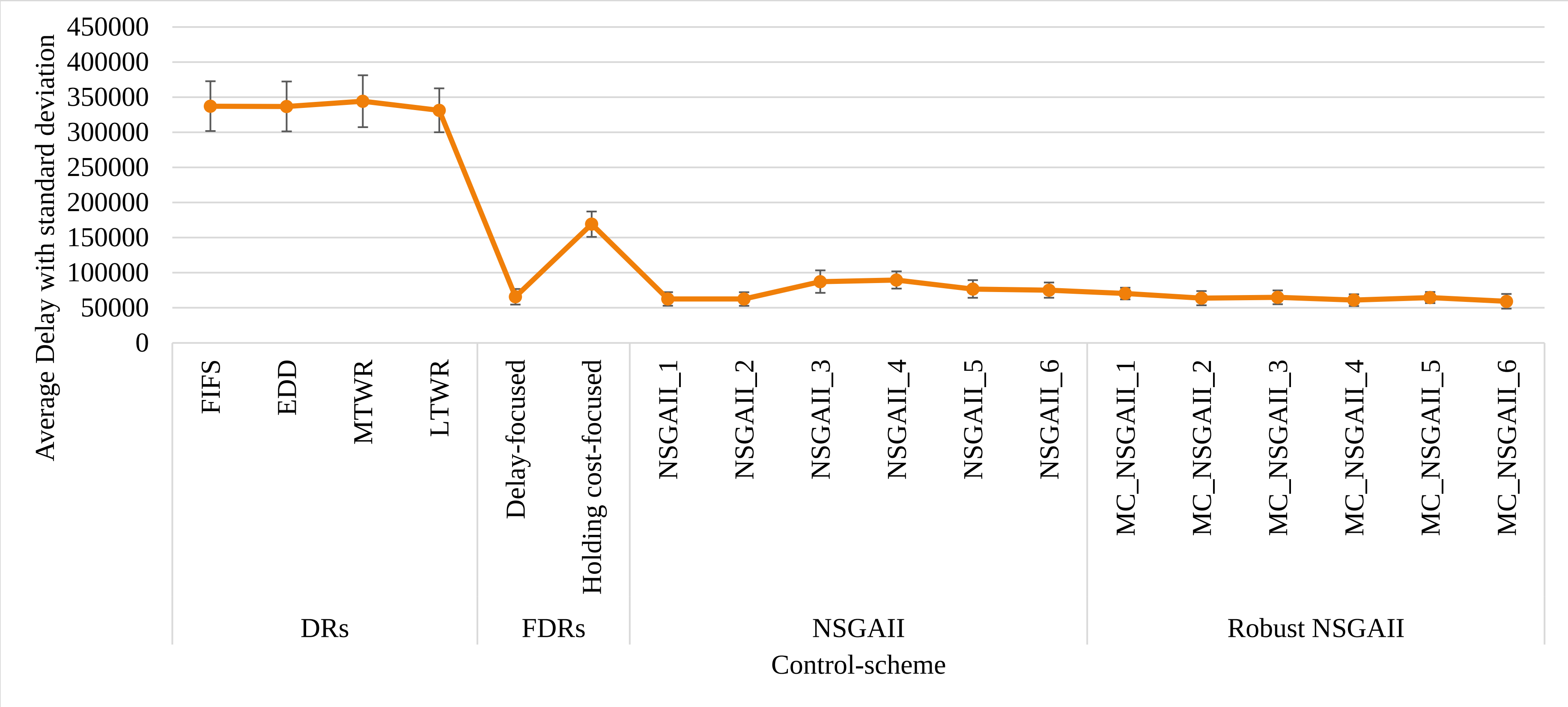

Figure 9.8 Performance of delay for different control-schemes with standard deviation for 20 scenarios

9.1.3 Conclusions

This chapter carried out a series of experiments that enabled comparison of various proposed control-schemes for considered SC problems. Initial experiment explored performance of proposed solutions on a benchmark scenario. Subsequent experiments allowed observation of control-schemes in the presence of simple, well controlled changes to the benchmark scenario. The observations focused mostly on the performance in terms of KPI values.

Finally, effects of uncertainty on performance and robustness were investigated using Monte-Carlo simulations involving longer a scenario with time-varying order intensity. The experiments involved crisp DRs, meant as a rather primitive and simplistic control scheme to serve as a reference or starting point and FDRs, that were developed to provide vastly improved performance. Among considered FDRs were those with manually designed rule bases and those obtained via means of optimisation. Finally, an extended optimisation approach based on NSGAII, MCNSGAII, with additional emphasis on robust performance was compared with FDRs generated by standard NSGAII.

The experiment confirmed the superiority of the MCNSGAII over standard NSGAII for the considered SC problem. Inclusion of the demand uncertainty in a form of Monte-Carlo simulation to help guide the optimisation, coupled with the robustness, represented by standard deviation, being included in the fitting function resulted in FDRs with improved performance for orders of various uncertainties.

The proposed scheme can extract knowledge from fairly limited data and as suggested by the standard deviation metric, it remains effective even in the presence of variable intensity of orders, despite the fact that the optimisation involved only the single, base intensity.

10 Conclusions and Future work

10.1 Conclusions

The work presented in this thesis considers a general-structure, multi-product, dynamic SC. For such a structure, integrated inventory control and scheduling problems in a presence of demand uncertainty are studied. The literature review exposed a few main limitations of the existing research on the topic. The state of the art on inventory and scheduling issues usually focuses on the internal activities of the single echelons. Very often, even when a multi-echelon structures are considered, additional echelons are not fully modelled on the operational level. They rather serve as an interface representing parameters which are used for decision-making in the main echelon.

SCM focuses on the SC as one system, recognizing that solutions for many problems must be proposed to secure flow of information and goods to satisfy specific objectives. Goal of SCM is optimisation of multiple echelons, but it is often assumed, that SC's participants are willing to implement complex procedures for information sharing. Such an assumption, together with unexamined uncertainties often leads to unrealistic or unpractical models. Finally, majority of research on the subject is restricted to mathematical modelling, which can guarantee the optimal solution. However, vast simplifications are often required to gain analytical feasibility. Moreover, the majority of mathematical models are missing on dynamic nature of SCs which is crucial for real-world applications. The considered SC is a generic structure which allowed modelling of various types of echelons. Each echelon has its own characteristics and processes. Therefore, the problems of scheduling and inventory control differs between echelon. Suppliers echelons do not hold inventory as they are the highest tier echelons. Suppliers schedule orders placed by the Manufacturer. The Manufacturer also schedule the production, where elements delivered by the Supplier are processed into the final products. This echelon holds an inventory of elements. The Distribution Centre echelon schedules deliveries by allocating the lorries. It also keeps an inventory of products. Simple DRs with crisp CRP are developed to deliver control across the SC's echelons. FIFS, EDD, LTWR and MTWR rules are used to prioritise orders and crisp CRP is used to control inventory levels.

This work builds on a simulation model that is inherently dynamic and enables observation of long-term effects of decision making. For this purpose, a custom simulation environment was developed in Python programming language. The simulation model was subsequently used for control-schemes development and simulation-based optimisation.

To better capture uncertain nature of a SC, the fuzzy logic was incorporated into the control scheme in the form of FDRs. The fuzzy logic offers a mathematical representation of human reasoning, which can be used to transfer expert knowledge into computer systems. Even more significantly, using fuzzy logic to represent incomplete or unknown data is well tried approach. Flexible nature of FIS and its ability to deal with nonlinearities are also well known.

Utilisation of more complex control-schemes may often result in substantial amounts of tuning parameters being introduced. As this makes manual parameter tuning a daunting task, the metaheuristics were used to aid optimisation of the proposed control-schemes. Well established NSGAII algorithm was chosen for its ability to directly work with multiple objectives.

The problem considered in this thesis was formalised and KPIs for assessing the performance across whole SC were determined including the total delay of orders in the final echelon and total holding cost at all echelons. The simulation environment was designed to support the development and testing of various control-schemes, with emphasis on flexibility and extendibility. The simulator was partitioned into two parts: echelon processes and decision making in accordance with the separation of concerns principle. The concept of scenario was introduced, as an input to the simulator, that encoded the SC structure and parameters, both related to echelon processes and decision making. A set of supporting tools with GUI were also developed to enable preparation of scenarios and aid subsequent analysis of the simulation results. Finally, a benchmark scenario was developed as basis for the experiments in the remainder of this thesis.

Next, an initial control scheme in the form of crisp DRs with pre-set replenishment levels was designed and evaluated. Decision making for dynamically changing demand was achieved by decomposing problem into scheduling and inventory control subproblems for each echelon. Implementation of these simple heuristics allowed observation of KPIs subject to different input values which are often unknown and allowed testing of how different changes in unknown demand can affect the behaviour of echelons under no information sharing policy. That initial decision making became a foundation for developing FDRs. The experiments conducted on crisp DRs allowed to observe the SC behaviour as a whole and guided the development of more advanced control-schemes. As expected, use of crisp DRs, which prioritises the orders by observing one input parameter, did not performed well in case of varying parameters. Both rules focused on processing time parameter for prioritisation (LTWR and MTWR) resulted in the worst performance for the delay KPI althoug LTWR was the most consistent amongst crisp rules for delay performance indicator as it eliminated delays in smaller orders. In case of overloading echelons with work by significant increase of order size, the processing time also increases, and orders are taking longer time to be manufactured. The delay KPI is similar for all the rules, with the best delay recorder by using LTWR rule but in a trade-off with holding cost KPI which is considerably higher than for all other rules which indicated that an arrival time and due date focused DRs provide better solutions when both KPIs are considered.

Subsequently, the FIS-based decision making in form of FDRs was proposed. Key inputs to the new system were determined based on experimental data obtained using crisp DRs. Fuzzy sets representing each of the uncertain inputs and outputs were defined for each echelon separately. Finally, two fuzzy rule bases were designed, each with focus on different KPI. Delay-focused FDR was developed to order more raw material to minimise possible shortages and to minimise the delay, while holding cost-focused FDR was design to minimise the holding cost by keeping lower levels of inventory. The FDRs were subsequently evaluated in the same setup as the crisp DRs. Substantial improvement of decision-making performance as measured by KPIs was observed, with further room for enhancement also being acknowledged. FDRs showed a degree of adaptability to uncertain demand, contrary to crisp DRs. Use of the delay-focused FDRs led to decrease of the total delay in benchmark scenario, decreasing by the 73% while increasing the holding cost only by 34% compared to best crisp DR.

Once FDRs with manually designed fuzzy rule bases were developed, the focus of the research moved into improving those rule bases by employing optimisation methods. Namely NSGAII algorithm was developed and implemented, using previously introduced KPIs as a fitting function. The evaluation of individual solutions was performed via simulation on the benchmark scenario. Two types of chromosomes were introduced: one encoding rule bases for the inventory control subproblem, and one encoding rule bases for the scheduling subproblem. Each echelon in SC was assigned both or only one of those rule bases as appropriate. Upon initial investigation it was decided to use independent chromosome(s) for each echelon, allowing for the rule bases to vary between echelons. The optimisation scheme allowed to extract knowledge obtained from simulation of the model in the form of rule bases. Furthermore, the obtained FDRs represented a range in trade-off between the two KPIs.

This method showed further improvement of the control-scheme performance, while the use of population-based NSGAII allowed to approximate the Pareto Front. However, it also raised concerns of robustness and role of the uncertainty in the performance of the proposed control-scheme. Since the extraction of knowledge was performed on a single benchmark scenario, a possible overfitting of rule bases to that specific instance had to be alleviated. This investigation led to introduction of Monte-Carlo simulation into the evaluation process within NSGAII.

A statistical model of incoming orders was developed and used to randomly generate customer orders of a specific intensity for each Monte-Carlo simulation.

As specific properties of customer orders, such as size, can greatly influence the KPIs levels, using them for comparison of performance between different trials of Monte-Carlo simulation became non-straightforward. To overcome this difficulty a normalisation scheme was proposed. This scheme made the delay and holding cost KPIs for different Monte-Carlo trials directly comparable. This led to the development of MCNSGAII. The evaluation based on KPI values from a single simulation of the benchmark scenario, was replaced with Monte-Carlo simulation of multiple random trials of similar scenarios. The KPIs values from each trial were normalised. The average values of normalised KPIs replaced raw KPIs as fitting function. To guide the optimisation towards achieving robustness, the standard deviation of each of the normalised KPIs were included in the fitting function, changing the initial two-objectives problem into four-objectives one.

In the final experiment all proposed control-schemes were evaluated on the benchmark scenario and all proposed control-schemes were tested against varying due dates, order sizes and processing time to assess the reaction to uncertain demand and different processing times. Additionally, a longer scenario with varying order intensity was simulated multiple times with randomly generated orders and average values of KPIs and the standard deviation of solutions were assessed. All the proposed FDRs achieved better performance in both KPIs than crisp DRs. Additionally, FDRs obtained by both NSGAII and MCNSGAII were shown to further enhance performance in the considered scenarios. When the robustness of the control-scheme, in form of the standard deviation of KPIs; is considered, then the MCNSGAII demonstrated more consistent performance in each case and the best overall results on the varying intensity scenario for individual FDRs, which led to creation of robust control-scheme.

In conclusion control-schemes based on FDRs were shown to obtain good performance with ability to adapt to uncertain environment. Multi objective optimisation capturing dynamics of the SC under no information sharing policy was developed. The trade-offs between holding cost and delays were optimised by MCNSGAII leading to successful inventory control and scheduling for multiple tiers of SC. Furthermore, in conjunction with simulation-driven optimisation, it forms a system capable of extraction of knowledge from the uncertain data. This approach was successfully applied to different echelons in SC, leading to robust performance across the SC. This thesis built a framework that was shown to be useful for solving problems in the domain of SCM, with inherent handling of uncertainty and extensible simulation model.

10.2 Future work

The SC model proposed in this work, although general is not a definitive one. It could be further extended by introducing uncertainty into the echelon processes, such as breakdown of machines, variable quality of supplies, changes to transportation time etc. New echelons and additional subproblems for existing ones can be introduced. An example of that would be an addition of supplier selection problem.

The proposed fuzzy control-scheme contains a fair number of parameters. Further research could focus on development and evaluation of automatic parameter selection, thus reducing the manual work required to set it up and potentially bringing additional benefits on performance.

Ability to make decisions ahead of time can bring further benefits. Thus, inclusion of forecasting into the proposed model, with the aim of improving overall SC performance and introducing information sharing and negotiations between echelons for further cooperative nature of a SC is an interesting avenue of research.

11 REFERENCES

Abdel-Aleem, A., El-Sharief, M. A., Hassan, M. A., and El-Sebaie, M. G. (2017) 'Implementation of Fuzzy and Adaptive Neuro-Fuzzy Inference Systems in Optimization of Production Inventory Problem'. Applied Mathematics and Information Sciences 11 (1), 289-298

Amaran, S., Sahinidis, N. V., Sharda, B., and Bury, S. J. (2014) 'Simulation Optimization: A Review of Algorithms and Applications'. 4OR 12 (4), 301-333

Assarzadegan, P., Rasti-Barzoki, M. (2016) 'Minimizing sum of the due date assignment costs, maximum tardiness and distribution costs in a supply chain scheduling problem'. Applied Soft Computing 47, 343-356

Balasubramanian, J., Grossmann, I. E. (2003) 'Scheduling optimization under uncertainty - an alternative approach'. Computers and Chemical Engineering 27(4), 469–490

Baptiste, P., Alsene, É., and Gaudimier, R. (2008) 'Integration of Production and Shipping Planning: A Co-Operative Approach'. Production Planning and Control 19 (7), 645-654

Bassett, M. H., Pekny, J. F., Reklaitis, J. F. (1997) 'Using detailed scheduling to obtain realistic operating policies for a batch processing facility'. Industrial and Engineering Chemistry Research 36 (5), 1717–1726

Baykasoglu, A., Göçken, T. (2008) ‘A review and classification of fuzzy mathematical programs.’ Journal of Intelligent and Fuzzy Systems 19, 205–229

Behnamian, J. (2016) 'Survey on Fuzzy Shop Scheduling'. Fuzzy Optimization and Decision Making 15 (3), 331-366

Bozutti, D.F., Espôsto, K.F. (2019) ‘Sales and Operations Planning: a comparison between the demand-driven and traditional approaches.’ International Journal of Production Management and Engineering 7(1), 23-38

Burrows, R.P. (2012) ‘The Market-Driven Supply Chain: a revolutionary model for sales and operations planning in the new on-demand economy.’ New York: AMACOM - American Management Association.

Cachon, G. (1999). 'Managing supply chain demand variability with scheduled ordering policies'. Management Science 45 (6), 843–856

Chen, J. M., Chen T. H. (2005) 'The multi-item replenishment problem in a two-echelon supply chain: the effect of centralization versus decentralization' Computers & Operations Research 32, 3191–3207

Chen, J., Huang, G.Q., Wang, J., Q. (2019) 'Synchronized Scheduling of Production and Outbound Shipping Using Bilevel-Based Simulated Annealing Algorithm'. Computers & Industrial Engineering 137. DOI:10.1016/j.cie.2019.106050

Cheng, B., Li, K., and Chen, B. (2010) 'Scheduling a Single Batch-Processing Machine with Non-Identical Job Sizes in Fuzzy Environment using an Improved Ant Colony Optimization'. Journal of Manufacturing Systems 29 (1), 29-34

Cheng, B., Leung, J. Y., and Li, K. (2015) 'Integrated Scheduling of Production and Distribution to Minimize Total Cost using an Improved Ant Colony Optimization'. Computers & Industrial Engineering 83, 217-225

Childerhouse, P., Towill, D. R. (2004) 'Reducing Uncertainty in European Supply Chains'. Journal of Manufacturing Technology Management 15 (7), 585-598

Chinello, E., Lee Herbert-Hansen, Z.N., Khalid, W. (2020) 'Assessment of the Impact of Inventory Optimization Drivers in a Multi-Echelon Supply Chain: Case of a Toy Manufacturer'. Computers & Industrial Engineering 141. DOI:10.1016/j.cie.2019.106232

Chu, Y., You, F., Wassick, J. M., and Agarwal, A. (2015) 'Integrated Planning and Scheduling Under Production Uncertainties: Bi-Level Model Formulation and Hybrid Solution Method'. Computers and Chemical Engineering 72, 255-272

Costantino, F., Di Gravio, G., Shaban, A., Tronci, M. (2015) 'The Impact of Information Sharing on Ordering Policies to Improve Supply Chain Performances'. Computers & Industrial Engineering 82, 127–142

Costantino, F., Di Gravio, G., Shaban, A., and Tronci, M. (2016) 'Smoothing Inventory Decision Rules in Seasonal Supply Chains.' Expert Systems with Applications 44, 304–319.

Croom, S., Romano, P., Giannakis, M. (2000) 'Supply Chain Management: An Analytical Framework for Critical Literature Review'. European Journal of Purchasing& Supply Management 6, 67-83

Cordón, O., Herrera, F., Hoffmann, F., Magdalena, L. (2001) 'Genetic fuzzy systems: Fuzzy knowlege extraction by Evolutionary Algorithms'. World Scientific. Advances in Fuzzy Systems - Applications and theory. Singapore, ISBN: 981-02-4016-3

Dai, Z., Aqlan, F., Gao, K. (2017) 'Optimizing Multi-Echelon Inventory with Three Types of Demand in Supply Chain.' Transportation Research Part E: Logistics and Transportation Review 107, 141–177

De Campos Souza, P.V. (2020) 'Fuzzy Neural Networks and Neuro-Fuzzy Networks: A Review the Main Techniques and Applications Used in the Literature'. Applied Soft Computing 92, 106-275

Deb, K. (2002) 'A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II'. IEEE Transactions on Evolutionary Computation 6, 182-197

Demirli, K., Cheng, S. X., and Muthukumaran, P. (2003) 'Subtractive Clustering Based Modeling of Job Sequencing with Parametric Search'. Fuzzy Sets and Systems 137 (2), 235-270

Duenas, A., Petrovic, D. (2008) 'Multi-Objective Genetic Algorithm for Single Machine Scheduling Problem Under Fuzziness'. Fuzzy Optimization and Decision Making 7 (1), 87-104

Eruguz, A. S., Sahin, E., Jemai, Z., Dallery, Y. (2016) 'A comprehensive survey of guaranteed-service models for multi-echelon inventory optimization'. International Journal of Production Economics 172, 110–125

Forrester, J. (1961) ‘Industrial Dynamics.’ M.I.T. Press, Cambridge, Wiley, New York

Framinan, J. M., Perez-Gonzalez, P., Fernandez-Viadas, V. (2019) 'Deterministic assembly scheduling problems: A review and classification of concurrent-type scheduling models and solution procedures'. European Journal Of Operational Research 273, 401-417

Hao, J., Cao, L., and Jiang, D. (2015) 'Integrated Production-Distribution Scheduling Problem with Multiple Independent Manufacturers'. Mathematical Problems in Engineering 2015. DOI:10.1155/2015/579893

Harjunkoski, I., Nyström, R., Horch, A. (2009) 'Integration of Scheduling and control—Theory Or Practice?'. Computers & Chemical Engineering 33 (12), 1909-1918

Hassanzadeh, A., Rasti-Barzoki, M., Khosroshahi, H. (2016) 'Two New Meta-Heuristics for a Bi-Objective Supply Chain Scheduling Problem in Flow-Shop Environment'. Applied Soft Computing Journal 49, 335-351

Hemmati, M., Fatemi Ghomi, S.M.T., Sajadieh, M.S. (2017) 'Vendor Managed Inventory with Consignment Stock for Supply Chain with Stock- and Price-Dependent Demand'. International Journal of Production Research 55 (18), 5225–5242

Horowitz, E., and Sahni, S. (1978) 'Fundamentals of Computer Algorithms'. Computer Science Press, Potomac 1978, Chapter 12

Hochbaum, D. S. (1997) 'Approximation algorithms for NP-hard problems'. ACM Sigact News 28, 40-52

Holland, J. H. (1975) 'Adaptation in Natural and Artificial Systems', University of Michigan Press, Cambridge, US. ISBN 0-262-58111-6

Holthaus, O., Rajendan, C. (1997) 'Efficient dispatching rules for scheduling in a job shop'. International Journal of Production Economics 48 (1), 87-105

Horn, J., Nafpliotis, N., Goldberg D. E. (1994) 'A Niched Pareto Genetic Algorithm for Multiobjective Optimization'. Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence 1, 82-87.

Hsu, C., Kao, B., Ho, V. L., Li, L., Lai, K. R. (2016) 'An Agent-Based Fuzzy Constraint-Directed Negotiation Model for Solving Supply Chain Planning and Scheduling Problems'. Applied Soft Computing Journal 48, 703-715

Huang, C., Huang, Y., Lai, P. (2012) 'Modified Genetic Algorithms for Solving Fuzzy Flow Shop Scheduling Problems and their Implementation with CUDA'. Expert Systems with Applications 39 (5), 4999-5005

Hugos, M. (2003) 'Essentials of Supply Chain Management'. John Wiley & Sons, Inc., Hoboken, New Jersey. ISBN 0-471-23517-2

Gao, L. L., Altay, N., Robinson, E.P. (2008) 'A Comparative Study of Modeling and Solution Approaches for the Coordinated Lot-Size Problem with Dynamic Demand'. Mathematical and Computer Modelling 47 (11), 1254–1263

Gao, S., Qi, L., and Lei, L. (2015) 'Integrated Batch Production and Distribution Scheduling with Limited Vehicle Capacity.' International Journal of Production Economics 160, 13–25

Grabot, B., Geneste, L. (1994) 'Dispatching rules in scheduling Dispatching rules in scheduling: a fuzzy approach.' International Journal of Production Research 32(4), 903-915.

Guo, Z., Shi, L., Chen, L., Liang, Y. (2017) 'A Harmony Search-Based Memetic Optimization Model for Integrated Production and Transportation Scheduling in MTO Manufacturing'. Omega 66, 327–343

Jain, S., and Foley, W. J. (2016) 'Dispatching Strategies for Managing Uncertainties in Automated Manufacturing Systems'. European Journal of Operational Research 248 (1), 328-341

Jia, Z., Ierapetritou, M. G. (2006) 'Uncertainty analysis on the right-handside for MILP problems.' AIChE Journal, 52, 2486–2495

Kaddoussi, A., Zoghlami, N., Hammadi, S., and Zgaya, H. (2013) 'An Agent-Based Distributed Scheduling for Crisis Management Supply Chain'. International Journal of Computational Intelligence Systems 6 (1), 156-173

Kim, T., Kim, Y., and Cho, H. (2020) ‘Dynamic Production Scheduling Model under Due Date Uncertainty in Precast Concrete Construction’. Journal of Cleaner Production 257. DOI: 10.1016/j.jclepro.2020.120527

Klosterhalfen, S., Minner, S. (2010) 'Safety Stock Optimisation in Distribution Systems: A Comparison of Two Competing Approaches'. International Journal of Logistics Research and Applications, 13 (2), 99–120

Kok, T. D., Grob, C., Laumanns, M., Minner, S., Rambau, J., Schade, K. (2018) 'A typology and literature review on stochastic multi-echelon inventory models'. European Journal of Operational Research 269, 955-983

Kreipl, S., Pinedo, M. (2004) 'Planning and Scheduling in Supply Chains: An Overview of Issues in Practice.' Production and Operations Management 13(1), 77-92

Lai, P., Wu, H. (2011) 'Evaluate the Fuzzy Completion Times in the Fuzzy Flow Shop Scheduling Problems using the Virus-Evolutionary Genetic Algorithms'. Applied Soft Computing Journal 11 (8), 4540-4550

Lai, P., Shu, M. (eds.) (2008) 'Tardiness in Fuzzy Flow Shop Scheduling Problems Based on Possibility and Necessity Measures'. Proceedings - 8th International Conference on Intelligent Systems Design and Applications. DOI: 10.1109/ISDA.2008.218

Lambert, D. M., Cooper, M. C. (2000) 'Issues in Supply Chain Management'. Industrial Marketing Management 29(1), 65-83

Lee, K. K. (2008) 'Fuzzy Rule Generation for Adaptive Scheduling in a Dynamic Manufacturing Environment'. Applied Soft Computing Journal 8 (4), 1295-1304

Li, K., Sivakumar, A. I., Ganesan, V. K. (2008) 'Complexities and Algorithms for Synchronized Scheduling of Parallel Machine Assembly and Air Transportation in Consumer Electronics Supply Chain'. European Journal of Operational Research 187 (2), 442-455

Li, J., Sun, K., Xu, D., and Li, H. (2010a) 'Single Machine due Date Assignment Scheduling Problem with Customer Service Level in Fuzzy Environment'. Applied Soft Computing Journal 10 (3), 849-858

Li, X., Zhang, C., Gao, L., Li, W., and Shao, X. (2010b) 'An Agent-Based Approach for Integrated Process Planning and Scheduling'. Expert Systems with Applications 37 (2), 1256-1264

Liao, T. W., Su, P. (2017) 'Parallel Machine Scheduling in Fuzzy Environment with Hybrid Ant Colony Optimization Including a Comparison of Fuzzy Number Ranking Methods in Consideration of Spread of Fuzziness'. Applied Soft Computing Journal 56, 65-81

Lima, C., Relvas, S., Barbosa-Póvoa, A. (2021) 'Designing and planning the downstream oil supply chain under uncertainty using a fuzzy programming approach' [online] 17 May. available from < https://www.sciencedirect.com/science/article/abs/pii/S0098135421001514?via%3Dihub 28 May 2020>

Liu, B., Chen, H., Li, Y., Liu, X. (2015) 'A Pseudo-Parallel Genetic Algorithm Integrating Simulated Annealing for Stochastic Location-Inventory-Routing Problem with Consideration of Returns in e-Commerce'. Discrete Dynamics in Nature and Society, vol 2015. DOI:10.1155/2015/586581

Longo, F. (2011) 'Supply chain management based on modeling & simulation : State of the art and application examples in inventory and warehouse management'. Supply Chain Management, Industrial Engineering and Management: Intech, 2011. DOI: 10.5772/15103

Mamdani, E.H., Assilian, S. (1975) 'An Experiment in Linguistic Synthesis with a Fuzzy Logic Control.' International Journal of Man-Machine Studies 7, 1-13.

Maravelias, C. T., Sung, C. (2009) 'Integration of Production Planning and Scheduling: Overview, Challenges and Opportunities'. Computers & Chemical Engineering 33 (12), 1919-1930

Moons, S., Ramaekers, K., Caris, A., Arda, Y. (2017) 'Integrating Production Scheduling and Vehicle Routing Decisions at the Operational Decision Level: A Review and Discussion'. Computers & Industrial Engineering 104, 224–245

Mulvey, J., Vanderbei, R., Zenios, S. A. (1995) 'Robust optimization of large-scale systems'. Operations Research 43, 264–281

Nakhaeinejad, M., Nahavandi, N. (2013) 'An Interactive Algorithm for Multi-Objective Flow Shop Scheduling with Fuzzy Processing Time through Resolution Method and TOPSIS'. International Journal of Advanced Manufacturing Technology 66 (5-8), 1047-1064

Negnevitsky, M. (2002) 'Artificial Intelligence: A Guide to Intelligent Systems', Pearson Education Limited, 1st edition, London, UK

Nikolopoulou, A., Ierapetritou, M. G. (2012) 'Hybrid Simulation Based Optimization Approach for Supply Chain Management'. Computers and Chemical Engineering 47, 183-193

Olugu, E. U., Wong, K., T. (2009) 'Supply Chain Performance Evaluation: Trends and Challenges'. American Journal of Engineering and Applied Sciences 2(1), 202-211

Othman, B. S., Zgaya, H., Dotoli, M., Hammadi, S. (2017) 'An Agent-Based Decision Support System for Resources' Scheduling in Emergency Supply Chains'. Control Engineering Practice 59, 27-43

Paul, S. K. (2015) 'Supplier Selection for Managing Supply Risks in Supply Chain: A Fuzzy Approach'. International Journal of Advanced Manufacturing Technology 79 (1-4), 657-664

Pedrycz, W., Gomide, F. (1998) 'An Introduction to Fuzzy Sets: Analysis and Design' A Bradford Book, The MIT Press, Cambridge, Massachusetts ISBN: 0-262-16171-0

Peng, H., Shen, N., Liao, H., Xue, H., Wang, Q. (2020) 'Uncertainty factors, methods, and solutions of closed-loop supply chain — A review for current situation and future prospects' Journal of Cleaner Production 254. DOI:10.1016/j.jclepro.2020.120032

Petrovic, D., Kalata, M. (2019) 'Multi-objective optimisation of risk and business strategy in real-world supply networks in the presence of uncertainty.' Journal of the Operational Research Society 70(11), 1869-1884

Petrovic, D., Xie, Y., Burnham, K., Petrovic, R. (2008) 'Coordinated Control of Distribution Supply Chains in the Presence of Fuzzy Customer Demand'. European Journal of Operational Research 185 (1), 146-158

Pickardt, C. W., Hildebrandt, T., Branke, J., Heger, J., Scholz-Reiter, B. (2013) 'Evolutionary generation of dispatching rule sets for complex dynamic scheduling problems'. International Journal of Production Economics 145, 67-77

Pistikopoulos, E.N. (1995) 'Uncertainty in Process Design and Operation's Computer Chemical Engineering, 19, p553

Schuster Puga, M., Tancrez, J. (2017) 'A Heuristic Algorithm for Solving Large location–inventory Problems with Demand Uncertainty'. European Journal of Operational Research 259 (2), 413-423

Pundoor, G., Chen, Z. (2009) 'Joint Cyclic Production and Delivery Scheduling in a Two-Stage Supply Chain'. International Journal of Production Economics 119 (1), 55-74

Rokni, S., Fayek, A. R. (2010) 'A Multi-Criteria Optimization Framework for Industrial Shop Scheduling using Fuzzy Set Theory'. Integrated Computer-Aided Engineering 17 (3), 175-196

Roozbeh Nia, A., Hemmati Far, M., Akhavan Niaki, S. T. (2014) 'A Fuzzy Vendor Managed Inventory of Multi-Item Economic Order Quantity Model Under Shortage: An Ant Colony Optimization Algorithm'. International Journal of Production Economics 155, 259-271

Sahin, F., Powell Robinson, E. (2005) 'Information sharing and coordination in make-to-order supply chains'. Journal of Operations Management 23, 578-598

Salem, R.W., Haouari, M. (2017) 'A Simulation-Optimisation Approach for Supply Chain Network Design under Supply and Demand Uncertainties.' International Journal of Production Research 55 (7), 1845–1861

Sarker, B.R. (2014) 'Consignment Stocking Policy Models for Supply Chain Systems: A Critical Review and Comparative Perspectives'. International Journal of Production Economics 155, 52–67

Sawik, T. (2011) 'Scheduling in Supply Chains Using Mixed Integer Programming', John Wiley & Sons Inc., Hoboken, Canada DOI: 10.1002/9781118029114

Sawik, T. (2014) 'Optimization of Cost and Service Level in the Presence of Supply Chain Disruption Risks: Single Vs. Multiple Sourcing'. Computers and Operations Research 51, 11-20

Sawik, T. (2016a) 'Integrated Supply, Production and Distribution Scheduling Under Disruption Risks'. Omega (United Kingdom) 62, 131-144

Sawik, T. (2016b) 'A Portfolio Approach to Supply Chain Disruption Management'. International Journal of Production Research, 1-22

Schoenmeyr, T., Graves, S. C. (2009) 'Strategic Safety Stocks in Supply Chains with Evolving Forecasts'. Manufacturing & Service Operations Management 11 (4), 657-673

Shaban, A., Costantino, F., Di Gravio, G., and Tronci, M. (2019) 'A New Efficient Collaboration Model for Multi-Echelon Supply Chains'. Expert Systems with Applications 128, 54-66

Shaw, M.J., Park, S., Raman. (1992) 'Intelligent Scheduling with Machine Learning Capabilities: The Induction of Scheduling Knowledge.' IIE Transactions 24 (2), 156–168.

Subramanian, K., Rawlings, J. B., and Maravelias, C. T. (2014) 'Economic Model Predictive Control for Inventory Management in Supply Chains'. Computers and Chemical Engineering 64, 71-80

Wei, J., Krajewski, L. (2000) 'A Model for Comparing Supply Chain Schedule Integration Approaches'. International Journal of Production Research 38, 2099-2123

Talibi, E-G. (2009) 'Metaheuristics. From Design to implementation' John Wiley & Sons, Hoboken, Canada ISBN: 978-0-470-27858-1

Tamannaei, M., Rasti-Barzoki, M. (2019) 'Mathematical Programming and Solution Approaches for Minimizing Tardiness and Transportation Costs in the Supply Chain Scheduling Problem'. Computers & Industrial Engineering 127, 643–656

Temiz, I., Erol, S. (2004) 'Fuzzy Branch-and-Bound Algorithm for Flow Shop Scheduling'. Journal of Intelligent Manufacturing 15 (4), 449-454

Van Laarhoven, P. J. M., Pedrycz, W. (1983) 'A fuzzy extension of Saaty's priority theory' Fuzzy Sets and Systems, 11(3) 229–241

Varthanan, P.A., Murugan, N., Kumar, G.M. (2012) 'A Simulation Based Heuristic Discrete Particle Swarm Algorithm for Generating Integrated Production–Distribution Plan'. Applied Soft Computing 12 (9), 3034–3050

Yan, H., Wan, X., Xiong, F. (2015) 'Integrated Production Planning and Scheduling for a Mixed Batch Job-Shop Based on Alternant Iterative Genetic Algorithm'. Journal of the Operational Research Society 66 (8), 1250-1258

Ye, F., Wang, Z. (2013). 'Effects of information technology alignment and information sharing on supply chain operational performance.' Computers & Industrial Engineering 65(3), 370–377.

Yeung, W., Choi, T., Cheng, T. C. E. (2011) 'Supply Chain Scheduling and Coordination with Dual Delivery Modes and Inventory Storage Cost'. International Journal of Production Economics 132 (2), 223-229

Yolmeh, A., Salehi, N. (2015) 'An Outer Approximation Method for an Integration of Supply Chain Network Designing and Assembly Line Balancing Under Uncertainty'. Computers and Industrial Engineering 83, 297-306

You, F., Grossmann, I. E. (2008) 'Design of Responsive Supply Chains Under Demand Uncertainty'. Computers and Chemical Engineering 32 (12), 3090-3111

Yu, Z., Yan, H., Edwin Cheng, T.C. (2001) 'Benefits of information sharing with supply chain partnerships'. Industrial Management & Data Systems 101 (3), 114-121

Zahran, S. K., Jaber, M. Y., Zanoni, S. (2016) 'Comparing Different Coordination Scenarios in a Three-Level Supply Chain System'. International Journal of Production Research, 1-21

Zhang, L., Xu, X., Tao, L. (2012) 'Some Similarity Measures for Triangular Fuzzy Number and Their Applications in Multiple Criteria Group Decision-Making'. Journal of Applied Mathematics vol. 2013 . DOI:10.1155/2013/538261

12 APPENDICES

Publications and academic activities

12.1.1 Published papers

- Petrovic, D., Kalata, M. (2019) 'Multi-objective optimisation of risk and business strategy in real-world supply networks in the presence of uncertainty.' *Journal of the Operational Research Society.* Vol 70(11) pp. 1869-1884

12.1.2 Papers under revision

- Petrovic, D., Kalata, M., Luo. J., 'A fuzzy scenario-based optimisation of supply network efficiency, robustness and flexibility.' *Computers and Industrial Engineering*

12.1.3 Other academic activities

12.1.3.1 Teaching

- Participation in *PhD Teaching Opportunities* program. I was teaching for one term part of module M23MSE (*Artificial Intelligence*) with a focus on fuzzy controllers

12.1.3.2 Workshops and conferences

- Participation in *'Risk, resilience and robustness of dynamic supply networks; bridging mathematical models and practice'* workshop organised by ICMS and UK Engineering and Physical Sciences Research Council project,11-13 January 2017, Edinburgh
- Conference presentation on '*New fuzzy dispatching rules for integrated planning and scheduling across supply chain'*. 30th European Conference on Operational Research, 23-26 June 2019, Dublin

ETHICAL APPROVALS

Certificate of Ethical Approval

Applicant:

Magdalena Ronge

Project Title:

Integrated supply chain planning and scheduling in uncertain environments - stage 1

This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Low Risk

Date of approval:

24 March 2017

Project Reference Number:

P52658

Coventry University

Certificate of Ethical Approval

Applicant:

Magdalena Kalata

Project Title:

Integrated supply chain planning and scheduling in uncertain environments - stage 2

This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Low Risk

Date of approval:

05 July 2018

Project Reference Number:

P72921

Certificate of Ethical Approval

Applicant:

Magdalena Kalata

Project Title:

Integrated supply chain planning and scheduling in uncertain environments - stage 3

This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Low Risk

Date of approval:

27 August 2019

Project Reference Number:

P93971