AN AGENT-BASED APPROACH TO INTELLIGENT MANUFACTURING NETWORK CONFIGURATION

By

DÉSIRÉ GUIOVANNI JULES

A thesis submitted to
The University of Birmingham for the
degree of
DOCTOR OF PHILOSOPHY

Department of Mechanical Engineering
School of Engineering
University of Birmingham

July 2015

Abstract

Small and medium manufacturers are among the most productive, in terms of small and medium enterprise value addition. Their participation in inter-firm collaboration can enhance their market reach while maintaining production lean. The planning of such collaboration has the propensity of being centralised, which is believed to be an unsustainable approach, in today's complex environment. The aim of the research is to investigate the configuration of manufacturing networks, where manufacturers maintain control over their scheduling activities and participate in a market-based event, to decide which configurations are retained.

The aim was achieved by introducing notions of flow shop system for networks, pairing for network configuration and bidding for network selection. The research modelled networks as a flow shop system, where n jobs and m manufacturers were involved. Next, the work investigated two pairing mechanisms, namely manufacturer pairing and operation pairing, where the intention was to capture and optimise collaboration at the granular level and then build up a network from those intermediate forms of organisation. Finally, the research looked at two bidding protocols where the first protocol involves manufacturers that bid for the operations that constitute the process plan of a job. The second protocol is concerned with networks that bid for a job in its entirety.

The methodology used consisted of identifying the boundaries of the problem, modelling the entities that contribute to a solution, simulating the proposed problem solving mechanisms and evaluating the merits of the mechanisms. The boundaries of the problem were set by an industrial use case and two operation research data sets. The problem was modelled as decentralised flow shop scheduling and the holonic paradigm was used to identify the problem solving agents. Agent-based modelling and simulation were used to investigate manufacturer pairing and the bidding protocols. These informed the development of a multi-agent system as well as a knowledge base with which the operation pairing mechanism was investigated.

Although manufacturer pairing outperformed operation pairing on lead time, it is strongly believed that the latter has potential to achieve true decentralisation of scheduling, with good performance on indicators of scalability, conflict resolution and schedule optimisation. Finally, the second bidding protocol was found to retain network configurations that were most apt to meet customer requirements, ranked by importance.

WITH LOVING GRATITUDE FOR

MY FAMILY &

HEDLEY &

AUNT VIOLETTE

Acknowledgement

I would like to thank all the people who made this research possible. My sincere gratitude goes to my research supervisor Dr Mozafar Saadat for the amazing support on the project.

I would like to acknowledge with thanks GFM s.r.l. for their contribution to the research and particularly Mr Lucas Pasinetti. The help from my colleagues Salman Saeidlou, Alireza Rastegarpanah, Amir Hajiyavand, Nan Li and Mohammad Owliya has been greatly appreciated. I would like to thank Dr Carl Anthony for the chairing of my Viva interview. For the valuable suggestions that enhanced this thesis, I would like to acknowledge with thanks Prof Ali Emrouznejad and Dr Marco Castellani.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

INTRODUCTION

1.1 Preamble

Small and medium manufacturing businesses contribute significantly to the economy. In 2013, 7% of SMEs in the UK, was in the manufacturing sector (Statista, 2015). The sector is segmented into specific industries such as food, drink, paper, apparel, and electronics, pharmaceutical, petrochemical, transport equipment and custom-built equipment. Although manufacturing SMEs account for a small share of the total number of SMEs in the UK, they are among the most productive. In 2014, manufacturing accounted for 15% of SME value added (European Commission, 2014).

Centralisation of manufacturing is great for capturing economies of scale but it is not always sustainable. For instance, in the automotive industry, all vehicle products share similar core production technologies and consequently productivity benefits from a centralised and automated production. Today however, many external factors at play have caused the cost of the distribution system to account for between 25% and 40% of the sales price of a vehicle (Wells, 2013). The creation of local manufacturing-distribution units would not have been profitable in the past economy. Today, it might be a necessity for manufacturers to reduce the relevance of those factors for better cost control.

Today's exponential progress, in technology, is enabling new market entrants to tool up, thrive and address new market opportunities, at smaller scales. Large companies are increasingly focusing on the large scale-and-scope roles and avoiding flaky roles that are best served by new entrants. Large companies are becoming infrastructure providers and agent businesses and continue to grow by being of service to new market entrants who are adding customised value (Hagel et al, 2015). Furthermore, competitiveness in the manufacturer industry is increasingly driven by soft power such as the ability of companies to innovate and be trusted (Ageron et al, 2012). In the steel industry, the investment would usually be focused on incremental improvements such as enhancement of steel properties. In another industry such as mechanical engineering, some new business opportunities are in the form of one-off productions, where even a large manufacturer would require prohibitive investments, in research and development, production infrastructures, supply chain reconfiguration and labour training.

Today, the pool of manufacturing SMEs provides significant opportunities for manufacturing services to evolve according to the specific requirements of customers and in response to the shifting economic environment. It is usually necessary for them to participate in an association called a virtual organisation breeding environment (Camarinha-Matos et al, 2009). The breeding environment acts as a platform for the SMEs to engage in collaboration. By combining unique and complementary knowledge as well as sharing resources as part of an inter-firm cooperative agreement, innovative solutions are achieved, that customers are willing to pay high prices for (Hanna and Walsh, 2002). Moreover, it is a cheaper alternative for SMEs to access new technologies, at the expense of some royalty fees paid and some efforts expended in searching, evaluating and coordinating the technology providers (Atuahene-Gima, 1992). The environment also encourages contractual self-enforcement among SMEs because of the compelling value involved in staying in a long term relationship (Mesquita 2008).

Moreover, the manufacturing market is moving towards individualisation on demand, also known as batch size one production (Lasi et al, 2013). An interesting perspective on batch-size-one production is that it allows marketplaces of production needs (Almada-Lobo, 2015) and of temporary under-capacities (Freitag et al, 2015) to emerge. A niche manufacturer may uncover the specialised production needs of an agent business and fulfil them, thus enabling the agent business to extend its range of product customisation. A manufacturer may market to a situation of overtime and overloaded production. This constitutes demand for production needs and demand for overloaded production. Then, there are marketplaces for unused production capacity or temporary over-capacity (Almada-Lobo, 2015). Excess capacity is capital intensive and normally exists for strategic reasons. A manufacturer invests into capacity for future increases in customer demand and into capacity for future product innovations (Freitag et al, 2015). With the right infrastructure, these scattered capacities can be summoned, to fulfil the production needs of multiple businesses. This constitutes supply of excess production capacity.

Current advanced technologies may assemble into the right infrastructure, under the framework of Industry 4.0, to support the complex manufacturing environment such as decentralised excess production capacity for batch size one production. The aim of the 4th industrial revolution is to develop manufacturing systems capable of self-organisation, self-optimisation and optimisation of production as a whole (Brettel et al, 2014). Decentralised machines will become increasingly capable of democratic decision making as well as capable of communicating with other machines wirelessly (Lasi et al, 2013). Excess capacity will no longer have to be controlled by a single company or even by multiple companies but it will be self-controlled and solicited by external demand. As well as the internet of things and distributed manufacturing, Industry 4.0 also addresses collaborative networks and internet of services (Brettel et al, 2014). A collaborative network exists, via the joint use of resources from different companies, to supply production competencies and capacities that match the demand of a custom production (Freitag et al, 2015).

Manufacturing execution system generates advanced production planning and scheduling in a centralised manner which requires overnight data crunching (Verstraete el al, 2008). The next day, production must occur with minimum deviation from the schedule because MES is not responsive enough to adjust the schedule in real time (Meyer et al, 2009). In distributed manufacturing, where temporary production over-capacities are summoned, operation scheduling is not performed in advance. It takes place just in time, so as to keep production going at the manufacturing station. This is real time scheduling which is event-driven and in a constant re-iteration (Kaihara et al, 2010). As previously mentioned, manufacturers have strategic reasons to maintain excess capacity. The schedules of a manufacturer and of its collaborative networks co-exist. When manufacturers need the excess capacity for their own production, the collaborative networks relying on the surplus capacities, may incur fluctuations in their schedules. Some scheduling problems involving such multiple interdependent issues cannot be effectively solved unless it is decomposed into self-contained sub-problems being individually addressed (Fujita et al, 2014). It is difficult for either a centralised or decentralised approach to find the optimal schedule for the aforementioned use case. However, pushing forward in the direction of decentralised scheduling, can help uncover useful insights in self-scheduling systems. The research described in this thesis addresses decentralised flow shop scheduling of manufacturing networks.

1.2 Aim and objectives of the research

The aim of the research was the investigation of the configuration of manufacturing networks where manufacturers control their own schedules and through a market-based mechanism, a configuration is selected. The aim is achieved through the following main objectives:

- A literature review of networks, scheduling techniques, modelling approaches and enablers of decentralised problem solving
- Identification of the problem, supported by industrial use case and operation research datasets
- Generation of innovative ideas for decentralised scheduling in the manufacturing network breeding environment
- Development of experiments for the validation and evaluation of ideas

The ideas proposed are mechanisms for network formation and network selection and the experiments were composed of agent-based modelling of network formation and selection as well as multi-agent system implementation of decentralised scheduling.

1.3 Methodology

The objectives were achieved using mostly open sourced development kits. The literature review was carried out using google scholar and the University of Birmingham library services as a source of peer-reviewed journals, articles and reference books. NVivo software was used to manage the data collected from the literature survey. The modelling of network formation, scheduling and selection was carried out with the following tools:

- Repast Simphony for agent-based modelling and simulations
- Workflow agent development environment (WADE), workflow lifecycle management environment (WOLF) supported by Eclipse Kepler, were the combination of tools used for developing a multi-agent system
- The multi-agent system is supported by a knowledge base system that was developed using Protégé as ontology and semantic rule editor.

1.4 Thesis layout

Chapter 1 provides the introduction to the opportunity addressed by the thesis. A general overview of the current environment, in which small and medium manufacturers operate, is presented. A brief account of industry 4.0, as the future trend in manufacturing, is provided. Also, the shortfalls of current manufacturing execution systems are pointed out. The research aim, objectives and methodology point out a proposed way to address a research gap.

Chapter 2 reviews the literature on the manufacturing network paradigm and the systems involved. A fundamental component of network configuration is scheduling. An overview of scheduling techniques is provided on analytical, heuristics, meta-heuristics and agent based systems as well as on some instances of integrated scheduling techniques that supported decentralised scheduling. The survey also elaborate on agent-based modelling as well as the enablers for multi-agent system implementation. Finally, a research gap is highlighted which the rest of thesis attempted to address.

Chapter 3 addresses the agent-based modelling of manufacturing network formation, scheduling and selection. An industrial use case and two flow shop scheduling case studies are presented, to define the scheduling environment. The chapter details the components that play a role in network configuration, two approaches to the formation and scheduling of networks and two approaches for the final selection of a network.

Chapter 4 presents the development of multiple simulation platforms. The manufacturer pairing approach was modelled as a decentralised genetic algorithm optimisation problem. For operation pairing, a multi-agent system and a knowledge base was developed to evaluate the local execution of distributed algorithms. Finally, a platform was developed to simulate a market-based environment for evaluating two proposed bidding protocols.

Chapter 5 presents the results and discussions about the proposed approaches to network configuration which consist of two pairing mechanisms and two bidding protocols. The results were compared to the benchmarks of the case study datasets, to evaluate the loss in optimality. Scenarios of disturbances were introduced, to evaluate the conflict resolution capabilities of proposed approaches, in terms of computation time and scalability.

Chapter 6 presents the concluding remarks of the thesis, with a summary of the research work, the aim and objectives, the contributions to the research field and some recommendations for future work.

LITERATURE REVIEW

2.1 Introduction

Due to volatility of the manufacturing industry, small and medium manufacturers may find some stability in participating in a manufacturing network breeding environment. The strength of the relationship between the manufacturers determine their preparedness to capture business opportunities as networked unit. A system of systems supports the environment in terms of configuration and coordination. One form of coupled configuration and coordination activity is scheduling. The techniques for solving scheduling problems fall under several categories. Some of them are more appropriate for the decentralised context set by manufacturing networks. Problem modelling and solving techniques are tightly coupled. The solving technique may also inform the solution implementation approach. Therefore this chapter presents the literature that helped define the flow shop scheduling problem of manufacturing networks as well as the modelling techniques for scheduling and the implementation opportunities for network scheduling. Finally, the review is concluded with the research gaps and a statement of purpose.

2.2 Manufacturing networks

For the last two decades, the study of manufacturing network has been a prominent field of research. A manufacturing network is a resource that exists when a group of manufacturers gather around a business opportunity and jointly coordinate their resources, skills and core competencies to achieve common goals. Due to the fact that networks consist of legally separate manufacturers, the latter can leave when there is no job and consequently reducing the production capacity of the network. This is a lean production concept (Womack and Jones, 1994).

2.2.1 Importance of participation

Manufacturers that participate in networks are autonomous, geographically dispersed and operate in a variety of environment and culture. It used to be that they join an association with a goal to gain access to resources and markets (Johanson and Mattsson, 1988), to be exposed to internationalisation, to acquire new knowledge and technology and learn best practices from other manufacturers (Chetty et al, 2000). Now it has become a culture in some regions of the world where manufacturers join an association backed by governmental measures and laws, as is the case in the Emilian model (Mosconi and Mantovi, 2012).

The need for participation in a network varies with time. Some manufacturers are fully equipped to serve the market at a particular point in time and are stable without a network. Some other times, when the market shifts, they may need to operate in sequence, dependent on outputs from others and contribute to the input of other manufacturers (Rudberg and Oulhager, 2003). To form part of a manufacturing network, manufacturers must have been in a cooperation agreement or association, long before they are asked to commit to a network. During downtime, they usually contribute to the steady ramping up of the infrastructure, enhancing their preparedness, for the moment when rapid network formation is solicited. This association of organisations is known as virtual organisation breeding environment (VBE) (Camarinha-Matos et al, 2009).

2.2.2 Organic growth of virtual organisation breeding environment

VBE is constantly evolving with new manufacturers integrating the association, strengthening and weakening relationships and new market opportunities being captured. It is important to introduce new partners as they provide competition within the VBE on lead time, price and capacity (D'amours et al, 1999). These are market qualifying criteria (Pires et al, 2001). Also, it encourages greater innovation performance (Thorgen, 2009). The entity that takes charge of the expansion of the VBE is called a network coach (Pluss et al, 2005). There are three levels of relationships which determines the preparedness of manufacturers to participate in a manufacturing network. In the first level, the manufacturer just joined and the transfer of knowledge is low and there is a general mistrust in the commitment of the new comer. The focus is on building confidence and contracting capacity. In the second level, the relationship is tested with short-term contracts in a manufacturing network. Intensive mutual learning takes place. In the third level, the manufacturer has standard network operating procedures, receives investments and key knowledge to tool up, and is fully committed to many manufacturing networks (Carbonara et al, 2002).

2.2.3 Support system for manufacturing network

A system integrator is a leader firm that acts as an interface between the VBE and the external customers (Danilovic and Winroth, 2005). It carries out the functions of sales and marketing on behalf of the VBE. It focuses on taking over the activities of the customers and building a service relationship with them (Spring and Araujo, 2013). A manufacturing network is formed from the initiative of a system integrator and therefore acts as the point of contact between manufacturing networks and customers (Pluss et al, 2005). The organisational arrangement is in such a way that power is symmetrically distributed among its members, where decision making is consensus based and where the nature of leadership is informal (Müller-Seitz and Sydow, 2012). There are four types of management control systems specific to a decentralized organization which are belief system, boundary system, diagnostic control system and interactive control system. Belief system is concerned with soft power such as mission statement and values while interactive control system is concerned with human intervention in the system (Karlos et al, 2011). Boundary systems represented explicit rules and boundaries for the behaviour of a network and its members. Examples of diagnostic control systems are task assignment, scheduling and fulfilling objective functions. The boundary and diagnostic control systems would work effectively if there is cognitive proximity in the network i.e. the network needs a medium through which members can exchange their piece of complementary knowledge. This medium would be in the form of a shared knowledge base (Li et al, 2013). An example of a widely accepted knowledgebase is the ontology.

2.2.4 Coordination mechanism of manufacturing networks

The configuration and coordination of networks, in which members operate in sequence and dependent on each other for complementary resources, have been a research issue (Fawcett et al, 1993; Rudberg et al, 2003). Rudberg et al. identified four basic network configurations namely plant, intra-firm (multi-site conglomerate), supply chain and inter-firm. Configuration and coordination are tightly related so that typical coordination mechanism can be identified as utilisation, optimisation, synchronisation and harmonisation (Cheng et al, 2011). The impact of information sharing on the time-price performance of network scheduling has been investigated (D'amours et al, 1999). In their studies, the authors investigate three bidding protocols that convey low, medium and high information intensive bids, depending on the relationship strength of the bidder with the system integrator. They bid on capacity and price, either as a one-off or on a daily basis. Better price-time trade-off performance was achieved with information intensive bids. The problem investigated consisted of single-product orders, linear sequencing, and no technical dependencies between products and imposed scheduled from system integrator. Another research has focused on the impact of VBE expansion on the speed of industrial innovation and innovation performance (Thorgen, 2009).

2.3 Finite capacity scheduling

Scheduling is usually concerned with the organization of tasks over time, against resources with finite capacity, in an environment that ranges from predictable to unpredictable states (Baptiste, 1996; Hermann, 2006). Much research has been focused on a set of scheduling problems that share some common properties. The first one is that the problems can be very complex for optimization. They have a complexity that is NP-hard which means that solutions are at best good approximations and the time frame required for optimal solutions to these problems is unfeasible (Chan & Chung, 2013). Next, there is room for improvement with new sets of constraints and objective criteria being considered. Finally, the problems are seldom static because the execution environment is changing frequently and can be unpredictable. Scheduling problems have been approached using tools and techniques broadly classifiable into four categories namely analytics, heuristics, meta-heuristics and agent based systems. The problem solving technologies and problem modelling techniques are tightly coupled.

Modelling enables a complex problem to be transformed into a format that is easier to solve with existing technologies (Simon, 1996).

2.3.1 Analytical approach

The analytical approach to scheduling has first come into use over 70 years ago and now involves matured techniques that are very popular with the operation research community. Spin-off techniques include Lagrangian relaxation, integer programming and constraint programming (Hermann, 2006). Combined with linear and non-linear programming, Lagrangian relaxation is aimed at reducing the complexity of a problem by relaxing the constraints of the solution space. Each relaxation instance is associated with a penalty cost called the Lagrange multiplier. The technique often was used in combination with problem decomposition into sub-problems, to increase a convergence to a feasible schedule that satisfied objective criteria (Luh et al, 1993; Liu et al, 1997; Jones et al, 1999). Integer programming for flow shop problems has been used for static and deterministic scheduling with common objective criteria of reducing make span, flow time and tardiness (Cheng et al, 2000; Tseng et al, 04). The key elements of a scheduling problem are generally constraints and objectives. Constraint programming is a technique of limiting the solution space of combinatorial optimization. Constraints can be temporal e.g. job process plan, capacity-related e.g. one job per machine at any time, and resource utilization related e.g. inventory depletion. A type of temporal constraint is disjunctive constraint which represent a condition where two entities sharing one resource cannot utilize the resource simultaneously. This was the premise for constraint programming in solving a couple of flow shop scheduling problems including MT10 and LA19 (Baptiste, 1996).

2.3.2 Heuristics approach

The design of heuristics has been subject to a significant battery of research (Morton and Pentico, 1993) and heuristics are consistently being used in the approximation of solutions for complex scheduling problems in real time. Heuristics are also known as dispatching or sequencing rules and include simple rules and combination of rules. Rule combinations can be weighted or un-weighted in order to vary the sensitivity of the rules to the requirements of the jobs. This is because simple rules have a myopic tendencies and may underperform with some job requirements (Morton and Pentico, 1993). Some studies have looked at executing different rules in parallel and building intermediate results from the cross-feeds of approximated data (Gones and Selman, 2001). The cornerstone rules include shortest processing time (SPT), earliest due dates (EDD), minimum slack, arrival times (FIFO) (Jones et al, 1999) and the Johnson's rule for 2-machine flow shop (Kurz and Askin, 2003).

2.3.3 Meta heuristics

Some scheduling problems are NP-hard and an exact solution cannot be achieved within a finite timeframe. Therefore, approximation techniques such as search algorithm and evolutionary algorithms have helped to alleviate some weaknesses of simple heuristics. Search techniques include branch and bound, hill climbing, simulated annealing and Tabu search (Brandimarte, 1993). Evolutionary techniques include genetic algorithms and ant colony optimization. However, search techniques need a wider time frame to operate and suffer from solution latency. To solve this issue, studies have used heuristics combined with meta-heuristics to guide the solution search process. This is often referred to as limited discrepancy search (Cicirello and Smith, 2002). Genetic algorithm applied to dynamic scheduling has shown better results than common dispatching rules (Chryssoluris et al, 2001). Evolutionary algorithms have been most effective at problem solving than other meta-heuristic search algorithms. The algorithm was used to minimize make span on n-job, m-machine flow shop sequencing problems and outperformed neighbourhood search and simulated annealing techniques (Reeves, 1995). In another study of resource-constrained scheduling problem, ant colony optimization, variations of genetic algorithm and simulated annealing have been compared on the basis of the standard deviation from the lower bound of make span results. The evolutionary algorithms outperformed simulated annealing. Genetic algorithm performed just as well as ant colony but with a decimal point higher deviation (Merkle et al. 2002).

2.3.4 Agent based systems

On one hand, if multi-agent system is used as enabler, simpler software could be designed for each agent and new agents would easily be integrated into the existing software network. On the other hand, knowledge-based system has huge potential in automating reasoning in a boundary system of soft and hard constraints. However, both multi-agent systems and knowledge based systems lack the ability to optimize scheduling objectives as well as a centralized approach (Ouelhadi and Petrovic, 2009). For scheduling activities to be delegated, there are three main components required, namely interaction protocol, negotiation mechanism and an inference system. The interaction protocol governs the timing and structure of data exchange between agents. Contract net protocol and modified ring protocol are useful examples (Owliya et al, 2013; Jules et al, 2015). The negotiation mechanisms can be categorized into market-based or threshold-based approaches. Market-based approach caters for agents with self-interested goals. Agents compete and are rewarded if they exhibit desirable system-wide behaviours. Threshold-based approach is based on the probability of an agent to accept a preferred type of task when some events take place. However, market-based negotiation which is a direct negotiation mechanism has problems with communication overhead due to the constant exchanges of bids and the processing of those bids (Shen, 2002; Goldingay and Van Mourik, 2013). Threshold-based mechanism which is often associated with indirect negotiation mechanisms such as stigmergy and bio-inspired coordination, do not suffer from communication scalability issues. The knowledge would be in the form of pheromone type traces in the case of threshold-based mechanism and in the case of market-based mechanism, knowledge would come from agent bids. Agents would operate within a context bounded by rules, implicit data and inferred data (Yilmaz, 2012).

2.3.5 Modelling techniques

The method of optimisation that is used, dictates the modelling approach of a scheduling problem. Mixed integer programming (MIP) formulations are prominently used for scheduling problem models (Harjunkoski et al, 2014). There are successful migration of models into industrial implementations. For instance, MIP solvers such as CPLEX® and GAMS have been integrated with relational databases and reporting software suites such as SAP-APO, to carry out routine schedule optimisation at an industrial scale (Lin et al, 2002). Generalised disjunctive program (GDP) was another form of representation of scheduling problems in terms of constraints and logical formulations. GDP would be converted into algebraic modelling language (AML) before being solved by a MIP solver (Castro et al, 2012). The next section reviews in more details, various approaches to manufacturing scheduling including analytic, heuristics and agent-based techniques.

2.3.6 Decentralisation as the default approach

The case for decentralised scheduling is supported on the basis of two reports from the consulting companies; McKinsey, and American Productivity and Quality Council (APQC). McKinsey (Campbell et al, 2011) proposed decentralisation as the default organisation structure unless one of three criteria is met. First criterion states that unless centralisation is mandated by law or external stakeholders, decentralisation is preferred. Second criterion states that if centralisation increase value by less than 10%, then decentralisation is preferred. The final criterion is concerned with the risks of increased bureaucracy, increased business rigidity and withered motivation. If implementing centralisation could not reduce the risks, then decentralisation is preferred. Based on a survey of 96 manufacturers, production schedule reliability was at most 5% better for centralisation compared to decentralisation (APQC, 2010). This fact was used to support the decision that production scheduling did not need to be centralised when the production sites were inherently decentralised.

2.3.7 Decentralised approach to scheduling

Yimer et al. (2010) proposed a two-phase mixed integer linear programming (MILP) model. The model looks a problem as a set of sub-systems that can be solved in a parallel and sequential manner. Each sub-system performed its own genetic algorithm optimisation. Thomas et al. (2013) presented a distributed mechanism to minimize three weighted objectives of tardiness, earliness and cost based on Lagrangian relaxation (LR), Volume and Wedelin algorithms. Acting like a shock absorber, the Volume algorithm was used to dampen resource-constraint violations, thus helping a faster convergence. The research reported a problem solving rate of 82.5% with distributed LR-based approach compared to 46% for a centralised MILP model (Thomas, 2014). A coordination mechanism was designed in Arena simulation platform according to the multi-agent paradigm (Renna, 2011). The decentralised mechanism allowed the simulated manufacturing system to achieve good performance under test conditions such as high/low workload, normal/rush due date, part-mix changes and arrival time changes. Another example of decentralisation in manufacturing is in revenue share negotiation between partners. Taghipour et al. (2013) proposed a dynamic mutual adjustment search heuristic. The concept consisted of a maximum discount plan (MDP) which incentivises manufacturers to compromise their capacity utilisation for better revenue on a manufacturing job. The simulated model of distributed incentives showed 9% profit improvement for partners and simplified job allocation. Baffo et al. (2013) proposed the Cascade Flow Shop (CFS) model which is decentralized so that several decision makers play their role in job scheduling and timing. Their model consisted of localized problem solving and downstream solution communications. The model was written in Algebraic Modelling Language (AML) intended for mainframe computing.

2.4 Agent-based modelling techniques

Multi-agent system can be modelled as an information exchange model and/or an optimisation model. Information exchange model is about the flux and evolution of information whereas optimisation model is about the system and their agents' objective functions (Nedic and Ozdaglar, 2009). Multi-agent systems need to be augmented with optimisation algorithms if problem solving capability is required. Agent-based modelling is the prototyping of the functions of a multi-agent system (Roorda et al, 2009). Multi-agent system can help solve naturally distributed problems that require an array of computation entities (Wang et al, 2009).

2.4.1 Framework for agent identification

The intelligent manufacturing system initiative (IMS) has introduced a conceptual framework for mapping legacy manufacturing systems, so as to transform or develop new systems for the dynamic manufacturing environment (Van Brussel et al, 1998). The initiative proposed the product, resource, order and staff architecture, known as PROSA, as part of a new holonic manufacturing system paradigm (HMS). HMS describes a group of software that autonomously cope with unforeseen disturbances, without having to wait for instructions from higher authorities (Leitão et al, 2013). The principle prescribes that manufacturing systems can be made up of four main entities that are mandatorily autonomous, cooperative and self-organised (Bussmann, 1998). Autonomy means that agents have control over their plans and actions. Cooperation is supported by agent interactions and negotiations for resources. A system is said to have strong self-organisation when it can reconfigure without central planning, compared to weak self-organisation which requires explicit central planning (Serugendo, 2006). The entity, called the holon, is self-contained with data input and output as well as a processing unit. The entities can specialise further to model a specific component of the manufacturing system. Systems of holons are also called holons and the structure that holds them together, is called a holarchy (Van Brussel et al, 1998, Giret and Botti, 2009). A holarchy represents an organisational duality of heterarchy and hierarchy. Heterarchy is a flat and distributed structure that gives flexibility to the system and empower entities have an equal stance in negotiation, cooperation and decision making. Hierarchy gives entities, in the upper levels, decision making power to pass down instructions to lower level entities. The four main entities are product, resource, order and staff holons. A product holon contains the product life cycle information, bill of materials, process plans and quality certification procedures. The order holon is the operation to be performed on time and against customer objectives. The resource holon is the facility offering production capability and also represents the component being produced (Van Brussel et al, 1998). The field of holonic manufacturing is key to decentralised manufacturing. Decoupling of system structure and control algorithm allow an agent interaction protocol to be independent of the proprietary programming language of individual resources. It helps towards making resources, pluggable into a multi-agent system, without installation downtime. Central to HMS, is the intensive reuse of sub-systems. This means that resource, order and product agents can be created and reused infinitely and if a sub system fails, it can easily be replaced (Van Brussel et al, 1998).

2.4.2 Configuration mechanism for multi-agent systems

Multi-agent system in manufacturing faces some challenges regarding its adoption by companies (Leitao, 2009). It was pointed out that a step change is needed in current reconfiguration mechanisms towards intelligent self-organisation (Oh and Smith, 2004). Fuelled by growing demands for reconfigurable manufacturing systems, recent literature has focused on functional developments. Owliya et al. (2013) investigated various interaction protocols governing agent communication patterns and developed a modified ring protocol for unsupervised task allocation in shop floors. The results showed a decrease in make span and improved utilization performance under the modified ring protocol compared to peer-to-peer and contract net protocol, even under rush task scenarios. Jules et al. (2015) investigated a modified Contract Net Protocol (CNP) for the formation of a collaborative network organization. The added interaction features include competency matchmaking, call for participation, combinatorial network formation and capacity release. These subsystems were individually optimised to improve the metrics of decentralised job allocation. A bio-inspired interaction mechanism was investigated for routing solutions in a flexible manufacturing system (Leitão et al, 2012). The mechanism used a concept of potential fields which are either attractive or repulsive fields, used to control behaviours of the system. The potential field is formulated as a matrix that correlates a service and its availability. It is claimed that better responsiveness of resources reallocation is achieved, when the potential field strengthens or weakens. Lim et al. (2013) used a currency-based iterative bidding mechanism, to facilitate the coordination of agents across geographically distributed facilities. The right currency incentive can be easily determined using Genetic Algorithm search and historical currency values. The coordination of agents can help to obtain optimised process plans and schedules by using a five steps approach of agent interaction. The goal was to optimise the cost-effectiveness of the solution. Aissani et al. (2012) proposed an online system to optimise scheduling across different sites using intelligent agents with reinforcement learning. They also tested a Mixed Integer Linear Program (MILP) as well as a Genetic Algorithm model of the multi-site problem. It was reported that the multi-agent system approach outperformed GA and MILP, in terms of project final date and computation times. Adhau et al. (2012) presented a multi-unit combinatorial auction mechanism to solve resource allocation negotiation. It allowed NP-hard scheduling problems to be approximately solved in six steps. They are initialisation, virtual and utility calculations, bid generations, provisional winner determination, bid modification and resource allocation. The mechanism can handle problems of any size regardless of the number of projects, activities and resources. Therefore, reconfiguration mechanism has received good research attention and is recognised as an important prerequisite to achieve autonomous and self-organisation behaviour in a system.

2.4.3 Agent incentive mechanism

Evolution and adaption in a distributed system of agents may trigger an endless cycle of chaotic behaviours. Barbosa et al. (2015) proposed a two layer stabilisation approach, for a system of self-organising agents, to reduce the nervous impulse of agents when reacting to perturbations. The research used a proportional, integrative and derivative (PID) controller derived from classical control theory. Applied to a manufacturing case study, a reduction in make-span performance degradation for behavioural self-organisation and a reduction in transportation times for structural self-organisation, were reported. Wooldridge et al. (2013) proposed a taxation scheme to impose different levels of costs on various agent actions, while the agent seeks to minimize its expenditures. This mechanism can provide an incentive for an agent to steer clear of some actions or steer towards some actions with respect to its goals. The social welfare of the system measures how well agents had their goals met. The notion of utilitarian social welfare is the sum of utilities of agents. In their work, the utilities were taxes.

Nguyen et al. (2014) performed a computational complexity survey on social welfare optimisation namely utilitarian, egalitarian and the Nash product. It was reported that on all three notions, the complexity of optimisation is NP-complete. In other words, an exact solution can be achieved but there is no known algorithm that can efficiently solve the problem. Therefore, the computation time significantly increases with the size of the problem.

To solve the highly complex utility space with improved efficiency, Fujita et al. (2014) proposed a mechanism to decompose the problem into distributed agents which, based on compatible issues, locally establish relationships with other agents, to form issue clusters. A mediator aggregates the clusters into issue groups, which undergo nonlinear optimisation, to produce the final solution. A measurement was proposed for issue interdependency strength, optimality rate of issue grouping and quality factor. They use centralised simulated annealing as control method. When the number of issues increased, the differential gradient of optimality rate as well as the quality factor improved.

2.4.4 Migration into multi-agent systems

Komma et al. (2011) developed an agent-based shop floor simulator for the manufacturing domain, modelling agents such as automated guided vehicle agent, machine-agent and part-agent. The components of the framework involved a knowledge base, reasoning capabilities and agent behaviours. The dispatch algorithm used, in part agents was 'first-come-first-serve'. The work was developed on the Java Development Framework (JADE). Barbosa et al. (2015) implemented stigmergy in the ADACOR multi-agent system, which allowed agents to pick upon message trails left in the environment. These messages signalled a plan deviation and an opportunity for self-reconfiguration. ADACOR is also built on the JADE infrastructure. Vrba et al. (2010) presented MAST which is a multi-agent system capable of structural reconfiguration when the layout of the factory floor changes. The factory consisted of a system of conveyor. The disturbance is simulated as a failed conveyor. The system reconfigured the virtual map and automatically searched for the shortest path for the product to reach its destination.

2.5 Opportunities from cloud manufacturing

Cloud manufacturing is a new paradigm that can enable full sharing of manufacturing resources across geographically distributed locations, resulting into Manufacturing as a Service (Montreuil et al, 2000). Moreover, immense computing power is at disposal for web services to utilise (Smith et al, 2002). This paradigm can also allow manufacturing systems to leverage collective expertise of large groups of people and communities to cost-effectively develop algorithms for problem solving of various complexities (Zhang et al, 2014). Moreover, the security developments, infrastructure maintenance and data loss safe guards are performed by cloud utility providers (Zhang et al, 2014). Cloud services can flexibly support the needs of manufacturers throughout the product life cycle including design, manufacturing, quality control and management (Saeidlou et al, 2014).

2.5.1 Application of ontology in manufacturing

An ontology is a type of knowledge base which allow data, objects and the relationship among them to be structured according to a set of international standards (Schalkoff, 2011). These standards ensure that the ontology can provide distinct expressions that are universally recognisable but also that reduces the mismatch between the model and the actual entity (Terlouw et al, 2013). Functionalities for mapping, aligning and merging multiple ontologies are being improved so that existing ontologies can be upgraded with ad hoc ontologies for specific applications (Hongbo et al, 2013). Kotulski et al. (2014) presented a graph transformation system for handling the storage and exchange of knowledge between agents. Due to the increasing complexity of data that needs to be handled and the increasing workload imposed by multi-agent systems, the authors are developing algorithms to maintain graph cohesion and to speed up graph processing.

Ontology has been used to capture the domain knowledge from shop floors (Lin, 2011), from production control and logistics (Jiang et al, 2010) and from customer, product and transport (Yan et al, 2010). To automate the estimation of the manufacturing cost of a product, MASON was developed to model the entities of the manufacturing domain that were relevant to costing (Lemaignan et al, 2006). To increase the effectiveness of inter-firm collaboration, PABADIS was developed as a comprehensive ontology modelling the manufacturing capability domain. (PABADISPROMISE, 2006). Based on the holonic manufacturing system paradigm, ADACOR was an ontology that modelled the control knowledge of a manufacturing system (Borgo and Leitão, 2004). Ontology follows a set of standards that enables its use in web applications. The production control web interface was developed to enable production devices to be modelled, their services invoked and their status updated (Puttonen et al, 2013). Wang et al. (2014) proposed a framework for expanding a cloud manufacturing (CMfg) task ontology. The framework consisted of three stages. The first stage was the application of sophisticated analytic on new task documents which prepared CMfg ontology for new data structure. Second, an ontology template was instantiated with real time task data. And third, via similarity analysis, the instantiated ontology was merged into CMfg ontology. To better understand the decision making process in the urban goods movement that is damaging the environment, Anand et al. (2014) developed an agent based model. The authors identified the stakeholder agents and their interactions and formed a model that worked in tandem with a knowledge base that represented the city logistics domain. Model simulation enabled the authors to understand how to consolidate goods and coordination of different types of goods movers to improve efficiency and reduce the environmental downsides of logistics. Companies are also developing their ontologies to increase the visibility of their services in the multi-agent environment (Wenyu et al, 2014).

2.5.2 Enabling cloud-based scheduling for manufacturing

Knowledge-based scheduling forms part of the core cloud service layer proposed in the cloud manufacturing architecture (Tao et al, 2011). Current efforts for scheduling in the cloud have used a system of data capture, cloud-based relational database and tools consisting of multi-objective optimization such as Monte Carlo simulation (Guo et al, 2015). Scheduling for manufacturing has not been fundamentally adapted for the cloud space where it can benefit from flexible data storage, fast data query and manipulation. Disjunctive graph is an existing representation where entities of production scheduling and their relationships can intuitively be modelled (Roy et al, 1964). Disjunctive graph can now be stored in graph databases. When dealing with linked data, studies have shown that data query on graph database is 20 to 30 fold faster than relational database. Also, graph database can be updated with new relationships by a simple addition of nodes and edges, without restructuring an entire schema (Batra et al, 2012). Disjunctive graphs consist of facts (i.e. raw data and relationships) that can best be stored in a graph database. This brings to mind manufacturing process plans which can benefit from this alternative. Fitting raw data with all their relationships on an ever-growing scale in a relational database can become a kludge.

2.5.3 Semantic reasoning algorithms

Graph database also stores rule-based representation which enables semantic reasoners to perform in-database inferences. The advent of state-of-the-art deductive mechanisms have enabled logic representation to be reliably interpreted. Chaining is a deductive mechanism that makes implicit knowledge explicit and therefore inferred data becomes query-able (Perez Urbina et al, 2012). There are forward, backward and bi-directional chaining used in engines such as Drools and Prolog based on the Rete algorithm (Kaiser et al. 2012). Tableaux algorithm, nominal absorption and nominal-based model merging constitute deductive mechanism of state-of-the-art inference engines such as Pellet and Stardog. Additional optimization techniques have endowed both inference engines with incremental reasoning capabilities. This is useful in a manufacturing context where there is a constant knowledge flux (Numao, 1994). Both inference engines can reason about knowledge consisting of concepts and facts written in ontology web language (OWL) and rules written in full semantic web rule language (SWRL) (Sirin et al, 2007). An aggregation of a mechanism, concept definitions and rule-based statements form an expert system (Genmari et al, 2003).

2.6 State of the art and research tools

The merits of the recursive porous agent simulation toolkit (REPAST) in the manufacturing domain have been conclusive especially when modelling distributed decision making, time scheduling and networks (Owliya et al, 2013). Moreover, the platform provides facilities for data collection, visualisation as well as an array of useful optimisation algorithms which outweigh similar platforms such as MASON, NetLogo and Swarm (North et al, 2013). Furthermore, REPAST is versatile in applications ranging from industrial analysis, to social systems and evolutionary systems (North et al, 2013). Workflows and Agent Development Environment (WADE) is the next generation of JADE. Coupled with Workflow Lifecycle Management Environment (WOLF), WADE allows scalable software systems to be visually programmed using workflows, actors, tasks, activities and relationships. Fundamentally, WADE enables the development of decentralized agents with unique behaviours which can send and receive synchronous and asynchronous messages, request agent services from the directory facilitator and event listening. WADE has been used in mission critical applications notably on projects for Telecom Italia (Bergenti et al. 2012). Conceptbase, Protégé (Gennari et al, 2003), Racer and Pellet are tools for storing knowledge and inferring new ones using logical reasoning. Ludwig drew a comparison between Deductive Database System and Semantic Web reasoning (Ludwig 2010). Though he demonstrated that in-database analytics is faster than semantic web reasoning, the file format used by Protégé, Racer and Pellet is more portable and particularly designed to be easily stored and retrieved from the Web. Also the reasoner Pellet is a mature software with Pellet API bindings for OWL API, the semantic file format, (Sirin et al. 2007) which enables its practical implementation in multi-agent systems.

2.7 Identification of research gap

Mixed integer linear programming (MILP) is commonly used for the modelling and simulation of manufacturing scheduling. However, there is not much evidence of MILP suitability for cloud-based application. There is a gap for the use of disjunctive graphs in manufacturing scheduling. Graphs can now be expressed in ontology web language and disjunctive constraints written in semantic web rule language. Graphs are already in format for implementation in dedicated cloud infrastructures such as graph databases. The literature review revealed the existence of manufacturing ontologies, however their scopes were not focused on the scheduling aspect of manufacturing. A focused ontology is important and can eventually be merged with other existing ontologies to widen its scope. There is also a gap for a comprehensive framework to investigate flow shop scheduling of manufacturing networks that combines agent-based modelling, ontology building and multi-agent system implementation. Such a framework would support the development of a scheduling ontology and new algorithms for agent-based systems. Research in the scalability of decentralised scheduling in manufacturing networks using market-based mechanisms, has been partially addressed and there is a gap for research in mechanisms enabling the emergence of a multi-manufacturer operation plan from a collection of simple local behaviours. This research attempted to investigate the formation of manufacturing networks, in which manufacturers have absolute control over their scheduling activities, followed by final network selection by market-based approach.

MODELLING OF NETWORK CONFIGURATION

3.1 Introduction

The flow shop is a special case of job shop systems where each job has a fixed process plan. The finite scheduling process that takes place in a flow shop system is called flow shop scheduling (Reeves, 1995). It is an optimisation problem where a job needs to be allocated to several resources according to its process plan. The MT10 and LA19 are instances of such scheduling problems that consist of unique jobs, unique operations and unique resources (Baptiste, 1996). Typically, jobs have process plans in which operations are technically inter-dependent. MT10 and LA19 were used in this research as case studies. The case studies of MT10 and LA19 have been used for the main reason that the optimum lead times of both are known from literature. The MT10 case study has been cited in more than 600 publications. LA19 is a similar sized problem that was demonstrated by SAS™. The optimum lead times were used as benchmarks to verify results and to determine the loss in optimality. For MT10 and LA19, the optimum lead time is known to be 930 (Park et al, 2003) and 842 (SAS, 2009) respectively. Also presented, was an industrial use case of a manufacturing network breeding environment (MNBE) that was modelled as a flow shop system. MNBE is defined as being fundamentally a virtual organisation breeding environment (Camarinha-Matos et al, 2009) for manufacturing. The process of flow shop scheduling consists of two main activities namely sequencing and timing which constituted the scope of the research. Sequencing is the decision making process of determining the operation that will be performed next and timing attributes a start time and a finish time to an operation.

The context, established by the industrial use case, led to the investigation of scheduling approaches for decentralisation. Decentralisation of scheduling can empower manufacturers to act as decision makers (Lasi et al, 2013) and play an important role in the sequencing and timing of manufacturing operations. Scheduling problems which consists of multiple inter-dependencies can be decomposed into self-contained sub-problems. Manufacturers would initiate problem solving procedures locally and would communicate the consequences of their solutions to their affected peers. The latter would make adjustments accordingly, negotiate or would refuse adjustments. This interaction among entities can cause the emergence of a multi-manufacturer operation schedule, from a collection of simple individual behaviours. In this research, the agent-based approach was used to emulate those manufacturers and simulate decentralised flow shop scheduling. As mentioned before, flow shop could consist of identical resources, in which case, multiple manufacturing networks could have prepared for the same job. For each job, only one network would be selected, on the basis of some criteria.

Agent-based modelling was chosen as the method to investigate network configuration by market-based mechanism where manufacturers control their own schedules. There is one main reason why this method is more beneficial to this investigation than other methods such as discrete event simulation (Law, 2007) and quantitative modelling techniques namely mixed integer linear programming (Aissani et al, 2012) and general algebraic modelling system (Lin et al, 2002). The reason resides in the useful characteristics of the modelling approach. First, the entities being investigated are self-contained (Macal and North, 2009). They have interlinked attributes that allow the local information about and around the entity to be navigable. Heuristics, as simple as 'if-then' rules and more complex rules, such as meta-heuristics, allow the attributes to control the actions of each individual entity. Second, the entities are modular (Macal and North, 2009). They may be made up of other entities. They follow strict rules of membership, in order to achieve desired end states. The rules may evolve when new memberships are created. The attributes and behaviours are inherited, become linked and navigable. The entities become one aggregated agent. Third, the entities are social (Macal and North, 2009). Their repeated interactions enable aggregation of entities, new behaviours to form and new attributes to be linked. The interactions have rules and attributes, just like agents, that maintain the scalability of the interacting system. The rules prevent superfluous interactions between agents. The attributes make the interactions tractable and available for analysis. In the context of decentralisation, manufacturing resources, devoid of centralised control, can be represented in terms of agents. Resources must adapt their behaviours when the system evolves and when disruptions occur. Resources must interact with other resources, strategically and not randomly. In order to prevent chaos, resources must form strategic organisations that learn and adapt at that level. Decentralisation often connotes the need for good scalability as the resources must temporarily plug into any production systems and operate. Therefore, agent-based modelling provides a natural representation of decentralised manufacturing.

This chapter presents the theoretical part of the research whereby active/passive agents and their knowledge requirements were identified and modelled. Several agent interaction techniques were proposed and their algorithms developed, to be later simulated within an agent-based environment and evaluated against the benchmarks.

3.2 Problem statement

3.2.1 Industrial use case

GFM s.r.l is a small and medium enterprise (SME) and has a strong market hold in the provision of mechanical engineering services. Its core market focus is the energy industry, providing custom-built steam and gas turbine components. With the ability to offer micro tolerance to multi-ton machining ISO-certified capability, it is also emerging in the aerospace, naval, oil and gas industry. It is also an original equipment manufacturer (OEM) and forms part of supply chains of market leading companies notably Siemens, Mapna and Ansaldo.

The company has an entrepreneurial background which started with the design of special equipment and machining. During its 40 years of existence, GFM established partnerships with more than 30 SME manufacturers and 500 suppliers. The latter are usually family-owned or cooperatives and are efficient and flexible at scale, innovative, independent and technologically specialised. Most of them are based in the northern Italian region. GFM eventually evolved into a company focusing on upstream and downstream business activities and for some time, outsourced all its production processes to its network of partners. Its business activities include customer service, engineering design, marketing, planning, procurement, purchasing, quality control, information technology, warehousing and logistics. However, it recently acquired a production facility, to add research and development (R&D)

and reinstate production to its expertise portfolio. Today, the main role of the company is service brokering and it acts as a single point of contact for customers (Danilovic and Winroth, 2005), leveraging a network of hundreds of manufacturing operation capabilities, at least a thousand technicians and more than a million operating hours per year. Figure 3.1 is a depiction of the activities performed by GFM and the departments involved when a customer order is placed.

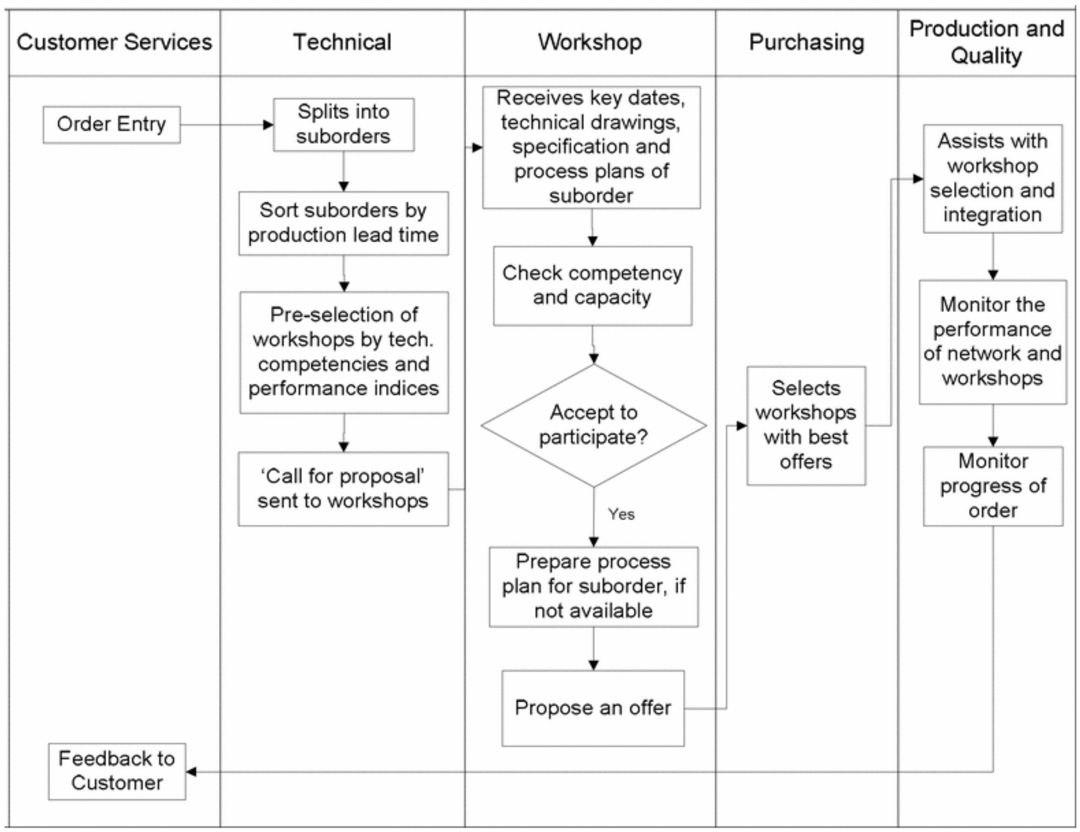


Figure 3.1: Flow of order processing in a manufacturing network breeding environment

Over the years, GFM and its SME partners have developed strong partnerships. There exists at least two manufacturers with similar competencies. This introduces an element of redundancy in the network, to encourage competition, among partners, on the basis of several criteria. Criteria includes trust, responsiveness, and lead time of remedial actions as well as product lead time, price and quality. Strong partnership goes beyond basic procurement, to include technology transfer, training and financial support. Strong partners operate on principles of reciprocity and weak partnerships operate on contract and profit principles. GFM acts as a provider of systemic support for co-operative and non-cooperative organisation, feeding global knowledge into local production and helping SMEs to compete on the global stage. Such global knowledge include technology developments, market trends and manufacturing best practices. GFM brings to a manufacturing network, the strengths of large

Table 3.1: Quantitative description of the industrial use case

Parameters	**Value**
Number of jobs per day	20 jobs
Number of manufacturers	30 manufacturers
Number of unique operations	45 operations
Number of manufacturers offering similar operations	At least two manufactures
Process plan maximum length	4 operations
Average operation processing time	10 hours
Processing time standard deviation	0.5 hours
Average capacity available per day	50 hours
Operation capacity standard deviation	10 hours
Average operation cost rate	£50/hour
Operation cost rate standard deviation	£10/hour
Average number of defects	100 dppm
Operation defect standard deviation	50 dppm
Lowest overhead cost per day	£500
Highest overhead cost per day	£10000

corporate structures, while respecting the strengths of SMEs. The characteristics of GFM and the manufacturing network breeding environment were quantified as shown in Table 3.1. In the scope of manufacturing operation scheduling, GFM and its partners present a good case for decentralised scheduling for the following reasons:

- First, in the spirit of cooperation, GFM cannot impose a schedule onto its partners.
- Second, the partners are independent and do not produce exclusively for GFM. Therefore they have other customers to consider, when agreeing on a schedule.

- Third, companies can be more responsive by solving scheduling disturbances themselves and communicating the solutions to affected partners. GFM can then focus on settling penalties incurred by the partners.

3.2.2 Flow shop scheduling problem case studies

The model was designed around flow shop scheduling problems and a specific case study format. An original case study consists of job numbers, machine numbers, the number of machines and the number of jobs. Each job has a row of machine numbers and a column represents a production step. Each step is performed by a machine with a processing time. The case studies used are the MT10 (Muth and Thompson) and LA19 (Lawrence) flow shop scheduling problem which are available from Universidad de Valladolid, school of industrial engineering (http://bit.ly/1JbNlTp). However, the proposed model was designed to accommodate a wider range of case studies, which are not limited to 10x10 data sets.

For the scope of the research, which investigated manufacturing networks, the structure of the data sets was modified to include operation and manufacturers as shown in Table 3.2, 3.3. The data itself was not tampered with, so that results could be validated against the benchmarks of the operation research community. Each production step was given a unique operation number which was the combination of a job number and a machine number. Then, the notion of machine was generalised as a type of resource which was re-specialised as a manufacturer. The process plan of a job with identity 1 would originally appear as {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} where the digits respectively represent individual manufacturer identity. In the modified version, Job 1 needs not be mentioned because {10, 11, 12, 13, 14, 15, 16, 17, 18, 19} implies the process plan of Job 1. Another example is {20, 22, 24, 29, 23, 21, 26, 25, 27, 28} which is the process plan of Job 2. In that structure, the data set consisted of 10 manufacturers, 10 jobs and 100 unique operations where each manufacturer offered 10 unique operations.

Table 3.2: Job process plans from MT10 problem

Job	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
1	10 (29)	11 (78)	12 (9)	13 (36)	14 (49)	15 (11)	16 (62)	17 (56)	18 (44)	19 (21)
2	20 (43)	22 (90)	24 (75)	29 (11)	23 (69)	21 (28)	26 (46)	25 (46)	27 (72)	28 (30)
3	31 (91)	30 (85)	33 (39)	32 (74)	38 (90)	35 (10)	37 (12)	36 (89)	39 (45)	34 (33)
4	41 (81)	42 (95)	40 (71)	44 (99)	46 (9)	48 (52)	47 (85)	43 (98)	49 (22)	45 (43)
5	52 (14)	50 (6)	51 (22)	55 (61)	53 (26)	54 (69)	58 (21)	57 (49)	59 (72)	56 (53)
6	62 (84)	61 (2)	65 (52)	63 (95)	68 (48)	69 (72)	60 (47)	66 (65)	64 (6)	67 (25)
7	71 (46)	70 (37)	73 (61)	72 (13)	76 (32)	75 (21)	79 (32)	78 (89)	77 (30)	74 (55)
8	82 (31)	80 (86)	81 (46)	85 (74)	84 (32)	86 (88)	88 (19)	89 (48)	87 (36)	83 (79)
9	90 (76)	91 (69)	93 (76)	95 (51)	92 (85)	99 (11)	96 (40)	97 (89)	94 (26)	98 (74)
10	101 (85)	100 (13)	102 (61)	106 (7)	108 (64)	109 (76)	105 (47)	103 (52)	104 (90)	107 (45)

S1 = first step, 90 = needed by Job 9, provided by Manufacturer 0, (76) = operation processing time

Table 3.3: Job process plans from LA19 problem

Job	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
1	12 (44)	13 (5)	15 (58)	14 (97)	10 (9)	17 (84)	18 (77)	19 (96)	11 (58)	16 (89)
2	24 (15)	27 (31)	21 (87)	28 (57)	20 (77)	23 (85)	22 (81)	25 (39)	29 (73)	26 (21)
3	39 (82)	36 (22)	34 (10)	33 (70)	31 (49)	30 (40)	38 (34)	32 (48)	37 (80)	35 (71)
4	41 (91)	42 (17)	47 (62)	45 (75)	48 (47)	44 (11)	43 (7)	46 (72)	49 (35)	40 (55)
5	56 (71)	51 (90)	53 (75)	50 (64)	52 (94)	58 (15)	54 (12)	57 (67)	59 (20)	55 (50)
6	67 (70)	65 (93)	68 (77)	62 (29)	64 (58)	66 (93)	63 (68)	61 (57)	69 (7)	60 (52)
7	76 (87)	71 (63)	74 (26)	75 (6)	72 (82)	73 (27)	77 (56)	78 (48)	79 (36)	70 (95)
8	80 (36)	85 (15)	88 (41)	89 (78)	83 (76)	86 (84)	84 (30)	87 (76)	82 (36)	81 (8)
9	95 (8)	92 (81)	93 (13)	96 (82)	94 (54)	97 (13)	98 (29)	99 (40)	91 (78)	90 (75)
10	109 (88)	104 (54)	106 (64)	107 (32)	100 (52)	102 (6)	108 (54)	105 (82)	103 (6)	101 (26)

S1 = first step, 90 = needed by Job 9, provided by Manufacturer 0, (75) = operation processing time

3.3 Modelling of networks as flow shop systems

Part of the research aim was to schedule a flow shop system, where manufacturers could control their scheduling activities. A generic flow shop system consists of a set of jobs and a set of manufacturers and more than one manufacturer are capable of fulfilling the operations required by the job. For instance, given a job requiring a welding operation, there may be several manufacturers with welding capabilities and enough capacity to fulfil the operation. Therefore, flow shop scheduling may result in more than one manufacturing network, for each job.

Decentralised scheduling could give manufacturers the autonomy to create their own operation plans as long as the loss in schedule optimality is kept to a minimum. As mentioned before, flow shop scheduling is made up of sequencing and timing. Sequencing determines which operation will be performed next. Operation plans are generated from the sequencing of operations. At this point, any operation is bounded by a job process plan as well as a manufacturer operation plan. Timing of operations is then carried out within those boundaries. The research proposed two approaches of decentralised flow shop sequencing. The first approach was manufacturer pairing with genetic algorithm (GA) optimisation. The second approach is operation pairing.

The first approach, manufacturer pairing, was tested in a recursive porous agent simulation toolkit also known as Repast. The toolkit is an agent based modelling and simulation platform that provides an array of optimisation techniques, one of which is GA optimisation, with comprehensive tutorials on how to use them. Repast also facilitates the collection and visualisation of simulation results. Repast carries benefits that outweighs toolkits like

MASON, NetLogo and Swarm. The second approach, operation pairing, was tested within a knowledge-based multi-agent system. Workflow and agents development environment (WADE) was used to develop the multi agent system and Protégé, an ontology and semantic web rule language editor was used to construct the knowledge base. With WADE and Protégé, it was possible to bridge the gap between an agent-based model and an actual implementation of a multi-agent system.

The entities that were relevant to the two sequencing approaches, the use case and the two case studies, were identified and their data structure were modelled. They are presented in Table 3.4. The knowledge of the entities was modelled as a set of relationships that interlinked them as well as a set of rules that calculated their attribute data. Both the approaches of manufacturing pairing and operation pairing involved the pairing of entities into new objects. The organisation rules that underpinned both approaches were identified and modelled.

Table 3.4: Nomenclature of agent properties

Entity	Attributes
Job (j)	(dt) Due time
	($o_{j,i} \in O_j$) Set of operations required
	($o_{j,i=1}$) First operation
	($o_{j,i=n}$) Last operation where n is the total number of operations
	(cg) Customer goals
Manufacturer (m)	($o_{m,i} \in O_m$) Set of operations offered
	($sp_{m,k} \in P$) Set of operation pairs selected
	(mqb_o) Manufacturer quality bid for operation o
	(mpb_o) Manufacturer price bid for operation o
Operation (o) where o stands for either $o_{m,i}$ or $o_{j,i}$	(ppp) Predecessor in process plan
	(spp) Successor in process plan
	(pop) Predecessor in operation plan
	(sop) Successor in operation plan
	(pt) Processing time
	(est) Earliest start time
	(eft) Earliest finish time
	(lst) Latest start time
	(lft) Latest finish time
	(pst) Proposed start time
	(pft) Proposed finish time
	(tb) Time budget
	($srpp$) Sequencing rank in process plan
	($srop$) Sequencing rank in operation plan
	(id) Operation identity
Operation pair (p) where ($p \in P$) is the set of all valid pairs	(po) Primary operation
	(so) Secondary operation
	($potb$) Primary operation time budget
	(sta) Start time adjustment
Manufacturer pair (mp)	(pm) Primary manufacturer
	(sm) Secondary manufacturer
	($tpit$) Total pair idle time
	(tbo) Time budget overdraft
	(mpc) Manufacturer pair compatibility
	(csc) Compatibility score cap
Manufacturing network (n)	(nc) Network compatibility
	($smp_{n,k} \in MP$) Set of manufacturer pairs selected
	(nqb_j) Network quality bid for job j
	(npb_j) Network price bid for job j
Verification (v)	($pdtt$) Passed due time test
	($pppt$) Passed process plan test
	($popt$) Passed operation plan test

3.3.1 Modelling of order agents

An order agent models the entities that capture the requirements of the customer for a job. Order agent were specialized into job agents, which has three attributes namely job identity (j), job due time (dt) and customer goals (cg). Together with resource agents, order agents possess production execution knowledge.

- Job identity is unique for every job.
- Job due time is used to determine the scheduling boundaries such as latest start time (lst) and latest finish times (lft).
- Customer goal is represented by the relative importance of lead time, quality and price.

3.3.2 Modelling of product agents

A product agent is an abstraction of the manufacturing operations, process plans and operation plans. A process plan (O_j) consists of technical dependencies among operations of a job (j). There is a predecessor operation (ppp)and a successor operation (spp). An operation plan (O_m) represents the line-up of operations at the production facility (m), in the sequence that they will be carried out. There is also a predecessor operation (pop)and a successor operation (sop). The operation agent is a specialization with various attributes namely operation identity (o), processing time (pt), the latest possible finish time (lft), the proposed start time (pst), proposed finish time (pft) and a time budget (tb). Together with order agent, product agents possess process knowledge.

- Every operation has a unique identity which is a combination of a job identity and a manufacturer identity. This means that for every new job, a new operation identity is generated.
- The processing time is the amount of time a manufacturer should allocate to an operation.
- The latest possible finish time is dependent on the due time of a job.
- The proposed start and finish times for an operation are dependent on the availability of a manufacturer and the finish time of the previous operation.
- The time budget is the difference between proposed finish time and latest possible finish time.

3.3.3 Modelling of resource agents

A resource agent models the entities that are responsible for controlling and executing production activities. Resource agents were specialized into three agents namely manufacturer agents, manufacturer pair agents and network agents. During the sequencing process, the role of the resource agents is to maximize their objective functions. The manufacturer agents were given four attributes namely manufacturer identity (m), an optimized operation plan and a proposed operation plan. Together with product agents, resource agents own production knowledge.

- The manufacturer identity distinguishes manufacturers from each other.
- An optimized operation plan is the plan that maximizes the objective function of the manufacturer $\{O_m|\max(tb)\}$.
- The proposed operation plan is the plan that satisfies the objective function of the

manufacturer pair as well as that of the manufacturer $\left\{O_m \middle| \underset{\max}{\longrightarrow}(tb), \underset{\max}{\longrightarrow}(mpc)\right\}$.

Manufacturer pair agents have four attributes namely pair identity (mp), a primary manufacturer (pm), a secondary manufacturer (sm), and pair compatibility (mpc).

- The pair identity is the combined identities of the primary and secondary manufacturers e.g. M1-M2
- Pair compatibility determines the amount of synchronization between the plans of both manufacturers. It is the quantity that the pair agent must maximize.

Network agents have two attributes namely network identity (n) and network compatibility (nc).

- Network identity is the combined identities of selected manufacturer pairs e.g. pair identities M8-M2 & M2-M5 become network identity M8-M2-M5.
- Network compatibility indicates the level of synchronization among selected pairs and it is the quantity that the network needs to maximize.

3.4 Manufacturer pairing approach to flow shop scheduling

The manufacturer pairing approach also known as resource pairing, basically takes a scheduling problem and decomposes it into a couple of self-contained optimisation sub-problems that share the same manufacturer. These sub problems are solved by genetic algorithm. This decentralised feature allows optimisation to take place in parallel at several manufacturers. At every phase, concurrent optimisation takes place. The approach is made up of four phases. The first phase encourages manufacturers to sequence their manufacturing operations as they would prefer it ideally. They maximise their objective functions regardless of how the welfare of the flow shop system is affected. In the second phase, the manufacturers have organised into pairs and are encouraged to cooperate. Pairs maximise their objective functions regardless of the welfare of the system but now with regards to the objectives of their primary and secondary manufacturers. The operation plans of manufacturers synergise and conflicts are reduced to maximise the objective function of the pair. Operation overlapping, idle time and being on the critical paths are considered to be scheduling conflicts. The manufacturer pairs that significantly stand out from the rest, proceed to the third phase. The third phase involves the manufacturing network optimising its own social welfare, by selecting manufacturer pairs with high compatibilities. Every two selected pairs that have a node in common are merged together to grow a network. In the final phase, cooperative scheduling takes place, where manufacturer pairs share their operation plans and cooperatively time their operation plans, for the benefit of the network.

3.4.1 Time budget optimisation

To maximise the chance of achieving job due date (dt), manufacturer agents implement a Time Budget Objective (TBO) function. The TBO function incentivizes the manufacturer to rearrange its operation plan, so that its schedule maximises the difference between proposed finish times (pft) and latest finish times (lft) of its operations. That difference is the time budget (tb) and each operation has one. The time budget is a measure of an optimised operation plan where:

TBO function: $\max \sum tb_o$

$tb_o = lft_o - pft_o$

$pft_o = pst_o + pt_o$

Sample problem:

Given Job 1 is defined as $j = 1$ with operations $o_{1,1} = op10, o_{1,2} = op11$ where $dt_{\text{op10}} = 1, lft_{\text{op10}} = 9$ and $dt_{\text{op11}} = 1, lft_{\text{op11}} = 10$

Given Job 2 is defined as $j = 2$ with operations $o_{2,1} = op20, o_{2,2} = op21$ where $dt_{\text{op20}} = 2, lft_{\text{op20}} = 6$ and $dt_{\text{op21}} = 4, lft_{\text{op21}} = 10$

Given Manufacturer 1 is defined as $m = 1$ with operations $o_{1,\text{i}} \in \{op10, op21\}$ and Manufacturer 2 is defined as $m = 2$ with operations $o_{2,\text{i}} \in \{\text{op11}, \text{op20}\}$

Worked solution:

If $o_{1,1} = op10, o_{1,2} = op21$, therefore $st_{\text{op10}} = 0$, $ft_{\text{op10}} = 1$, $tb_{\text{op10}} = 8$ and $st_{\text{op21}} = 2$, $ft_{\text{op21}} = 6$, $tb_{\text{op21}} = 4$, i.e. $\sum tb_{o_{1,\text{i}}} = 12$

If $o_{1,1} = op21, o_{1,2} = op10$, therefore $st_{\text{op21}} = 2$, $ft_{\text{op21}} = 6$, $tb_{\text{op21}} = 4$ and $st_{\text{op10}} = 6$, $ft_{\text{op10}} = 7$, $tb_{\text{op10}} = 2$, i.e. $\sum tb_{o_{1,\text{i}}} = 6$

If $o_{2,1} = op11, o_{2,2} = op20$, therefore $st_{\text{op11}} = 1$, $ft_{\text{op11}} = 2$, $tb_{\text{op11}} = 8$ and $st_{\text{op20}} = 2$, $ft_{\text{op20}} = 4$, $tb_{\text{op20}} = 2$, i.e. $\sum tb_{o_{2,\text{i}}} = 10$

If $o_{2,1} = op20, o_{2,2} = op11$, therefore $st_{\text{op20}} = 0$, $ft_{\text{op20}} = 2$, $tb_{\text{op20}} = 4$ and $st_{\text{op11}} = 2$, $ft_{\text{op11}} = 3$, $tb_{\text{op11}} = 7$ i.e. $\sum tb_{o_{2,\text{i}}} = 11$

The selected operation plan for Manufacturer1 is where $max\left(\sum tb_{o_{1,\text{i}}}\right)$ and therefore $o_{1,1} = op10, o_{1,2} = op21$ is chosen.

The selected operation plan for Manufacturer2 is where $max\left(\sum tb_{o_{2,\text{i}}}\right)$ and therefore $o_{2,1} = op20, o_{2,2} = op11$ is chosen.

3.4.2 Pair compatibility optimisation

To maximise manufacturer pair compatibility (mpc), pair agents implement a Pair

Compatibility Objective (PCO) function. Pair compatibility is inversely proportional to the sum of idle time between operations ($tpit$) and the overdraft of time budget (tbo). An overdraft occurs when a manufacturer has proposed, for at least one operation, a finish time that exceeds the latest possible finish time of the operation. Pair compatibility is also a measure of optimality loss. Pair compatibility ideally would be equal to 1. The PCO function incentivizes a pair of operation plans to rearrange, to reduce operation lateness and idle time between operations. This results in an optimized arrangement of operations where:

PCO function: $\max mpc_p$

$a = \left|\sum(st_{o_{pm}} - ft_{o_{sm}})\right|$ where $o_{pm} \equiv o_{j,x}$ and $o_{sm} \equiv o_{j,y}$ and $x > y$

$b = \left|\sum(st_{o_{sm}} - ft_{o_{pm}})\right|$ where $o_{pm} \equiv o_{j,x}$ and $o_{sm} \equiv o_{j,y}$ and $x < y$

$tpit_p = a + b$

$a = \left|\sum TB_{o_{pm}} + \sum TB_{o_{sm}}\right|$ where $o_{pm} \equiv o_{j,x}$ and $o_{sm} \equiv o_{j,y}$ and $x > y$ and $TB < 0$

$b = \left|\sum TB_{o_{sm}} + \sum TB_{o_{pm}}\right|$ where $o_{pm} \equiv o_{j,x}$ and $o_{sm} \equiv o_{j,y}$ and $x > y$ and $TB < 0$

$tbo_p = a + b$

$mpc = \frac{csc}{x}$ where $x = 1 + tpit_p + tbo_p$

Sample problem:

Given Job 1 is defined as $j = 1$ with operations $o_{1,1} = op10, o_{1,2} = op11, o_{1,3} = op12$

Given Job 2 is defined as $j = 2$ with operations $o_{2,1} = op20, o_{2,2} = op21, o_{2,3} = op22$

Given Manufacturer 1 is defined as $m = 1$ and has a selected operation plan where $o_{1,1} =$

$op10, o_{1,2} = op21$

Given also that $dt_{\text{op10}} = 1, lft_{\text{op10}} = 2, pst_{\text{op10}} = 0,\ pft_{\text{op10}} = 1,\ tb_{\text{op10}} = 1$ and $dt_{\text{op21}} = 4, lft_{\text{op21}} = 5, pst_{\text{op21}} = 2,\ pft_{\text{op21}} = 6,\ tb_{\text{op21}} = -1, \sum tb = 0$

Given that Manufacturer 2 is defined as $m = 2$ and has a selected operation plan $o_{2,1} = op11, o_{2,2} = op20$

Given also that $dt_{\text{op11}} = 1, lft_{\text{op11}} = 2, pst_{\text{op11}} = 1,\ pft_{\text{op11}} = 2,\ tb_{\text{op11}} = 0$ and $dt_{\text{op20}} = 2, lft_{\text{op20}} = 5, pst_{\text{op20}} = 2,\ pft_{\text{op20}} = 4,\ tb_{\text{op20}} = 1, \sum tb = 1$

Given that Manufacturer 3 is defined as $m = 3$ and has a selected operation plan $o_{3,1} = op22, o_{3,2} = op12$

Given also that $dt_{\text{op22}} = 3, lft_{\text{op22}} = 10, pst_{\text{op22}} = 6,\ pft_{\text{op22}} = 9,\ tb_{\text{op22}} = 1$ and $dt_{\text{op12}} = 1, lft_{\text{op12}} = 10, pst_{\text{op12}} = 9,\ pft_{\text{op12}} = 10,\ tb_{\text{op12}} = 0, \sum tb = 1$

Worked solution:

$m_{1,1} = 1; m_{2,1} = 3; n = 1; p = 1;\ csc = 100$

$a = pst_{op22} - pft_{op21} = 6 - 6 = 0$; $b = pst_{op12} - pft_{op10} = 9 - 1 = 8$; $tpit_1 = |a + b| = 8; tbo_1 = 0; mpc_1 = \frac{100}{1+8+0} = 11$

$m_{1,2} = 1; m_{2,2} = 2; n = 1; p = 2; csc = 100$

$a = pst_{op21} - pft_{op20} = 2 - 4 = -2$; $b = pst_{op11} - pft_{op10} = 1 - 1 = 0$; $tpit_2 = |a + b| = 2; tbo_2 = 0; mpc_2 = \frac{100}{1+2+0} = 33$

$m_{1,3} = 2; m_{2,3} = 3; n = 1; p = 3; csc = 100$

$a = pst_{op22} - pft_{op20} = 6 - 4 = 2$; $b = pst_{op12} - pft_{op11} = 9 - 2 = 7$; $tpit_3 = |a + b| = 9; tbo_3 = 0; mpc_3 = \frac{100}{1+9+0} = 10$

3.4.3 Network compatibility optimisation

Finally, to maximise the network compatibility (nc), the manufacturing network agent selects a group of manufacturer pairs ($smp_{n,k}$) that will create a valid network (n). The network compatibility is the sum of pair compatibilities. A valid network will have a much higher compatibility value (nc) that an invalid network and lower optimality loss. The valid network with the highest compatibility value will be selected according to the Network Compatibility Objective (NCO) function.

NCO function: $\max \sum mpc_p$

3.5 Operation pairing approach to flow shop scheduling

Operation pairing approach allows manufacturers to generate their own operation plans. An operation plan would be constructed from a set of operation pairs that were selected according to some criteria. The properties of the operation pair is derived from properties of its primary and secondary operations. The indicators tb_o and sta_p help to keep the operation planning in check with various boundary constraints. To generate operation plans, two approaches are proposed.

- One approach gathers operation pairs that consists of primary operations, with similar positions in the process plans. This position is indicated by $srpp_{o_{j,i}}$.
- The other approach is a pre-selection of operation pairs that allow their primary operations to conform to a time-based position which is given by $srop_{o_{m,i}}$.

It is possible to have more than one operation pairs with the same $srpp$ or same $srop$ positions. Those pairs will form part of the same preselection. From the pre-selected operation pairs, $sp_{m,k}$ is selected, where the indicators tb_o and sta_p or a combination of both, informed the selection. At every stage of the process, the validity tests $pppt, popt, pdtt$ enable the monitoring of schedules generated so that the schedules conform to the boundary conditions.

3.5.1 Rules of sequencing

In this approach, all manufacturer agents must use the same sequencing rules. These rules ensure that if all generated manufacturer operation plans were superposed and timed, their plans would emerge into complete flow shop schedules. Below are the rules that define the relationships between the attributes of the modelled agents.

Relationship between $o_{j,i}$ and $o_{j,i+1}$ in a process plan

$o_{j,i} \in O_j$

$o_{j,i+1} = spp_{o_{j,i}}$ where $i \leq n$

$o_{j,i} = ppp_{o_{j,i+1}}$ where $i \leq n$

Relationship between $o_{m,i}$ and $o_{m,i+1}$ in an operation plan

$o_{m,i} \in O_m$

$o_{m,i+1} = sop_{o_{m,i}}$ where $i \leq n$

$o_{m,i} = pop_{o_{m,i+1}}$ where $i \leq n$

Selection of operation pairs out of all possible operation pairs

$sp_{m,k} \in SP_m \subset P$

Relationship among $sp_{m,k}$, $o_{m,i}$, $o_{m,i+1}$ in creating an operation plan

$pop_{so_p} = po_p$ where $p = sp_{m,k}$

All boundary conditions for the generation of operation pairs

$lft_{o_{j,n}} = dt_j$

$lst_{o_{j,i}} = lft_{o_{j,i}} - pt_{o_{j,i}}$

$lft_{o_{j,i-1}} = lst_{o_{j,i}}$

$est_{o_{j,1}} = 0$

$eft_{o_{j,i}} = est_{o_{j,i}} + pt_{o_{j,i}}$

$est_{o_{j,i+1}} = eft_{o_{j,i}}$

Proposed property of an operation

$pst_o = lst_o - sta_o$

$pft_o = pst_o + pt_o$

Relationship between operations of an operation pair

$sta_p = pft_{po_p} - pst_{so_p}$

Relationship between operations and boundary conditions

$tb_o = pft_o - eft_o$

$sotb_p = tb_{so_p}$

$potb_p = tb_{po_p}$

Relationship between operations and their fixed positions in a process plan

$srpp_{o_{j,1}} = 1$

$srpp_{o_{j,i+1}} = srpp_{o_{j,i}} + 1$

Relationship between operations and their proposed positions in an operation plan

$$srop_{o_{m,i}} = \left\| n \times {}^{lst_{o_{m,i}}}\!/_{max(lst_o)} \right\|$$

Validity tests between the two operations of an operation pair

$$v = \left(id_{o_{j,i}}, id_{o_{j,i+1}}\right)$$

$$pppt_v = pft_{o_{j,i}} \leq pst_{o_{j,i+1}}$$

$$v = \left(id_{o_{m,i}}, id_{o_{m,i+1}}\right)$$

$$popt_v = pft_{o_{m,i}} \leq pst_{o_{m,i+1}}$$

$$pdtt_{v=j} = pst_{o_{j,i}} \geq 0$$

3.5.2 Last operation heuristic rule

To create an operation plan of (n) operations, $(n-1)$ operation pairs are needed. An operation plan of five operations would consist of four operation pairs as shown in Figure 3.2. Generation of operation pairs, for the purpose of creating an operation plan can be started in two ways. In the first approach, the secondary operation $so_{sp_{m,1}}$ can be any operation $o_{m,i}$. In the second approach, the secondary operation is limited to $\{o_{m,i} | o_{j,5} \equiv o_{m,i}\}$ which represents the last operation in the process plan of a job (j). This is called last operation heuristic rule.

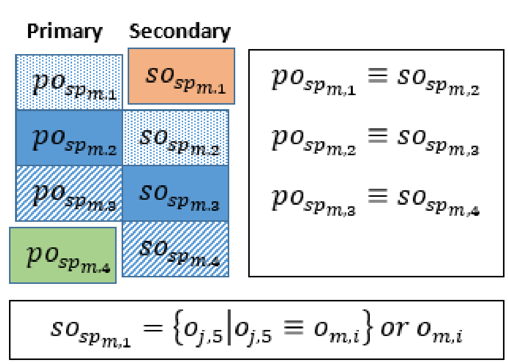


Figure 3.2: Template for a four-pair operation plan

3.5.3 Sequencing recovery from disturbance

A delay of an operation would decrease its boundary value $lst_{o_{m,i}}$ so that its time-based position $srop_{o_{m,i}}$ may be affected. The larger the delay is, and the greater the total number of operation is, the more likely the operation would be moved forward in position. A new position means that the operation pair that has a delayed primary operation would fall in a different pair preselection list. This makes sense as a late operation needs all the priority it can get for completion. A rush job could be simulated by shortening the job due time. The time-based positions of all operations involved would shift forward into a different pair preselection list. The pair selection is not directly affected by disturbances as the pair preselection is because the algorithm is sensitive to neither processing time nor due time. However, the pair preselection list that had the pair before the delay, no longer has it. From that list, a new best pair would be selected. The rest of the operation plan would be updated similarly. This is how disturbance could be managed in this system.

3.6 Customer-driven selection of final manufacturing network

It was mentioned in an earlier section that a flow shop system might consist of more than one manufacturer that can fulfil an operation required by a job. Therefore, there might be more than one network capable of fulfilling a job. The next part of the research aim addresses the market-based selection of a network. The final decision about which network is awarded the job, is left to the order agent. There are two ways for the agent to discover the best manufacturing networks for customer orders. The first approach selects a network where most of its manufacturers offered best bids for each operation of a job process plan. The second approach selects the network that offers the best bid for the job.

3.6.1 Call for bids

When a customer places an order, the system integrator disaggregates the order into several jobs. A job agent represents a job and carries out functions on its behalf. The agent gets the process plan from the product agent. Then it broadcasts a call for bids with the wider community of resource agents. All possible manufacturing networks that can satisfy the job, are generated by either the manufacturing pairing approach or the operation pairing approach.

3.6.2 Timing of operations and bidding

At this stage, operation timing is performed. The proposed start time and finish time are generated by the forward scheduling of an operation within the constraints imposed by a job process plan and by a manufacturer operation plan. In the first approach, the manufacturer agent bids for the operations with a proposed start time (pst), proposed finish time (pft), quality (mqb) and price (mpb). Bids are visible to other manufacturers and this encourages competition. In the second approach, the network agent bids for the whole job. The individual manufacturer scheduling information is kept confidential and is known only within the network. Only the bid from the network is visible. The quality bid from the network is given by the sum of the quality of its manufacturers: $nqb = \sum_i^n mqb_{o_{j,i}}$ where n is total number of operations for job j. The price that the network can deliver the job for, is given by the sum of the price of its manufacturers: $npb = \sum_i^n mpb_{o_{j,i}}$ where n is total number of operations for job j.

3.6.3 Network selection

Networks are selected on the basis of main customer-driven objective metrics such as quality, cost and lead time. The outcome of network selection is affected by the relative importance of these objectives. The relative importance can be determined by pairwise comparison. For quality (Q), cost (C) and lead time (L), there are six such comparisons i.e. QC, CQ, CL, LC, QL, LQ. From the industrial use case with GFM, it was determined that lead time was the most important criteria to the customer. Quality is the most important criteria to GFM even though the product, requested by the customer, may not have critical quality attributes. Production output is high-value and their customers expect high price tags. Therefore, cost is of relatively lower importance to the customer. Therefore, it was determined that network selection would be carried out by the LQC decision algorithm. This particular algorithm operates by filtering the bidders on the basis of **L**ead time first, then **Q**uality and finally **C**ost. The algorithm is executed by a job agent and is performed with either the first or second approach of network selection. In the first approach, manufacturers bid for the individual operation that constitutes a complete job. Each manufacturer belongs to a manufacturing network and bids on behalf of it. A job agent retains the best bidding manufacturer for each of its operations. Eventually, it results into a list of preferred manufacturers. The network that has the most of its manufacturers on that list, is selected by the job agent. In the second approach, job agent selects the manufacturing network that offers the best bid for the job.

3.6.4 Summary of proposed approaches

The approaches can be summed up into three main functions namely formation, pairing and selection as illustrated in Figure 3.3. Each function has to comply with a set of constraints and achieve some objectives to a certain extent. The outcome of a formation function is an operation plan from which a manufacturer can gather enough information to bid on time-based metrics, quality and cost. For approach 1.1, the time-based metric is the optimised time budget (TBO) and is a pre-requisite to the pairing function in approach 1.2. For approach 2.2, the time-based metrics of time budget, start time adjustment and a due time test must be achieved without schedule overlap in operation plan and process plan. Moreover, the last operation of a process plan must be the last operation of an operation plan. The outcome of a pairing function is a manufacturer pair, for approach 1.2, and operation pair, for approach 2.1. With the optimised time budget, at the level of approach 1.2, it is possible to determine the optimality loss which is incurred when manufacturers are compelled to re-sequence their operation plans in order to reduce idle time and time budget overdraft. These metrics contribute to the manufacturer pair compatibility which is a pre-requisite for the selection function in approach 1.3. As for approach 2.1, operation pairs must be part of the same manufacturer and have the same position in either their process plans (denoted as srpp) or on a common time scale (denoted as srop). Also, the new generated operation pair must link to the previous pair used in operation plan formation. Finally, the selection function involves approach 1.3 which has the objective of selecting networks of manufacturer pairs with high compatibility, leading to high network compatibility. From the set of highly compatible networks, the list of network is narrowed to those with high production quality scores. Finally, a highly compatible network with high quality scores is selected on the basis of price. The means for prototyping involved agent-based modelling and multi-agent systems. The input to the prototype consisted of a scheduling problem involving manufacturers, jobs, job process plan, operations, processing times and operation disturbances. The main outcomes were manufacturer operation plans, job schedules, manufacturer schedules and manufacturing networks.

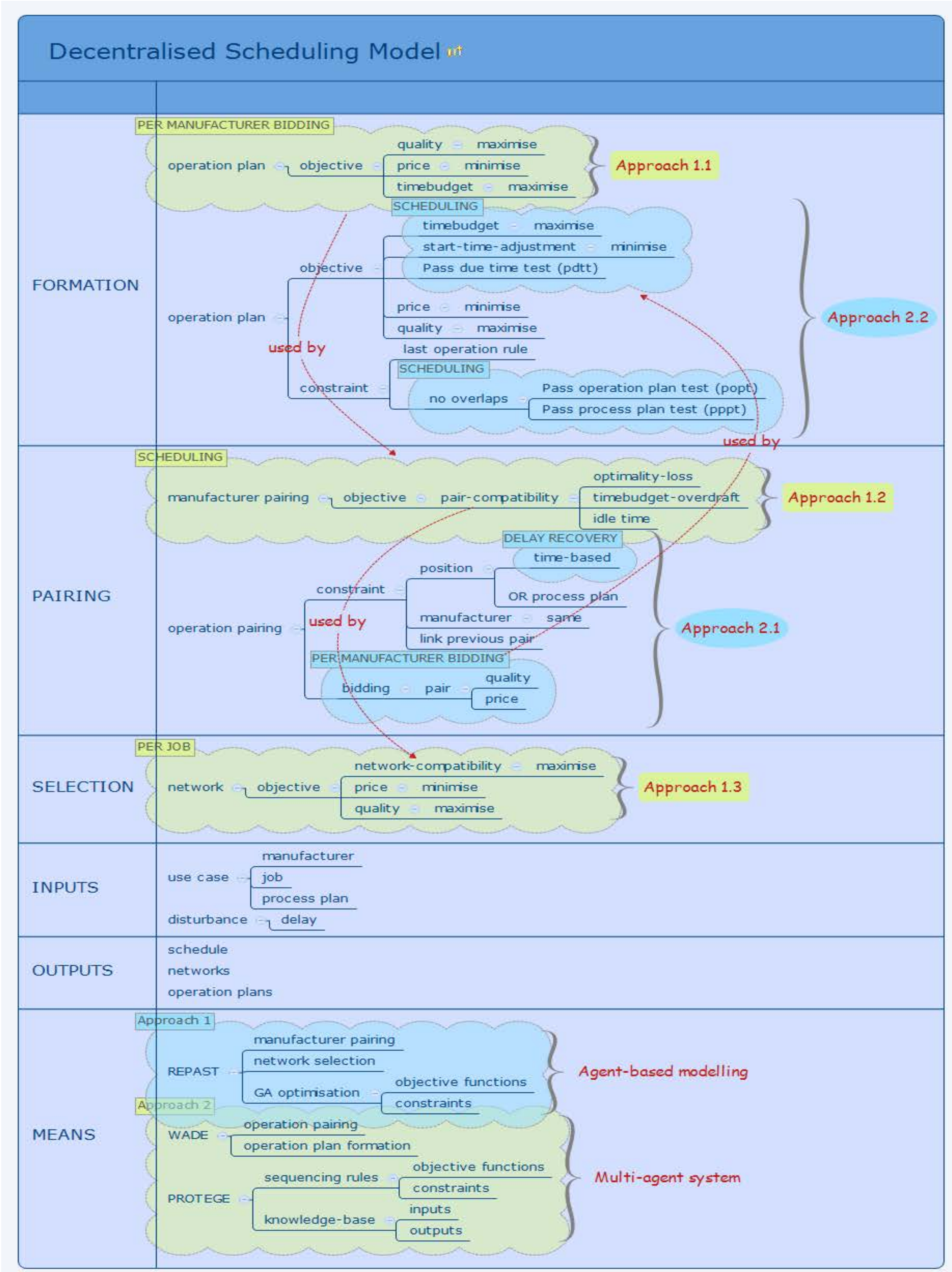


Figure 3.3: Functional analysis of proposed scheduling approaches

3.7 Conclusion

The chapter describes a flow shop scheduling problem for an industrial use case and presented two operation research case studies including datasets and benchmarks. Four models of decentralised scheduling have been developed so that they can be simulated with the available datasets and their performance evaluated against some benchmarks. The models of decentralised flow shop scheduling include the manufacturer pairing and the operation pairing approaches. The approaches, for selecting a network out of a pool of potential networks, were modelled and are presented. The approaches include network selection with respect to manufacturer bids versus network selection with regards to network bids. The chapter presented a holistic approach to flow shop scheduling for a manufacturing network breeding environment.

DEVELOPMENT OF SIMULATION SYSTEMS

4.1 Introduction

The software infrastructure was developed to test the various approaches presented in chapter 3, for the decentralised scheduling of a flow shop system. All approaches aim at giving manufacturers control over their scheduling activities. The approaches involve the formation of manufacturing networks by manufacturer pairing or operation pairing as well as the selection of the best network on the basis of individual manufacturer bids or network bids. The software should effectively use the data sets from the case studies so that the results of the experiments can be benchmarked and comparison of approaches can be drawn. However, the results should allow the validation of the research hypotheses. The first hypothesis is that the selection of manufacturing networks, for a set of jobs, is better performed by a selection based on network bids rather than a selection based on manufacturer bids. The performance is measured according to the criteria of production lead time, cost and quality. The second hypothesis is whether genetic algorithm can be decentralised onto individual manufacturers and still manage to maximise the utilitarian welfare of the manufacturing network through cooperation. The third hypothesis is whether an inference-based system would effectively detect conflicts when subject to various disturbances. The fourth hypothesis inquires about the optimality loss of scheduling with decentralised heuristic algorithms. The fifth hypothesis inquires about the scalability of conflict resolution performed by a multi-agent system in terms of solution space size and computation time.

4.2 Simulation of the manufacturer pairing approach

The agent-based model of manufacturer pairing is ran Recursive Porous Agent Simulation Toolkit (REPAST) using the MT10 data set and results plotted into a Gantt chart using Microsoft Project. Repast supports a set of independent third-party applications such as Java Genetic Algorithm Package (JGAP) and Microsoft Excel spreadsheet. Repast uses Eclipse as the primary development environment. Genetic algorithm (GA) is a search algorithm relying on natural selection to evolve a sample set of potential solution to a set of more optimal solutions. The advantage of GA lies within its ability of experimenting on many potential optima and not stagnating on one solution but boasting a set of potentially optimal solutions. GA optimisation mechanism consists of three main stages namely reproduction (solution copy), crossover (solutions mating) and mutation (random alteration within solution). Two criteria help guarantee that the algorithm works effectively. First, an objective fitness function is necessary to compare potential solutions called chromosomes. The higher the fitness value, the better the solution. Second, discrete parts, called genes, must make up the potential solution, parts which can be independently manipulated.

4.2.1 Encoding of the operation plan of a manufacturer agent

The manufacturer agent has a population of sample potential solutions called genotypes. These solutions are encoded in chromosomes made up of integer genes. The integer within the genes are called alleles and are bounded from 1 to 10 because the operation plans consist of 10 operations. Genotypes are made of chromosomes where genes, in Figure 4.1a, have been shuffled around for each chromosome. Figure 4.1b represents operation identity substitution for genes of a chromosome where allele 1 represents Operation 11, allele 2 represents Operation 21 and so on.

a)

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**		**10**

b)

11	**21**	**31**	**41**	**51**	**61**	**71**	**81**	**91**	**101**

Figure 4.1: a) Encoded operation plan and b) Decoded operation plan of Manufacturer 1

4.2.2 Encoding of the combined plan of a manufacturer pair agent

The integer within genes are bounded from 1 to 20 because two operation plans are being dealt with. The manufacturer pair agent does not perform the same operation several times and therefore the chromosome will not contain duplicate integer genes. Figure 4.2 represents the combined operation plan of two manufacturers encoded into 20-genes.

a)

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**

b)

16	**26**	**36**	**46**	**56**	**66**	**76**	**86**	**96**	**106**
19	**29**	**39**	**49**	**59**	**69**	**79**	**89**	**99**	**109**

Figure 4.2: a) Encoded combined plan and b) Decoded combined plan of two manufacturers

4.2.3 Encoding of manufacturer pairs for a manufacturing network agent

The integer within the genes are bounded from 1 to 10 because there are 10 manufacturers. The manufacturing network agent does not use the same manufacturer pair twice and therefore the chromosome will not contain duplicate integer genes. Furthermore, it is very important that no more than two pairs contain the same manufacturer. Figure 4.3 represents the relationship between alleles and the decoded manufacturer pair identities whereby the combination of two adjacent alleles represent a manufacturer pair.

a)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
b)	**1-2**	**2-3**	**3-4**	**4-5**	**5-6**	**6-7**	**7-8**	**8-9**	**9-10**	**10-1**
c)	**4**	**2**	**9**	**1**	**5**	**8**	**6**	**7**	**3**	**10**
d)	**4-2**	**2-9**	**9-1**	**1-5**	**5-8**	**8-6**	**6-7**	**7-3**	**3-10**	**10-4**

Figure 4.3: Chromosomes (a, c) representing the decoded manufacturing networks (b, d)

4.2.4 Genetic algorithm embedded within a resource agent

In order for agents to have the right information at the right time for their optimisation process, an interaction protocol was developed. The protocol is made up of six main functions, as illustrated in Figure 4.4, which enable the following actions to be performed in series.

- Optimisation of an operation plan by the manufacturer agent according to the TBO fitness function (optimiseOperationPlanLocally)
- Generation of manufacturer pairs by the manufacturer agent (generatePairs)
- Optimisation of combined operation plans of the pair by the manufacturer pair agent based on the PCO fitness function (optimisedCombinedPlan)
- Generation of manufacturing networks by the manufacturer pair agent (generateManufacturingNetworks)
- Optimisation of the manufacturing network by the manufacturing network agent based on the NCO fitness function (optimiseManufacturingNetwork)

- Selection of the best manufacturing network by the scheduler agent (selectBestManufacturingNetwork)

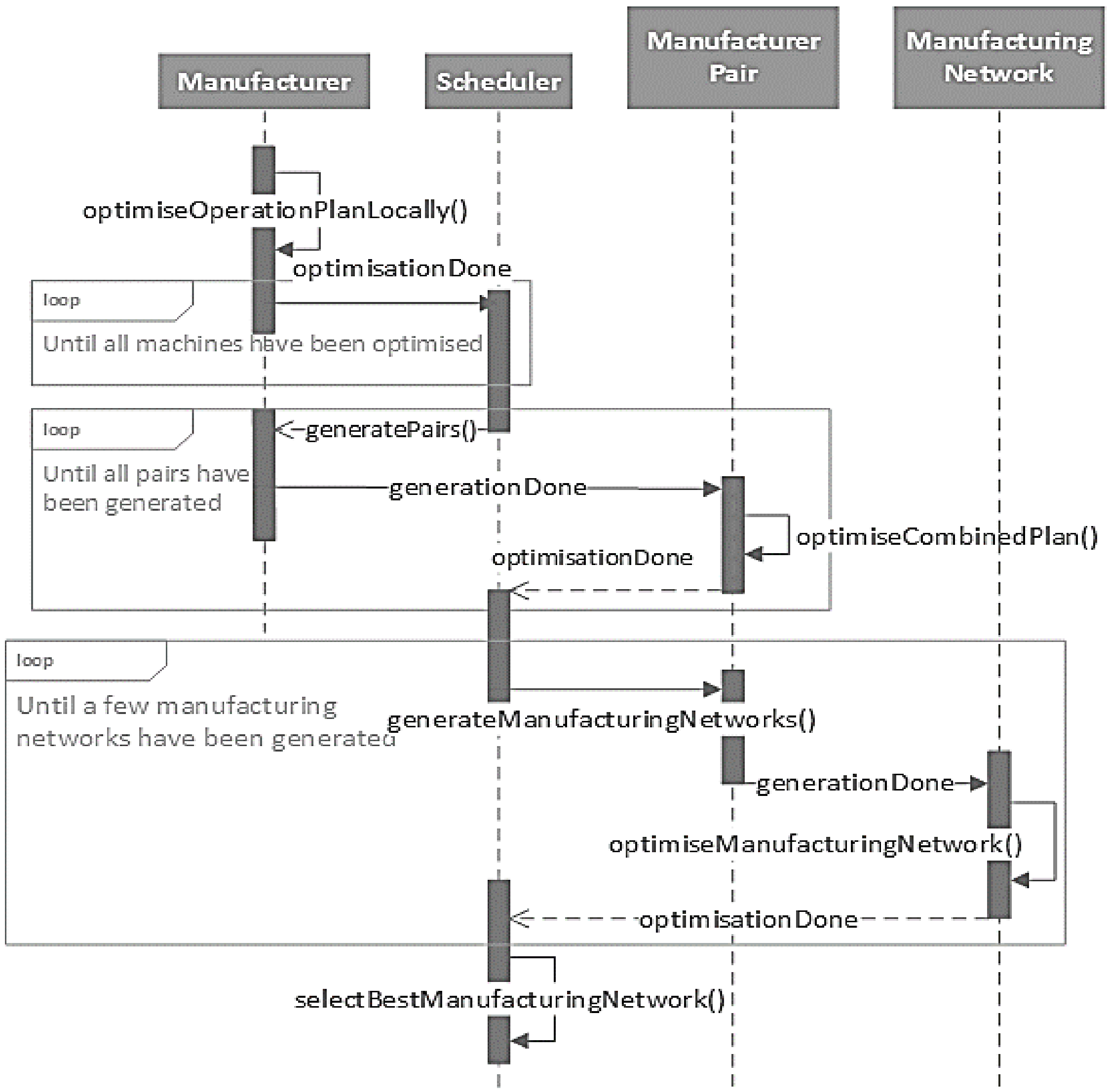


Figure 4.4: UML activity diagram of the interactions between agents

The generic algorithm, shown in Figure 4.5, is used by all resource agents, manufacturer agent, manufacturer pair agents and network agent, with minor modifications for each agent type. A parameter, called chromosome application data, is an object that passes information to the fitness function enabling chromosome genes to be decoded. For manufacturer agent, an operation plan is passed via the parameter, for manufacturer pair agent, a combined operation plan is passed and for manufacturing network, a list of manufacturer pairs is passed. The GASolver is an object that each agent implements to execute GA optimisation.

```
data = new ChromosomeApplicationData(operationPlan or combinedPlan or listOfPairs)
solver = new GASolver(populationSize, data)
solver.createConfiguration(new X-FitnessFunction(), new GreedyCrossoverOperator(),
new SwappingMutationOperator())
population = solver.generatePopulation()
do {
    model.setPreviousPopulation(population)
    population = solver.generatePopulation()
    population.evolve()
    j++
} while (j <= cycles)
```

Figure 4.5: Pseudo-code for the optimisation process of the resource agents.

The solver is configured with a TBO fitness function for manufacturer agents, PCO fitness function for manufacturer pairs, and NCO for manufacturing network agents. The solver is further configured with the greedy crossover operator for the crossover stage and the swapping mutation operator for the mutation stage. They are the only operators which will avoid duplicate alleles and are often found applied to the Travelling Salesman Problem (TSP) for precisely that reason. Furthermore, an offset parameter can be specified to keep a part of a chromosome fixed and devoid of mutation and swapping manipulation. The offset parameter is used for manufacturing network optimisation. For instance, given pairs 1-6, 6-3 and 3-5. On one hand, for pair 1-6, the combined operation plan is given no offset so that operation plans for manufacturer 1 and manufacturer 6 are optimised. On the other hand, the optimised operation plan of manufacturer 6 is kept fixed during the optimisation of pair 6-3. The combined plan is offset so that only the operation plan of manufacturer 3 is allowed to be manipulated. The same takes place for the pair 3-5. This ensures that the optimised manufacturing network is congruent and schedules of the pairs are aligned.

The population of solutions is initially created and passed to a loop function. The population is evolved for a set number of cycles. To increase the problem solving efficiency, the previous population is passed back to the solver at each iteration and a fraction of solutions are replaced by a new population. Best solutions have higher chances to be retained.

4.3 Simulation of operation pairing approach to decentralised flow shop scheduling

The agent-based model of operation pairing was implemented into a multi-agent system. To support implementations in a multi-agent system applications, a knowledge base and an inference based system were investigated. The latter enable multi-agent system to gather information, reason about their environment and take actions. The opportunity arose to revisit issues of scheduling from the perspective of these systems.

4.3.1 Knowledge-based system overview

Ontology editor Protégé 5.0 beta was used to design, in OWL2, a knowledge base with classes 'Manufacturer', 'Job', 'Operation' and 'Verification'. SWRL editor, which is provided with the ontology editor, enabled constraint calculation, scheduling and verification rules to be developed as mentioned in the previous section. The framework is depicted by Figure 4.6. In this framework, job process requirements, manufacturer services and job process plans come from the modified MT10 and LA19 scheduling problems. These documents form the majority of the facts asserted to the knowledge base. The input of the facts, the instantiation of the data objects and the linking of data are performed manually. On the other hand, constraints calculation and plan scheduling are autonomous where time adjustments are inferred and performed effectively by a multi-agent system. Furthermore, the manufacturer agent automatically generates the manufacturer service plan by operation sequencing. The time adjustments are suggested via a series of rules that flag up violations of constraints, sources of conflicts, magnitude and direction of required adjustments. More details about the calculation, scheduling and verification rules are in the Appendix.

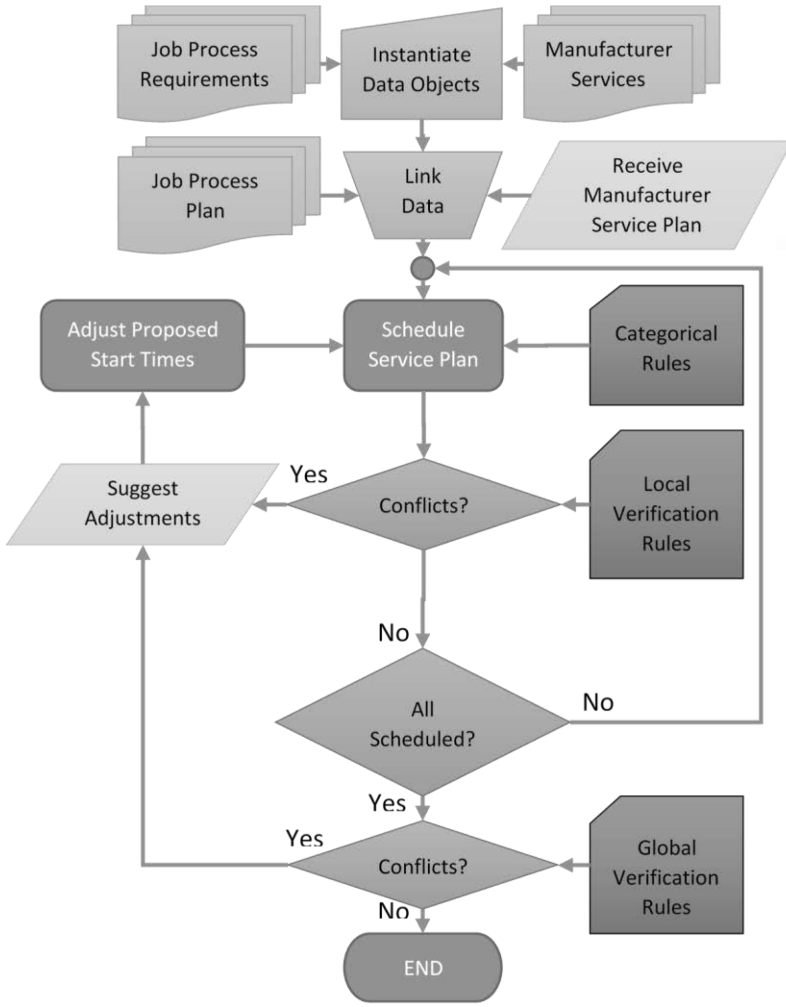


Figure 4.6: Framework for utilizing Inference-based System for scheduling

4.3.2 Inference-based system overview

Modus ponens (MP) is one possible basis for logical manipulation. Given an MP as such ‘P implies Q’, if P is true, then Q must be true. The consequent of a rule like ‘P → Q’ in this case Q, can be an antecedent to another rule ‘Q → R’. This forms a chain of logic within the rule base which an inference engine exploits to manipulate its database of facts and rules. Such an inference engine is Pellet which provides sound-and-complete reasoning for facts written in OWL 2 and rules written in SWRL. The example below demonstrates an application of the system. The schedule rules from Chapter 3 have now been written in SWRL as shown below. More examples of useful rules are in the Appendix. Equations $o_{j,i} = ppp_{o_{j,i+1}}$ and $o_{j,i+1} = spp_{o_{j,i}}$ can be written in SWRL as `(precedesJobwise ?o1 ?o2)` where `precedesJobwise` is an object property of an operation which takes another operation as its parameter. In this case, the operation `o1` precedes operation `o2` within a job process plan. Parameters $est_{o_{j,i}}$ can be written as `(hasEarliestPossibleStartTime ?o1 ?est)` so that an operation has an earliest possible start time of `?est`. Parameters $pt_{o_{j,i}}$ can be written as `(hasProcessingTime ?o1 ?pt)` so that an operation has a processing time of `?pt`.

Let:

$m =$ `(precedesJobwise ?o1 ?o2)`

$n =$ `(hasEarliestPossibleStartTime ?o1 ?est)`

$p =$ `(hasProcessingTime ?o1 ?pt)`

Facts:

$m^1 =$ `(preceedsJobwise operation_10 operation_11)`

$m^2 =$ `(preceedsJobwise operation_11 operation_12)`

$n^1 =$ `(hasEarliestPossibleStartTime operation_10 0)`

$n^2 =$ `(hasEarliestPossibleStartTime operation_11 null)`

$p^1 =$ `(hasProcessingTime operation_10 29)`

$p^2 =$ `(hasProcessingTime operation_11 78)`

When equations $eft_{o_{j,i}} = est_{o_{j,i}} + pt_{o_{j,i}}$ and $est_{o_{j,i+1}} = eft_{o_{j,i}}$ are written in the form (m, n, p) → (q, r), the following modus ponen is obtained:

```
preceedsJobwise(?o1,?o2),hasEarliestPossibleStartTime(?o1,?st),
hasProcessingTime(?o1,?pt),add(?ft,?st,?pt)->
hasEarliestPossibleFinishTime(?o1,?ft),hasEarliestPossibleStart
Time(?o2,?ft)
```

Substitution 1:

`preceedsJobwise(operation_10,operation_11),hasEarliestPossibleStartTime(operation_10,0),hasProcessingTime(operation_10,29),add(29,0,29)` $= m^1, n^1, p^1$

Which yields:

$q^1 =$ `(hasEarliestPossibleFinishTime operation_10 29)`

$r^1 =$ `(hasEarliestPossibleStartTime operation_11 29)`

Substitution 2:

`preceedsJobwise(operation_11,operation_12),hasEarliestPossibleStartTime(operation_11,29),hasProcessingTime(operation_11,78),add(107,29,78)` $= m^2, n^2, p^2$

Which yields:

q^2 = `(hasEarliestPossibleFinishTime operation_11 107)`

r^2 = `(hasEarliestPossibleStartTime operation_12 107)`

4.3.3 Multi-Agent System overview

The decentralised flow shop scheduling of manufacturing tasks was performed by a multi agent system. The multi-agent system would read and write to the ontology in the Ontology Web Language (OWL). Moreover, the system implemented Pellet which enabled agents to reason about the knowledge base. The reasoning is dictated by rules that were written in Semantic Web Rule Language (SWRL). Technologically, OWL supersedes Resource Description Framework (RDF) and is currently the best way of storing facts in an ontology. SWRL supersedes Drools and is the main language for Pellet.

Case studies of scheduling problems such as MT10 and LA19 were used to test the proposed heuristic scheduling algorithms that were embedded in manufacturing agents. The support agent provides support for the formation of manufacturer agents and job agents. It utilises manufacturer and job information from the ontology and register the agent details to the resource management agent (RMA) in WADE. RMA acts as a yellow page facility where agents can look up operations and touch basis with providers. Once the agents are created, the support agent terminates.

At this point, the manufacturing scheduling process is triggered and Figure 4.7 depicts functions that are performed by the system. Manufacturer and job agents generate a set of valid operation pair agents, agents that handle pairs of primary and secondary operations that are related by precedency. Job agents create operation pairs that are derived from process plans. The ontology holds the data about operations and the predecessor-successor relationships between them.

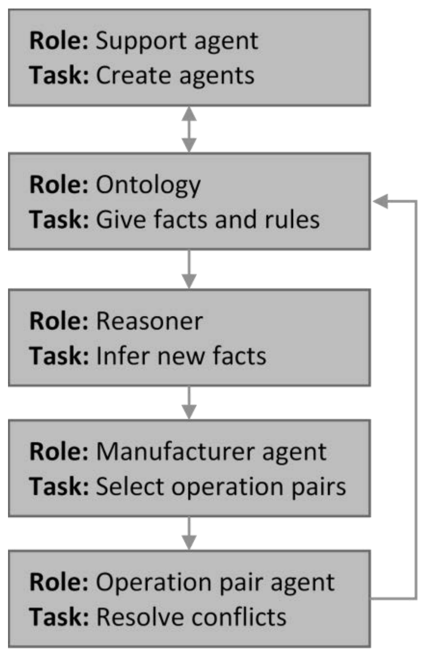


Figure 4.7: Interaction flowchart of system components

Manufacturer agents, on the other hand, do not have operation plans yet. An operation plan is derived from the operation pairs that an agent decides to select. The creation process of operation pairs follows four distinct steps. These steps are empirically investigated and presented in greater details later in the chapter. The steps are:

- Pair set generation
- Pair subset preselection
- Pair selection
- Pair conflict resolution

Job agents then assert the properties of their operation pair agents into the ontology. The manufacturer agents do the same with theirs where operation plans are automatically inferred from the pair data. There are explicitly written rules that the reasoner Pellet interpret to infer such data. Manufacturer agents and operation pair agents implement Pellet.

4.4 Simulation of final network selection

4.4.1 Formal Proof to the Best Network Organisation

Considering a scenario of two jobs and three manufacturers. Job 1 requires two operations where the first operation can be performed by manufacturer M1 or M3 and the second operation by manufacturer M2. Job 2 requires one operation where the operation can be performed by manufacturer M1 or M3. In terms of capacity, manufacturer M1 is available between unit times 1 and 6, manufacturer M2 between 4 and 10 and manufacturer M3 between 1 and 6. A manufacturer is symbolized as M (s, f, q, c) where M is the manufacturer label, s and f are the dynamic attributes, start time and finish time, q and c are the static attributes, quality and cost. A network is symbolised as N (s, f, q, c) where N is a label aggregate of the manufacturers that constitute the network.

Concept 1 – Manufacturer Selection

Job 1 → M1 (1, 3, A, £20) + M2 (5, 7, A, £40) → M1M2 (1, 7, AA, £60)

Job 2 → M3 (1, 3, C, £50)

Concept 2 – Network Selection

Job 1 → M1 (3, 5, A, £20) + M2 (5, 7, A, £40) → M1M2 (3, 7, AA, £60)

Job 2 → M1 (1, 3, A, £20)

As proven above, concept 2 logically performs better than the conventional concept. Concept 1 uses local optimisation, where manufacturer having better delivery time are selected while concept 2 looks at the performance of the whole network and chooses the group of networks with the most benefit in terms of lead time M1M2 (7) + M1 (3), quality M1M2 (AA) + M1 (A), cost M1M2 (£60) + M1 (£20) as mentioned in Section 3.6.2.

4.4.2 Overview of simulation platform

To compare the two approaches of network selection by manufacturer bids or by network bids, a simulation was developed using the agent-based modelling framework. The platform was developed in Java, using the NetBeans integrated development environment. The activity diagram of the platform as shown in Figure 4.8 integrates all the modelled entities, their roles, interactions and outcomes. The network formation is performed by the aforementioned manufacturer pairing or the operation pairing approaches, where operation plans are also formed. The resources for network formation come from the manufacturing network breeding environment. The blueprints for forming the networks are the process plans of job agents.

Consequently, within each formed network, all operations are bounded by the process plans and operation plans that they are part of. The staff agent controls when a job agent is released, using a priority rule. Once released, a job agent selects and allocates itself to a manufacturing network. The selection of a network is dependent on static and dynamic attributes. The times at which a resource agent can start and finish an operation or a job, are dynamic attributes. Quality and price are static attributes. The dynamic attributes are determined when resource agents perform the timing of operations for each released job, within the constraints of the job process plans and the manufacturer operation plans.

4.4.3 Experiment overview

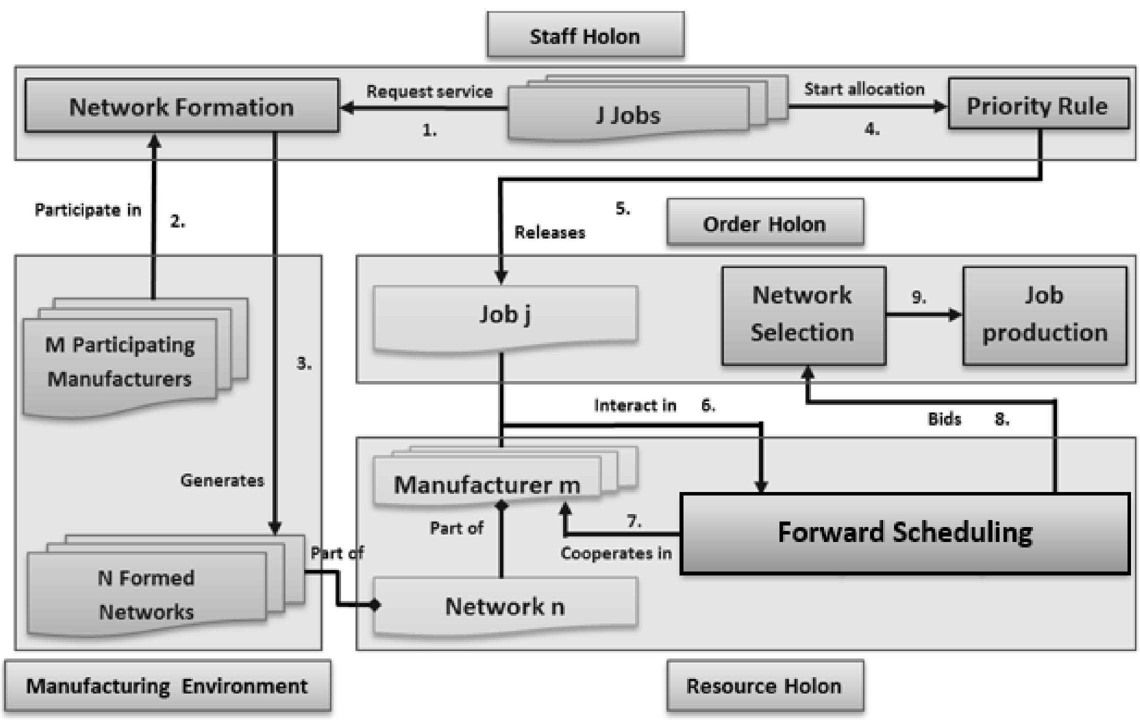


Figure 4.8: Activity diagram of resource, order, staff elements in the platform

The final stage of the flow shop scheduling process is performed by the selection of a network for each job. The platform accommodates several simulation settings for different tests to be carried out as shown in Figure 4.9. Eight output (R1-R8) were generated with combinations of simulation settings involving the following:

- Network selection by manufacturer bids
- Network selection by network bids
- Network selection by cost, quality, time selection criteria
- Job agent release priority

The priority rules tested are shortest processing time (SPT), shortest slack time (SLACK) and earliest due date (EDD). The simulation results are measured in terms of lead time, cost, and quality as well as delivery-on-time reliability. The best network is considered to be the one that is most compliant with the job requirements on lead time, quality and cost.

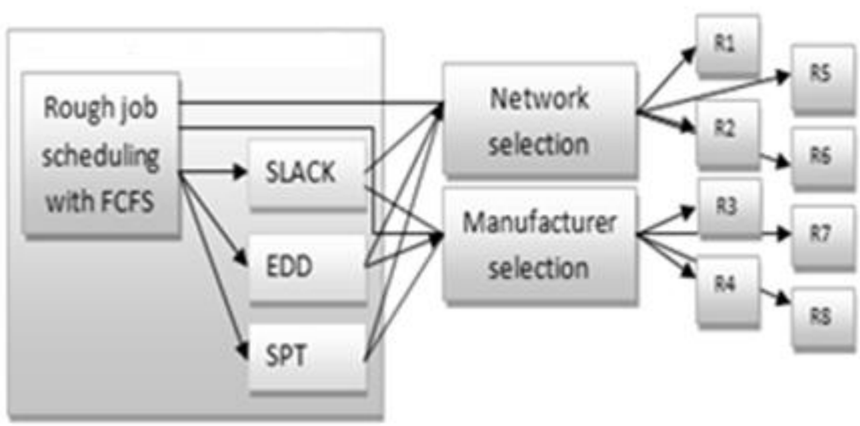


Figure 4.9: Combination of simulation settings

The LA19 and MT10 scheduling problems were not used in this case because the platform was not designed for these problems. However, the platform used the data collected from the use case company, which was presented in chapter 3, as a template for generating realistic data sets for the experiment. Each test has a fresh new data set of jobs and manufacturers generated. Five instances of simulation were carried out where each test resulted in eight outputs (R1-R8). Therefore, 40 results were collected.

4.4.4 Statistical tool for result evaluation

The tools available, for statistically comparing networks, are the one-way sensitivity analysis and the tornado chart. By varying one parameter by a specific amount, it is possible to evaluate the impact on the performance of the model. If more than one parameter is investigated, it is possible to find out which parameter has the greatest influence on the performance of the model. The performance impact can be recorded and graphically represented in a tornado chart (Taylor, 2009). In this research, the model used the scheduling algorithm. The output of the scheduling algorithm was measured by an objective metric. The objective metric for comparison was job lead time. In the case of a scheduling problem such as LA19 and MT10, there were 10 job lead times. The value, that influenced job lead times for this model, was operation variability such as delay, and remaining capacity. Also, the remaining capacity is dependent on job lead times. Providers of capacity are manufacturers and manufacturing networks. In order to compare networks, networks were used as the parameters of the sensitivity analysis. The median job lead time, of an undisturbed scheduling problem, was used as the base value of the tornado chart. From the tornado chart, the networks were compared on the following:

1) Number of jobs that have lead times below base value

2) Number of jobs that outperformed on lead time

3) Total lead time of all jobs

4) Lead time of last scheduled job

Results of the comparison were used to increase confidence about the correctness of the decisions taken by the network selection algorithm.

4.5 Detailed design of the knowledge based system

The objective of developing the knowledge base was to provide a means for a user of a multi-agent system to feed data into the system. The knowledge base designed should fulfil some specifications as follows:

- Scalability, not only in terms of the amount of data it can hold but also in terms of the amount of data structure it can contain as well as the number of links among data.
- Accessibility so that the knowledge base can be manipulated in various ways by the user as well as a multi-agent system. Functions for reading, writing, saving and copying the knowledge base should be easily integrated into the behaviour of agents.

Complex functions such as data linkage, data structuring and data merging should be accessible to agents.

- Portability so that the knowledge base can be uploaded, sent and downloaded from the web on any machines that can read and write to a text file.
- Self-contained so that all rules required to handle data in a knowledge base should be embedded in it.

Ontology web language (OWL) can fulfil those requirements and it was used to develop the knowledge base for the multi-agent system. Protégé ontology editor was the tool used for developing the knowledge base.

4.5.1 Developing the object and data structure of agents

An agent can have only two types of properties namely object property and data property. Both properties have two main features which are the domain and the range. The domain is the agent i.e. the object that owns the property. The range is the value of the property. For an object property, the value is an object and for a data property, the value is data e.g. an integer, a string. The identity of the property itself is called a predicate. Most of the object properties of agents identified in chapter 3 are shown in Figure 4.10 and most if the data properties are presented in Figure 4.11. For instance, the highlighted predicate 'precedes Jobwise' represents the equations $o_{j,i} = ppp_{o_{j,i+1}}$ and $o_{j,i+1} = spp_{o_{j,i}}$ so that the domain is an operation $o_{j,i}$ and the range is another operation $o_{j,i+1}$. The predicates can be linked e.g. 'precedes Jobwise' is the inverse of 'succeeds Jobwise'. For example, when the fact that operation_10 precedes operation_11, is asserted, the fact that operation_11 succeeds operation_10, is inferred. Figure 4.12 shows the object linkages that took place when the agents had their object properties defined.

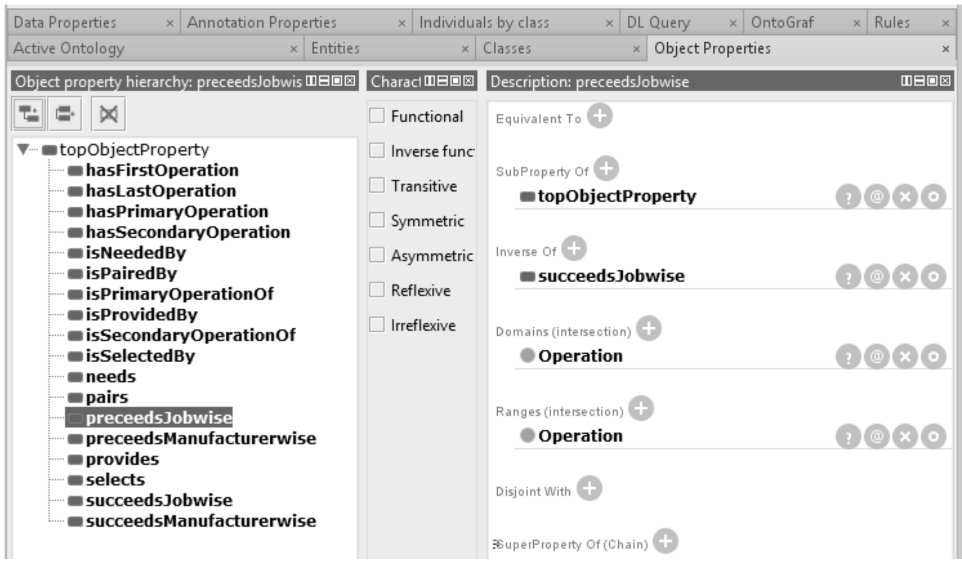


Figure 4.11: Object property of knowledge base in Protégé

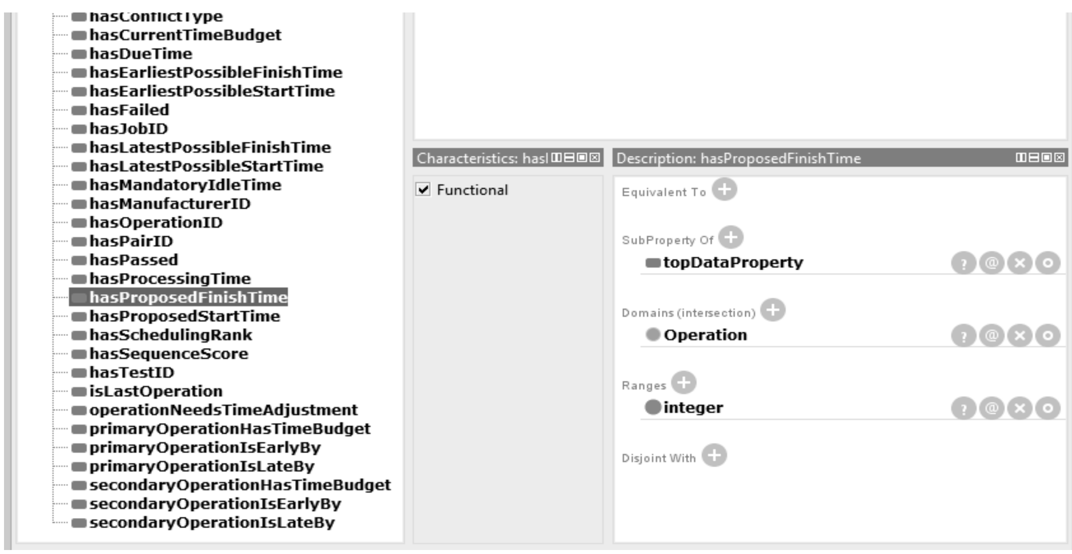


Figure 4.10: Data property of knowledge base in Protégé

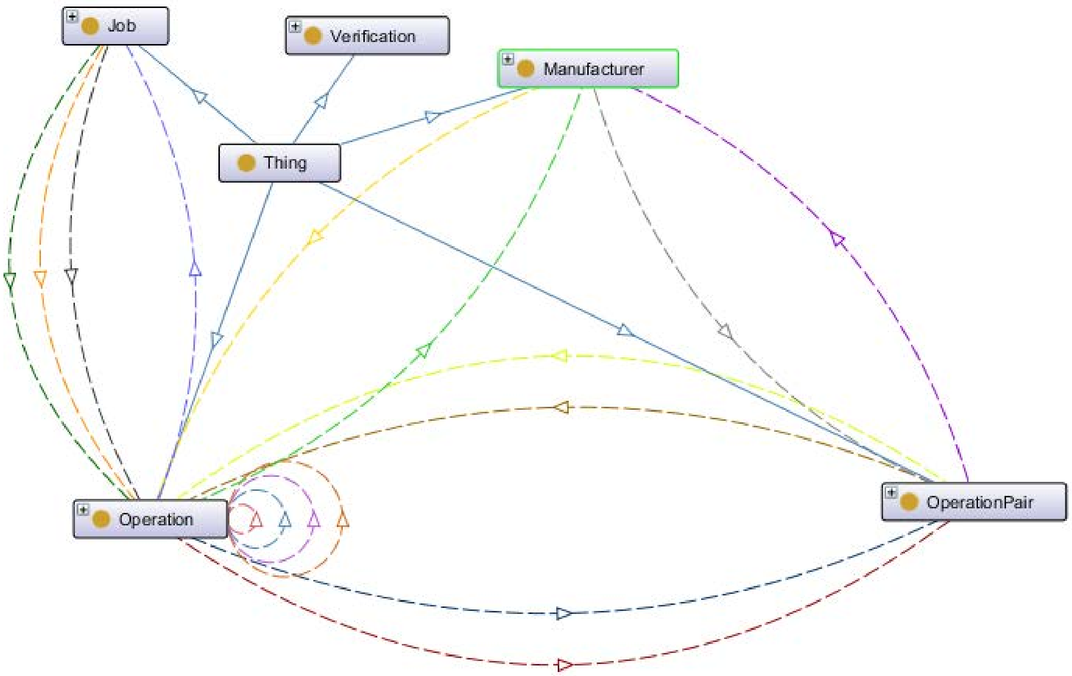


Figure 4.12: Relations among objects in the ontology as defined in chapter 3

4.5.2 Developing the data properties linkage

The knowledge base becomes a powerful source of knowledge for the multi-agent system when the data properties are linked. With linked data, a change in the value of an agent would also cause several other agents to see the change in their data properties. Data are linked by means of explicitly written rules. Data in OWL are linked by semantic web rule language (SWRL) as shown in Figure 4.13 and in the Appendix. Also data within an agent itself can be inferred from other data that it holds. SWRL were written from equations presented in chapter 3.

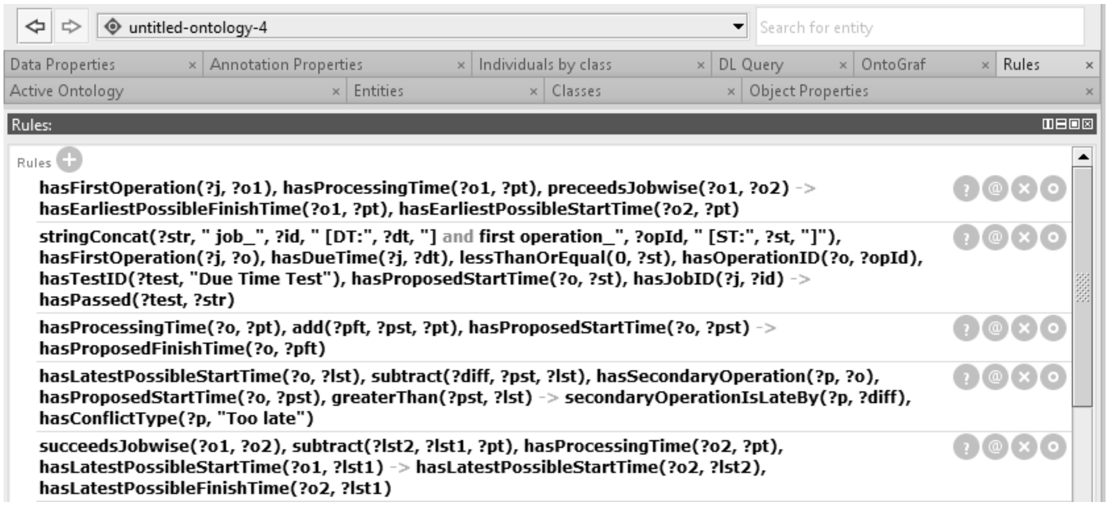


Figure 4.13: Relation between the data in the ontology as defined in chapter 3

SWRL rules are interpreted by a reasoner called Pellet, to infer the data that are highlighted in Figure 4.14. For instance 'operation_10 succeeds Jobwise operation_14' was inferred from 'operation_14 precedes Jobwise operation_10'. The aforementioned fact is not visible here but exists in the instance of operation_14.

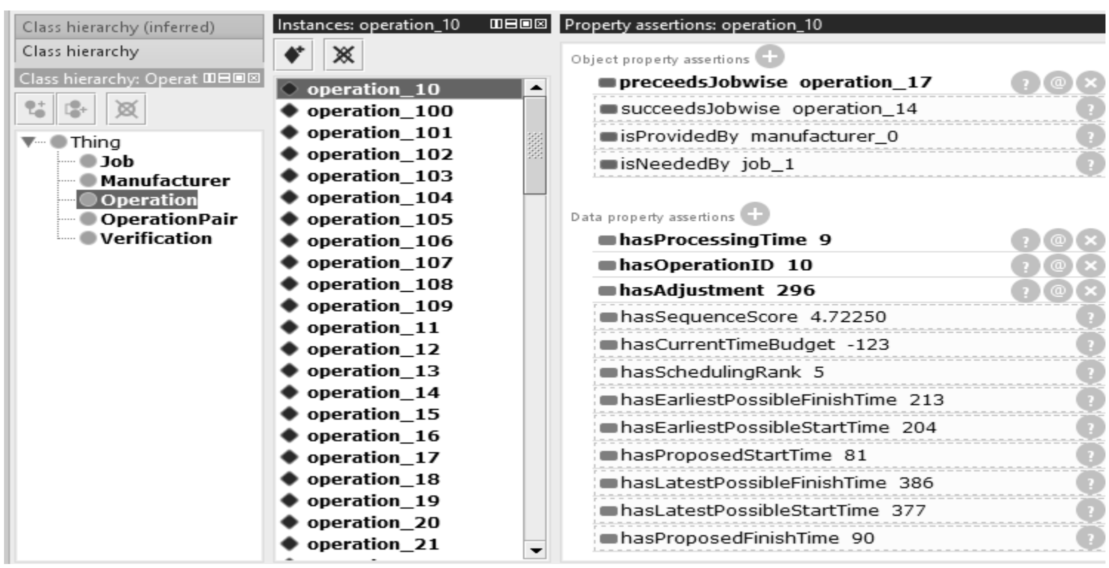


Figure 4.14: Data properties inferred from asserted data, linked by SWRL

Another use of SWRL was for the verification tests which were embedded in the knowledge base. The verification tests infer useful messages for the user when the knowledge base meet some criteria such as $pdtt_{v=j} = pst_{o_{j,i}} \geq 0$. The logic for due date test is shown in Figure 4.15.

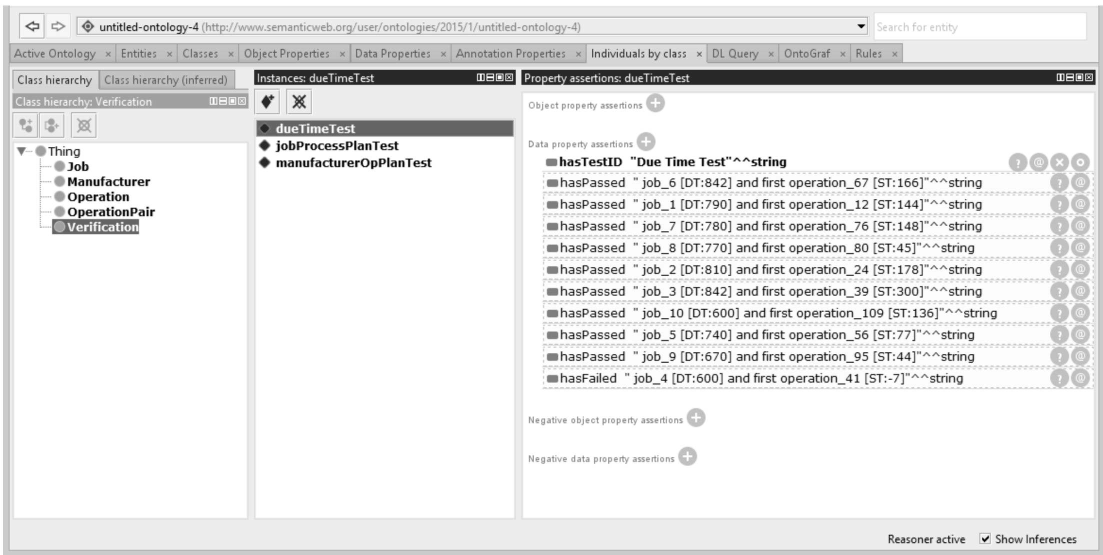


Figure 4.15: Messages inferred from asserted and inferred data

4.5.3 Linking the knowledge base to a multi-agent system

The knowledge base is linked to a multi-agent system for two purposes. The first purpose of the system is the generation of operation pairs. Some of the pairs are eventually selected by manufacturer agents to form operation plans. The second purpose is for adjusting the start times of operations during operation timing as well as conflict resolution. Figure 4.16 shows the operation pairs asserted into the ontology with object properties so that pair_100_50 has primary operation operation_100 and has secondary operation operation_50 as well as has pair identity pair_100_50. From that assertion, a number of data is inferred. The inferred data is used by the manufacturer agents to decide on which pairs to keep and which one to discard. The selected pairs form the operation plan of the manufacturer.

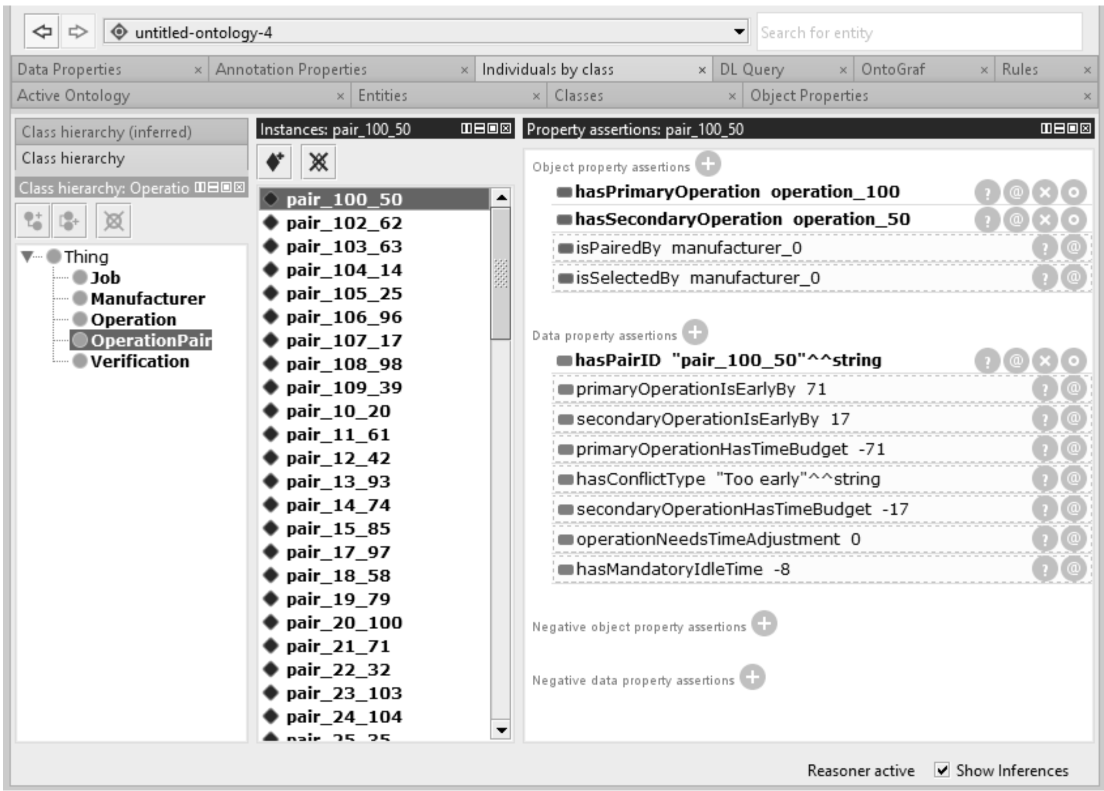


Figure 4.16: Operation pair data asserted into the knowledge base by a multi-agent system

Figure 4.17 shows an operation bounded by an operation plan and a process plan so that operation_10 precedes jobwise operation_17, succeeds jobwise operation_14, precedes manufacturer-wise operation_20 and succeeds manufacturer-wise operation_80. The pair agent, of the multi-agent system, performs an adjustment of 296 to correct the start time of operation_10.

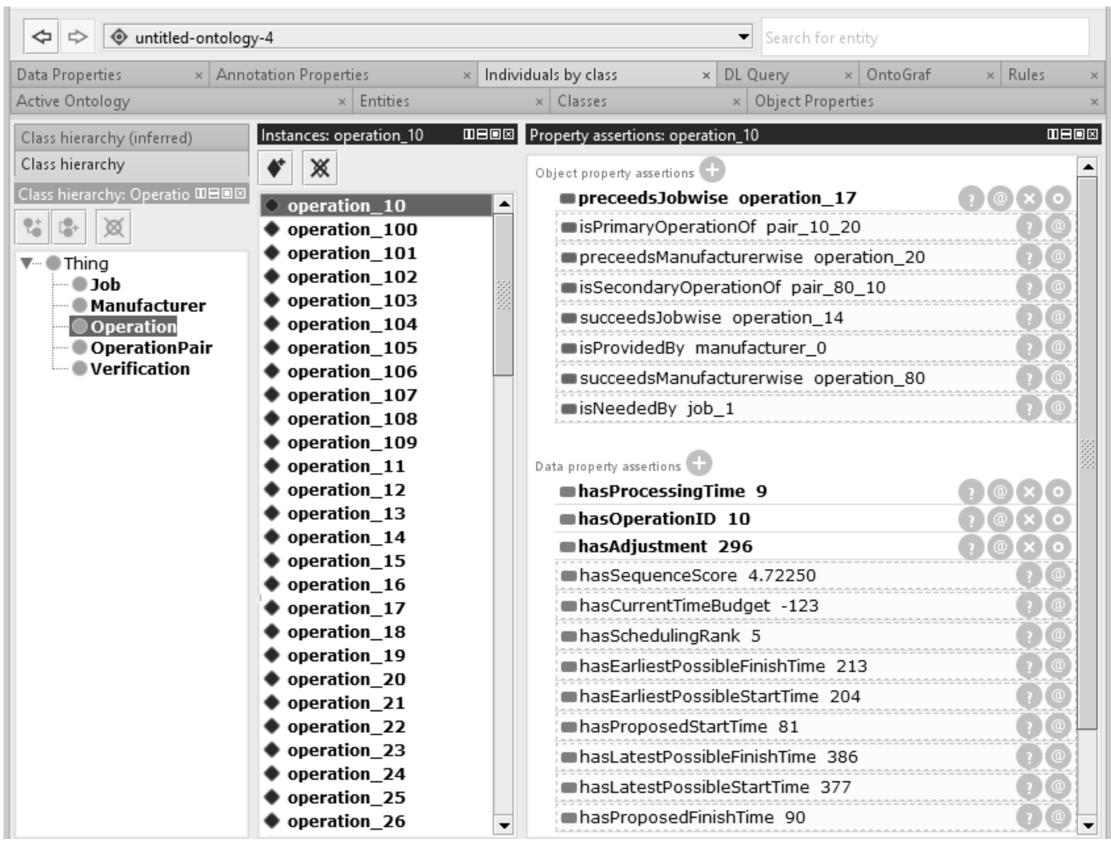


Figure 4.17: Inferred data giving rise to pieces of an operation plan

4.6 Detailed design of the multi agent system

The aim for developing the multi-agent system was to address the weaknesses of the knowledge base reasoner. Sorting functions cannot be expressed in SWRL and therefore the capability is outsourced. The system perform heuristically the preselection of operation pairs, followed by a final selection. The heuristic algorithms are basically a set of sorting functions that is executed by each manufacturer agent. The system was developed in the workflow agent development environment (WADE) and is presented as workflows. Figure 4.18 describes the functions of the support agent. The data from the ontology enables the agent to generate job agents and manufacturer agents. Agents live in containers created by the support agent and there can be several containers on a WADE platform. Also generated are the operation pair agents from the process plans of job agents. For n operations, $(n-1)$ pairs are created. In the case of manufacturer agents, the support agent generates all the possible operation pairs. For n operations, $n(n-1)$ pairs are created.

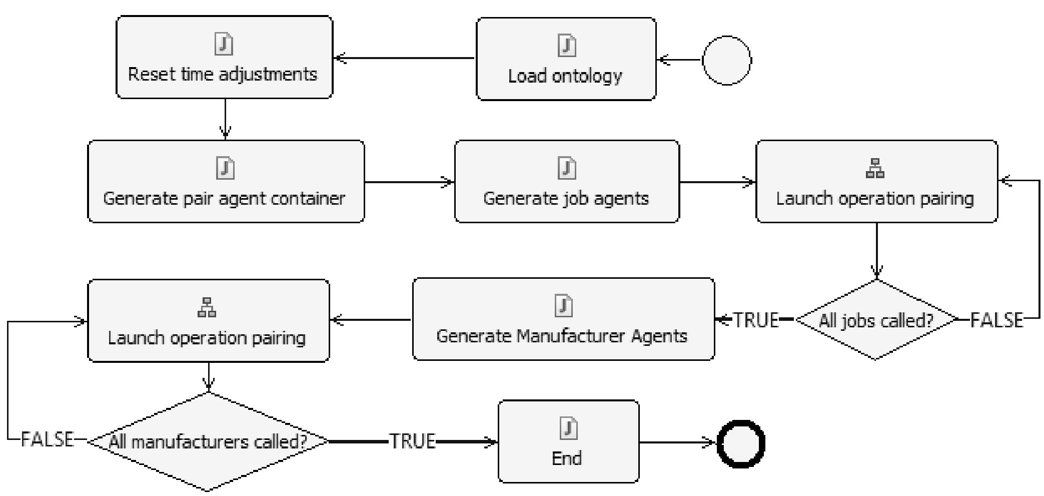


Figure 4.18: Workflow of the support agent

4.6.1 Operation pair generation for job agents

The support agent launches the sub flow for the generation of operation pairs from the process plan of the job agent. This is shown in Figure 4.19. The job agent is the performer of the sub-flow and therefore the latter is asynchronous i.e. the support agent can proceed with its workflow and does not wait for the sub-flow to complete. The sub-flow loads information from the ontology and generate the pair identities from the process plan. It does so by combining the identities of the operations adjacent to each other. A process plan of n operations will result in the generation of $(n-1)$ operation pairs. The knowledge base is checked for the existence of the pair. If the pair does not exist, an operation pair object is created, with object properties. The object properties are the primary operation and the secondary operation.

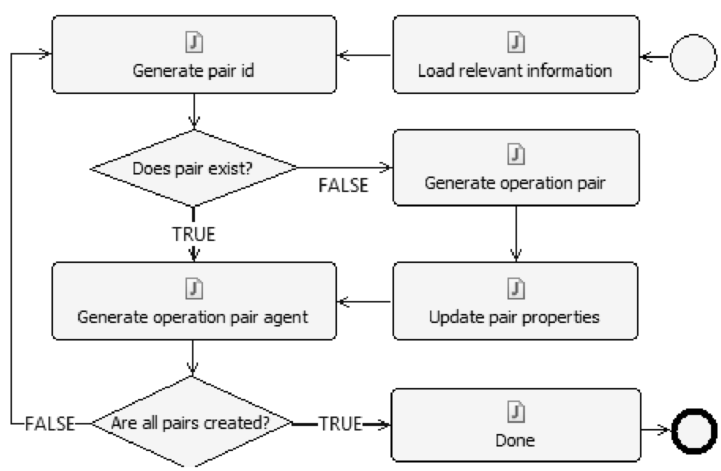


Figure 4.19: Sub-flow for operation pairing on behalf of job agents

An operation pair agent is created to represent each operation pair that exists in the knowledge base. The agents as well as their services are registered with the directory facilitator of the WADE platform as illustrated by Figure 4.20.

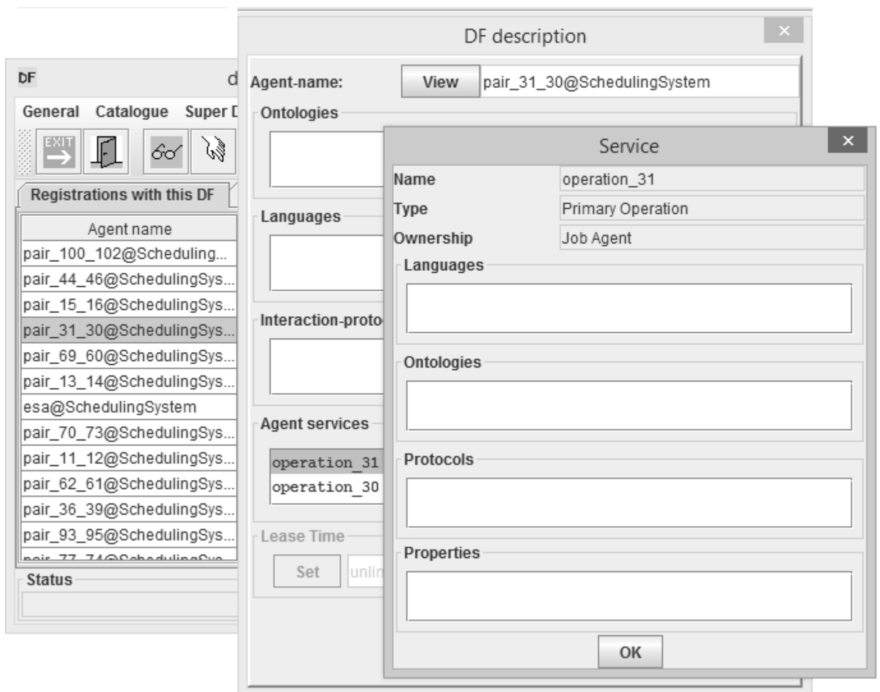


Figure 4.20: Registered operation pair agents with the directory facilitator in WADE

4.6.2 Pre-selection and selection of operation pairs by manufacturer agents

The support agent launches the sub-flows of the manufacturer agents. The latter load the ontology information and launches an asynchronous sub-flow for the generation of operation pairs as shown in Figure 4.21. This process is explained in more details in the next section. The knowledge base is updated with the operation pairs. An inference-based reasoning operation is performed on the knowledge base so as to infer the operation pair and operation properties as shown in Figure 4.22.

The pellet reasoner is embedded within each manufacturer agent using the Pellet API and OWL API. The seamless integration allows the agents to reason about the knowledge and infer the operations and pair properties. The operation pairs are pre-selected on the basis of their scheduling rank $srpp_{o_{j,i}}$ or their rounded sequence score $srop_{o_{m,i}}$. A pair is finally selected with a combination of the time adjustment sta and the primary operation time budget $potb_p$

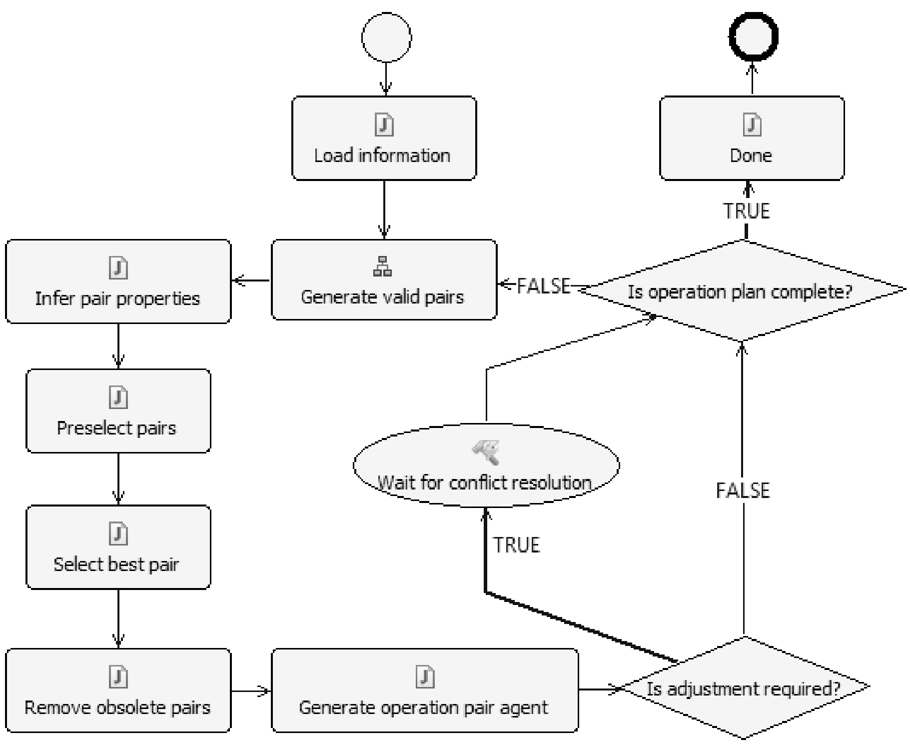


Figure 4.21: Workflow of manufacturer agent for pre-selecting and selecting operation pairs

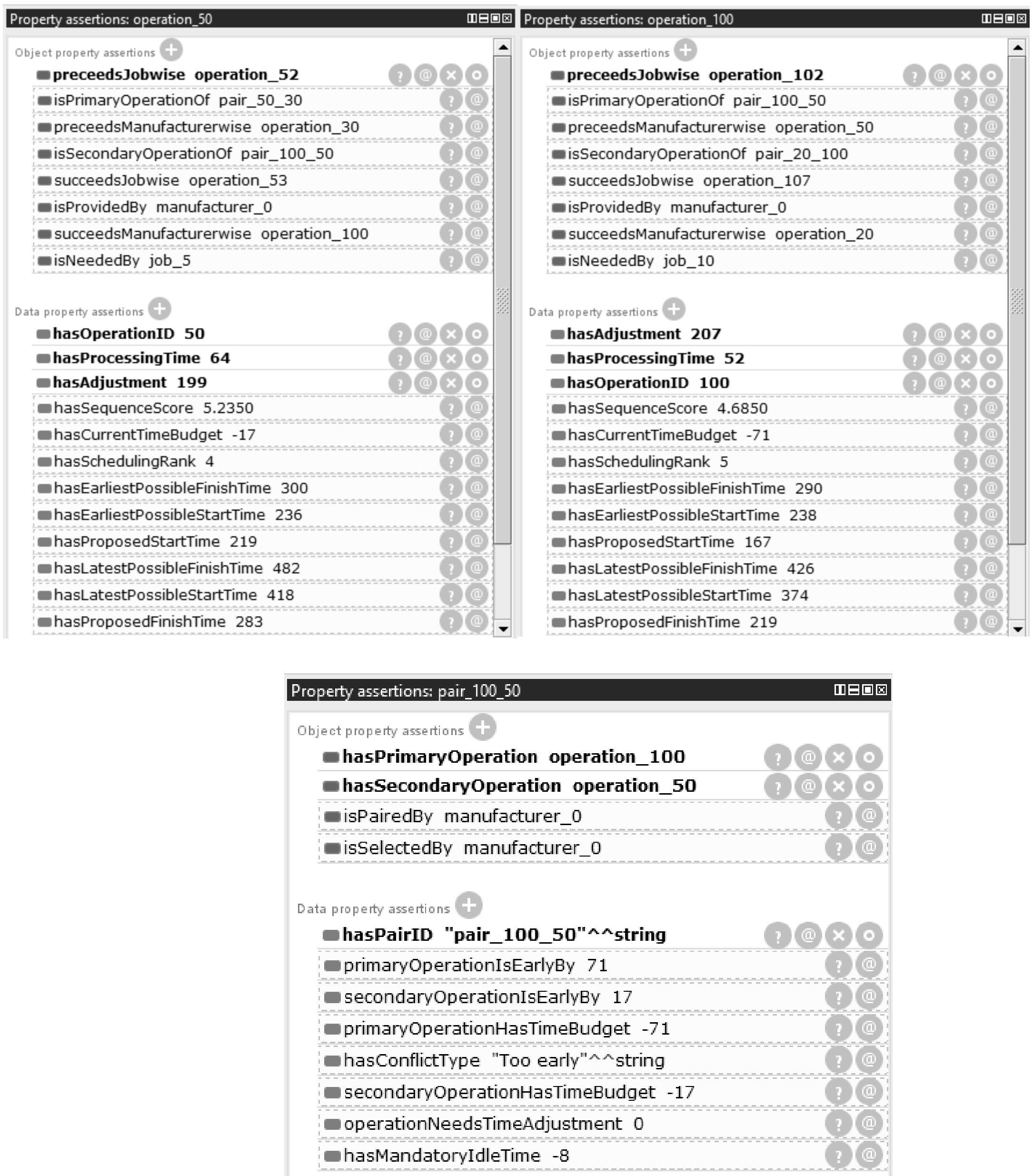


Figure 4.22: Operation pair properties inferred by reasoning agent

4.6.3 Operation pair generation for manufacturer agents

The manufacturer agent launches the sub-flow of generating operation pairs as shown in Figure 4.23. The agent loads the ontology and creates an operation pair identity by combining the identities of a primary operation and of a secondary operation offered by the manufacturer. The first set of pairs will consist of $n(n-1)$ pairs if the secondary operation has not been pre-defined i.e. the secondary operation is $o_{m,i}$. Otherwise, the first set of pairs will consist of n pairs. The case where the secondary operation is pre-defined is where the last operation heuristic rule has been applied so that the secondary operation is limited to $\{o_{m,i} | o_{j,5} \equiv o_{m,i}\}$. This is the case for the first set of operation pairs. The rest of the sub-flow consists of making sure that the pairs do not already exist, in which case they are destroyed.

When the sub-flow completes and the manufacturer has selected the first operation pair, the sub-flow is launched again until a stopping condition is reached. In the subsequent sub-flows, the secondary operation is defined as the primary operation of the previously selected pair. The set of potential pairs will continue to decrease in size so as to consist of $(n-x-1)$ pairs, where x is the number of pairs already selected by the manufacturer agent. The agent stops producing pairs when $x \geq n-1$.

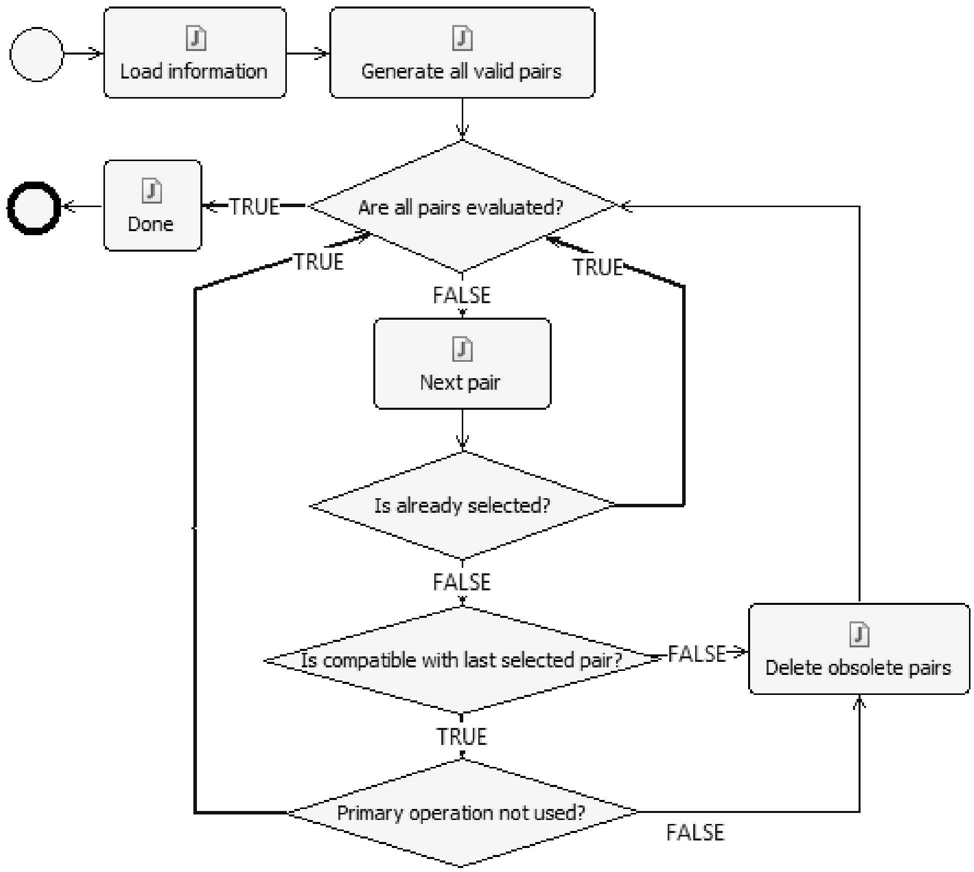


Figure 4.23: Sub-flow of operation pairing on behalf of manufacturer agents

4.6.4 Conflict resolution by operation pair agents

When an operation pair is selected by the manufacturer agent, the operation agent launches a conflict resolution sub-flow where start time adjustment is performed as shown in Figure 4.24. From the directory facilitator of the WADE platform, the operation agent calls on all operation agents whose secondary operation is equivalent to its primary operation, to adjust their start times. Upon adjusting theirs, those affected agents call on the next batch of agents, that they affected and ask them to adjust their start times. The workflow is the most time consuming to execute because of the chain reaction of adjustments and concurrent access to the knowledge base.

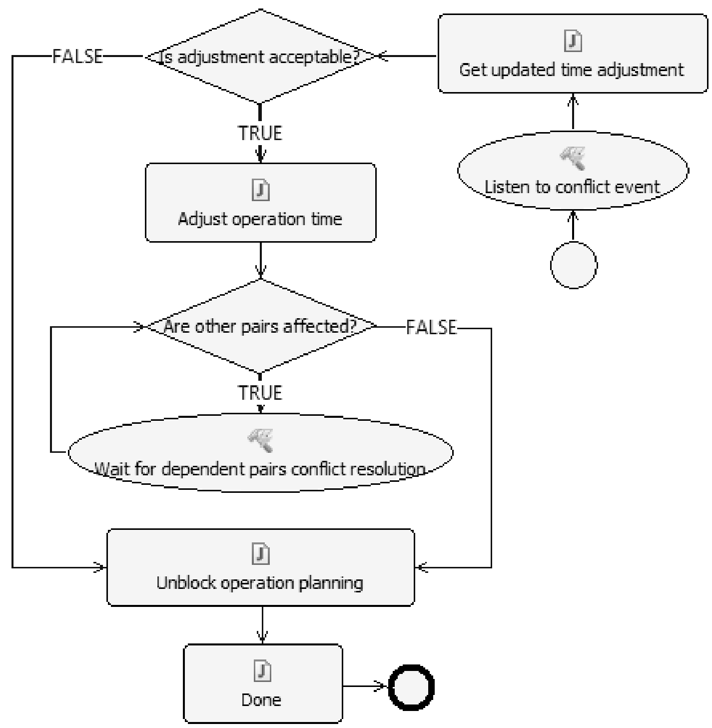


Figure 4.24: Workflow of operation pair agent in resolving schedule conflicts

4.7 Conclusion

The chapter presented the development of the experiment software for simulating different approaches to decentralised flow shop scheduling in a manufacturing network breeding environment. The different approaches involved manufacturer pairing, operation pairing and network selection on the basis of manufacturer bids or by network bids. Three main techniques were used including genetic algorithm optimisation, heuristic rules and agent based systems such as inference-based system, knowledge base and multi-agent system. Furthermore, the chapter presented thoroughly the design of a multi-agent system and the development of a knowledge base based on the entities defined in chapter 3.

EVALUATION OF CONFIGURATION MODELS

5.1 Introduction

The centralised approach to scheduling flow shop systems has been the preferred method for sometimes. Nowadays, the manufacturing environment has become more dynamic, horizontally integrated and the manufacturing entities more interdependent. Strong coordination is necessary but autonomy of management is a strength to be leveraged. Therefore, this chapter evaluate some models of manufacturing entities that define their own operation schedules and come together to integrate those schedules. The chapter empirically compares three approaches for scheduling a flow shop system in a decentralised manner. They are manufacturer pairing, operation pairing, network selection by network bidding and network selection by manufacturer bidding. The results are followed by discussion and recommendations for future improvements.

5.2 Local meta-heuristic optimisation versus welfare of manufacturing network

The manufacturer pairing approach to flow shop scheduling consists of four phases. Each phase undergoes genetic algorithm optimisation. In the first phase, the time budget objective function, of each manufacturer is maximised. The second phase involves manufacturer pairs and their goal is to maximise the pair compatibility objective function. The third phase involves the manufacturing networks and the network compatibility objective function is maximised. The fourth phase addresses the resolution of conflicts present in the network schedule.

In the first phase, the time budget objective function maximised the total distance of the operations from the critical time path. As shown by Figure 5.1, the proposed operation finish times (pft_o) were kept as far away as possible, from the green arrows which indicate latest possible finish times (lft_o). Manufacturer 1 achieved a total distance of 1960 hrs away from the critical time path, in other words, a maximum time budget ($max \sum tb_o$) of 1960 hrs was achieved. During phase, the manufacturer has no idle time scheduled in. Variants for optimisation were set for a population size of 500 and a swapping mutation rate of 50%. After 35 iterations, a maximum time budget was achieved as shown in Figure 5.2.

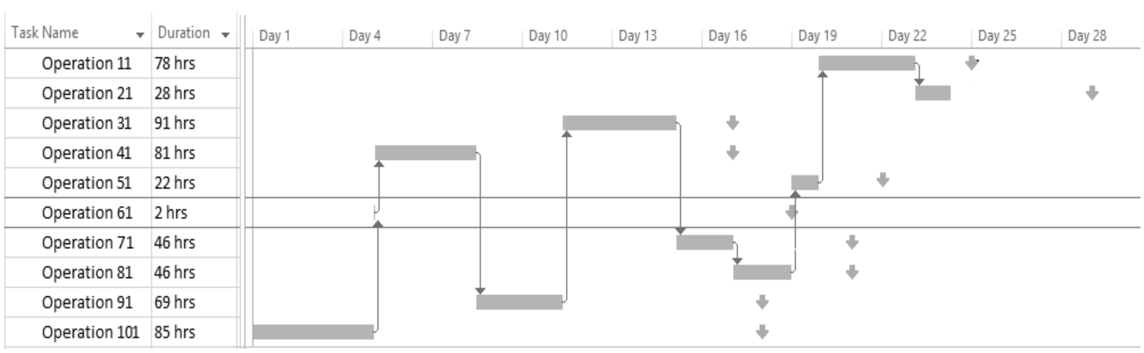


Figure 5.1: Operation plan of Manufacturer 1 with latest finish times

In the second phase, the pair compatibility objective function reduces the likelihood of invalid relations to prevail among the operations of the primary manufacturer $(o_{pm,x})$, those of the secondary manufacturer $(o_{sm,y})$ and the jobs $\{j \mid \{o_{pm,x}, o_{sm,y}\} \subset O_j\}$. Invalid situations include for example, the first operation of job 1 being sequenced as the last operation of manufacturer 6.

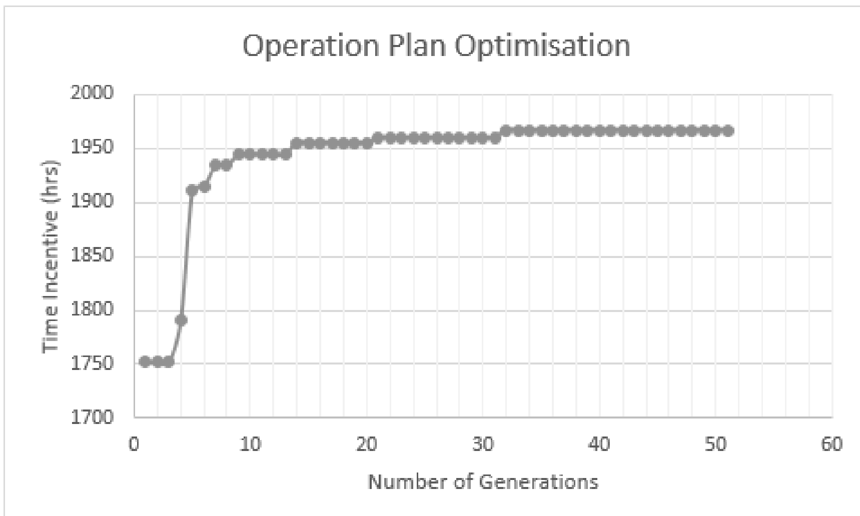


Figure 5.2: GA optimization of the time budget for Manufacturer 1

The focus of this optimisation has shifted from the manufacturer alone to jobs and two manufacturers. In this phase, constraints involve the process plan of jobs, operation plans of manufacturers and the manufacturer time budgets. Pair compatibility is an inverse function of the total idle time and the total negative $(-)$ time budget incurred by the pair of manufacturers, as the pair sequences its operations. The pair having the highest compatibility is likely to contribute to a good schedule. Figure 5.3 shows the optimisation plateaued at a pair compatibility of 17 with a compatibility cap of 100. The pair solutions are competing along a curve of type $\frac{1}{x}$.

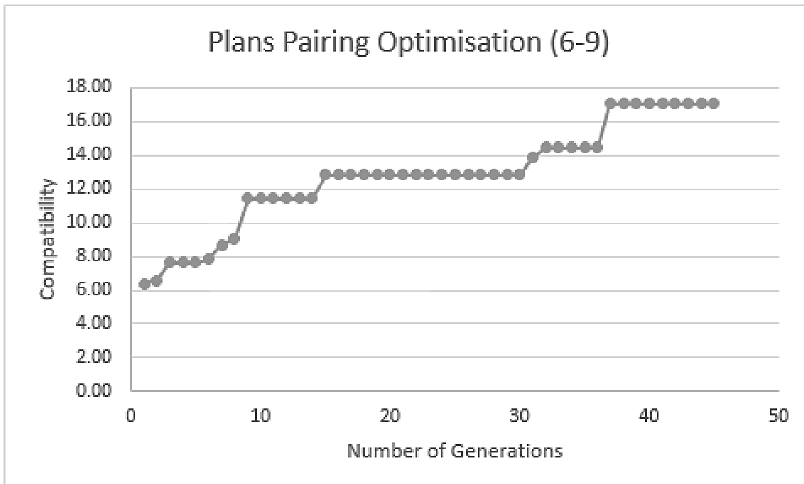


Figure 5.4: GA optimisation of operation plans for Manufacturer 6 and 9

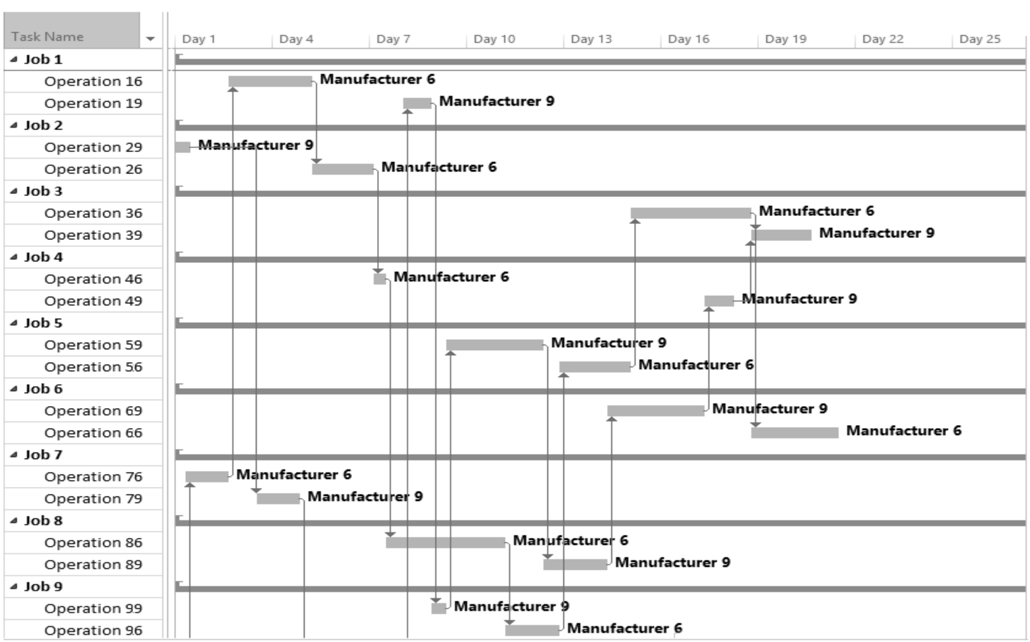


Figure 5.3: Combined operation plans of Manufacturer 6 and 9 after pairing optimisation

A population size of 200 and a swapping mutation rate of 25% were set and returned the best compatibility after 40 generations. The result was quite good where the budget time deficit was zero and the total idle time was 6 hrs as shown by the Gantt chart in Figure 5.4.

The third phase gathers valid manufacturer pairs with high compatibility so as to maximise the network compatibility objective. A manufacturing network is formed by matching adjacent pairs that share a similar manufacturer. For instance, given the machine pairs like the ones presented in Table 5.1, the pairs aggregate to form a network 0-5-7-2-3-8-1-9-6-4-0.

Table 5.1: Final scheduling of operations by network

Manufacturer	**TD (h)**	**AD (h)**	**I (h)**
Manufacturer 0	493	585	92
Manufacturer 1	548	754	206
Manufacturer 2	556	730	174
Manufacturer 3	631	1006	375
Manufacturer 4	534	1047	513
Manufacturer 5	416	919	503
Manufacturer 6	491	1022	531
Manufacturer 7	499	1125	626
Manufacturer 8	531	1085	554
Manufacturer 9	410	1047	637

TD = Theoretical Duration, AD = Actual Duration, I = Idle Time, h = hours

Finally, the fourth phase re-uses the objective function that was employed in the second phase to synchronise and re-optimise the operation plans of the network. New pair compatibilities ($mpc = 39 \pm 24; n = 10$) are generated with coefficient of variation 0.6 compared to their initial optimisation round. The resulting schedule is presented in Table 5.2 and the make span achieved for the MT10 problem was 1125 hrs.

Table 5.2: Operation plan sequencing to form a manufacturing network

Pairs	Operation Plan 1	Operation Plan 2	C
0-5	90-70-100-80-10-40-20-50-60-30-	65-75-105-85-95-55-15-35-25-45-	79
7-5	47-77-97-37-57-87-17-27-67-107-	65-75-105-85-95-55-15-35-25-45-	70
2-7	82-62-52-102-42-72-22-12-92-32-	47-77-97-37-57-87-17-27-67-107-	61
3-2	73-63-93-103-13-33-53-23-43-83-	82-62-52-102-42-72-22-12-92-32-	42
8-3	108-68-48-78-88-38-58-98-18-28-	73-63-93-103-13-33-53-23-43-83-	32
1-8	71-101-61-41-91-81-31-11-51-21-	108-68-48-78-88-38-58-98-18-28-	32
9-1	109-69-79-29-99-89-49-59-39-19-	71-101-61-41-91-81-31-11-51-21-	26
6-9	106-76-46-86-96-16-26-66-36-56-	109-69-79-29-99-89-49-59-39-19-	19
4-6	44-84-24-14-54-94-104-64-74-34-	106-76-46-86-96-16-26-66-36-56-	15
0-4	90-70-100-80-10-40-20-50-60-30-	44-84-24-14-54-94-104-64-74-34-	13

5.3 Local heuristic sequencing versus overall scheduling optimality

Manufacturing pairing with GA optimisation was an important concept of subdivision of a scheduling problem and distributed problem solving. However, it is believed that the concept was over-engineered and that the concept of decentralisation, itself, can be powerful enough to solve flow shop scheduling problems. Therefore, this experiment investigated about the optimality loss of scheduling with simple and light heuristic algorithms that are self-contained in independent manufacturer agents. These algorithms were embedded in four functions of the manufacturer agent namely 1) pair generation, 2) pair pre-selection, 3) pair selection and 4) pair conflict resolution. This approach is called operation pairing. The approach is a degree of granularity higher than the manufacturer pairing approach which revealed better insight in flow shop scheduling.

5.3.1 Observations from operation pair generation

To create an operation plan of (n) operations, $(n-1)$ operation pairs are needed. Generation of operation pairs, for the purpose of creating an operation plan can be started in two ways. In the first approach, the secondary operation $so_{sp_{m,1}}$ can be any operation $o_{m,i}$. In the second approach, the secondary operation is limited to $\{o_{m,i} | o_{j,5} \equiv o_{m,i}\}$ which represents the last operation in the process plan of a job (j). This is called the last operation heuristic rule.

It was observed that the first approach was likely to generate operation sequences with discrepancies so that at least one operation was linked to another operation and the latter was part of a chain that linked back to the former operation. When these operations underwent scheduling, they perpetually undermined each other's timing, the so called chicken-and-egg problem. On the other hand, it was observed that the second approach, combined with appropriate pair preselection would always produce manufacturer operation plans in harmony with other manufacturers' plans.

5.3.2 Observations from operation pair pre-selection

The ranking of the primary operation (po) determines which operation pairs, of all operation pairs generated, will be pre-selected next by the manufacturers. Ranking is defined by $srpp_{o_{j,i}}$ or $srop_{o_{j,i}}$. A preselection will always contain one or more operation pairs whose primary operations share the same rank. Table 5.3 shows that algorithm D_SROP performed best. The comparison was on the basis of how well the pre-selection algorithm, together with the pair selection algorithm, reduced the make spans of the MT10 and LA19 problem.

The preselection algorithms experimented with were:

- D_SRPP → descending sequencing rank in process plan
- D_SROP → descending sequencing rank in operation plan
- N_SR → no sequencing rank

Table 5.3: Comparison of pre-selection algorithms

Algorithm	D_SRPP	D_SROP	N_SR	
D_SRPP better than		0	1	1
D_SROP better than	1		1	2
N_SR better than	0	0		0

Due to the last operation heuristic rule, sequencing is performed from the last operation, down to the first operation. Combined with the last operation heuristic rule, the pre-selection process successfully mitigated the likelihood of discrepant sequencing of operations.

5.3.3 Observations from operation pair selection

Within each preselection, one operation pair is selected, contributing one piece to the development of an operation plan. Table 5.4 shows that algorithm MIN_POTB_STA is best for the pair selection. The algorithms were compared on the basis of make spans for the MT10 and LA19 problems.

Table 5.4: Comparison of selection algorithms

Algorithm	MAX STA	MIN STA	MIN POTB	MAX POTB	MIN POTB STA	
MAX_STA better than		0	0	0	0	0
MIN_STA better than	1		1	1	0	3
MIN_POTB better than	0	1		1	0	2
MAX_POTB better than	0	0	0		0	0
MIN_POTB_STA better than	1	1	1	1		4

The algorithms experimented with were:

- MAX_STA → maximum adjustment $\max(sta)$
- MIN_STA → minimum adjustment $\min(sta)$
- MIN_POTB → minimum primary operation time budget $\min(potb)$
- MAX_POTB → maximum primary operation time budget $\max(potb)$
- MIN_POTB_STA → $\min(potb - sta)$

For the LA19 problem, a maximum make span of 1174 hours was achieved, which amounts to an optimality loss of 39% while for MT10 problem, a make span of 1239 hours was achieved which amounts to an optimality loss of 33%. When job due times were specified for LA19, the optimality improved, where optimality loss was reduced from 39% down to 35%. These results were obtained with an algorithm combination of D_SROP and MIN_POTB_STA. Figure 5.5 shows the manufacturer operation schedule with operation pair conflict resolution applied.

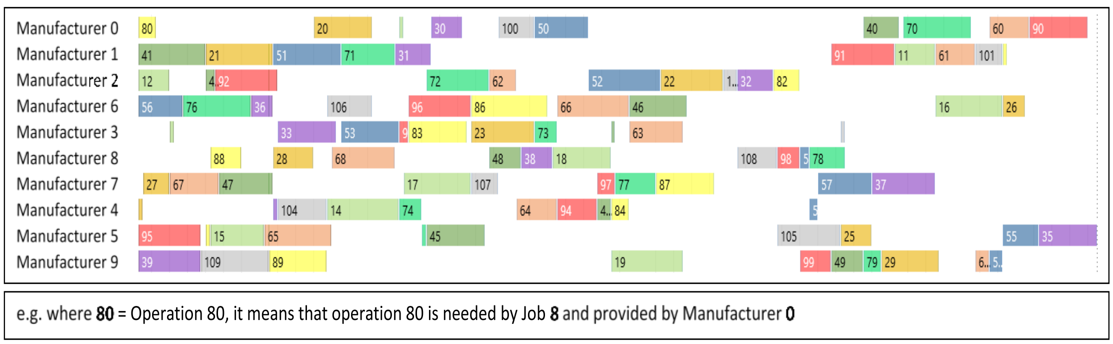


Figure 5.5: Manufacturer schedules for LA19 with achieved make span of 1174 hours

5.3.4 Discussion about the pre-selection algorithm

In a process plan, the position of operations is fixed and is represented by $srpp_{o_{j,i}}$. On the other hand, $srop_{o_{m,i}}$ gives a time-based position to operations, a position that is independent of a process plan. When building up an operation plan, there are equal benefits of pre-selecting operation pairs based on the value $srpp_{o_{j,i}}$ or $srop_{o_{m,i}}$. First, the preselection reduces the solution space for operation pair selection. Second, any operation pair within the preselection list would contribute to a valid operation plan. Third, the resulting operation plan will not cause timing conflicts with other operation plans. D_SROP outperformed D_SRPP on the basis of its positive sensitivity to job due times as well as processing times of operations. This is the reason why the optimality of the LA19 solution increased when job due dates were specified.

5.3.5 Discussion about the selection algorithm

The preselected set of operation pairs is then reduced to one operation pair. All operation pairs in that set have the same secondary operation but different primary operations. The algorithm MIN_POTB_STA utilizes data from both the primary and secondary operations. The equation of the algorithm which is $tb_{po_p} - sta_p$ is also equivalent to $pst_{so_p} - eft_{po_p}$, where pst_{so_p} is the proposed start time of the secondary operation and eft_{po_p} is the earliest finish time boundary of the primary operation. In the context of backward sequencing i.e. when the last operation heuristic rule is used, eft_{po_p} is not sensitive to processing time, and therefore neither is algorithm MIN_POTB_STA. When the algorithm was changed to $tb_{po_p} - sta_p - pt_{po_p}$, the presence of processing time decreased the effectiveness of the algorithm.

MIN_POTB_STA is also a measure of conflict. The pair which scored the least was selected because it is the pair where the primary and secondary operation needed the least or no adjustment of start time. Also, it prioritises the primary operation that has the least remaining budget. This is why MIN_STA and MIN_POTB are among the most effective algorithms.

For each selection round, the primary operation time budgets decrease, not only for the selected pair but also for the other unallocated pairs. If the primary operation was not to be selected as part of the current pair and instead was to be selected as part of the next pair, it would eventually have a negative budget. A negative budget means that the boundary system has been exceeded.

5.4 Order auctioning with network bidders versus manufacturer bidders

This experiment explored the scenario where a number of manufacturing networks are available for job allocation. The hypothesis is that selecting networks, on the basis of network bids, yields better performance than a selection based on manufacturer bids. For the LQC objective, performance was measured in terms of lead time, quality and then cost as well as delivery reliability. Moreover, the impact of job release priority was investigated. Release priority rules determine the next job to be allocated by, first-come-first-served (FCFS), earliest due date (EDD), shortest theoretical lead time (SLT), and shortest slack time (SLACK). The job slack time is the difference between the job due date and the job theoretical lead time.

5.4.1 Results about network selection strategies

Datasets for five orders of 20 jobs and a number of manufacturing networks were generated from a template of industrial data. 40 results were generated as presented in Figures 5.6 -5.9.
The best order lead time was obtained when jobs were fulfilled by manufacturing networks that were selected based on network winner bids. Also, the performance of selected networks, based on manufacturer bids, was found to be the most sensitive to job release priorities. The rules, earliest due date (EDD) and shortest theoretical lead time (SLT) were the best job release strategy

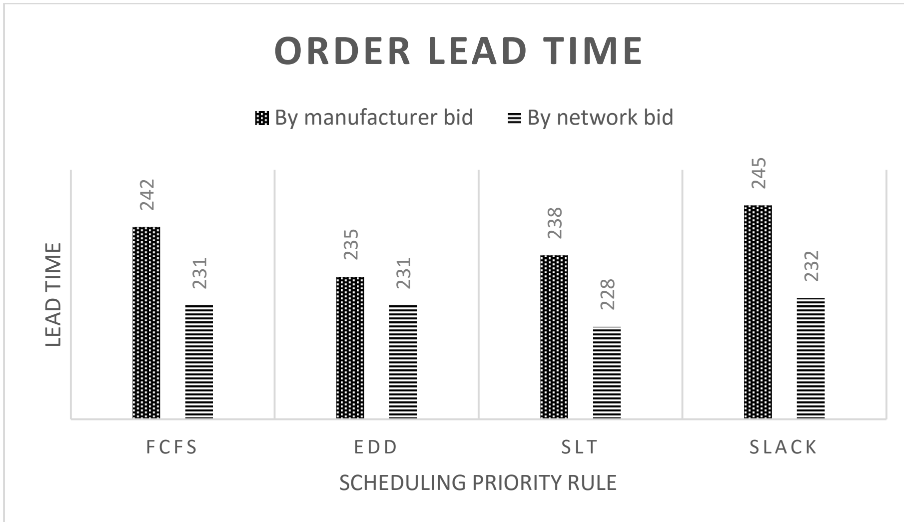


Figure 5.6: Comparison of average order lead time (LQC)

.

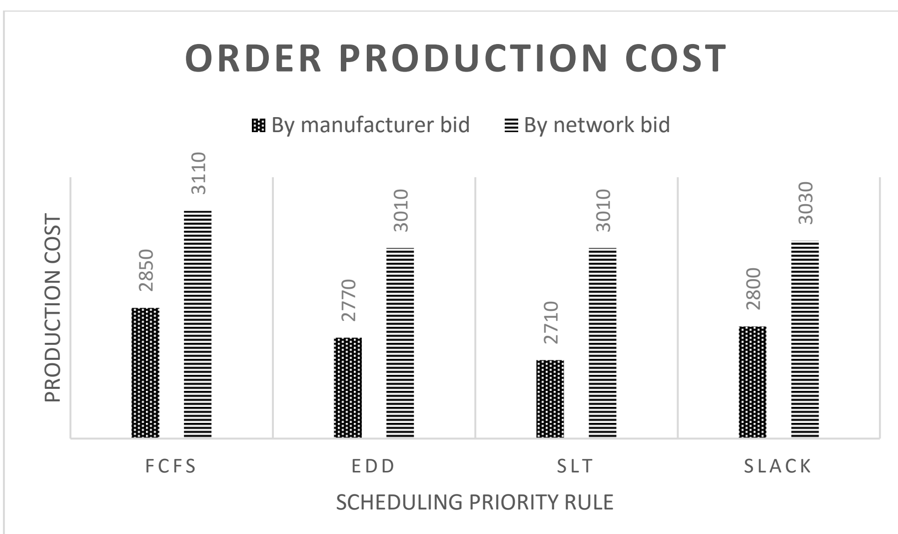


Figure 5.7: Comparison of average production cost (LQC)

Manufacturers bidding for jobs were found to be more competitive on price than bidding networks. The best job release strategies were EDD and SLT. Both network selection strategies were equally sensitive to the release strategy used.

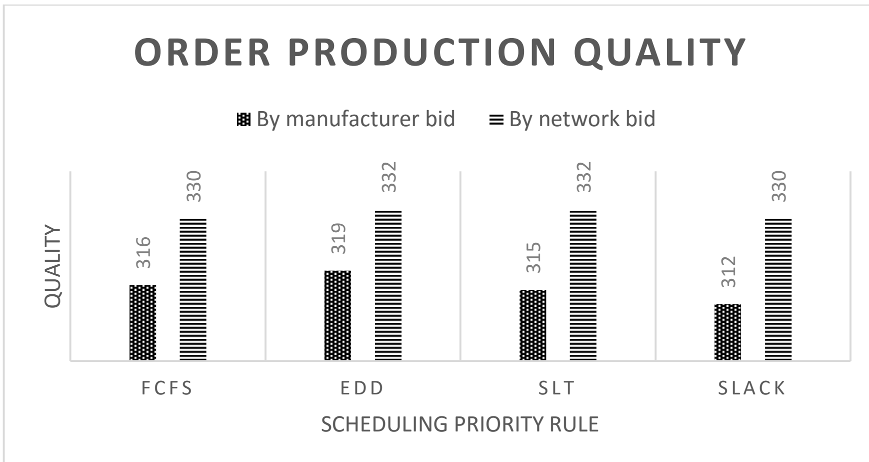


Figure 5.8: Comparison of average production quality (LQC)

On the basis of quality, bidding networks outcompeted the bidding manufacturers. SLACK and FCFS were the best job release strategies. Network bidding strategies were negligibly affected by job release strategies. Networks selected on the basis of network bids achieved their due dates more reliably than networks selected based on bidding manufacturers. FCFS and SLACK were the best job release strategies. The job release strategies affected negligibly the performance of selected networks.

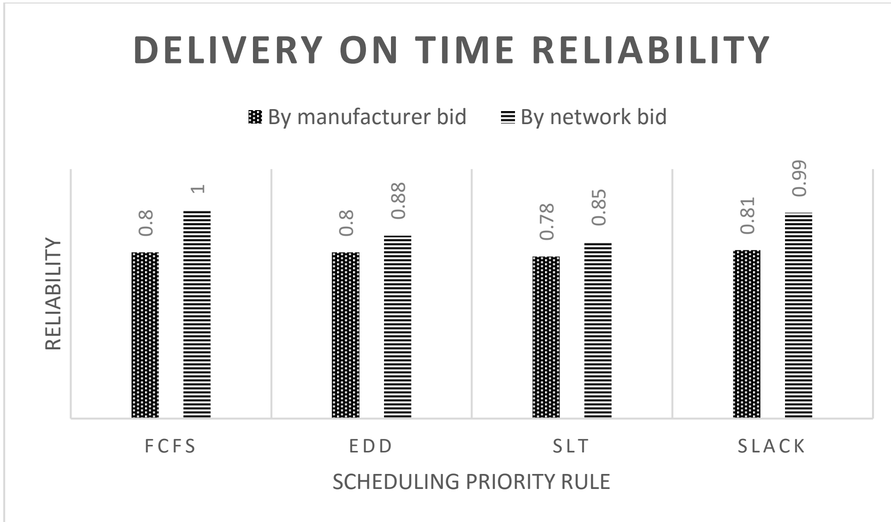


Figure 5.9: Comparison of average delivery on time reliability (LQC)

5.4.2 Discussion about job release and bidding strategies

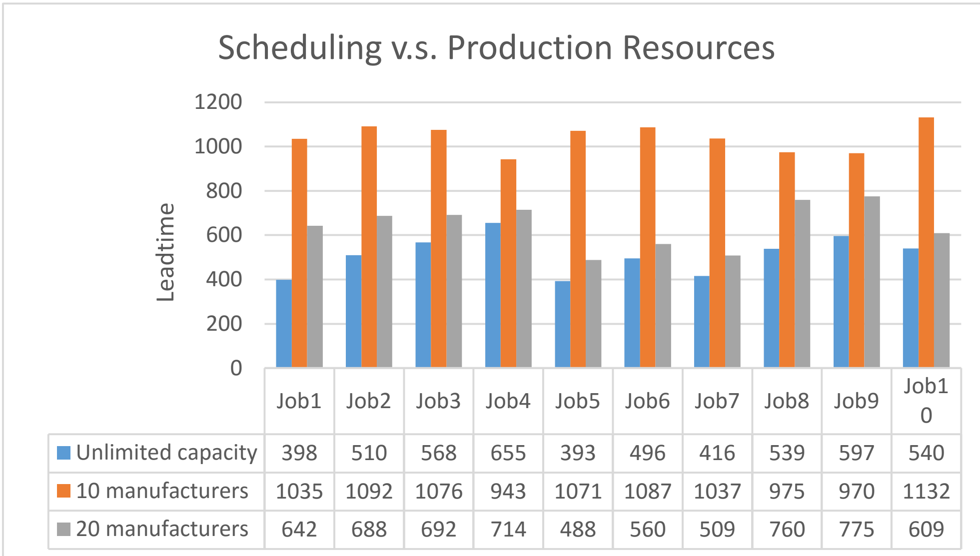
Each job had the LQC objective, which means that network or manufacturer bids were considered, if they were within lower and upper bounds of acceptable lead time, quality and cost thresholds. The first approach, which is network selection by manufacturer bidding, performed best on criteria of production cost. The second approach, which is network selection, by network bidding, performed best on criteria of lead time, quality and delivery on time. For lead time and quality, the second approach was the most stable, with respect to any job release priority used. For cost and delivery reliability, the approach became more sensitive to job release strategies. Therefore, for jobs with LQC objectives, the second approach combined with SLACK job release strategy, would yield the best network performance.

5.5 Impact of delays on network lead time performance

An investigation was performed on the lead time sensitivity of the scheduling and network selection algorithms, when an operation is delayed. Three characteristics were considered when choosing which operation would undergo a delay. First, the operation which is in first position in its operation plan is considered and thus called first-planned operation. Second, the operation which is succeeded by a slack, in its operation plan, is called single-slacked operation. And finally, the operation which is succeeded by slacks, in its operation plan as well as in its process plan, is called double-slacked operation. The slacked operations underwent delays, as a percentage of their processing times. Moreover, redundant manufacturers are pre-requisites for the existence of alternative networks. Therefore, in addition to the 10 manufacturers from the MT10 problem, twin manufacturers were introduced thereby bringing the total to 20. All operations of each manufacturer can be performed by its twin manufacturer and the two manufacturers are labelled by the suffix A and B. For instance, manufacturers 1A and 1B are what were called twin manufacturers.

5.5.1 Setup of MT10 problem for the network context

When twin manufacturers were introduced, there were more capacity to handle the MT10 problem. The proposed algorithm, which consisted of scheduling by operation pairing and network selection by network bidding, allocated the extra resources, which significantly reduced job lead times. This is illustrated in the Figure 5.10.



	Job1	Job2	Job3	Job4	Job5	Job6	Job7	Job8	Job9	Job10
Unlimited capacity	398	510	568	655	393	496	416	539	597	540
10 manufacturers	1035	1092	1076	943	1071	1087	1037	975	970	1132
20 manufacturers	642	688	692	714	488	560	509	760	775	609

Figure 5.10: Impact of twin manufacturers on job lead times

With twin manufacturers introduced, the workload is shared, so that manufacturers are each able to handle 10 or less operations. Also, each job now has a choice between alternative networks. Each network offers all the operations needed by the job. The network selection algorithm selects and allocates one network per job. An order consists of a number of jobs and is fulfilled by a group of networks. In Table 5.5, it is detailed where the operations required by jobs were allocated, with the objective of reducing total lead times.

Table 5.5: Current operation plans of twin manufacturers (MT10)

Manufacturer	Operation plan	Manufacturer	Operation plan
M0A	90-20-10-40-100-70-60	**M0B**	80-30-50
M1A	41-31-101-71-11	**M1B**	91-81-61-51-21
M2A	82-42-62-52-102-92-72	**M2B**	22-32-12
M3A	93-33-73-23-103-83	**M3B**	63-53-13-43
M4A	24-44-54-104-94-64-74	**M4B**	84-14-34
M5A	85-65-75-105-45	**M5B**	95-55-15-35-25
M6A	106-46-76-96-66	**M6B**	86-26-16-36-56
M7A	47-37-97-87-77-67-107	**M7B**	57-17-27
M8A	108-68-88-58-78-98	**M8B**	48-38-18-28
M9A	29-109-79-69	**M9B**	99-89-59-49-39-19

5.5.2 Sensitivity analysis of scheduling and network selection algorithms

In order to investigate the sensitivity of scheduling and network selection algorithms when an operation is delayed, a specific operation, to be delayed, was selected for each of three experiments. The analysis was focused on lead time and did not consider cost and quality. The algorithm selects the best network group based on the total lead time of the jobs that constitute an order. This is the best group of networks for the whole MT10 order. Specific operations namely first-planned, single-slacked and double-slacked operations are presented in Figure 5.11. Operations 90, 41 are first planned operations. Operations 33, 70 are single-slacked operations. Operations 10, 95 are double-slacked operations.

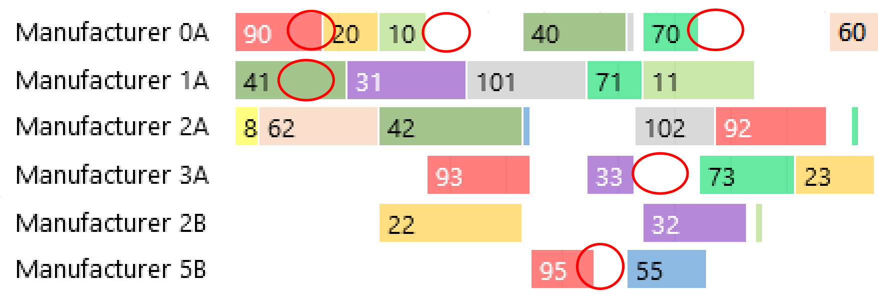


Figure 5.11: Delayed operations from MT10 schedule

When the first-planned operation **90** acquired a delay of 10% of its processing time, the current network for Job 9 was replaced by a new network where manufacturer M0A was substituted for M0B. Operation 90 offered by manufacturer M0B had no delay.

In this case, action was taken to replace the faulty manufacturer M0A but action may not have been taken. The case, for going for the next best network or not, for Job 9, is illustrated in Figure 5.12. The base value, for all the following tornado charts, is 665 which is the median of the job lead times when there is no delay, as previously presented in Figure 5.10.

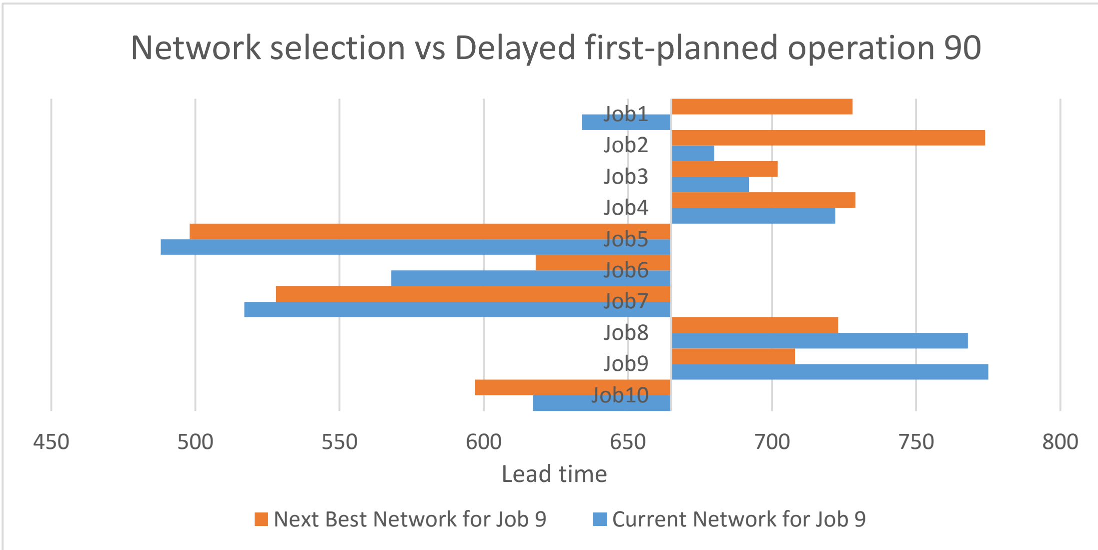


Figure 5.12: Sensitivity analysis of delayed first-planned operation 90

From the results, the case, for keeping the current network, is observed as follows in Table 5.6. Based on the total lead time of jobs, the algorithm kept the current network for Job 9. It could be observed that it was the right decision with respect to other metrics indicating that the current network yielded better job lead times overall.

Table 5.6: Comparison of networks for delayed first-planned operation 90

Metric	Next best network	Current network
Below median	Jobs 5, 6, 7, 10	Jobs 1, 5, 6, 7, 10
Outperformed	Jobs 8, 9, 10	Jobs 1, 2, 3, 4, 5, 6, 7
Last lead time	774	775
Total lead time	6605	6461

A second example of first-planned operation was investigated when manufacturer M1A acquired a delay on the first-planned operation **41** of 10% of its processing time. The comparisons of network update versus network status quo, are shown in Figure 5.13.

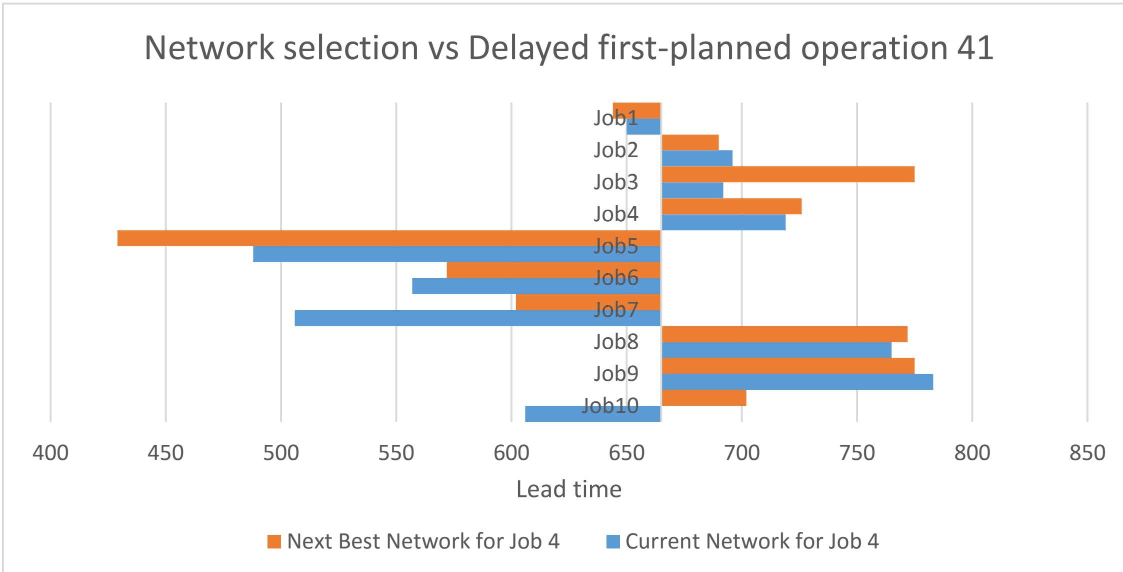


Figure 5.13: Sensitivity analysis of delayed first-planned operation 41

From the results, the case for updating the network group is as shown in Table 5.7. In this case, the algorithm which is triggered by the total lead time of jobs, took the right decision to keep the original network for Job 4 because mostly other metrics concurred with this decision.

Table 5.7: Comparison of networks for delayed first-planned operation 41

Metric	Next best network	Current network
Below median	Jobs 1, 5, 6, 7	Jobs 1, 5, 6, 7, 10
Outperformed	Jobs 1, 2, 5, 9	Jobs 3, 4, 6, 7, 8, 10
Last lead time	775	783
Total lead time	6687	6462

Next, the impact of delay on the single-slacked operation 33 of 230% of its processing time, was investigated. Any lower percentage did not affect any of the job lead times. The case for replacing manufacturer or not i.e. M3A with M3B, is illustrated in Figure 5.14.

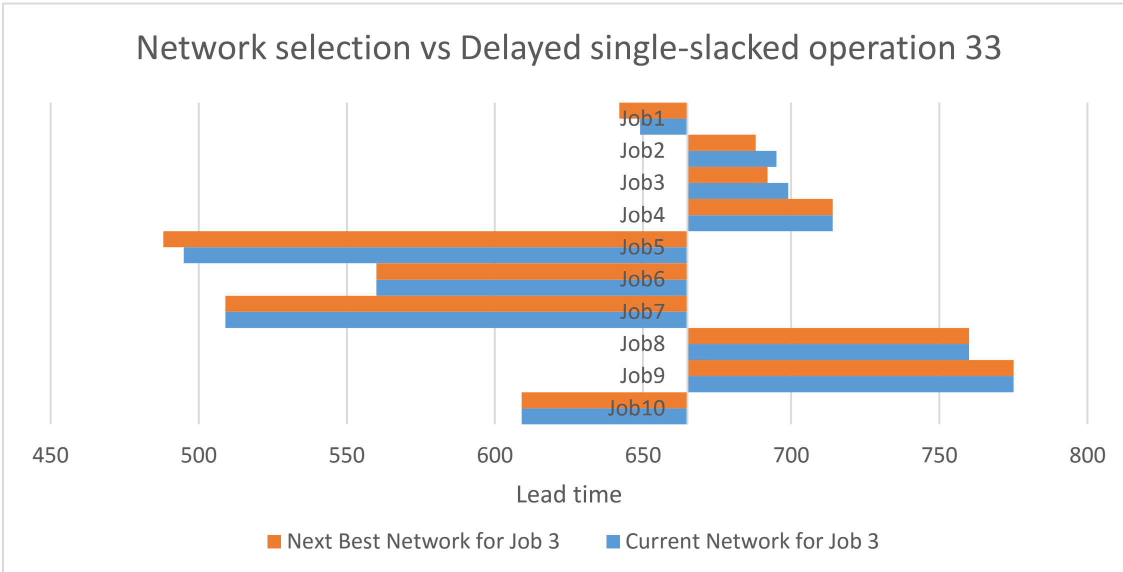


Figure 5.14: Sensitivity analysis of delayed single-slacked operation 33

There is a case for the algorithm to take action with regards to the delayed operation so that the network group is updated. The total job lead time for the current network group was high. When action was taken to replace manufacturer M3A, it resulted in the next best network for Job 3 to be selected and the total lead time was reduced.

As for the single-slacked operation **70**, a delay of 50% of its processing time, was not mitigated by neither taking action nor no action. Therefore, a network change did not mitigate the impact of job lead times when a single-slacked operation is delayed. The current network group was kept and this decision is supported by the observations from Figure 5.15 where the job lead times are almost equal.

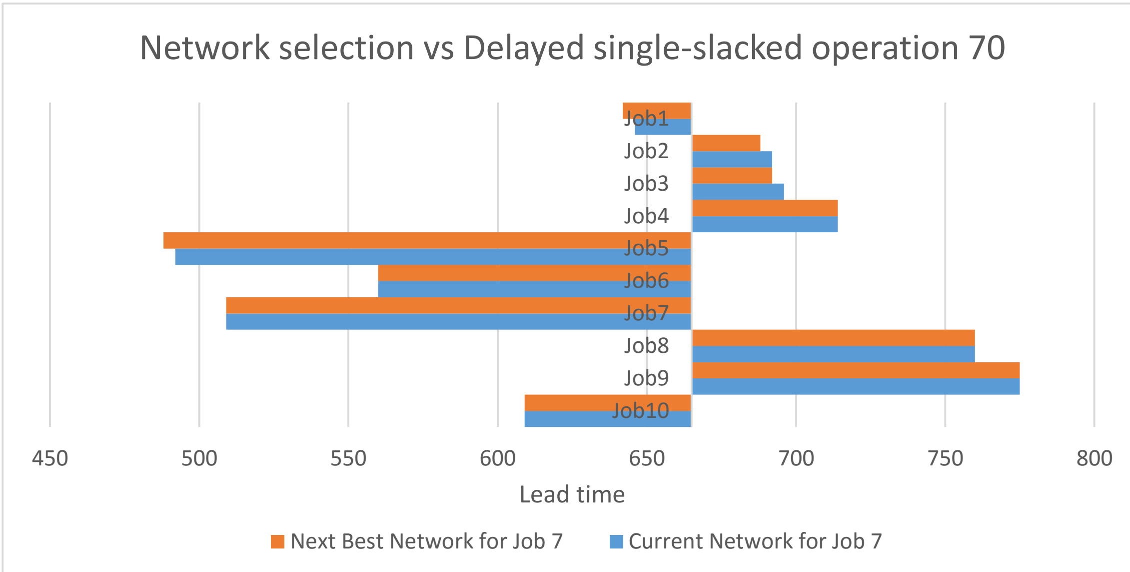


Figure 5.15: Sensitivity analysis of delayed single-slacked operation 70

Finally, it was investigated how the delayed double-slacked operation **10** of 500% of its processing time, could be mitigated by a network change. Any delay, below the prescribed percentage did not significantly influence the job lead times. The comparison of action vs no action is illustrated in Figure 5.16.

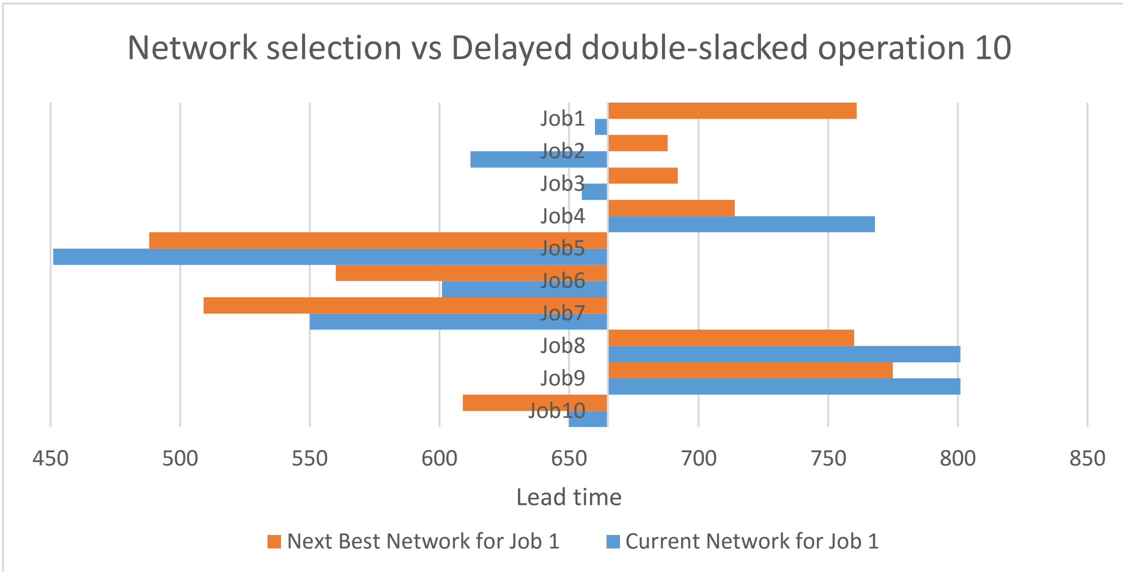


Figure 5.16: Sensitivity analysis of delayed double slacked operation 10

The algorithm replaced the current network for Job 1, with the next best one because of a better total lead time, resulting from the updated network group. This decision is also supported by the fact that changing the network yielded better performance overall. The case for changing the network or not, is as follows in Table 5.8.

Table 5.8: Comparison of networks for delayed double-slacked operation 10

Metric	Next best network	Current network
Below median	Jobs 5, 6, 7, 10	Jobs 1, 2, 3, 5, 6, 7, 10
Outperformed	Jobs 4, 6, 7, 8, 9, 10	Jobs 1, 2, 3, 5
Last lead time	775	801
Total lead time	6549	6556

A second example of double-slacked operation was investigated when manufacturer M5B acquired a delay on the double-slacked operation **95** of 10% of its processing time. The comparisons of network update versus network status quo, are shown in Figure 5.17.

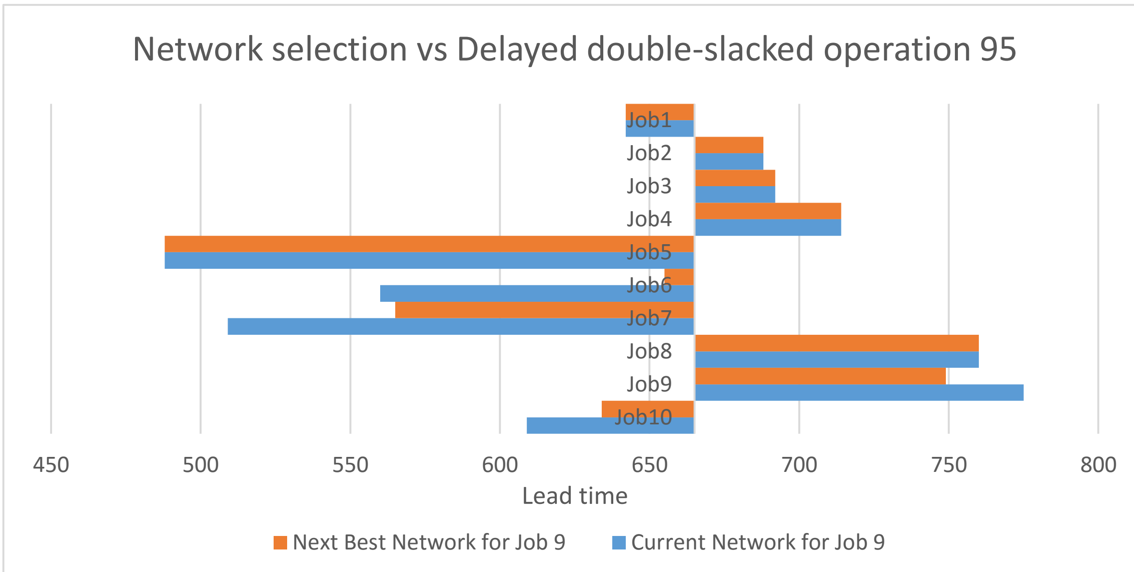


Figure 5.17: Sensitivity analysis of delayed double-slacked operation 95

The network selection algorithm chose to keep the current network group because of the shorter total lead time. This is supported by the fact that the next best network was outperformed on Job 6, 7 and 10 among the metrics presented in Table 5.9.

Table 5.9: Comparison of networks for delayed double-slacked operation 95

Metric	Next best network	Current network
Below median	Jobs 1, 5, 6, 7, 10	Jobs 1, 5, 6, 7, 10
Outperformed	Jobs 9	Jobs 6, 7, 10
Last lead time	760	775
Total lead time	6587	6437

5.5.3 Discussion about why network replacement may harm lead times

Considering the experiment of first-planned operation 90 that is delayed by 10% of its processing time, the algorithm replaced the faulty manufacturer M0A and replaced it with M0B. When M0B was allocated operation 90, the operation original processing time was restored. However a few repercussions occurred to other manufacturers as shown in Table 5.10. A comparison with Table 5.5 revealed the changes that occurred when the network was changed.

Table 5.10: Updated operation plans of twin manufacturers (MT10)

Manufacturer	Operation plan	Manufacturer	Operation plan
M0A	20-10-40-100-70-60	**M0B**	**90-80**-30-50
M1A	41-31-101-71-11	**M1B**	91-81-61-51-21
M2A	82-42-62-52-102-92-72	**M2B**	22-32-12
M3A	93-33-73-23-103-83	**M3B**	63-53-13-43
M4A	24-44-**54-94-104-64**-74	**M4B**	84-14-34
M5A	**65-85-75**-105-45	**M5B**	95-**55-35-15-25**
M6A	106-46-76-96-66	**M6B**	86-26-16-36-56
M7A	47-37-97-87-**77-107-67**	**M7B**	57-17-27
M8A	108-**68-58-88-78**-98	**M8B**	48-38-18-28
M9A	29-109-79-69	**M9B**	99-89-59-49-39-19

The operation lead times of M0A were reduced and those of M0B increased. This had a compound effect on the operation plans of M4A, M5A, M5B, M7A and M8A as well as their lead times. This compound effect may be more harmful than the initial operation delay. For instance, it can be seen that for first-planned operations 90 and 41, current network group with its first-planned operation delay, had better total lead time than the next best network which was without operation delay. The same was true of single-slacked operation 70 and double-slacked operation 95. One reason is that sequencing recovery, mentioned in Section 3.5.3, was triggered so that several operations changed positions in the operation plans. Different operation pairs were generated as part of new pair pre-selection lists for the network selection algorithm to choose from. However, the next best network yielded a better network group lead time for single-slacked operation 33 and double-slacked operation 95. One obvious reason is that the next best network had no operation delay. Another possible reason is that the next best network had less slack compared to the delayed current network. These possibly contributed to the updated network group having a total lead time that was the lowest.

5.6 Detection of schedule conflicts

The next hypothesis was whether resolving disturbances is synonymous to resolving conflicts of resource-constrained scheduling and if so, what type of disturbances. Four types of disturbance were investigated in Protégé and observations were made from the inferred data of three tests. Types of disturbance include 1) cancelled order, 2) delayed operation, 3) collapsed manufacturer and 4) rush operation. Tests include a job process plan conflict test, a manufacturer operation plan conflict test and a due time conflict test.

5.6.1 Job cancellation disturbance

In this experiment scenario, the processing times of all operations of Job 1 were zeroed out, to simulate a job cancellation. Except for Operation 10, all nine remaining operations of Job 1 were flagged as a fail as shown in Figure 5.18. Manufacturer M4 is affected by collapsed schedule of Operation 14, the degree of which is measured as overlap -152 hours, Manufacturer M7 is affected by collapsed schedule of Operation 17, with an overlap -274 hours and so on.

5.6.2 Manufacturer collapse disturbance

In order to simulate the collapse of a manufacturer, operations performed by manufacturer M0 are given an undefined schedule i.e. its operations' processing times are zeroed out. Verification tests flagged up the disturbance. Consequently, almost one third of tests failed and all jobs are affected due to the fact that the manufacturer provides operations to all jobs.

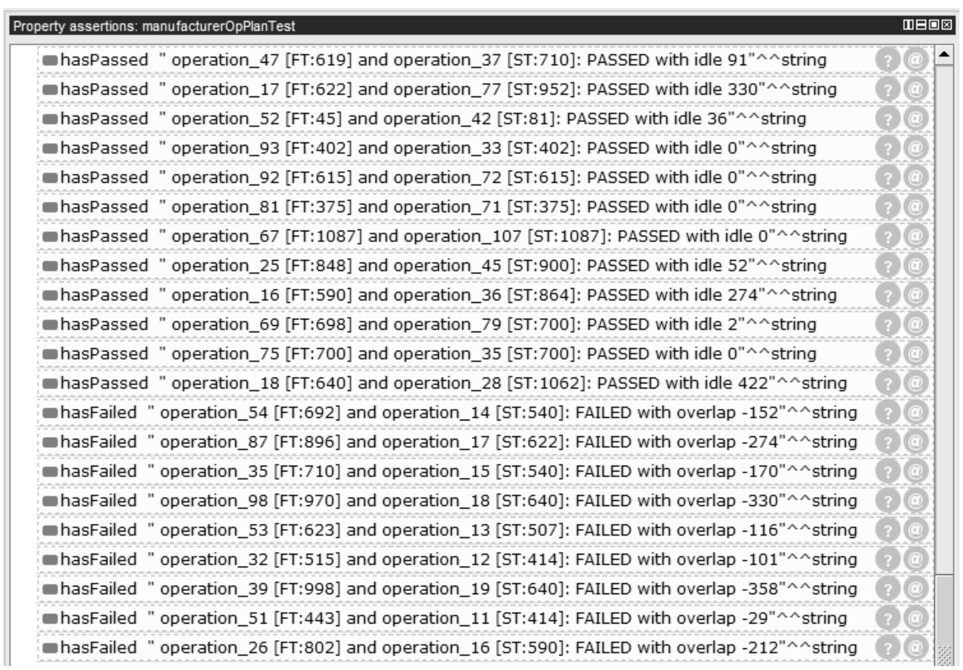


Figure 5.18: Manufacturer operation plan test performed due to cancelled job

5.6.3 Operation delay disturbance

The processing time of Operation 16 was tripled, to simulate a delay and the due time test showed that on-time delivery of Job 1 has failed, incurring a lateness of 30 hours. It was also noted that finish times of Operation 17, 18, 36, 77 and 28 inflated to cause an average overlap of 108 hours. Initially, all the other jobs seemed unaffected. However, when conflict resolution process took place, Job 2, 3, 5, 6, 7 and 10 were affected as well, as shown by Figure 5.19. The figure also depicts paths, along which the delay spread to 15 other operations. For instance, operation_16 affects operation_17 and operation_16 also affects operation_36.

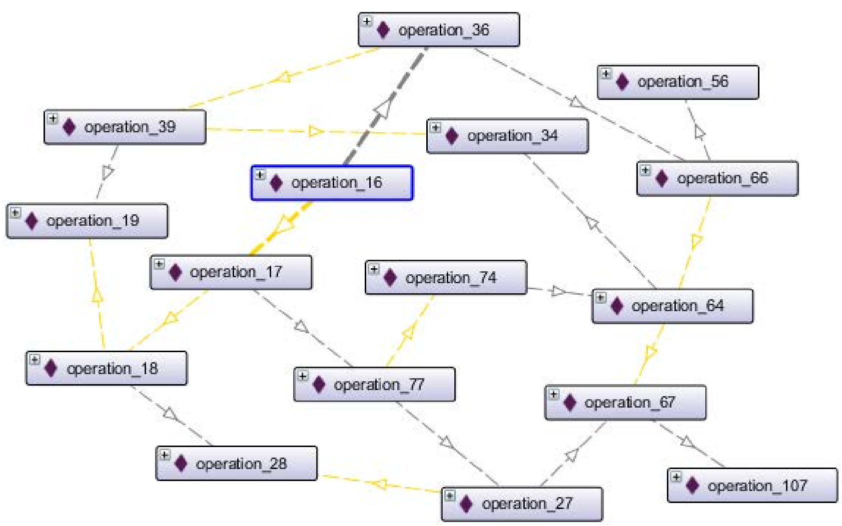


Figure 5.19: Interdependencies among operations emerging from Operation 16

5.6.4 Rush operation disturbance

To introduce a rush scenario, operations 16 and 26 are swapped with each other, within the operation plan of Manufacturer 6. Conflicts are adjusted, as soon as they get flagged in the 'manufacturing operation plan' test and the 'job process plan' test. Needs for adjustments propagate to other operations and jobs as depicted in Figure 5.20. Jobwise, operation_16 affects operation_17 and now manufacturer-wise, operation_16 affects operation_26. Same logic applies for other operations in Figure 5.20. Consequently, jobs 1, 3, 4, 5, 6 and 10 had longer lead times, as shown in Table 5.11.

Table 5.11: Job lead times with and without rush operation

Job	**Lead time (h)**	
	W/O Rush	**With Rush**
1	1035	1065
2	1092	1092
3	1076	1103
4	943	999
5	1071	1117
6	1087	1095
7	1037	1037
8	975	975
9	970	970
10	1132	1147

5.7 Scalability of conflict resolution

5.7.1 Operation pair conflict resolution

In the MT10 problem, when pair 16_36 was selected for the operation plan of Manufacturer 6, the start time adjustment in operation 16 caused conflicts along the dependency chain as previously shown in Figure 5.20. Without conflict resolution, no operation is adjusted.

The conventional approach would sequence an operation and would resolve conflicts immediately after. The start time adjustment creates a cascade of conflicts and the sequencing of next operation would wait until the outstanding conflicts were resolved. However, it was observed that the resultant operation plans were exactly the same, irrespective of whether conflict resolution was performed or not.

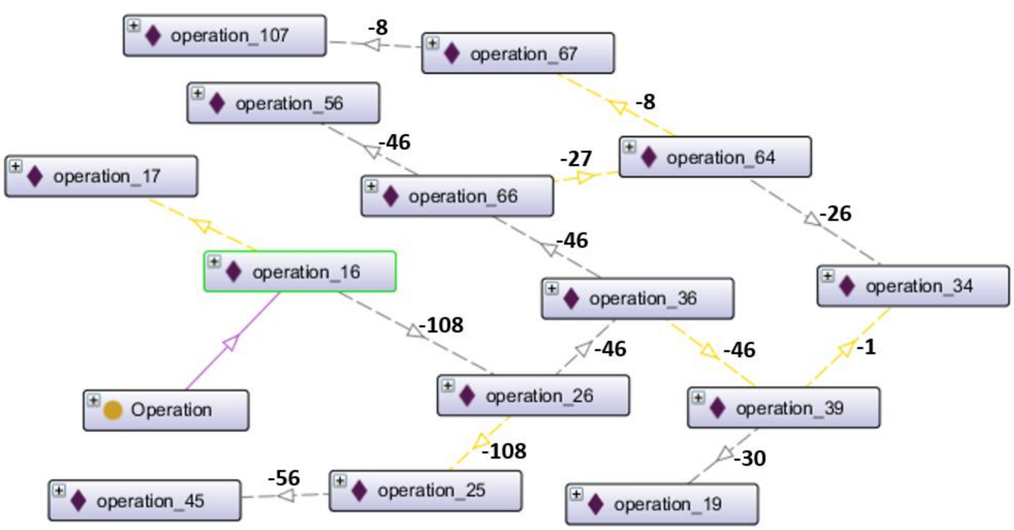


Figure 5.20: Paths of disturbance propagation, with adjustments on edges, due to rush scenario

For the MT10 and LA19 problems, it was observed that carrying out sequencing, timing and conflict resolution, for every pair, required 480 interactions between pair agents compared to 90 interactions if no conflict resolution is performed as shown in Figure 5.21. This resulted in a computation time reduction, from 288 minutes down to 43 minutes. Figure 5.22 compares the distribution of computation times across manufacturer agents, with/out conflict resolution.

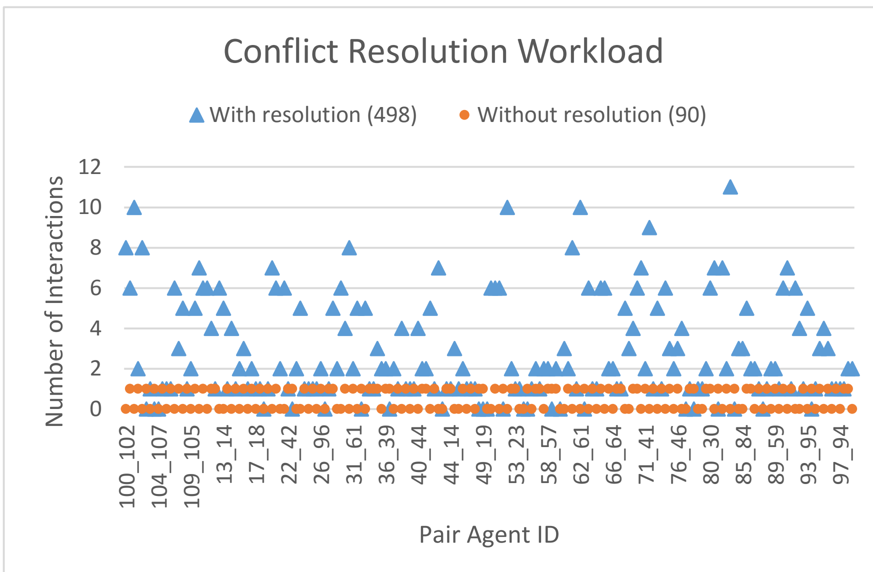


Figure 5.21: Amount of conflict resolution work requested from pair agents

5.7.2 Discussion about conflict propagation and resolution

Manufacturers provide operations to several job process plans and consequently their operation plans are interlinked. Operation plans are made of selected operation pairs. Within the pair, the timings of primary and secondary operations would usually overlap. Therefore, the start time of the primary operation would be adjusted to solve the conflict. The adjustment incurred by the primary operation of the operation plan would also propagate down other operation plans as well as process plans. However, the selection of operation pairs by manufacturers were not influenced by whether conflicts were resolved or not if the pre-selection and selection algorithms follow some important rules. The algorithms operate in such a way that operation sequencing is decoupled from and not influenced by conflict

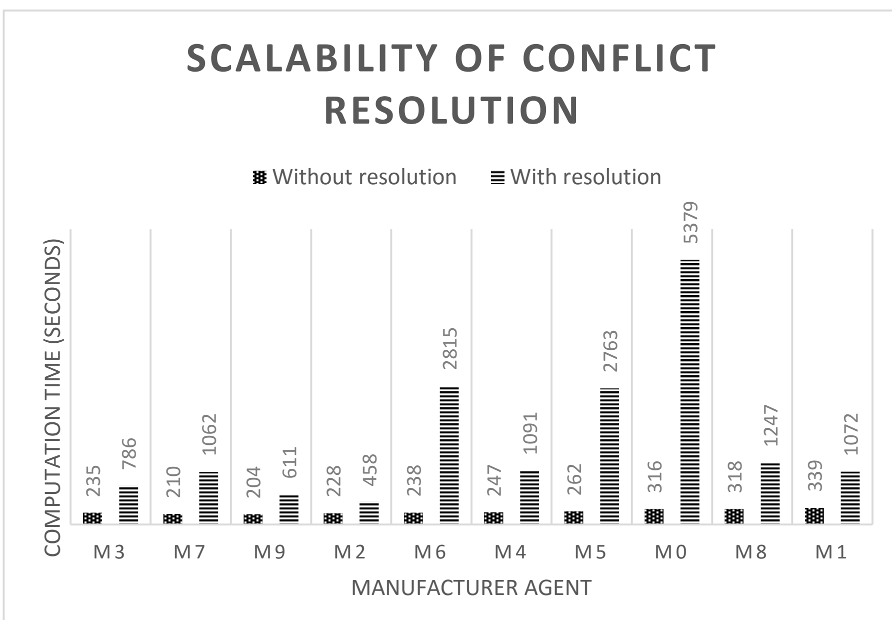


Figure 5.22: Computation time distribution among manufacturing agents

resolution. In scalability context, it meant that the manufacturer agents, using pair pre-selection and pair selection algorithms, can work in parallel.

5.8 Scalability of operation pairing approach

Two approaches for the generation of operation pairs were investigated. The first approach generates all possible operation pairs and the manufacturer agent performs pair pre-selection and selection. The second approach limits the number of pairs generated. Only pairs with a particular secondary operation (So_p) are generated. The operation plan is developed from end to front. For the end pair, the last operation heuristic rule is used, whereby the end operation pair must have a secondary operation that is equivalent to the last operation of a job process plan. When the end pair is selected, the next generated set must contain pairs that have secondary operations that are equal to each other and equal to the primary operation of the previously selected pair. Figure 5.23 shows the total number of pair generations for the first and second approaches. Without the heuristics, total number of pairs generated is $n(n-1)$. With the heuristics, the total number of pairs generated is $\sum_{i=1}^{n} n - i$ equivalent to $\frac{n}{2}(n-1)$. Therefore, scalability can be improved. Future work will develop better mechanisms to further reduce the number of generated pairs and some suggestions are presented in the next section.

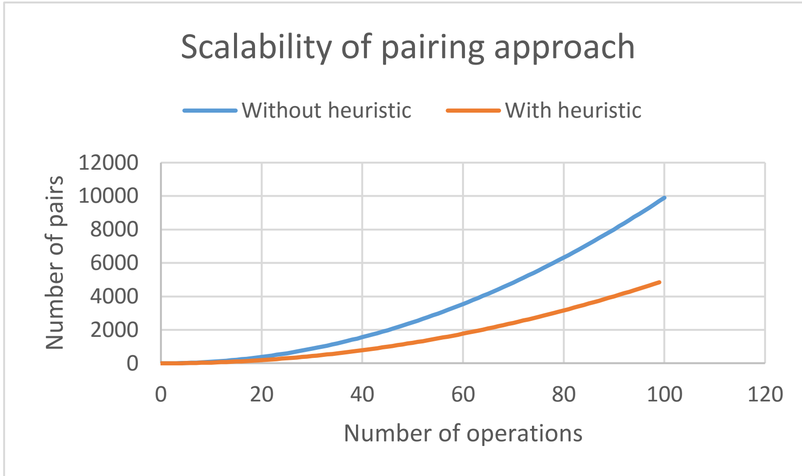


Figure 5.23: Size of the solution space for the operation pairing approach

5.9 Future work

5.9.1 Enhancing the resilience of scheduling

Given a case where two manufacturers, M1 and M2, have two unique operations each. There are three possible schedules for Job A and B, as shown in Figure 5.24. The first and second schedules, which are more resilient to operation disturbances, share an important planning characteristic. In both schedules, each consisting of two operation pairs, the primary operations of the operation pairs are from the same job. So is the case for the secondary operations which are from the same job. For instance, OpA1 and OpA2 belong to Job A and they are both primary operations in their respective pairs. Same goes for OpB1 and OpB2 which are secondary operations which belong to Job B. This characteristic ensures that an operation is kept away from the critical path. This operation is more resilient and makes the job and manufacturer, it forms part of, resilient as well.

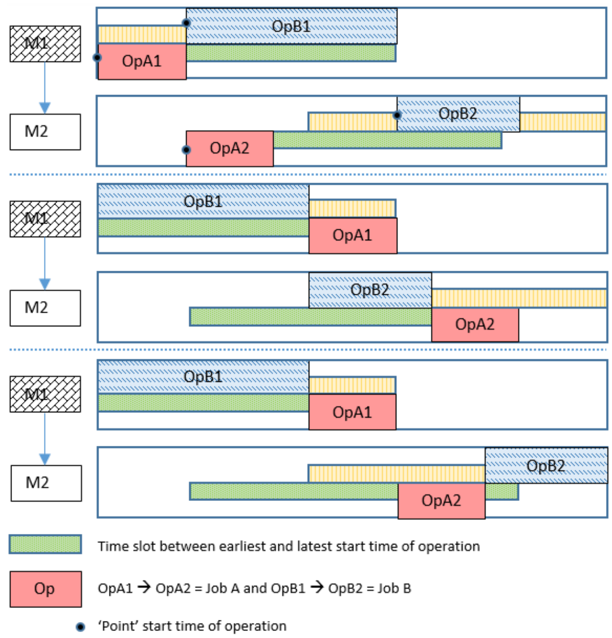


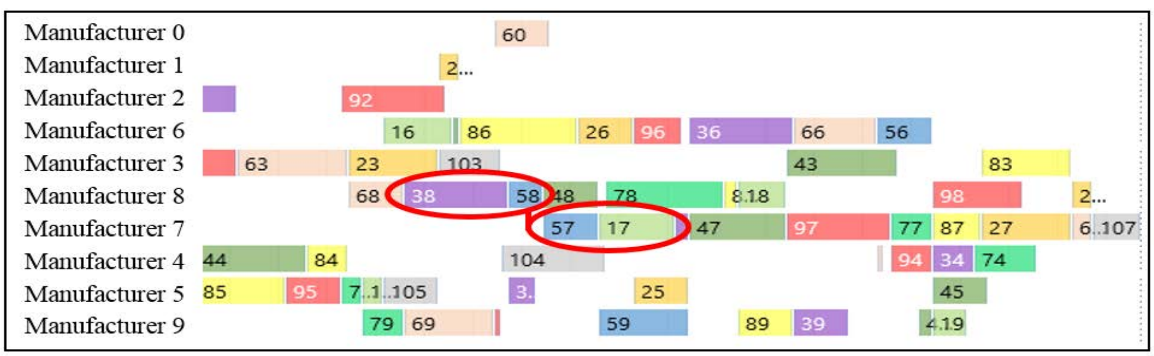
Figure 5.24: Three possible scheduling outcomes for Job A and Job B

For instance, the first schedule has no idle time within the process plans of Job A and Job B. The idle time in the operation plan of M1 is absent but present in M2. Also, OpA2 is not on the critical path. Therefore M2 becomes more resilient to possible disturbances on OpA2. This benefit is consistent with the fact that operations OpA1 and OpA2 are both primary operations from the same job.

The second schedule for the manufacturer pair has no idle time in both manufacturers as well as Job B while there is some idle time in Job A. Operation OpA1 is not on the critical path which makes Job A more resilient to disturbances on OpA1. This benefit is consistent with the fact that operations OpA1 and OpA2 are both secondary operations from same job.

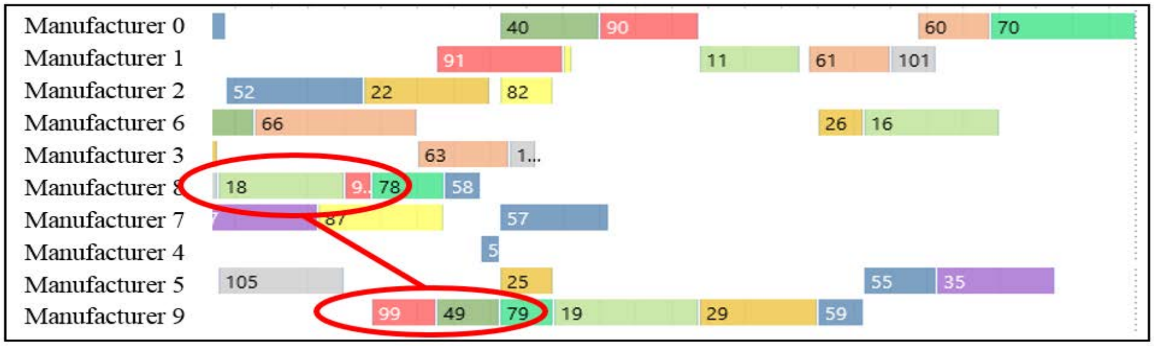
This is in contrast with the third schedule where Job A, manufacturers M1 and M2 have no idle time but Job B has idle time. All operations are on the critical path so that the manufacturer pair would not be resilient to disturbances. This problem arises due to the fact that even though they come from the same job, operation OpA1 is a secondary operation while OpA2 is a primary operation.

This problematic characteristic can be noted in the MT10 schedule as shown in Figure 5.25. The operation pairs $sp_{8,4}$ = 38_58 and $sp_{7,5}$ = 57_17 have a primary operation $po_{57\_17} = 57$ and a secondary operation $so_{38\_58} = 58$ which belong to the same job $j = 5$. Operation 17 and 38 should have been primary operations while operations 57 and 58 should become secondary operations. This would have reduced the lead time of manufacturer 7 and that of job 1. As well, such characteristics can be noted in the LA19 schedule as shown in Figure 5.26.



Where **80** = operation 80, it means that operation 80 is needed by Job **8** and provided by Manufacturer **0**

Figure 5.25: Schedule inefficiency highlighted for scheduling MT10



Where **80** = operation 80, it means that operation 80 is needed by Job **8** and provided by Manufacturer **0**

Figure 5.26: Schedule inefficiency highlighted for scheduling LA19

Future work will develop a heuristic algorithm that will allow only the good planning characteristic to prevail. It will not only produce better schedules but the size of the solution space, for pairs, will be reduced. As a consequence, scalability can improve.

5.9.2 Enhancing scalability of conflict resolution

Conflict resolution will have to be performed in a more intelligent and efficient manner. For a 10x10 scheduling problem like LA19 and MT10, the number of operation pairs that are finally retained was 180 pairs. 90 pairs were inherent to the job process plans. The other 90 pairs were selected from a larger pool of solutions. However, it was observed that 480 interactions between pair agents took place, during the conflict resolution process. This meant that some pairs performed conflict resolution more than once also indicated by Figure 5.21. Future work will aim at reducing interactions between pair agents and allow them to resolve conflicts in parallel.

5.9.3 Implementation of the operation pairing approach on distributed machines

Future work will investigate how to migrate OWL ontology and SWRL rules from desktop-based Protégé into a graph database server with full SWRL reasoning support. This will enable a framework for the distributed creation, rapid deployment, contained maintenance and versioned updates of scheduling knowledge bases. Furthermore, implementation of multi-agent system will be addressed in three phases as shown in Figure 5.27.

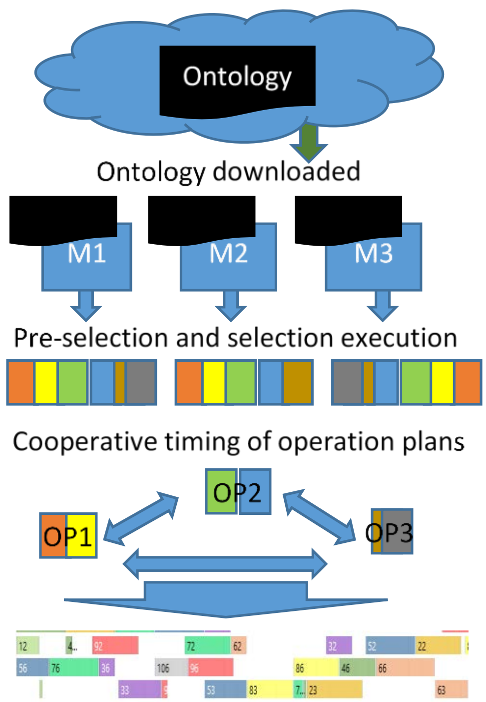


Figure 5.27: Implementation framework for decentralized scheduling

First, new ways of implementing agents on machines will be investigated, so that machines have the ability to connect to the cloud and compute the operation scheduling algorithms. Second, MAS-based query and update of a graph database will be the cornerstone of an efficient and resilient scheduling process. Third, functions of a multi-agent system (MAS) will be developed to optimise the creation of manufacturer operation plans and generation of scheduling priority.

5.10 Conclusion

It was expected that the manufacturing pairing approach would achieve better make spans than the operation pairing approach. However, there was concern about the scalability of the first approach. The results were achieved by iterations on two control parameters, namely swapping mutation rate and population size. There were three stages of optimisation, each using different parameters. Also as the number of operations increased, the optimisation performance deteriorates, with higher computation time and a lower rate of convergence. On the other hand, the operation pairing approach operated at a level of granularity that enabled simple algorithms to be used and kept the optimality loss at 35%, with respect to make spans. Furthermore, with the ability of pair agents to communicate with each other, the algorithms were able to effectively propagate solutions, during conflict resolution. Finally, the use of network bids, rather than manufacturer bids, as criteria for the selection of manufacturing networks, yielded the best performance, with regards to job lead time and quality. The performance of the cost criteria was improved by the use of job dispatching rules.

CONCLUSION

6.1 Introduction

Manufacturing small and medium enterprises play an important role in the economy accounting for 15% of SME value added in the UK. The SMEs, as part of a virtual organisation breeding environment, have significant potential for resilience and productivity in the volatile economic environment. Combining their unique resources, expertise and innovative solutions, SMEs can meet the demand for customised high value products, as part of manufacturing networks. The shop floor, as well, is undergoing a transition towards decentralisation of manufacturing systems. Industry 4.0 recognises decentralised autonomous system as the sustainable solution for complex environments and production demand. Current manufacturing execution systems generate central plans that have limited life span and are too rigid to re-iterate their plans in real time.

6.2 Aim and objectives

The research aim was to investigate, within the context of flow shop systems, the formation of networks, where manufacturers generate their own schedules and where a final network is selected, following a bidding process. The main objectives of the research were as follows:

- A literature review of networks, scheduling techniques, modelling approaches and enablers of decentralised problem solving

- Identification of the problem, supported by industrial use case and operation research datasets
- Generation of innovative ideas for decentralised scheduling in the manufacturing network breeding environment
- Development of experiments for the validation and evaluation of ideas

The ideas for decentralised flow shop scheduling have also addressed network formation and network selection. Two models were investigated for network formation namely manufacturer pairing and operation pairing. Two other models were investigated for network selection namely selection on the basis of manufacturer bids and selection on the basis of network bids. The experiments were composed of agent-based modelling of network formation and selection as well as multi-agent system implementation of decentralised scheduling.

6.3 Summary of the work presented in the thesis

This research has looked at the scenario where small and medium manufacturers participate in manufacturing network breeding environments to find some stability against the volatile market. A literature review of the systems that supports network configuration and coordination, was carried out and presented in chapter 2. Fundamental techniques of scheduling in flow shop systems and their innovative integration for localised scheduling, were reviewed. An architecture for modelling an industrial use case of a manufacturing network and techniques for modelling flow shop scheduling problems, were also reviewed. Furthermore, technological enablers for migration from model into actual industrial implementation, were highlighted. The scope of the problem was developed with respect to, the research gap, an industrial use case as well as operation research case studies and benchmarks. A manufacturing network was modelled as a flow shop system and inspired by the holonic paradigm, the entities relevant to scheduling were identified and modelled.

In chapter 3, the research proposed two ways for forming a network namely manufacturer pairing and operation pairing. The manufacturing pairing approach is concerned with manufacturers cooperating to produce their respective operation plans that did not conflict with each other. Each applied genetic algorithm to help them converge to an agreement. The operation pairing approach affords a manufacturer control over its operation planning, within heuristic boundaries that are globally acceptable. Moreover, in the case that more than one network bid for the same job, two network selection approaches were proposed. One approach consists of selecting a network based on the bids of its manufacturers. The other approach selects a network based on its bid for the overall job.

Simulation platforms were developed, in chapter 4, around the proposed approaches. Manufacturer pairing achieved better make spans than operation pairing. However, operation pairing provided better insights into how decentralised scheduling can be performed effectively. The use of simple rules for local planning, emerged into complete schedules for the flow shop scheduling problems MT10 and LA19, with an optimality loss of 33% and 35% respectively. Furthermore, the approach is tractable and scalable and scalability has room to be enhanced even further. The use of network bids as the criteria for network selection is intuitive even though the industrial use case showed that the company entirely relies on manufacturer bids to form manufacturing networks. The experiments indicated the superiority of the former approach with respect to customer requirements for lead time, quality and cost.

6.4 Contributions of the thesis

This research has contributed to the field of operation research in the following ways:

- A novel interaction system for manufacturer agents and job agents, in an agent-based representation of a flow shop system.

  The proposed operation pairing mechanism introduced operation pair agents which act as mediators between two manufacturer agents, two job agents or between a manufacturer agent and a job agent. All agents are able to share data via a pair agent. Some of the shared data are universal and some are unique and owned by two agents and their pair agent. This novelty enables interactions to be developed in new ways. For instance, interaction is now an agent with scalable data structure. Next, the interaction has the ability to reason about data. Also, interaction is tractable and therefore can be scientifically enhanced.

- A novel methodology for manufacturer agents and job agents, to acquire algorithms that limit the population of operation pair agents.

  A multi agent system was developed to provide a controlled environment for job agents, manufacturer agents and operation pair agents. The system was developed in workflow agent development environment (WADE), which has not yet been used in the operation research field. To ensure the effectiveness of the interaction system, the population of pair agents is controlled by manufacturer agents and job agents. If there is no pair agent, there is no interaction. The methodology uses a workflow approach to develop, combine and implement control algorithms into manufacturer agents and job agents, allowing them to control interactions.

- A self-similar approach for the configuration of manufacturing networks and decentralised operation planning.

  By controlling interactions, manufacturer agents and job agents limit the pool size of operation pair agents. In doing so, they also limit the solution space for possible configuration of networks. In the context of this thesis, operation planning consists of manufacturer agents that select operation pair agents, according to some criteria. Pair agents can influence the decisions of manufacturer agents, through bidding. By principle of self-similarity, network configuring consists of a job agent that selects a network of manufacturer agents, according to some criteria. This approach allows the execution of simple heuristics in a decentralised manner, to emerge into high quality network configurations.

6.5 Limitation of analysis

6.5.1 Comparison of centralised and decentralised scheduling

The three main functions of the decentralised scheduling model namely formation, pairing and selection were described in Chapter 3. The performance of each function was measured according to objective metrics such as time budget, start time adjustment, pair compatibility and network compatibility. The limitation of the analysis was in determining the significance of the metrics. Were they the most appropriate input to the next function? To what extent did they help to achieve the scheduling objectives? This limitation might be overcome by comparing the outcomes of centralised, and decentralised scheduling, in terms of the objective metrics. The outcomes of centralised scheduling could be decomposed into operation plans, manufacturer pairs, operation pairs and manufacturing networks and measured with the objective metrics. This would result into a specific benchmark for each objective metric and therefore strengthen the metric usefulness.

6.5.2 Impact of incomplete information on pair compatibility

Manufacturer pairs were proposed as the outcomes of the pairing function. A manufacturer pair consists of two manufacturers and also two operation plans. The case studies used, have jobs which did not pass more than once with the same manufacturers. However, multiple manufacturers could work on the same job, more specifically, one manufacturer per operation required by the job. Therefore, within each manufacturer pair, each job was always represented by only two of its required operations. This meant that any single manufacturer pair had incomplete data about its jobs. This caused the calculation of pair compatibilities to be initially volatile and eventually to settle down as and when more manufacturer pairs were being formed. Even though this would not affect network selection, however, it affected the type of networks available for selection. This limitation might be overcome if the pairing function became aware of and requested the missing data.

6.5.3 Complementarity of operation pairing and manufacturer pairing

Also proposed were the operation pairs as the outputs of the pairing function. This is the smallest structure of a manufacturer operation plan and of a job process plan as a matter of fact. It has two objective metrics namely time budget and start time adjustment. A limitation exists with regards to how the operation pair can assist a manufacturer pair in decreasing the volatility of pair compatibilities. Are the operation pair metrics appropriate and can they be summoned to replace the missing information needed by a manufacturer pair? There is a need to link the two approaches that were proposed for the pairing function, to address this limitation.

6.6 Future works

When evaluating the resulting operation plans of the LA19 and MT10 scheduling problems, it was noted that the plans share some characteristics that cause inefficiencies in the final schedules as indicated in Figures 5.17 and 5.18. A simple algorithm, for the operation pairing approach, similar, in simplicity, to the last operation heuristic rule, could prevent those characteristics from being generated. This has the potential to narrow the boundaries within which manufacturers can generate high quality operation plans. Furthermore, scalability will be increased due to the smaller size of the solution space.

Another issue that future work can address, is the inefficient conflict resolution routine. When a disturbance takes place, the routine propagates the solution to other operation pairs. In the experimentation, it was noticed that some operation pair agents are called multiple times during one conflict resolution routine. However, a pair agent should be able to wait for the solution to aggregate and then apply the final solution once. Furthermore, some pair agents should be able to perform the resolution routines in parallel.

The research was limited to 10x10 scheduling problems. However, the operation pairing approach has shown good scalability and potential for further enhancement in that aspect. Therefore, it is strongly recommended that future work investigates the approach under larger scheduling data sets. Once the aforementioned is achieved, future work can address the demonstration of the approach using a cluster of cloud-enabled machines.

Appendix

Constraints calculation rules

Rule A1: Earliest possible start time for the FIRST operation is zero.

hasFirstOperation(?j,?o)→hasEarliestPossibleStartTime(?o,0)

Rule A2: Earliest possible finish time for the FIRST operation is its processing time which is also the earliest possible start time for the NEXT operation.

hasFirstOperation(?j,?o1),hasProcessingTime(?o1,?pt),preceedsJobwise(?o1,?o2)→hasEarliestPossibleFinishTime(?o1,?pt), hasEarliestPossibleStartTime(?o2,?pt)

Rule A3: Earliest possible finish time for ANY operation is its start time plus its processing time which is also the earliest possible start time for the NEXT operation.

hasEarliestPossibleStartTime(?o1,?est),add(?eft,?est,?pt),preceedsJobwise(?o1,?o2),hasProcessingTime(?o1,?pt) -> hasEarliestPossibleFinishTime(?o1,?eft),hasEarliestPossibleStartTime(?o2,?eft)

Rule A4: Earliest possible finish time for the LAST operation is its start time plus its processing time.

hasLastOperation(?j, ?o), hasEarliestPossibleStartTime(?o, ?est), hasProcessingTime(?o, ?pt), add(?eft, ?est, ?pt) → hasEarliestPossibleFinishTime(?o, ?eft)

Rule A5: Latest possible finish time for the LAST operation is the due time of the job and latest possible start time for that same operation is the same due time minus the operation's processing time.

hasLastOperation(?j,?o),hasDueTime(?j,?dt),hasProcessingTime(?o,?pt),subtract(?lst,?dt,?pt)→hasLatestPossibleFinishTime(?o,?dt), hasLatestPossibleStartTime(?o,?lst)

Rule A6: Latest possible start time for ANY operation is its latest possible finish time minus its processing time. Its latest possible finish time is the latest possible start time of the succeeding operation.

hasLatestPossibleStartTime(?o1, ?lst1), subtract(?lst2, ?lst1, ?pt), hasProcessingTime(?o2, ?pt), succeedsJobwise(?o1, ?o2) →hasLatestPossibleStartTime(?o2,?lst2),hasLatestPossibleFinishTime(?o2, ?lst1)

Local verification rules

Rule B1: Compatibility is positive one when proposed start time of an operation lies in the interval of its earliest and latest possible start time.

hasProposedStartTime(?o,?pst),hasEarliestPossibleStartTime(?o,?est),hasLatestPossibleStartTime(?o,?lst),greaterThanOrEqual(?pst,?est),lessThanOrEqual(?pst,?lst)→hasCompatibility(?o,1)

Rule B2: Compatibility is zero when the schedules for a pair of operations overlap.

preceedsJobwise(?o1,?o2),hasProposedFinishTime(?o1,?ft),

hasProposedStartTime(?o2,?st),greaterThan(?ft,?st)→hasCompatibility(?o1,0), hasCompatibility(?o2,0)

Rule B3: Compatibility is negative one when an operation violates Rule D1 and proposed start time of operation is earlier than the earliest permissible start time.

hasProposedStartTime(?o,?pst),hasEarliestPossibleStartTime(?o,?est), lessThan(?pst,?est) → hasCompatibility(?o,-1)

Rule B4: Compatibility is negative two when an operation violates Rule D1 and proposed start time of operation is later than the latest permissible start time.

hasProposedStartTime(?o,?pst),hasLatestPossibleStartTime(?o,?lst), greaterThan(?pst,?lst) → hasCompatibility(?o,-2)

Rule B5: Operation earliness message when compatibility is negative one.

hasEarliestPossibleStartTime(?o,?est),subtract(?overlap,?est,?st),equal(?c,-1),hasProposedStartTime(?o,?st), hasOperation ID(?o,?id),hasCompatibility(?o,?c),stringConcat(?str," operation_",?id," is too early by ",?overlap)→ hasOperationConflict(?o,?str)

Rule B6: Operation lateness message when compatibility is negative two.

equal(?c,-2),subtract(?delay,?st,?lst),hasProposedStartTime (?o,?st),hasCompatibility(?o,?c),hasOperationID(?o,?id),stringConcat(?str," operation_",?id," is too late by ",?delay), hasLatestPossibleStartTime(?o,?lst)→hasOperationConflict(?o,?str)

Adjustment suggestion rules

Rule C1: Operation start time adjustment message when compatibility is zero.

equal(?c,0),subtract(?overlap,?ft,?st),preceedsJobwise(?o1,?o2),hasCompatibility(?o2,?c),hasProposedFinishTime(?o1,?ft),hasProposedStartTime(?o2,?st),hasOperationID(?o1,?id1), hasOperationID(?o2,?id2),hasCurrentTimeBudget(?o2,?ct),subtract(?diff,?ct,?overlap),stringConcat(?str,"operation_",?id1," finish time overlaps operation_",?id2," start time, adjust operation_",?id2," [", ?diff, "] by ",?overlap)→hasOperationConflict(?o1,?str),hasOperationConflict(?o2,?str)

Rule C2: Succeeding operation start time adjustment messages for all possible successors of current operation

stringConcat(?str, "If operation_",?ID2," [RB", ?diff, ", PT", ?pt, "]"," succeeds operation_",?ID1, ",adjust operation_", ?ID2, " by ", ?overlap), hasOperationID(?o2,?ID2), hasProcessingTime(?o2,?pt),provides(?m,?o1),hasProposedStartTime(?o2,?pst2),provides(?m,?o2),hasOperationID(?o1,?ID1),subtract(?overlap,?pft1,?pst2),hasProposedFinishTime(?o1, ?pft1), hasCurrentTimeBudget(?o2,?ct), notEqual(?ID1,?ID2), Manufacturer(?m),subtract(?diff,?ct,?overlap)→hasOperationConflict(?o1,?str)

Plan scheduling rules

Rule D3: Proposed start time of ANY operation is its earliest possible start time plus time adjustment required for conflict resolution and proposed finish time is its proposed start time plus its processing time.

hasEarliestPossibleStartTime(?o,?est),hasAdjustment(?o,?adj), hasProcessingTime(?o,?pt),add(?pst,?est,?adj), add(?pft,?pst, ?pt)→hasProposedStartTime(?o,?pst),hasProposedFinishTime(?o, ?pft)

Rule D4: Current time budget is the time left before a proposed finish time creates a conflict with an operation latest possible finish time.

hasProposedFinishTime(?o,?pft),subtract(?diff,?lft,?pft),hasLatestPossibleFinishTime(?o,?lft)→hasCurrentTime Budget(?o, ?diff)

Global verification rules

Rule E1: Passes test if finish time of last operation of job process plan is not greater than due time of job and displays the earliness.

hasTestID(?test,?testId),equal(?testId,"Due Time Test"),

hasLastOperation(?j,?o),hasOperationID(?o,?opId),hasDueTime(?j,?dt),hasProposedFinishTime(?o,?ft),lessThan OrEqual(?ft,?dt),hasJobID(?j,?id),subtract(?earliness,?dt,?ft),stringConcat(?str," job_",?id," [DT:", ?dt,"] and last operation_",?opId," [FT:", ?ft,"]: PASSED with earliness ",?earliness)→ hasPassed(?test,?str)

Rule E2: Passes test if operations in a job process plan do not overlap and displays idle time between adjacent operations.

hasTestID(?test,?testId), equal(?testId, "Process Plan Test"),

preceedsJobwise(?o1,?o2),hasOperationID(?o1,?id1),hasOperationID(?o2,?id2),hasProposedFinishTime(?o1,?ft), hasProposedStartTime(?o2,?st),lessThanOrEqual(?ft,?st),subtract(?idle,?st,?ft), stringConcat(?str," operation_",?id1, " [FT:",?ft, "] and operation_",?id2," [ST:",?st,"]: PASSED with idle ",?idle)→ hasPassed(?test,?str)

Rule E3: Passes test if operations in a manufacturer operation plan do not overlap and displays idle time between adjacent operations.

hasTestID(?test,?testId), equal(?testId,"Operation Plan Test"), preceedsManufacturerwise(?o1,?o2),

hasOperationID(?o1,?id1), hasOperationID(?o2,?id2), hasProposedFinishTime(?o1,?ft),

hasProposedStartTime(?o2,?st),lessThanOrEqual(?ft,?st),subtract(?idle,?st,?ft), stringConcat(?str," operation_",?id1, " [FT:",?ft,"] and operation_",?id2, " [ST:",?st, "]: PASSED with idle ",?idle) →

hasPassed(?test,?str)

Rule E4: Fails test if operations in manufacturer operation plan overlap and displays overlap time.

hasTestID(?test,?testId), equal(?testId, "Operation Plan Test"), preceedsManufacturerwise(?o1,?o2), hasOperationID(?o1,?id1), hasOperationID(?o2,?id2), hasProposedFinishTime(?o1,?ft), hasProposedStartTime(?o2,?st),greaterThan(?ft,?st),subtract(?overlap,?st,?ft),stringConcat(?str, " operation_", ?id1, " [FT:",?ft, "] and operation_",?id2," [ST:",?st, "]: FAILED with overlap ",?overlap) → hasFailed(?test,?str)

Rule E5: Fails test if finish time of last operation of job process plan is greater than job due time.

hasTestID(?test,?testId),equal(?testId,"Due Time Test"), hasLastOperation(?j,?o), hasOperationID(?o,?opId),hasDueTime(?j,?dt),hasProposedFinishTime(?o,?ft),greaterThan(?ft,?dt),hasJobID(?j, ?id),subtract(?lateness,?ft,?dt), stringConcat(?str, " job_",?id, " [DT:",?dt, "] and last operation_",?opId, " [FT:",?ft, "]: FAILED with lateness ",?lateness)→ hasFailed(?test,?str)

Rule E6: Fails test if operations in job process plan overlap and displays overlap time.

hasTestID(?test,?testId),equal(?testId,"Process Plan Test"), preceedsJobwise(?o1,?o2), hasOperationID(?o1,?id1),hasOperationID(?o2,?id2),hasProposedFinishTime(?o1,?ft),hasProposedStartTime(?o2 ,?st),greaterThan(?ft,?st),subtract(?overlap,?st,?ft), stringConcat(?str, " operation_",?id1, " [FT:",?ft, "] and operation_",?id2, " [ST:",?st,"]: FAILED with overlap ",?overlap) → hasFailed(?test,?str)

List of References

Adhau, S., Mittal, M. L. and Mittal, A. (2012). A multi-agent system for distributed multi-project scheduling: An auction-based negotiation approach, *Engineering Applications of Artificial Intelligence*, 25(8), 1738-1751

Ageron, B., Gunasekaran, A., Spalanzani, A. (2012). Sustainable supply management: An empirical study. *International Journal of Production Economics*, 140(1), 168-182

Aissani, N., Bekrar, A., Trentesaux, D. and Beldjilali, B. (2012). Dynamic scheduling for multi-site companies: A decisional approach based on reinforcement multi-agent learning, *Journal of Intelligent Manufacturing*, 23(6), 2513-2529

Almada-Lobo, F. (2015). The Industry 4.0 revolution and the future of Manufacturing Execution Systems (MES). *Journal pf Innovation Management*, 3(4), 16-21.

Anand, N., Van Duin, R. and Tavasszy, L. (2014). Ontology-based multi-agent system for urban freight transportation, *International Journal of Urban Sciences*, 18(2), 133-153

APQC (2010). Manufacturing: Centralization versus Decentralization: Which is better? Target Third Issue [Online] Available: http://bit.ly/2146UDK. Accessed on 16/01/2016

Atuahene-Gima, K. (1992). Inward Technology Licensing as an alternative to Internal R&D in New Product Development: A Conceptual Frame-work. *Journal of Product Innovation Management*. 9(2), 156–167

Baffo, I., Confessore, G. and G. Stecca. (2013). A decentralized model for flow shop production with flexible transportation system, *Journal of Manufacturing Systems*, 32(1), 68-77

Baptiste, P. (1996). Disjunctive constraints for manufacturing scheduling: Principles and extensions, *International Journal of Computer Integrated Manufacturing*, 9(4)

Barbosa, J., Leitão, P., Adam, E. and Trentesaux, D. (2015). Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution, *Computers in Industry*, 66, 99-111

Batra, S. and Tyagi, Batra. (2012). Comparative analysis of relational and graph databases, *International Journal of Soft Computing and Engineering (IJSCE)*, 2(2), 509-512

Bergenti, F., Caire, G. and D. Gotta. (2012). Interactive Workflows with WADE, *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2012 IEEE 21st International Workshop on*, 10-15

Berry, W.L., Hill, T.J., Klompmaker, J.E. (1995). Customer-driven manufacturing. *International journal of Operations and Production Management*, 15(3), 4-15

Borgo, S. and Leitão, P. (2004). The role of foundational ontologies in manufacturing domain applications, *In On the Move to Meaningful Internet Systems: CoopIS, DOA, and ODBASE*, 670-688

Brandimarte, P. (1993). Routing and scheduling in a flexible job shop by tabu search, *Annals of Operations Research*, 41, 157–183

Brettel, M., Friederichsen, N., Keller, M., & Rosenberg, M. (2014). How virtualization, decentralization and network building change the manufacturing landscape: An Industry 4.0 Perspective. *International Journal of Science, Engineering and Technology* 8(1), 37-44.

Bussmann, S. (1998). An agent-oriented architecture for holonic manufacturing control, *In Proccedings of the 1st international workshop on Intelligent Manufacturing Systems, IMS-Europe*.

Camarinha-Matos, L. M., Afsarmanesh, H., Galeano, N. and Molina, A. (2009). Collaborative networked organizations – Concepts and practice in manufacturing enterprises, *Computers & Industrial Engineering*, 57(1), 46-60

Campbell, A., Kunish, S., Müller-Stewens, G. (2011). To centralize or not to centralize? McKinsey Quarterly [Online] Available: http://bit.ly/1ORFwVL. Accessed on 17/01/2016

Carbonara, N., Giannoccaro, I. and Pontrandolfo, P. (2002). Supply chains within industrial districts: a theoretical framework, *International Journal of Production Economics*, 76, 159–176

Castro, P. M., Zeballos, L. J. and Méndez, C. A. (2012). Hybrid time slots sequencing model for a class of scheduling problems, *AIChE Journal*, 58(3), 789–800.

Chan, H. and Chung, S. (2013). Optimisation approaches for distributed scheduling problems, *International Journal of Production Research*, 51(9), 2571-2577

Cheng, T. C. E., Gupta, J. N. D. and Wang Guoqing. (2000). A review of flowshop scheduling research with setup times, *Production and Operations Management*, 9 (3), 262–282

Cheng, Y., Farooq, S. and Johansen, J. (2011). Manufacturing network evolution: a manufacturing plant perspective, *International Journal of Operations & Production Management*, 31(12), 1311-1331

Chetty, S. and Holm, D. B. (2000). Internationalisation of small to medium-sized manufacturing firms: a network approach, *International Business Review*, 9(1), 77-93

Cicirello, V. A. and Smith, S. F. (2002). Amplification of search performance through randomization of heuristics, *In Principles and Practice of Constraint Programming-CP 2002*. Springer Berlin Heidelberg, 124-138

D'Amours, S., Montreuil, B., Lefrançois, P. and Soumis, F. (1999). Networked manufacturing: The impact of information sharing, *International Journal of Production Economics*, 58(1), 63-79

Danilovic, M. and Winroth, M. (2005). A tentative framework for analyzing integration in collaborative manufacturing network settings: a case study, *Journal of Engineering and Technology Management*, 22(1–2), 141-158

European Commission. (2014). Enterprise and Industry: 2004 SBA fact sheet, United Kingdom, Retrieved from http://bit.ly/1FEdW56. [Accessed 03/05/2015]

Fawcett, S. E., Birou, L. M. and Taylor, B.C. (1993). Supporting global operations through logistics and purchasing, *International Journal of Physical Distribution & Logistics Management*, 23(4), 3-11

Freitag, M., Becker, T., Duffie N. A. (2015). Dynamics of resource sharing in production networks. *CIRP Annals - Manufacturing Technology*, 64(1), 435-438.

Fujita, K., Ito, T., Klein, M. (2014). Efficient issue-grouping approach for multiple interdependent issues negotiation between exaggerator agents. *Decision Support Systems*, 60, 10-17.

Gennari, J. H., Musen, M. A., Fergerson, R. W., Grosso, W. E., Crubézy, M., Eriksson, H. and Tu, S. W. (2003). The evolution of Protégé: an environment for knowledge-based systems development, *International Journal of Human-computer studies*, 58(1), 89-123

Giret, A. and Botti, V. (2009). Engineering Holonic Manufacturing Systems, *Computers in Industry*, 60(6), 428-440

Goldingay, H. and Van Mourik, J. (2013). The effect of load on agent-based algorithms for distributed task allocation, *Information Sciences*, 222, 66-80

Guo, Z. X., Ngai, E. W., Yang, C. and Liang, X. (2015). An RFID-based intelligent decision support system architecture for production monitoring and scheduling in a distributed manufacturing environment, *International Journal of Production Economics*, 159, 16-28

Hagel, J., Brown J. S., Kulasooriya, D., Giffi, C., and Chen, M. (2015). The future of manufacturing: Making things in a changing world. *Deloitte University Press.* Retrieved from http://bit.ly/1FQIkv3. [Accessed 25/01/2016]

Hanna, V. and Walsh, K. (2002). Small firm networks: a successful approach to innovation? *R&D Management*, 32, 201–207

Harjunkoski, I., Maravelias, C. T., Bongers, P., Castro, P. M., Engell, S., Grossmann, I. E., Hooker, J., Méndez, C., Sand, G. and Wassick, J. (2014). Scope for industrial applications of production scheduling models and solution methods, *Computers & Chemical Engineering*, 62(5), 161-193

Jiang, Y., Pang, G. and Liu, W. (2010). Research on ontology-based integration of product knowledge for collaborative manufacturing. *The International Journal of Advanced Manufacturing Technology*, 49(9), 1209-1221

Johanson, J. and Mattsson, L. (1988). Internationalization in industrial systems — a network approach P.J. Buckley, P.N. Ghauri (Eds.), The internationalization of the firm: a reader, *Academic Press*, London, 303–321

Jones, A., Rabelo, L. C. and Sharawi, A. T. (1999). *Survey of job shop scheduling techniques*, Wiley Encyclopedia of Electrical and Electronics Engineering.

Jules, G. D., Saadat M. and Saeidlou, S. (2015). Holonic Ontology and Interaction Protocol for Manufacturing Network Organization, *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, 45(5), 819-830

Kaihara, T., Fujii, N., Toide, S., Ishibashi, H., Nakano. T. (2010). A proposal of socio-inspired manufacturing scheduling concept and its application into flexible flow shop. *Proceedings of the 43rd CIRP Conference on Manufacturing Systems*, 813–820

Kaiser, R., Weiss, W., Falelakis, M., Michalakopoulos, S. and Ursu, M. F. (2012). A rule-based virtual director enhancing group communication, *In Multimedia and Expo Workshops (ICMEW), IEEE International Conference on*, 187-192

Karlos, A., Kulvik, A., Poskela, J. and Turkulainen, V. (2011). The integrative role of the project management office in the front end of innovation, *International Journal of Project Management*, 29(4), 408-421

Komma, V. R., Jain, P. K. and Mehta, N. K. (2011). An approach for agent modeling in manufacturing on JADE™ reactive architecture, *The International Journal of Advanced Manufacturing Technology*, 52(9-12), 1079-1090

Kotulski, L., Sędziwy, A. and B. Strug. (2014). Translation of Graph-based Knowledge Representation in Multi-agent System, *Procedia Computer Science*, 29, 1048-1056

Kurz, M. E. and Askin, R. G. (2003). Comparing scheduling rules for flexible flow lines, *International Journal of Production Economics*, 85(3), 371-388

Lasi, H., Fettke, P., Kemper, H., Feld, T., & Hoffmann, M. (2014). Industry 4.0. *Business & Information Systems Engineering*, 6(4), 239-242.

Law, A. M. (2007). Simulation modelling and analysis, 4th ed. New York: McGraw-Hill.

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

Leitão, P., Barbosa, J. and Trentesaux, D. (2012). Bio-inspired multi-agent systems for reconfigurable manufacturing systems, Engineering Applications of Artificial Intelligence, 25(5), 934-944

Leitão, P., Marik, V. and Vrba, P. (2013). Past, present, and future of industrial agent applications, *Industrial Informatics, IEEE Transactions on*, 9(4), 2360-2372

Lemaignan, S., Siadat, A. and Dantan, J.Y. (2006). MASON: A Proposal for an Ontology of Manufacturing Domain, In *Distributed Intelligent Systems: Collective Intelligence and Its Applications, IEEE Workshop*, 195-200

Li, W., Veliyath, R. and Tan, J. (2013). Network Characteristics and Firm Performance: An Examination of the Relationships in the Context of a Cluster, *Journal of Small Business Management*, 51, 1–22

Lim, M., Tan, K. and Leung, S. (2013). Using a multi-agent system to optimise resource utilisation in multi-site manufacturing facilities, *International Journal of Production Research*, 51(9), 2620-2638

Lin, L. F. (2011). Developing manufacturing ontologism for knowledge reuse in distributed manufacturing environment, *International Journal of Production Research*, 49(2), 343-359

Lin, X., Floudas, C. A., Modi, S. and Juhasz, N. M. (2002). Continuous-time optimization approach for medium-range production scheduling of a multiproduct batch plant, *Industrial & Engineering Chemistry Research*, 41, 3884

Liu, G., Luh, P. B. and Resch, R. (1997). Scheduling permutation flow shops using the Lagrangian relaxation technique, *Annals of Operations Research*, 70, 171-189

Ludwig, S. A. (2010). Comparison of a Deductive Database with a Semantic Web reasoning engine, *Knowledge-Based Systems*, 23(6), 634-642

Luh, P. B. and Hoitomt, D. J. (1993). Scheduling of manufacturing systems using the Lagrangian relaxation technique, *Automatic Control, IEEE Transactions on*, 38, (7), 1066, 1079

Macal, C. M., and North, M. J. (2009). Agent-based modeling and simulation. Winter simulation conference.

Merkle, D., Middendorf, M. and Schmeck, H. (2002). Ant colony optimization for resource-constrained project scheduling, *Evolutionary Computation*, *IEEE Transactions on*, 6(4), 333, 346

Mesquita, L.F. and Lazzarini, S.G. (2008). Horizontal and Vertical Relationships in Developing Economies: Implications for SMEs' Access to Global Markets. *The Academy of Management Journal*, 51(2), 359-380

Meyer, H., Fuchs, F., Thiel, K. (2009). Manufacturing Execution Systems (MES). Optimal Design, Planning, and Deployment. McGraw-Hill, New York

Montreuil, B., Frayret, J. M. and D'Amours, S. (2000). A strategic framework for networked manufacturing, *Computers in Industry*, 42 (2-3), 299–317.

Morton, T. and Pentico, D. W. (1993). *Heuristic scheduling systems: with applications to production systems and project management*, John Wiley & Sons.

Mosconi, F. and Mantovi, A. (2012). *The 'Emilian Model' for the Twenty-First Century*, Retrieved from http://ssrn.com/abstract=1986036. [Accessed 05/06/2013]

Müller-Seitz, G. and Sydow, J. (2012). Maneuvering between Networks to Lead: A Longitudinal Case Study in the Semiconductor Industry, *Long Range Planning*, 45(2–3), 105-135

Nedic, A. and Ozdaglar, A. (2009). Distributed subgradient methods for multi-agent optimization, *Automatic Control, IEEE Transactions on*, 54(1), 48-61

Nguyen, N. T., Nguyen, T. T., Roos, M. and Rothe, J. (2014). Computational complexity and approximability of social welfare optimization in multiagent resource allocation, *Autonomous agents and multi-agent systems*, 28(2), 256-289

North, M. J., Collier, N. T., Ozik, J., Tatara, E. R., Macal, C. M., Bragen, M. and Sydelko, P. (2013). Complex adaptive systems modeling with repast symphony, *Complex adaptive systems modeling*, 1(1), 1-26

Numao, M. (1994). Development of a cooperative scheduling system for the steel-making process, in *Intelligent Scheduling*, Zweben, M., and Fox, M.S., eds., Morgan Kaufmann Publishers, San Francisco.

Oh, J. and Smith, S. F. (2004). Learning calendar scheduling preferences in hierarchical organizations, In *International Workshop on Preferences and Soft Constraints*.

Ouelhadj, D. and Petrovic, S. (2009). A survey of dynamic scheduling in manufacturing systems, *Journal of Scheduling*, 12(4), 417-431

Owliya, M., Saadat, M., Jules, G., Goharian, M. and Anane, R. (2013). Agent-Based Interaction Protocols and Topologies for Manufacturing Task Allocation, *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, 43(1), 38-52.

PABADISPROMISE consortium. *Development of a product and production process description language*. (2006). Retrieved from http://www.uni-magdeburg.de/iaf/cvs/pabadispromise/dokumente/Del_3_1_Final.pdf. [Accessed 12/09/2013]

Park, B. J., Choi, H. R. & Kim, H. S. (2003). A hybrid genetic algorithm for the job shop scheduling problems. *Computers & industrial engineering*, 45(4), 597-613.

Pérez-Urbina, H., Rodrıguez-Dıaz, E., Grove, M., Konstantinidis, G. and Sirin, E. (2012). Evaluation of query rewriting approaches for OWL 2. *Proc. of SSWS+ HPCSW*, 943

Pires, S. R., Bremer, C. F., De Santa Eulalia, L. A. and Goulart, C. P. (2001). Supply chain and virtual enterprises: comparisons, migration and a case study, *International Journal of Logistics*, 4(3), 297-311

Plüss, A. and Huber, C. (2005). Virtuellefabrik. Ch—A Source Network for VE in Mechatronics, In *Virtual Organizations*. Springer US, 255-264

Puttonen, J., Lobov, A. and Martinez Lastra, J. L. (2013). Semantics-Based Composition of Factory Automation Processes Encapsulated by Web Services, *Industrial Informatics, IEEE Transactions on*, 9, 2349-2359

Reeves, C. R. (1995). A genetic algorithm for flowshop sequencing, *Computers & operations research*, 22(1), 5-13

Renna, P. (2011). Multi-agent based scheduling in manufacturing cells in a dynamic environment, *International Journal of Production research*, 49(5), 1285-1301

Roorda, M., Cavalcante, R., McCabe, S. and Kwan, H.. (2009). A conceptual framework for agent-based modelling of logistics services, *Transportation Research Part E: Logistics and Transportation Review*, 46(1), 18–31

Roy, B. and Sussmann, B. (1964). *Les problems d'ordonnancement avec constraintes disjonctives*, Note DS No. 9 bis, SEMA, Montrouge.

Rudberg, M. and Olhager, J. (2003). Manufacturing networks and supply chains: an operations strategy perspective, *Omega*, 31(1), 29-39

Saeidlou, S., Saadat, M. and Jules, G. (2014). Cloud manufacturing analysis based on ontology mapping and multi agent systems, *Systems, Man and Cybernetics (SMC)*, *IEEE International Conference on*, 2019-2024

SAS (2009). SAS/OR(R) 9.2 User's guide: Constraint Programming, Retrieved from http://bit.ly/216nIct [Accessed 11/02/2016]

Schalkoff, R. J. (2011). *Intelligent Systems: Principles, Paradigms and Pragmatics*, Jones and Bartkett publishers.

Serugendo, G. D. M., Irit, M. P. and Karageorgos, A. (2006). Self-organisation and emergence in MAS: An overview. *Informatica*, *30*(1)

Shen, W.M. (2002). Distributed manufacturing scheduling using intelligent agents, *IEEE Intelligent Systems*, 17 (1), 88–94

Sirin, E., Parsia, B., Grau, B., Kalyanpur, A. and Katz, Y. (2007). Pellet: A practical OWL-DL reasoner, *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2), 51-53

Smith, A. D. and Rupp, W. T. (2002). Application service providers (ASP): moving downstream to enhance competitive advantage, *Information Management and Computer Security*, 10 (2–3), 64–72

Smith, R.G. (1980). The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver, *IEEE Transactions on Computers*, C-29(12), 1104-1113.

Spring, M. and Araujo, L. (2013). Beyond the service factory: Service innovation in manufacturing supply networks, *Industrial marketing management*, 42(1), 59-70

Statista. (2015). Manufacturing small and medium enterprise with growth plans in the UK Q1 2011-Q4 2013, Retrieved from http://bit.ly/1ergVaB. [Accessed 03/05/2015]

Taghipour, A. and Frayret, J. M. (2013). Dynamic mutual adjustment search for supply chain operations planning co-ordination, *International Journal of Production Research*, 51(9), 2715-2739

Tao, F., Zhang, L., Venkatesh, V. C., Luo, Y. and Y. Cheng. (2011). Cloud manufacturing: a computing and service-oriented manufacturing model, Proceedings of the Institution of Mechanical Engineers, Part B, *Journal of Engineering Manufacture*, 225 (10), 1969–1976

Taylor, M. (2009). What is sensitivity analysis? *Consortium YHE: University of York*, 1-8.

Terlouw, L. I. and Albani, A. (2013). An Enterprise Ontology-Based Approach to Service Specification, *Services Computing, IEEE Transactions on*, 6, 89-101

Thomas, A. (2014). Approaches to Supply Chain Coordination: Decomposed and Decentralised Decision Making Models, *Ph.D. dissertation synopsis*, Indian Inst. Of Tech., Bombay.

Thomas, A., Singh, G., Krishnamoorthy, M. and Venkateswaran, J. (2013). Distributed optimisation method for multi-resource constrained scheduling in coal supply chains, *International Journal of Production Research*, 51(9), 2740-2759

Tseng, F. T., Stafford Jr, E.F. and Gupta, J.N.D. (2004). An empirical analysis of integer programming formulations for the permutation flowshop, *Management Science*, 32, 285–293

Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L. and Peeters, P. (1998). Reference architecture for holonic manufacturing systems: PROSA, *Computers in industry*, 37(3), 255-274

Verstraete, P., Valckenaers, P., Van Brussel, H., Saint Germain, B., Hadeli, K. & Van Belle, J. (2008). Towards robust and efficient planning execution. *Engineering Applications of Artificial Intelligence*, 21, 304–314

Vrba, P. and Marik, V. (2010). Capabilities of dynamic reconfiguration of multiagent-based industrial control systems. Systems, Man and Cybernetics, *Part A: Systems and Humans, IEEE Transactions on*, 40(2), 213-223.

Wang, M., Wang, H., Vogel, D., Kumar, K. and Chiu, D. K. W.. (2009). Agent-Based Negotiation and Decision Making for Dynamic Supply Chain Formation, *Engineering Applications of Artificial Intelligence*, 22 (7), 1046–1055

Wang, T., Guo, S., and Lee, C. G. (2014). Manufacturing task semantic modeling and description in cloud manufacturing system, *The International Journal of Advanced Manufacturing Technology*, 71(9-12), 2017-2031

Wells, P. (2013). Sustainable business models and the automotive industry: A commentary, *IIMB Management Review*, *25(4)*, 228-239

Wooldridge, M., Endriss, U., Kraus, S. and Lang, J. (2013). Incentive engineering for Boolean games, *Artificial Intelligence*, 195, 418-439

Yan, J., Ye, K., Wang, H. and Hua, Z. Ontology of collaborative manufacturing: Alignment of service-oriented framework with service-dominant logic, *Expert Systems with Applications*, 37(3), 2222-2231

Yılmaz, Ö., Erdur, R.C. (2012). iConAwa – An intelligent context-aware system, *Expert Systems with Applications*, 39(3), 2907-2918

Yimer, A. D. and Demirli, K. (2010). A genetic approach to two-phase optimization of dynamic supply chain scheduling, *Computers & Industrial Engineering*, 58(3), 411-422

Zhang, L., Luo, Y., Tao, F., Hu Li, B., Ren, L., Zhang, X., Guo, H., Cheng, Y., Hu, A. and Liu, Y. (2014). Cloud manufacturing: a new manufacturing paradigm, *Enterprise Information Systems*, 8(2), 167-187

Zhang, W., Zhang, S., Qi, F. and Cai, M. (2014). Self-Organized P2P Approach to Manufacturing Service Discovery for Cross-Enterprise Collaboration, *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, 44, 263-276.

Publications by the author

Peer-reviewed journal papers

Jules G. D., Saadat M, "Agent Cooperation Mechanism for Decentralised Manufacturing Scheduling", IEEE Transactions on Systems, Man and Cybernetics: Systems, Under revision.

Jules G. D., Saadat M, Saeidlou S, "Knowledge and Agent Based System for Decentralised Scheduling in Manufacturing", IEEE Transactions on Systems, Man and Cybernetics: Systems, Under revision.

Jules, G. D., Saadat M. and Saeidlou, S. (2015). Holonic Ontology and Interaction Protocol for Manufacturing Network Organization, *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, 45(5), pp. 819-830

Owliya, **M.**, Saadat, M., **Jules, G. D.**, Goharian, M., and Anane R. (2013). Agent-Based Interaction Protocols and Topologies for Manufacturing Task Allocation, *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* (43), pp. 38-52.

Peer-reviewed conference papers

Jules, **G. D.,** Saadat, M., and Saeidlou, S. (2013). Holonic goal-driven scheduling model for manufacturing networks, in *2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1235-1240.

Jules, G. D., Saadat, M., and Saeidlou, S. (2013), Ontological extension of PROSA for manufacturing network formation, in *Industrial Applications of Holonic and Multi-Agent Systems,* Springer, pp. 140-151.

Jules, G. D., Saadat, M., and Li, N. (2013), On designing a unified ontology for holonic manufacturing networks, in *Integration of Practice-Oriented Knowledge Technology: Trends and Prospectives*, Springer, pp. 207-220.

Jules, G. D,, Saadat, M. and Li, N. (2012), Holonic multi-agent system: Simulation of manufacturing network formation, in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC),* pp. 686-691.

Jules, **G. D**., Saadat, M., Tan, M. C. and Owliya, M., (2011). Holonic-based flexible allocation of workforce in labour intensive manufacturing shop floors, in *2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp.* 434-439.

Li, N., Saadat, M. and **Jules, G. D.,** (2012), Supplier selection using modified fuzzy AHP-based approach in manufacturing networks, in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp.* 1847-1852.

Jules, **G. D**., Saadat, M. and Owliya, M. (2010), A holonic systems approach to the formation of manufacturing networks, in *2010 IEEE 9th International Conference on Cybernetic Intelligent Systems (CIS),* pp. 1-6.