An Agent-based Approach for Manufacturing Production Scheduling with Emission Consideration

Submitted by ***Ruiqiang Lu****, to the University of Exeter*
as a thesis for the degree of
Doctor of Philosophy in Engineering
In December 2012

I certify that all material in this thesis which is not my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other University.

Signature: ………………………………………

ACKNOWLEDGEMENT

I would like to firstly thank to my supervisor, Professor David Zhang, for his kind encouragement, motivation, suggestion and advice throughout my PhD research. It is David who provided me with guidance and help in every stage of writing this dissertation.

I acknowledge my colleagues in Exeter Manufacturing Enterprise Centre and all the staff in College of Engineering, Mathematics and Physical Sciences for their kind assistance.

I would also like to thank all my friends, especially Yunfeng (Deve) Shang, Aidi Xu, Naihui He, Liyi Chen, Junlu Zhu, Zhaotong Lei, Rundong Wang, Mark Errington and Xiao O, who shared every single joyful and sorrowful moment with me throughout the period of this PhD study. Your friendship makes my life a wonderful experience.

I owe my deepest gratitude to my parents, Junping Cui and Fengru Lu for their endless love, support and encouragement throughout these years.

At last but not the least, I am grateful to Ms Qing He, who gave me help and inspiration during the past few years.

ABSTRACT

In the current business climate with increasingly changing customer requirements and strong business competition, manufacturing organisations need to enhance their productivity and adaptability in order to survive in the current business environment and raise their competitiveness. As a result, the optimisation of production scheduling in manufacturing systems has attracted increasing attention by manufacturers. The optimisation of manufacturing scheduling can be simplified as an optimisation problem for minimising processing cost and time with a set of constraints reflecting the technical relationships between jobs or job features and the resource capability and capacity. Conventional optimisation approaches including mathematical approaches, dispatching rules, heuristics and meta-heuristics have been applied in this research area but optimal solutions cannot be achieved in a reasonable computational time. In this PhD research, an agent based approach is developed for solving the manufacturing production optimisation problem. There is an agent iterative bidding mechanism coordinated by a Genetic Algorithm (GA) which facilitates the search for optimal routing and sequencing solutions for processing an entire job with shared manufacturing resources. A shop agent in the system works as a mediator which announces bidding operations, collects bids and decides winner machines according to a weight-based function. Machine agents with specific technical capability calculate the total production cost and lead time for job operations according to the predesigned operational sequence, and decide whether to submit their bids based on local utility. Another agent self-adjusting mechanism is employed for resource agents updating the priorities of unprocessed jobs in their buffers. The objective of each machine agent is to maximise local utility, i.e., to increase individual profit. After genetic generations for updating parameters with agent self-adjusting, the

near optimal schedule plans can be found.

On the other hand, the use of energy in all organisations has become a key issue worldwide. Carbon emissions from manufacturing processes of a company are under the pressure of government and also affect the public opinion. In the previous works from the literature, however, economic and environmental issues are not considered simultaneously in manufacturing production scheduling. Based on the basic agent based optimisation mechanisms, two extensive models with the consideration of the carbon emission during production are built in this research work, where the emission factor is set to be a constraint and another objective respectively. Numerical tests are utilised in order to examine the effectiveness and efficiency of the proposed approaches. Furthermore, two previous approaches from the literature for solving the same problems are rebuilt and results are compared for testing the comparative performance of the proposed approaches. Test results show that near optimal schedule plans can be achieved in a reasonable computational time.

CONTENTS

LIST OF TABLES

LIST OF FIGURE

LIST OF ABBREVIATIONS

ACI: Ant Colony Intelligence

ACL: Agent Communication Language

ACO: Ant Colony Optimisation

ADDYMS: Architecture for Distributed Dynamic Manufacturing Scheduling

AVG: Automated Guided Vehicle

BATs: Best Available Technologies

CAMPS: Constraint-based Architecture for Multiagent Planning/Scheduling

CNP: Contract Net Protocol

DAS: Distributed Asynchronous Scheduler

DIMS: Dynamically Integrated Manufacturing Systems

DM: Decision Maker

DR: Dispatching rules

EA: Evolutionary Algorithm

ED: Earliest Dispatch

EDD: Earliest Due Date

EFR: Emission Factor Rating

FCFS: First Come First Serve

FIFO: First In First Out

FIPA ACL: FIPA Agent Communication Language

GA: Genetic Algorithm

ILC: Iterative Learning Control

JDK: Java Development Kit

KQML: Knowledge Query Manipulation Language

MAS: Multi-agent System

MASS: Multi-agent Scheduling System

MDD: Modified Due Date

MINSLACK: Minimum Slack

MOEA: Multi-objective evolutionary algorithms

MOO: Multi-objective Optimisation

MTO: Make-to-order

MTS: Make-to-stock

NP-hard: Non-deterministic Polynomial-time hard

NSGA – II: Non-dominated Sorting Genetic Algorithm – II

oHAN: Online Hybrid Agent-based Negotiation

P: Progress Measure

P-TATO: Precedence cost Tatonnement

PA: Part Agent

SD: Shortest Distance

SETG-Plan: Strategic Energy Technology Plan

SOCT: Shortest Operation Completion Time

SP: Spacing

SPEA – 2: Strength Pareto Evolutionary Algorithm 2 (SPEA–2)

SPT: Shortest Process Time

SSLP: Shortest Set-up with reference to the Last Part in queue

SST: Shortest Set-up Time

TSC: Two-set coverage

UNFCCC: United Nations Framework Convention on Climate Change

USEPA: United States Environmental Protection Agency

WA: Workstation Agent

WSA: Weighted-sum Aggregation

PUBLICATIONS

Lu, R. and Zhang, D.Z., (2010) A novel agent bidding based optimization approach in manufacturing planning and scheduling, 40th International Conference on Computers and Industrial Engineering, Awaji Yumebutai, Hyogo, Japan, 26th – 28th Jul 2010, 1-6.

Lu, R. and Zhang, D.Z., (2012) An agent based optimization approach for manufacturing process planning and scheduling with emission consideration, 17th International Working Seminar on Production Economics, Innsbruck, Austria, 21st – 24th Feb 2012, 1, 351-362.

CHAPTER 1

INTRODUCTION

1.1 Introduction

The first Chapter of this thesis provides an introduction to the study. This chapter is organised as follows. Section 1.2 presents the background in relation to the study. Section 1.3 states the research questions, followed by the discussion on objectives to be achieved in Section 1.4. Section 1.5 provides an overview of the remaining chapters of this thesis.

1.2 Research Background

With the economic globalisation, today's manufacturing organisations are facing a dynamic market with increasingly changing customer requirements and strong business competition. The ability of manufacturing organisations to adapt to the changing of customer needs has become an essential issue for them to survive in the current business environment and raise the competitiveness (Ouelhadj and Petrovic 2009). Production scheduling is one of the most important manufacturing activities which influence the productivity and adaptability of manufacturing organisations.

Scheduling is a manufacturing decision-making process, which is defined as the assigning of manufacturing resources and time to the set of requested jobs or tasks according to process plans with the goal of optimising certain criteria such as operational cost, job lateness and the job delivery time (Graves 1981). It determines the most appropriate time to execute each job operation with the consideration of a set of rules or constraints, which reflect the technical relationships between jobs or job features, as well as the capability and capacity limitations of the shared manufacturing resources (French 1982).

Scheduling optimisation problems are considered as NP-hard problems, meaning that it is impossible to find an optimal solution without the use of an enumerative method and the computational time grows exponentially with increase of the problem size, i.e., the number of requested jobs and manufacturing resources (Bagchi 1999). When unforeseen situations are considered in scheduling, the problem of seeking an optimal solution for performing a required job becomes more complex. The manufacturing system may be asked to produce emergent tasks, to adapt to changes on existing tasks or to cancel tasks. Manufacturing resources, on the other hand, may not be always available or reliable, as machine failures and unavailability of tools or materials happen unpredictably during production.

A wide variety of algorithms and approaches have been used in the manufacturing scheduling optimisation problems. Mathematical approaches such as Integer Programming and Lagrangian Relaxation were applied in non-complex scheduling problems. Due to their simplicity and extremely high computational complexity, although they can find optimal solutions for small-scale scheduling problems, they can hardly be practicable to solve complex problems in a reasonable time. Some other researchers developed several dispatching rules for establishing priorities of scheduling jobs/operations, which are assigned to jobs waiting to be processed in a common resource buffer queue based on certain criteria. Good solutions can be found by adopting dispatching rules but optimal ones cannot be reached, because such comparatively fixed rules cannot meet the need of a whole processing line in a dynamic manufacturing environment, particularly when instant events occur (Wu et al. 1993). In recent decades, many other approaches search for near-optimal schedule plans using meta-heuristics such as Genetic Algorithms, simulated annealing, Tabu search and neural networks. Satisfied solutions are achieved in relatively complex problems but these approaches lack flexibility due to their centralised structure (Shen 2006). Nowadays, increasing efforts are made in applying distributed intelligent agent techniques to solve scheduling optimisation problems. Multi-agent systems are decentralised and autonomous systems made up of distributed agents, which can act/react locally and have the abilities to cooperate and coordinate their actions dynamically motivated by local and global objectives. Such techniques have proven a major success in providing more efficient, flexible and adaptable optimisation approaches to complex and dynamic scheduling problems (Badr 2011).

On the other hand, global warming has recently become a key issue worldwide with the development of the global industry. Traditional fossil fuels such as coal and petroleum are still the first choice of energy sources for manufacturing systems. Manufacturing emissions have a significant effect on the composition of the atmosphere in the earth and therefore gradually change the global climate. The increasing number of emission related policies and legislations enacted by local governments and international organisations, together with the rising price and demand for such nonrenewable fuels, have resulted in greater efforts toward the reduction of exhausted emissions during manufacturing processes. Moreover, less waste during production also reflects a better utilisation of input material in production activities, and a lower waste disposal cost (Schipper 2006). Therefore, emission reduction can help manufacturers increase their productivities and production efficiency. Many approaches have been proposed for reducing the amount of production emissions. Searching for alternative energy sources to replace fossil fuels has been stated impracticable in the near future. The improvement of manufacturing equipment technological efficiency succeeds in certain specific types of manufacturing systems but cannot provide generically applicable approaches for reducing production emissions (Wang et al. 2011). Other methods with the enhancement of the system operational efficiency can provide more general and flexible approaches in emission reduction problems, where the emission factor is considered simultaneously with the overall production performance of the manufacturing systems (Mouzon et al. 2007).

To sum up, together with the objective to increase the productivity, flexibility and production efficiency, the current manufacturers also need to consider the problem as how to decrease the emissions released in the production processes. To date, there has not been a sophisticated method for enhancing cost-efficiency for manufacturers and simultaneously reducing production emissions. In this research work, an approach for optimising manufacturing scheduling with both economic and environmental consideration is developed with intelligent agent techniques.

1.3 Research Questions

Based on the background of this research area introduced in the last section, three research questions need to be answered in this study:

- Question 1: What and how the inherent attributes of intelligent agents, such as the abilities of learning, cooperation and coordination, can be involved in an agent based approach to search for optimal or near optimal scheduling solutions?
- Question 2: What does the manufacturing emission include and what problems relate to emissions that a general manufacturer faces in the current business climate?
- Question 3: How to introduce the manufacturing emission factor into the agent based scheduling optimisation problem, in which the primary objective is to be cost-efficient and flexible to fulfill customer orders?
- Question 4: How to solve the multi-objective optimisation problem if production cost and emission need to be minimised simultaneously?

1.4 Research Objectives

In the research area of manufacturing scheduling, the primary objectives in the optimisation problems are often defined as the minimisation of operational cost or the production lead time. Emissions are rarely taken into consideration simultaneously. Therefore, the problem addressed in this study aims to investigate the use of intelligent agent techniques for the manufacturing scheduling problem, with the consideration of manufacturing emissions. When the manufacturing emission factor is considered together with production scheduling problems, the amount of exhausted emission in production processes are aimed to be minimised. In this context, in this problem there may be more than one objective to be optimised under certain technical and system constraints.

Therefore, the main objectives of this PhD research are listed as follows:

- To define and to represent the scheduling problem with certain criteria and constraints to be solved by using equations and inequations.
- To review contributions and shortages of all the existing approaches for solving manufacturing scheduling problems including conventional and intelligent agent based approaches problems, and to find the research gap or to suggest what can be further developed.
- To build an agent based model to represent manufacturing systems composed of different agent types representing various manufacturing activities and facilities, and to develop an effective and efficient algorithm used in the model to search for optimal or near-optimal scheduling solutions.
- To clarify and to analyse the emission problems related to manufacturing and manufacturing scheduling as well as the emission measurement methods.
- To find a method or methods for involving manufacturing emissions into the standard production scheduling problem.
- To implement the solution algorithms based on Java programming, and to validate the algorithms using test cases and comparisons.

1.5 Thesis Organisation

Apart from this chapter, an overview of the rest of the chapters in this thesis is given as follows:

- Chapter 2: A Literature Review on Manufacturing Production Scheduling and Agent-based Optimisation Approaches. This chapter provides a literature review of existing approaches for solving manufacturing production scheduling problems. General definitions of manufacturing systems and production scheduling are given at the beginning. Conventional optimisation approaches for scheduling including mathematical approaches, dispatching rules, heuristics and meta-heuristics are reviewed. Agent technology is then introduced. The concepts of agents, multi-agent systems and their applications to manufacturing control and scheduling optimisation are discussed. The strengths and limitations of the existing approaches are also analysed in this chapter.

- Chapter 3: An Agent Based Approach for Optimisation of Manufacturing Production Scheduling. In this chapter, an agent based approach for solving the production scheduling problem is described. The basic problem is firstly formulated mathematically and the computational complexity is discussed. The proposed approach includes two parts: the first part is an agent bidding based process controlled by Genetic Algorithms, and the second part is an agent self-adjusting process where each resource agent updates the job priorities in local buffer waiting lists according to its local utility and the information from the previous bidding round. The new combinative algorithms aim to find the optimal or near optimal routing and sequencing plans for carrying out a requested job with the lowest overall production cost and satisfied job due date.

- Chapter 4: Multi-agent Modelling Architecture. This chapter introduces the architecture and functions of the model based on multi-agent systems for implementing the algorithms introduced in Chapter 3. Different agents in the model including job agent, shop agent, machine agent and material handling agent and their architectures and functions are described.

- Chapter 5: Manufacturing Emissions. In this chapter, the factor of manufacturing emissions is involved into the standard agent based algorithms for the production scheduling problem. Background, related policies and measurement methods of manufacturing emissions are firstly introduced, followed by the existing approaches to reduce emissions during production. This work provides two possible methods to consider the scheduling optimisation in terms of improving the cost-efficiency simultaneously with emission reduction. The first one is to assume the emission factor as a constraint, i.e., the emission exhausted during production cannot exceed a certain amount. The second method is to consider the emission reduction as a second objective. When there is more than one objective to be optimised in a problem, it belongs to the multi-objective optimisation (MOO) problems. MOO problems are briefly introduced and a potential algorithm for solving this particular problem is given.

- Chapter 6: Numerical Tests and Results. This chapter introduces the numerical tests to validate the proposed approaches. The first test is to examine the effectiveness and efficiency of the standard agent based model, together with the constraint of exhausted emission amount. In the second test, the multi-objective optimisation algorithm for minimising both production cost and emission is implemented. Results obtained from the experimental tests under different conditions are presented and discussed.

- Chapter 7: Comparative Experimentation. This chapter is a continuation of the previous chapter. The results from the tests presented in Chapter 6 are further discussed by comparing them with the results obtained by two other approaches from literature with the same objectives. Two models are rebuilt based on another agent bidding based mechanism and a combinative dispatching rule based mechanism, in order to investigate the comparative performance of the proposed approach. Tests and results are discussed using standard performance measuring metrics, and based on the results a discussion is given in the end.

- Chapter 8: Conclusion and Future Work. In the last chapter of this thesis, the conclusions of this PhD research are summarised, the major contributions of this work are clarified and the limitations are pointed out. In the end, suggestions for future work extensions are proposed.

CHAPTER 2

A LITERATURE REVIEW ON MANUFACTURING PRODUCTION SCHEDULING AND AGENT-BASED OPTIMISATION APPROACHES

2.1 Introduction

This chapter provides a literature review of the existing approaches for solving the manufacturing production scheduling optimisation problem. As one of the main objectives of this thesis is to solve the scheduling problem by adopting agent techniques, the literature review of the conventional approaches and agent-based approaches are listed and discussed respectively. In the rest of this chapter, Section 2.2 introduces the definitions and types of manufacturing systems and production scheduling. Section 2.3 reviews the conventional approaches for solving production scheduling problems including mathematical approaches, dispatching rules, heuristics and meta-heuristics. Section 2.4 discusses agent, multi-agent systems and their applications to manufacturing control and production scheduling optimisation. In Section 2.5, the issues that the existing approaches are not able to address are discussed. A summary of this chapter is given in Section 2.6.

2.2 Manufacturing Production Scheduling

Manufacturing systems include a series of organised activities devoted to the transformation of raw materials into products (or services) for use or sale (Hill 1983). The factors of production in the manufacturing systems input consist of production objects, productive labour, production means and production information. The production output, on the other hand, includes profit, quality, delivery and reputation. In brief, what a manufacturer does is to add value between the production input and output. The processes involved in manufacturing are market forecasting/receiving order, planning to fulfill market/order, executing plans, purchasing converting materials, controlling and execution and monitoring manufacturing and finance (Hitomi 1996).

There are two types of manufacturing systems distinguished by the mode of operation. One is the make-to-stock (MTS) systems, which produce goods for stock replenishment, along with sales forecast information. It has comparatively high certainty of product specification, and the exact specification and volumes are predicted in advance. In this type of manufacturing systems, production plans can be easily made but when the number of products is large, producing to stock becomes costly. Meanwhile, it is risky when demand is variable and products have short life cycles (Handfield R.B. 1994). The other type is the make-to-order (MTO) systems, which begin to carry out products once a confirmed order for these products is received. This type of manufacturing systems has relatively low stock cost and highly customised products, but the certainty of product specifications and volumes are comparatively low, since the exact job specification is only established on the requirement of individual customer's order. Furthermore, it usually spells long production lead time and large order backlogs (Arreola-Risa and DeCroix 1998).

According to Stecke (1985), there are three main categories of manufacturing functions concerning the management of production, namely process planning, production scheduling and control. Process planning specifies how a product in a manufacturing system is produced with resource limitation as well as technical and product constraints. It is a process which selects and sequences manufacturing processes and parameters to convert a product from raw material to a final form. The aim is to produce a process plan which maximises the utilisation of the production system (Zijm 1995). Based on the given plan, a schedule is generated by assigning jobs or job operations to shared manufacturing resources, and it is then delivered to the shop fool control for execution (Smith et al. 2000). Shop floor control monitors the production processes and collects data of shop floor status in order to ensure that the schedule meets the specification of the process plan. When problems such as machine breakdown, machine maintenance or unavailability of material are detected in production processes, actions are taken by shop floor control in order to minimise the deviation from the planned schedule (Vieira et al. 2003). In addition, quality assurance is also a responsibility of the control process.

For production scheduling, there are two sub-problems: routing and sequencing. In the routing sub-problem, different job operations are allocated to a set of shared machines in the system according to the operational order in the process plan. In the sub-problem of sequencing, the most appropriate arrangement of the allocated jobs at a given machine needs to be decided (Badr and Gohner 2010). When a job or job part is loaded into the system, it needs to be transported to the current machine and stored in its input buffer. Hence, the operational priorities of different parts in the buffer have to be determined, which is followed by setting up the machine and processing the part. Based on the work plan of the job, the finished part is stored in the output buffer to wait for the following operation being processed, or transported to the input buffer of another machine (El Khayat et al. 2006). The solutions of production scheduling problems are always required to meet certain predesigned performance criteria such as minimising the total production cost and satisfying the job delivery due date. In the make-to-order type of manufacturing systems, due to the uniqueness of individual product batches from different customers, two important performance indicators of a manufacturer are the total production cost and the delivery time for a single customer order. By adopting a good schedule, the production lead time for completing a job order is expected to satisfy the delivery due date. Meanwhile, resources in the manufacturing system can be utilised effectively and the production cost is kept low.

2.3 Conventional Approaches for Manufacturing Scheduling

2.3.1 Mathematical Approaches

Before the techniques of computer modelling and heuristics were involved in the production scheduling optimisation problem, some previous researchers focused on mathematical techniques and algorithms, including Integer Programming (Balas 1965) and its extension Dynamic Programming (Srinivasan 1971) and Lagrangian Relaxation (Shapiro 1979). These methods usually involve mathematical formulations with their associated parameters, constraints and objectives in order to seek the optimal schedules. Mathematical techniques have been extensively applied in the production scheduling optimisation problems, but due to the simplified assumptions in these approaches, it was stated that the generation of optimal solutions dramatically growing computational time with the increase of the number of jobs and manufacturing resources (Lenstra et al. 1977). The other drawback of the mathematical approaches is that they can hardly meet the need of the real manufacturing systems in terms of fast response to environmental changes, such as new job arrive, machine break down and changed operational requirements (Wu 1994).

2.3.2 Dispatching Rules

In order to find solutions to complex production scheduling problems, many fixed dispatching rules have been developed for establishing priorities of scheduling jobs/operations, which are assigned to jobs waiting to be processed in a common resource buffer queue based on a certain criterion. According to Wu (1987), Xu and Randhawa (1996) and Jeong and Kim (1998), dispatching rules are categorised into three classes. The first class includes simple priority rules, which are based on certain information of jobs or sub-jobs. For example, Shortest Process Time (SPT) rule is based on job processing times, i.e., jobs with shorter processing times are assigned with higher priorities. Earliest Due Date (EDD) and Modified Due Date (MDD) follow the required job due dates, Minimum Slack (MINSLACK) is based on slack conditions, and First-In-First-Out (FIFO) is based on the arrival times of jobs or sub-jobs. The dispatching rules belonging to the second class are combinations of rules in the first class. Particular rules are implemented according to different conditions of the existing information in the system. The third class consists of the rules that use more than one type of job information to determine the scheduling solutions. “Weights” are used to distinguish different importance of different types of information.

Kim and Kim (1994) presented a simulation based scheduling system, in which the simulation mechanism evaluates different dispatching rules and selects the best one among them. There was another reactive control mechanism monitoring the system operation periodically and determines the start of new simulation runs. Jeong and Kim (1998) adopted dispatching rules for real-time production scheduling with the dynamic environment of urgent jobs, machine breakdown and tool breakage. Several dispatching rules were evaluated and a best one which satisfied certain criteria was selected by the scheduler. Rajendran and Holthaus (1999) analysed and compared the performance of 13 different dispatching rules in job shop scheduling with the objectives of minimising the mean and variance of production flow-time, and mean and variance of job tardiness respectively. Kutanoglu and Sabuncuoglu (1999, 2001) presented a comparative study of more than 20 dispatching rules in job shop scheduling problems with weighted tardiness criterion for dynamic job arrivals. The simulation was based on rerouting the jobs on their alternative machines.

Due to their simplicity and low computational complexity, dispatching rules have been widely used to optimise the performance of a manufacturing system in terms of different criteria, such as total tardiness, the number of late jobs, system utilisation and resource usage. Good solutions can be found in terms of minimal total processing cost, shortest lead time or other criteria. Optimal schedules, however, are rarely obtained, because dispatching rules are comparatively fixed but the manufacturing systems are commonly dynamic. A single dispatching rule or a combination of a few rules may not be satisfied for a whole processing line, particularly instant events occur such as machine breakdown, new product types adding in, and system rescheduling. In order to overcome this obstacle, Smith (2003) presented an agent-based scheduling system which applied different dispatching rules for generating and updating schedules. Based on a knowledge base, those rules are selected dynamically according to the current status on the shop floor. Due to their highly integrative ability, dispatching rules based approaches have been combined with other techniques to deal with the manufacturing production problems, which will be introduced in the following sections.

2.3.3 Heuristics and Meta-heuristics

Heuristics are schedule repair methods, including the well-known right-shift schedule repair (Yamamoto and Nof 1985, Mehta and Uzsoy 1999), match-up schedule repair (Bean et al. 1991, Akturk and Gorgulu 1999), and partial schedule repair methods (Kutanoglu and Sabuncuoglu 2001). The optimal schedule is not guaranteed to be found with heuristic methods, but good solutions can be reached in a relatively short time. A variety of combination schedule repair heuristics have been proposed in the previous literature. Abumaizar and Svestka (1997) discussed the comparative performance of the schedule repair heuristics, affected operations schedule repair and right-shift schedule repair and completed rescheduling with respect to measures of the global system efficiency and stability. The results showed that the affected operations schedule repair reduced much of the deviation as well as the computational complexity associated with right-shift schedule repair and complete rescheduling. Jain and Elmaraghy (1997) proposed various schedule repair heuristics for production rescheduling in flexible manufacturing systems with the consideration of different criteria such as machine breakdown, arrival of emergent jobs, job priority adjustment and job cancellation.

It has been discussed that one of the drawbacks of simple heuristics is that the generated solutions may be trapped in a local optimum with poor performance (Dorn et al. 1995). Meta-heuristics, which are high level heuristics guiding local neighborhood search heuristics to escape from local optima, have been successfully applied to solve production scheduling problems (Pham and Karaboga 2000). In a general step of the most neighbourhood search methods, a different solution is iteratively constructed from the previous solution, and it is examined whether the search terminates or attempts another step. With meta-heuristics such as simulated annealing, Tabu search and Genetic Algorithms, good solutions to complex scheduling problems can be found within a reasonable computation time.

Simulated annealing is inspired from the physical process of annealing, in which a lattice structure of a solid is being heated up until it melts, and then the system temperature is gradually lowered until it solidifies into a steady low-energy state. In each step of the simulated annealing algorithm, the current solution is attempted to be replaced by another solution randomly constructed to sample from the solutions near the current one. Whether a new solution is accepted depends on the difference between the two objective function values and also on a parameter representing the global temperature, whose value is gradually decreased during annealing steps. Simulated annealing has been successfully applied in many manufacturing scheduling problems (Kirkpatrick et al. 1983, Palmer 1996, Song et al. 2012).

Tabu search is another local search method which uses memory structures to enhance the searching performance. If a potential solution is visited previously within a certain term period, it is marked as “Tabu” (or Taboo) and will not be visited repeatedly in the future. The deterministic iterative improvements are combined with the possibility of accepting cost-increasing solutions in order that the local optimum can be avoided (Glover and Laguna 1997). Nowicki and Smutnicki (1996) applied the Tabu search algorithms in job shop scheduling problem and an improved algorithm with a path relinking technique was provided by the same authors in 2005. Zhang et al. (2008) introduced a hybrid Tabu search and simulated annealing algorithm, in which the simulated annealing was used to generate the initial set of high-quality solutions, and these solutions were then processed via Tabu search-driven intensification. Vinicius (2012) applied Tabu search in flow shop scheduling problems for minimising total tardiness. Tabu search is used to expore the solution space, which is composed by analysing different due date scenarios, where diversification, intensification and neighbourhood restriction strategies are evaluated.

Inspired by natural selection and evolution, Genetic Algorithms (GA) are developed for iteratively searching optimal or near optimal solutions from solution populations. In GAs, candidate solutions are represented by natural chromosomes (Holland 1975). In each GA generation, the chromosomes with highest fitness, which are selected to be parents, produce offspring chromosomes by using different pieces from each parent, which is called crossover. By the processes of selection and crossover, the search improvements are small and local rather than huge jumps in the solution space. Additionally, an arbitrary bit or bits of an offspring chromosome can be mutated from its original state with a pre-designed possibility, in order that the search can be directed away from local optimum. The new chromosome population is formed according to the fitness of individual chromosomes, and only the ones with highest fitness can survive in the next generation. The genetic process stops until a termination condition is reached, for instance, the successive iterations can no longer produce better solutions, or a fixed number of GA generations are finished. Wu et al. (1993) compared the performance of GAs and local search heuristics with respect to generating robust schedules. Comparative results showed that the optimised solutions of GAs are much better than local search heuristics in terms of the production lead time and stability. GA applications to manufacturing production scheduling problems can be found in a huge number of previous literatures including (Davis 1985, Morad and Zalzala 1999, Hajri et al. 2000, Zhao and Wu 2001, Baskak and Erol 2005, Chaudhry and Drake 2008).

These meta-heuristic search approaches including simulated annealing, Tabu search and GAs, as well as other experience and learning based methods such as artificial neural networks (Sabuncuoglu and Gurgun 1996) and fuzzy logics (Sauser et al. 1998) can provide near optimal solutions in complex optimisation problems of production scheduling, but they encounter great difficulties when they are applied in real manufacturing. This is because such theoretical models share the same disadvantage with other conventional scheduling optimisation approaches that all computations are carried out in a central controlling unit. It leads these traditional approaches to inefficient, inflexible, inadaptable and expensive to satisfy the need of real-world scheduling problems (Shen 2002). The intelligent agent technique, on the other hand, provides distributed approaches, in which manufacturing resources are represented as distributed agents. Without a centralised controller, each agent in the system can make autonomous decisions and handle local schedule individually. Through agent individual actions and negotiation among agents, global schedule solutions can be obtained. Agents and agented based approaches in scheduling will be discussed in the following section in detail.

2.4 Agents and Agent-based Optimising Approaches for Manufacturing Scheduling

2.4.1 Agents and Multi-Agent Systems

Computer simulation has become a widely used tool for modelling many fields of natural systems, in order to observe their behaviour and to gain the insight into the operation of the system. Agent-based modelling, which is a type of computer technique for simulating discrete events with individual entities and their interactions and relationships, has been successfully applied in many different academic fields including manufacturing production scheduling. This section will deliver an introduction of agents, multi-agent systems and their applications to the manufacturing scheduling problems.

2.4.1.1 Definition and Properties of Agents

Agents are entities represented by computer systems with the ability of making autonomous decisions, interacting with others and responding to the environmental changes (Wooldridge and Jennings 1995). Figure 2.1 shows a simple agent's architecture and the relationship between the agent and its environment. An agent is able to perceive its environment through its sensor and acts upon that environment through the actuator. The information received from sensor is transformed by the interpreter to formative languages which the agent can read and deal with. The decision maker or controller keeps the agent internal knowledge and rules, based on which the agent makes decisions and selects an action to be taken for the next step.

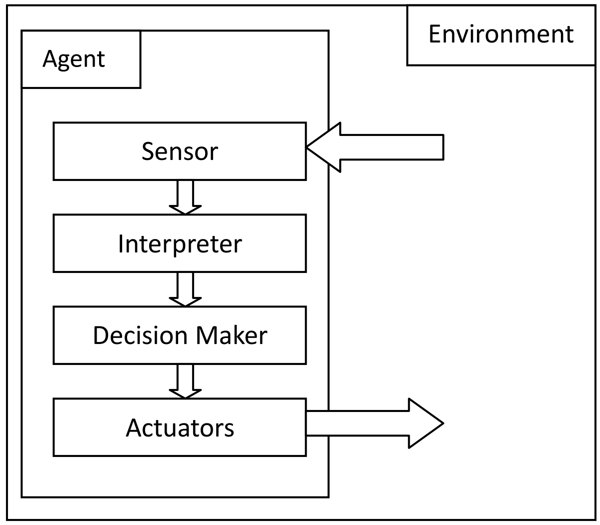


Figure 2.1 Simple Agent Architecture

According to Wooldridge (1997), the key properties of agents are:

- Agents take actions to achieve a goal which is decided by its designer or user.
- Agents are autonomous in terms of controlling both their internal states and external behaviours in the environment.
- Agents are able to perform intelligent behaviours, such as learning, calculating and communicating.
- Agents can interact with their environment and with other agents.
- Agents are adaptive. They can respond to the environmental changes.

2.4.1.2 Agent Types

According to different types of behaviour, agents can be classified into reactive agents, deliberative agents and hybrid agents. Based on the functionality of agents, they can be classified into interface agents and information/internet agents. Furthermore, based on the mobility of agents, they can also be classified into mobile agents and stationary agents (Stone and Veloso, 2000).

Deliberative agents with a planning system are able to undertake a sequence of actions with the intent of achieving individual goals. Their actions are based on the search within a space of behaviours, maintaining internal states and the prediction of the actions effects. When the complexity of the problem is increasingly high, deliberative agents cannot behave satisfactorily due to their low reaction in real time. Reactive agents, on the other hand, retrieve pre-set behaviours without maintaining internal state. The actions of reactive agents are taken based on their knowledge acquired from sensing of the environment. The combination of deliberative agents and reactive agents is called hybrid agents, which overcome the weaknesses of these two types of agents (Muller 1998).

2.4.1.3 Multi-Agent Systems

The environment of an agent may also include other agents. The community composed by a population of agents is called a multi-agent system (MAS) (Wooldridge 1997). In other words, an MAS consists of several agents and their environment. Important properties in the aspect of individual agent are concluded by Wooldridge (2000):

- The agents' own knowledge on the environment, which may be incomplete or incorrect.
- The interaction between agents, and between an agent and its environment.
- The individual goals of various agents.

- The actions that agents take and effects of these actions.

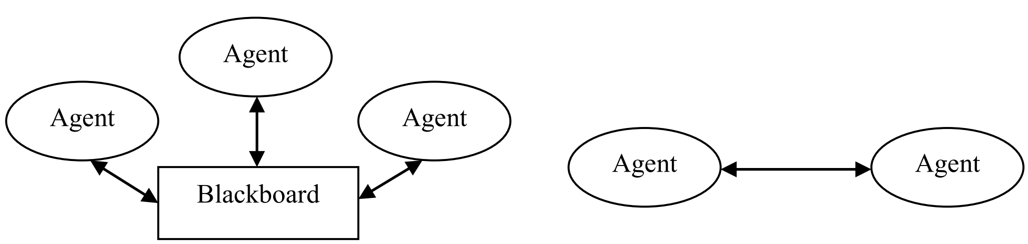
MASs provide agents a decentralised environment where executions and controls can be performed collaboratively by agents. Nevertheless, agents are autonomous. Although being affected by others and the environment, an agent in an MAS makes decisions and takes actions based on its own utility. This means there are several threads of task control and execution in the system simultaneously without internal dependency. If an agent quits or a new agent enters into the system, others are not affected. Meanwhile, agents are able to solve individual problems locally rather than waiting for a solution provided by the system central controller, which leads to the time reduction when solving global problems (Jennings et al. 1998).

The architectures of MASs can be categorised into centralised architectures, distributed architectures and hybrid architectures based on different agent collaboration methods. In centralised multi-agent architectures, there is always a master-slave relationship between different types of agents. Agents at the lower level in the system always take actions followed the command from the higher level agents. In distributed or decentralised multi-agent architectures, agents take actions based on their own knowledge and information observed from the environment rather than orders from higher level controllers. The combination of centralised and distributed MAS architecture is called hybrid architectures, in which there is usually a mediator who coordinates the communication among distributed agents (Ferber 1999).

2.4.1.4 Agent Communication and Coordination

One of the most significant characteristics of agents in a multi-agent system is the ability to coordinate their knowledge, skills, goals and plans with others. According to Jennings (1996) and Shen et al. (2001), agents need to communicate and coordinate with each other because firstly, many complex problems require different expertise, resources and information, which cannot be solved by individual agents in isolation; secondly, since in a decentralised multi-agent system, agents with different local views, goals and partial information of the system environment may conflict with others, a global view of the entire system can be formed by communication with individual agents; thirdly, when a task is being performed by a group of agents, some certain global constraints can be satisfied; fourthly, a decision made by a single agent may have an impact on other members in the community, so agents can be interdependent; finally, agent coordination can significantly increase the system efficiency, since local knowledge, information and behaviour can be shared or observed by different agents.

There are two popular techniques for agents' coordination: blackboard and message-passing (Brenner, W. et al. 1998). A blackboard is a device which receives the agents' messages and passes them to other ones as shown in Figure 2.2a, where agents do not communicate with others directly. One of its advantages is that any new agents entering into the system can join the communication immediately based on the blackboard; also, if any existing agent quits, others would not be influenced in the system. Blackboard works well when the number of agents in a system is relatively small, because the blackboard (or mediator) agent should have necessary information of all agents in the system, including their competency and availability. If there is too much information in a blackboard, it is difficult for the blackboard agent to find the information required by agents in a reasonable time. Different from blackboard method, agents in message-passing communicate with another one directly (Figure 2.2b). In such a distributed system, each agent is able to individually carry out the communication without any degree of centralisation. For example, it is possible for A sending a message to B if the B's address is known by A. Thus, it requires more intelligence of agents.



(a) (b)

Figure 2.2 Blackboard and message-passing structure for agent communication, adjusted from (Brenner et al. 1998)

Contract net protocol (CNP) is a very common agent coordination mechanism introduced by Smith (1980). Agent negotiation is achieved by granting contracts, where agents act as managers or contractors. A manager agent first decomposes the required task into sub-tasks and sends out an announcement of the sub-task information. After receiving the announcement, the remaining agents decide whether and how they should bid for the task or sub-task based on their capability, availability and utility, etc. Then the manager agent needs to evaluate all received bids and decides which candidate agent to award the contract to. If an agent is selected, it becomes a contractor and performs the task that has been awarded. CNP is able to provide high reliability, flexibility and extensibility for different types of systems. However, when the system is complex and the number of agents is large, agents in CNP need to spend more time on dealing with messages than performing actual works. The basic contract net protocol has been extended several different versions including multistage negotiation approach, bidding-based approach and market-based mechanism.

Additionally, in order for these agent coordination methods to work effectively, the communicating specification must be formatted, that is, the communication protocol. Therefore, agent communication languages with precisely defined syntax, semantics and pragmatics are necessary. Agent Communication Language (ACL) is developed based on query languages. Knowledge Query Manipulation Language (KQML) (Finin et al. 1994) and FIPA Agent Communication Language (FIPA ACL) (FIPA 1999) are two of the famous formalised ACLs for agents to exchange information and knowledge. Based on these communication tools and methods, multi-agent systems are more suitable for implementation of coordinated and inductive decision making. For the purpose of achieving the global objectives effectively and efficiently, it requires intelligent agent coordination mechanisms.

2.4.2 Design and Implementation of Agent Applications in Manufacturing Systems

From the previous literature, it has been argued that the agent based scheduling approaches may reduce the system efficiency because the myopic view of individual agents obscures the schedule optimisation at a global level. However, according to Shen (2002), agent based approaches in distributed manufacturing scheduling have several advantages which lead to a better performance than the conventional scheduling approaches in terms of the solution efficiency. Firstly, inter-agent communication processes are performed in parallel through a number of processors, which may provide high efficiency and robustness to production scheduling problems. Therefore, the computational complexity of agent based approaches is much lower than the traditional centralised optimisation approaches. Secondly, in manufacturing systems, resource agents can be connected directly to the physical equipments they represent so as to realise the real-time dynamic rescheduling, which may lead to high reliability of the system and the resource fault tolerance. Thirdly, with the predesigned common system goal, agents make local decisions and take individual actions and as a whole the global performance is able to be enhanced, leading to cooperative scheduling. Fourthly, local changes can be handled by the corresponding agents individually. Unexpected disturbances such as machine breakdown can be automatically handled by other agents responsible for other machines with the same capabilities. Finally, the agent based structured systems can be extended from shop floor level to the supply chain level by connecting manufacturing capabilities of different manufacturing enterprises, providing approaches for the whole supply chain optimisation (Shen 2002; Badr 2010). In addition, other techniques such as meta-heuristics including GAs and simulated annealing and knowledge based techniques including ant colonies can be combined with agent based approaches for decision making or accelerating the optimising speed (Shen 2006).

Among various agent based approaches in manufacturing systems, there are two major types: functional decomposition approaches and physical decomposition approaches (Shen et al. 2006). In functional decomposition approaches, agents are applied to encapsulate models connected to functions such as process planning, production scheduling, material handling and transportation management. There are no direct relationships between agents and physical entities. Examples of this type of systems are introduced by Dornfeld et al. (1999) and Zhao et al. (2000). With functional decomposition approaches agents share state variables in across different manufacturing functions, which may lead to problems of consistency and unintended interactions. In a system using this kind of approaches, the basic agent mechanisms such as coordination mechanisms are usually predefined, which is similar to traditional approaches as discussed above. In physical decomposition approaches, on the other hand, agents are used to represent physical entities such as machines, tools, products, operations, etc. Sikora and Shaw (1997), McDonnell et al. (1999) and Wang et al. (2003) involved this type of agent based approaches as typical examples. Physical decomposition approaches usually define distinct sets of state variables, thus the interactions among agents are limited, but the large number of resource related agents may result in high computational complexity.

In the physical decomposition approaches, agent communication protocols are required, with which agents are able to reply to received information and to compete or to negotiate with each other. Each agent should have at least the knowledge about the information of its represented physical resource such as capability and availability, and some others may be able to observe the information and actions of other agents in the system. Local decisions are made by individual agents based on their own knowledge. For instance, with a Contract Net protocol, agents need to reply to an offer with requested information such as production cost, start time and process lead time (Parunak 1987). A game theory based multi-agent system requires agents to negotiate following the game rules (Guan et al. 1995; Jeong and Leon 2002). Moreover, to update agents' knowledge, learning mechanisms are required. Previous agent learning mechanisms may range from case-based reasoning to neural network and fuzzy logic-based reasoning. Detailed agent coordination and learning mechanisms in manufacturing systems will be discussed in the following section.

2.4.3 Agent-based Applications in Manufacturing Production Scheduling

Research on agent based applications in manufacturing production scheduling has a rich literature base. Based on different agent system architectures, agent based approaches for solving the problems of production scheduling can be classified into three categories: hierarchical, distributed and federated.

Some researchers adopt agents to present multi-layer hierarchical architecture systems, since a manufacturing system is always physically structured with a centre which controls, manages and makes decisions for the entire system. Agents at the lower level in the system take actions followed by the command of the higher level agents according to the system global strategies. Some others, on the contrary, use MASs to present pure distributed (decentralised) systems without a central controller. Agents with individual goals have the ability to make local decisions autonomously and the global performance can be enhanced by distributed interactions among agents. In federated structured system, agents are also able to take autonomous actions without a central controller, but there is a mediator or facilitator agent (or agents) who coordinates the behaviour of the local agents to reduce inter-agent communication overhead and to ensure the system stability and scalability.

2.4.3.1 Hierarchical Structured Approaches

In most manufacturing enterprises, there are a number of physically distributed units (i.e. workshops) positioned at different hierarchical levels of the system, and the relationships between two adjacent levels are formed as master-slave / controller-subordinate relationships. Thus, in many agent applications in manufacturing systems, there is a central unit which makes decisions for its low level subordinates based on the global strategies of the system (or sub-systems). In such hierarchical structured systems, it is not allowed to exchange information between agents at the same level, and command data is generated from the central computer and sent in a downward direction to the lower level units.

A typical example of hierarchical agent based architecture is the architecture for distributed dynamic manufacturing scheduling (ADDYMS) introduced by Butler and Ohtsubo (1992). There are two levels in the systems. The site agents at the first level are responsible for the assignment of a received task to its corresponding work cell. In the second level, the resource agents determine local resource schedules under the control of the site agents, with the consideration of operators and tools which may be shared among different work cells. Another three tier hierarchical system called distributed asynchronous scheduler (DAS) was proposed by Burke and Prosser (1994). The strategic agent attached to the top tier is responsible for the system strategic decisions and the resolution of potential conflicts between its subordinate agents. Agents in the middle tier are responsible for allocating tasks to individual resources, who act as group controllers within a manufacturing unit. The lowest tier agents in the operational level are responsible for the task execution on individual resources. All agents in the hierarchy have the property of reactiveness, and they work collaboratively to generate satisfied production schedules. Inter-agent backtracking is involved to offer a scheduling decision making method, and the earliest dispatch (ED) strategy is adopted in generating schedules.

Jacobi et al. (2007) reported a multi-agent application in real manufacturing industry. An agent based planning and scheduling system was developed in a German steel manufacturing company for enhancing the throughput within real uncertainties and changes during production. The system is structured in a three tier hierarchy consisting of a database, a schedule planner and several resource agents. The higher level database captures the updated snapshot of the real-time steelwork in the factory. The schedule planner is responsible for the correction of the system overall schedule, formulating the local scheduling rules which are passed to resource agents. The lower level distributed agents represent the local system resources, which can receive and send message from/to the planner, but cannot interact with each other directly.

CAMPS (Constraint-based Architecture for Multiagent Planning/Scheduling) was developed by Miyashita (1998). The system for solving process planning and scheduling problems is based on agent information exchange with the objective of minimising lead time tardiness, work-in-process inventory and maximising resource utilisation. There are three types of hierarchical agents in the system, namely, a manager agent, planner agents and scheduler agents. When a job order arrives at the system, the manager agent first decomposes it into a set of tasks and each planner agent is responsible for each task with the job constraints such as due dates. Tasks are then decomposed into operations by planner agents and assigned to scheduler agents. Scheduler agents administrate resources and sequence the operations assigned by planners. Each scheduler agent seeks a local schedule according to the requirements from the planner as well as optimising its own local objectives. When the schedule is found, resources are reserved for the operations so as to avoid potential conflicts of overlapping reservations with other scheduler agents. As each job has its own operational precedence constraints, coordination among agents enables the system to deal with the problem of deadlock (e.g. an operation is scheduled to be processed before its preceding one). The scheduler agent firstly tries to adjust inter-agent constraints such as shifting job positions in the job queue if the deadlock problem occurs. If it cannot be solved locally by the scheduler agent, the planner agent will relax some constrains such as deferring the job due date in order to offer the scheduler agent more flexibility in modifying its local schedule. In CAMPS, there is no interaction between scheduler agents unless the deadlock problems occur, but they only respond to the planner agent which they are assigned to. Similarly, planner agents only follow the fixed process plan distributed by the manager agents. In other words, the communication among agents in CAMPS follows the strict hierarchy, so there are not any other alternative resources for processing the job being considered, which leads to less flexibility in searching for an optimal solution.

As introduced above, meta-heuristic search approaches have been proven to be effective and practical in production scheduling. Although the optimal solutions cannot to be guaranteed, meta-heuristics has the capability of finding a sufficient solution in a reasonable computational time. In some research works agent based approaches are combined with meta-heuristics, which are applied to equip agents with more sophisticated scheduling and optimisation techniques. Passos et al. (2010) proposed a multi-agent approach combined with GA and Tabu search algorithms to solve job shop scheduling problems. Agents in the system share the same algorithm for scheduling locally under the supervision of a higher level manager. Other recent agent hierarchical structured approaches combined with meta-heuristics can be found in Ennigrou and Ghedira (2008), Cheng et al. (2009) and Nejad et al. (2011).

In the approaches with hierarchical control architecture, a consistent global view of the entire manufacturing system is provided and the models are comparatively easy to build. However, although agents are able to take actions based on knowledge and information from others, the control decisions are always made by a higher level controller within a unit. Subordinate agents rigidly follow the orders from the high level agents, and command information flow is always from top to bottom hierarchically. Therefore, the speed of response to changes in this type of architectures is slow, since any adjustment to the system requires the update of the whole model. According to Heragu (2002), the centralised hierarchical approaches have the following disadvantages: Firstly, since all the decisions are made by a single controller, any adjustment to the system requires the update of the whole model and the speed of response to changes is slow. Secondly, it is difficult for the controller design, because there is a large number of interrelationships among system resources with respect to possible failures that need to be redesigned in order to model a fault tolerant system. Thirdly, centralised hierarchical approaches have lack of flexibility and responsiveness to environmental changes since the control decisions are made without considering the real-time dynamic situations in the shop floor. As hierarchical structured approaches are found to be inefficient with increasing size of the scheduling problems and in a highly dynamic environment, they cannot satisfy the requirement of today's complex and distributed manufacturing environment.

2.4.3.2 Distributed Structured Approaches

In distributed or decentralised structured agent based approaches, there is not a global controller who makes decisions for its subordinates. Instead, local decisions are made by distributed agents based on their own knowledge and objectives. The global control decisions are reached by autonomous interactions such as negotiation and cooperation between agents representing manufacturing resources and job tasks. In this category of approaches, agents usually have the following properties: Firstly, they are able to make autonomous decisions rather than being controlled or managed by any other agents. Secondly, they can communicate directly with other agents in the system. Thirdly, they have complete or incomplete knowledge of their environment. Finally, they all have individual goals and objectives which motivate their actions.

Different agent based decentralised approaches have been proposed in the literature. A heterarchical system was proposed by Maione and Naso (2001, 2003) for solving the shop flow routing and sequencing problem. In the manufacturing system, individual parts and resources are represented by autonomous agents: Part Agents (PA) and Workstation Agents (WA). Each PA gathers the information of the corresponding part and controls the decision algorithm. It determines the mechanism to select resources (workstations), followed by the combinational rules of Shortest Operation Completion Time (SOCT), Shortest Distance from next station (SD) and Shortest Set-up with reference to the Last Part in queue (SSLP). WAs make local decisions for job sequencing within their own buffer based on a fuzzy weighted combination of dispatching rules, including First Come First Serve (FCFS), Earliest Due Date (EDD) and Shortest Set-up Time (SST). Weight parameters control the preferences of different rules in the combination, which are updated dynamically based on a GA tuning method. The integration and coordination results show that the dispatching rule based mechanism is effective in the optimisation problem of task scheduling and job sequencing in resource buffer, as well as facing disturbances and changes in the workload. As introduced in Section 2.3, a single dispatching rule or a combination of a few rules cannot produce optimal schedules which satisfies the whole processing line, particularly when instant events occur such as machine breakdown, new product types adding in, and system rescheduling. Therefore, the only method to search for the optimal schedule is to "break" the rules and to list all the possible solutions.

The distributed multi-agent scheduling system (MASS) was developed by Kouider and Bouzouia (2012) to solve static and dynamic job shop scheduling problems. There are two types of agents in the system, namely job agents and resource agents. The job agent decomposes the scheduling problem into interrelated sub-problems and distributes them to the group of resource agents. Resource agents co-operate with each other through a distributed approach of local idle time minimisation and make individual decisions of local machine schedules. Simulation results show that the approach of MASS outperforms the methods with common dispatching rules in terms of effectiveness and stability. In these two agent based distributed structured scheduling methods introduced above, the operational precedence constraints of the required jobs are not considered. Each resource agent only manages to produce optimal local schedules but the final scheduling solution for the entire job may not be optimised. In addition, the cost and time for material transporting between machines in the job shop are rarely taken into consideration either.

An agent bidding-based multi-agent system for solving integrated job shop scheduling problem was developed by Shukla et al. (2008), where agents are capable of communicating and bidding with each other. In this model, the tool cost including tool-using cost and tool-repairing cost is considered as a dynamic quantity rather than a constant traditionally. Tool-repairing cost depending on the breaking probability of the tool is predicted by data-mining agents based on the virtues of C-fuzzy decision tree. When a new job arrives at the system, the first feature of the job is announced by the component agent for bid. After a machine agent with the capability of performing the first feature is set to be winner, it becomes a leader which groups other machine agents for processing the rest features of the job. When all features are assigned to machine agents, the allocation route is sent to the optimisation agent by the leader. The optimisation agent then searches for the optimal solution among all bids received based on a hybrid Tabu-Simulated Annealing algorithm.

Li et al. (2008) proposed a dynamic job shop scheduling system based on multi-agent systems and a GA process to improve the solution performance. Distributed agents in the system include task management agent, process management agent and resource management agent. The allocation of received tasks is firstly decided by inter-agent negotiation based on contract net protocol. Another rescheduling process is performed by individual agent for selecting local optimal tasks, according to a hybrid-GA mechanism. In this GA mechanism, the generation of initial population is based on certain heuristic rules rather than randomly generated, with which the whole optimisation process can be accelerated. Since the GA based rescheduling process is only executed by individual resource agents, there is no linkage among different agents during rescheduling. Although good solutions are found after the two step scheduling, the global optimal or near optimal solution cannot be achieved.

Xiang and Lee (2008) introduced an agent coordination approach combined with ant colony intelligence (ACI) for dynamic scheduling with various products, resources and disturbances. Autonomous agents are able to make adaptive decisions to change local circumstances so as to enhance the global performance of the system. The proposed model consists of multiple product types (represented by job agents) and multiple multi-functional machines with sequence-dependent setup constraints (represented by machine agents) under the influence of dynamic disturbances. ACI algorithm is used to guide both machine agents and job agents to solve the task allocation and sequencing problems. Coordination between machine agents and job agents is performed concurrently and continuously at the same level to deal with scheduling problems including job order release, job or sub-job allocating and sequencing, job or sub-job process and machine breakdown. The proposed method is compared to a counterpart approach using dispatching rules and the results show that the new coordination approach performs better, particularly when jobs are with high tightness due dates and machine breakdown is considered.

In distributed structured approaches, the scheduling problem is usually decomposed into several sub-problems, which are handled by a set of autonomous agents representing manufacturing resources. These agents are able to communicate with each other but any of them cannot control others. Distributed structured approaches offer the flexibility and responsiveness to environmental changes for the manufacturing systems, since all control decisions are made by individual local agents based on their knowledge from experience and from interactions with others. It is easy to modify systems in terms of adding, adjusting and removing agents. Meanwhile, agent decentralised decision-making models are fault-tolerant to unforeseen situations. Any problem occurred on a single agent does not affect the whole system. Nevertheless, the pure distributed approaches are not well suited for large scale scheduling problems, since the global performance of the optimisation solutions can be unpredictable in a complex environment.

2.4.3.3 Federated Structured Approaches

To overcome the drawbacks of hierarchical and distributed structured approaches, in some literature a mediator or facilitator is involved in distributed approaches, i.e., the federated approaches. In the federated structured approaches, there is a facilitator or mediator at a higher level over distributed resource agents responsible for coordinating the communication amongst resources and collecting messages and information based on the results from agent communication.

Gu et al. (1997) presented a bidding-based approach to the integration of process planning and real time scheduling. A model involving administrative information of product specification, batch size and due dates is structured. There are four types of agents including a part agent representing a task part with the information of its required material, production due date and penalty for tardiness, a shop manager agent representing the shop floor with information regarding the hierarchical level of machines, machine agents representing machines with their own information such as capabilities, functions, schedules and tooling, and tool agents representing cutting tools with the information about their machining quality for different machines and resources. In the beginning of the bidding process, when the part arrives at the system, the part agent registers with a shop manager, and the manager announces the information of the part to machine agents at a high level of hierarchy. Machine agents then connect with the lower lever tool agents and evaluate whether they can provide a tool for performing an operation of the part as well as achieving the best tolerance specification with shortest processing time. If tool agents cannot offer a tool for the operation, and there is more than one machine offering the same process for the part, machine agents will negotiate with each other. A subcontract may be made for finishing the part collaboratively, and different machine agents submit their schedules (or collaborative schedules) to the manager agent. The bid of machine agent and its corresponding tool agent and partner agents with the lowest total production (including process time, setup time, tool change time and cost of tooling) will be chosen by the shop manager to process the part. Then it will request raw material from the manager to process the part. All parts are allocated in the same way and a contract net is then composed. With this approach the total production cost can be reduced by machine agents' bidding processes.

An agent coordination based scheduling approach was introduced by Ou-Yang and Lin (1998). Shop floor controller and cell controller are two levels of controllers in the system. Distributed cell controllers determine the production schedules for their own corresponding resources and receive bid calls for job operations announced by the shop floor controller. The shop floor controller, which plays the role of the mediator, is responsible for releasing production jobs and announcing them to cell controllers, managing the cell controllers' coordination, selecting appropriate cells for jobs based on the coordination results.

Another agent bidding based mechanism in manufacturing systems called MetaMorph was presented by Maturana and Norrie (1996, 1999). In the system, agents are classified into two types: resource agents and mediator agents. Resource agents represent the physical manufacturing devices, and mediator agents are responsible for coordinating the interactions among agents. When the bidding process starts, first the mediator agent broadcasts the tasks to all machine agents. After receiving the call, machine agents ask tool agents to bid and the ones with the best processing quality and shortest processing time will be selected. Machine agents then report their bids to the mediator with the information on production cost. The best bid with minimum production cost is selected by the mediator and the task will be allocated to the selected machine based on its earliest available time. Another type of mediator agents called data agent manager evaluates and deals with any potential conflicting schedules between features according to latest start times of the features. In this approach, machines with minimum production cost are selected by mediators, but the job features are allocated to selected machines based on their earliest available time. Therefore, the overall optimisation may still not be achieved.

An Online Hybrid Agent-based Negotiation (oHAN) approach was proposed by Wong et al. (2006). It employed a rescheduling approach to reschedule operations affected by disruptions including machine breakdown and new machine arrival. Agents in this approach comprise part agents, machine agents and a supervisor agent. It aims to minimise a part's flowtime and to maximise machine utilisation in a manufacturing environment with disturbances. This process is negotiated between a part agent with the part information including due date and a set of alternative processing routes, and a machine agent with the machine loading information. The supervisor agent in the higher level is responsible for coordinating and monitoring the rescheduling process with the supervisory controls to part and machine agents. An online hybrid contract-net negotiation protocol is involved to control the negotiation process among agents. Experimental results show that this approach is able to deal with a large-scale rescheduling problem and achieve a good global performance in a reasonable time.

A market-based multi-agent mechanism was developed by Lee et al. (2003). It aimed to solve the job-shop scheduling problem in an enterprise in terms of minimising the total production cost. There is a dynamic economic rule controlling the establishment and clearance of local markets over time. Each task agent is in charge of a single task, and each resource agent controls a single resource. Individual resource agents communicate with each other by bidding for a particular task, and the bidding auction is managed by the task agent. The whole system follows the rule of precedence cost tatonnement (P-TATO). There is a coordinator agent responsible for coordinating multiple resource agents according to the market rule in the virtual market model. Based on this approach, the production system is able to achieve flexibility, scalability and adaptability.

Another example of federated structured system is a five-level reconfigurable production planning and scheduling method presented by Bruccoleri et al. (2003, 2005). The multi-level MAS structure is shown in Figure 2.3. Different planning levels of negotiation between different groups of agents are structured hierarchically. The decisions in different levels are made by individual agents' internal reasoning strategies and agent negotiation process with the mediation of the corporate agent. The negotiation mechanism is based on the contract-net protocol and there are different designed strategies for agent's internal reasoning. It was demonstrated that an agent's internal reasoning based on fuzzy reasoning enhances the system global performance under a certain environmental condition. This proposed model is compared with a centralised alternative, and the results show that the agent-based decentralised model is relatively more efficient.

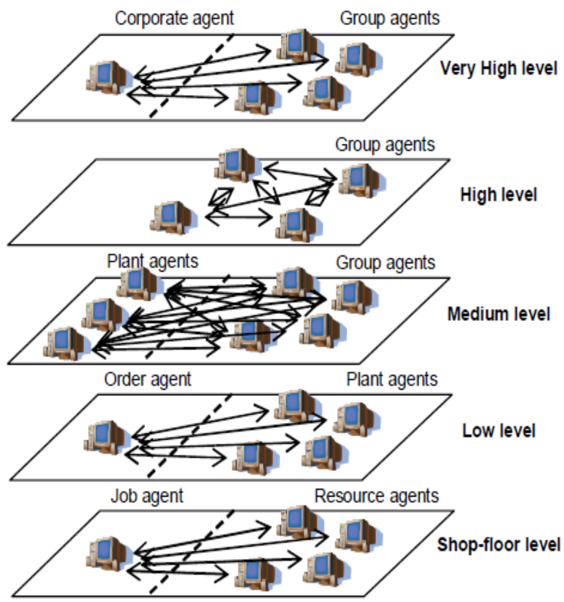


Figure 2.3 Negotiation levels for production planning and scheduling. (Bruccoleri et al. 2005)

Kang et al. (2007) developed a MAS approach in scheduling optimisation problem with ant colony optimisation (ACO) algorithm in dynamic job shop. The system consists of task agents, resource agents and management agents. Job orders from customers are assumed to be continuously arriving in the job shop and productions schedules are being generated via the communication among task and resource agents based on the ACO algorithm. Meanwhile, the management agent is responsible for the system global control, which provides the communication strategy to distributed agents, collects local agent information, selects the best solution and arranges the process of rescheduling. This approach is validated in a simulated job shop and generated production schedules were proven to be efficient and robust in terms of near optimal production lead time.

Sheng et al. (2010) proposed a hybrid architecture modelling method with multi-agent technology in agile intelligent production scheduling system. A production scheduling-oriented multi-agents architecture is constructed, where agent based hierarchical architecture and federal architecture are combined for the purpose of ensuring the system stability and scalability. Agents at different system levels are responsible for different tasks. At the higher level, static agents are created for long-term who provide certain services and guides to lower level dynamic agents, who are immediately created according to the system operation needs. Another kind of agent called cooperative support agent is involved in the hybrid architecture to generate, manage and adjust the ongoing collaboration among agents. Cooperative support agents are able to improve the system adaptability to the production objectives and the dynamic environment. A coordinator is also introduced with a higher global view and coordination capacity. Local or partial cooperative support agents can communicate and request certain support from the coordinator in some more complex collaborative processes.

A distributed multi-agent system was proposed by Ahn et al. (2011) with iterative learning control (ILC) for uncertain dynamic scheduling. There are two coordination algorithms with MAS introduced in the system. In the first algorithm a group of non-holonomic mobile agents are formatted and there is a centralised controller which calculates and memorises all the updates of local agent information. Individual agents update local schedules after receiving the feedback from the controller. Another decentralised coordination strategy is then described, where the control updates are computed locally by each agent through the iterative update process based on inter-agent autonomous communication. Numerical simulation results show that for the small size scheduling problems, the centralised algorithm outperforms the other one in terms of the global minimal production lead time. For the large size scheduling problems, the two algorithms produce similar final schedules, but the decentralised algorithm performs better as it requires much less computational time with highly flexibility and stability.

Dynamically Integrated Manufacturing Systems (DIMS) was developed by Zhang et al. (2007), which introduced a hierarchical multi-layer structured system and a heterarchical multi-agent model for planning and scheduling of integrated manufacturing systems in a dynamic environment. The whole system is modelled as a hierarchical structure but the subsystems interact heterarchically with product orders to achieve optimal process planning and scheduling. The structure constraints are relaxed gradually during agent interaction, in order to provide alternative configurations if the solution cannot be found with existing structure constraints. There is an agent coordination bidding approach to search the optimal solution which involves two adjustable parameters: the virtual price of each operation of the job and the minimal virtual profit of each resource in the shop. The general bidding process is based on contract net protocol, and the two parameters are generated randomly in the beginning by the shop agent as a coordinator. Job operations are announced to bid following the pre-designed operational precedence. In each bidding round, a resource agent is willing to bid for a job, if its profit for processing the operation is not less than its minimal virtual profit. More than one bid may be submitted by the same resource agent since the new job operation may be arranged in different position in its buffer queue. Genetic Algorithm is adopted to adjust the two set of parameters over bidding iterations. The set of "winner" resources can be found in the end, and the global optimum with minimum total cost and satisfied production time is able to be achieved. A similar currency based agent optimisation approach was proposed by (Lim et al. 2009), where a simulated annealing optimisation technique is involved to optimise the currency values over iterations.

2.5 Discussion

Based on the review of the conventional approaches and agent based approaches for manufacturing scheduling, some discussions are stated in this section.

In the conventional optimisation approaches, all computations are carried out in a central controller, which leads these approaches to inefficient, inflexible and inadaptable to satisfy the need of real-world scheduling problems (Shen 2006). Among different types of conventional approaches for scheduling, meta-heuristics have been proved to be the most effective approaches in terms of the computational time when the problem size increases exponentially. Agent based approaches have shown apparent advantages for solving scheduling problems based on their inherent flexibility (Leitao 2008). Scheduling problems with different decision-making tasks are decomposed into self-contained entities represented by agents with the abilities of autonomy and interactivity.

Based on different organisational structures, agent based approaches for solving manufacturing scheduling problems are divided into three categories, namely hierarchical structured approaches, distributed structured approaches and federated structured approaches. In hierarchical agent based approaches, the system is modelled with the similar structure of conventional centralised approaches and there is a controller who makes all the operational decisions for the system. It has been argued that these approaches can hardly achieve flexibility and responsiveness to environmental changes due to the hierarchical structure. In distributed structured approaches, local decisions are made by local agents rather than by a high level controller. Agents with high decision making autonomy are distributed in the manufacturing system and are allowed to observe their environment and to communicate and cooperate with each other, which can lead to the globally desirable schedules. Local autonomy increases the robustness and flexibility of the system. For example, a single agent's arrival or quit does not affect others. However, pure distributed structured approaches have the problems in providing globally optimised solutions. Since each individual agent makes decisions based on its own local objective, the performance of the solution for the whole system can be unpredictable with a large number of agents in a complex environment. A combination of above-mentioned two types of approaches called federated approaches shows improved performance relative to autonomous architecture for developing distributed manufacturing systems. It provides a multi-layer platform for agent interaction and making local decisions within a similar hierarchical structured system.

In decentralised and federated agent based approaches, agents are able to communicate directly or indirectly with each other. Many existing research involved agent coordination methods such as bidding mechanisms to optimise production schedules. Some approaches attempted to achieve optimal or near optimal solutions by only considering the allocation of operations to resources with best performance based on certain given criteria such as processing cost and time. Because the precedence constraints between job operations were not taken into consideration, the global optimum for the routing and sequencing solution of the entire job may not be found. For example, in the agent bidding process, choosing a bid with lower processing cost (or shorter lead time) for the preceding operation may result in higher cost (or longer lead time) bids for the following bidding operations. Therefore, in order to obtain the optimal solution of the allocation plan, alternative operation allocations have to be considered and evaluated simultaneously. The bidding process needs to be implemented iteratively and certain bidding criteria need to be adjusted to generate different bidding iterations. It would attempt to improve the performance for the current iteration compared to the previous one. Some previous researchers adopted meta-heuristic optimisation methods such as GA and Tabu search to solve this problem, where parameters were being adjusted over iterations to facilitate the search for optimal solutions.

On the other hand, since there is a buffer for each resource which stocks unprocessed jobs, when a new job enters into the resource buffer, it needs to be decided which position it should be added in the buffer queue. It is a sequencing problem as introduced above. Some previous approaches selected certain dispatching rules or adopted combinations of dispatching rules, which might lead to good results but not the optimal one. Some others attempted every possible position for the new entered job in the buffer queue. Optimal or near optimal solutions could be found but the computational time was always long especially for a complex system with heavy workload. However, since agents have the abilities of learning and making autonomous decisions, it can be expected that there are methods that agents can learn from previous experience and adjust their buffer condition based on their own knowledge and the observed information in the system.

2.6 Summary

This chapter firstly provided general definitions and characteristics of manufacturing systems and production scheduling. Some traditional methods for solving the scheduling problems were listed and discussed, including mathematical approaches, dispatching rules, heuristics and meta-heuristics. This chapter then introduced the concepts of agents and multi-agent systems which have become an increasingly popular subject in distributed system research area. Existing agent applications in production scheduling were reviewed including hierarchical structured approaches, distributed structured approaches and federated structured approaches. The approaches with centralised hierarchical structure are able to obtain optimal solutions but cannot deal with large scale problems and can hardly achieve flexibility and responsiveness to environment changes. Pure distributed structured approaches may not generate global optimised solutions as all decisions are autonomously made by local agents. Federated structured approaches provide agents a higher level mediator or facilitator to support inter-agent communication and collaboration. Some of the agent coordination based mechanisms in previous approaches do not consider the precedence constraints of job operations or resource agents' learning ability from the previous experience. As a result, in order to accelerate the optimising speed to find optimal or near optimal production schedules as well as enchaining the flexibility and responsiveness of the manufacturing systems, a new agent based approach for manufacturing production scheduling is needed.

CHAPTER 3

AGENT BASED APPROACH FOR OPTIMISATION OF MANUFACTURING PRODUCTION SCHEDULING

3.1 Introduction

This chapter proposes an agent based model for solving the optimisation problem of manufacturing production scheduling. A production job order consisting of a sequence of sub-jobs is required to be allocated to a set of resources within a manufacturing unit. For the manufacturer, it aims to complete the entire job at the lowest cost with a given time of the job order. This novel approach includes two parts: agent bidding process controlled by a Genetic Algorithm and an agent self-adjusting process for each resource agent changing the priorities of jobs in its buffer queue. The rest of this chapter is organised as follow: Section 3.2 formulates the basic problem mathematically and discusses the computational complexity. Section 3.3 including two sub-sections, respectively describes the mechanism of agent iterative bidding and the Genetic Algorithm process, and agent self-adjusting mechanism for changing priority of unprocessed jobs in buffer. A brief summary is provided in Section 3.4.

3.2 Problem Formulation

When an order of a job *j* containing *n* sub-jobs (or operations) ($O_1, O_2, \ldots, O_n$) enters into a manufacturing system, its operations will be allocated to resources by the system manager, i.e., the shop agent. Job operations need to be processed according to a certain sequence followed by the precedence constraints. In other words, the finish processing time of the preceding operation has to be earlier than the start time of the subsequent operation. Due date is assigned to the job as the latest finishing time. In the manufacturing system, there are *m* resources ($R_1, R_2, \ldots, R_m$) with different technical abilities and capacities for carrying out different types of job operations. Some of the resources have multiple capabilities which enable them to perform more than one type of operation. Meanwhile, there is a job buffer for each resource containing a list of unprocessed jobs from previous orders. It should be noted that each resource is able to perform at most one job operation at the same time. Therefore, in this problem, it should be decided which resource will carry out which job operation, as well as which resource at what time will carry out each operation. The objective of this basic problem is to find the optimum allocation of a production job containing a sequence of operations to a pool of resources in the system in order that the entire job can be finished within a given due date at the minimum production cost.

When a job operation *i* is being processed in a resource *j*, it needs a processing cost C_{pij} and a lead time T_{pij} and a corresponding setup cost C_{sij} and time T_{sij} before processing if the type of previous finished operation is not the same as the current one. Unprocessed jobs are stocked in the resource buffer, so there is a waiting cost C_{wij} and time T_{wij} if the resource is not idle when the new job arrives. Before being carried out for the current operation, the job is transported from another resource where the preceding operation is carried out, so a transporting cost C_{tij} and time T_{tij} are needed.

After a new job operation is loaded to the resource buffer, it has to be decided which position the new job is arranged in the buffer list, since there may be some existing jobs already waiting in the queue. If the new job is not put at the end of the queue, it would affect other previous jobs in terms of their waiting cost and time. In that case, there will be an additional cost C_{rij} paid by the current job for rescheduling others in the waiting list. If the new job is set to the end of the buffer queue, it would wait to be carried out until all the existing jobs completed, where C_{rij} is equal to 0.

To sum up, the whole operational cost and lead time for completing a single job operation i are:

$$T_{ij} = T_{tij} + T_{pij} + T_{sij} + T_{wij}\text{ ,} \tag{1}$$

$$C_{ij} = C_{tij} + C_{pij} + C_{sij} + C_{wij} + C_{rij} \tag{2}$$

The problem that the manufacturing system faces is to allocate each job operation to a proper resource in order to achieve optimal routing and sequencing solution, i.e., to find an allocation with minimal production cost and satisfied due date, as shown in (3),

$$Min(C = \sum_{i=1}^{n}\sum_{j=1}^{m} C_{ij} y_{ij})$$

Subjected to $\quad T = \sum_{i=1}^{n}\sum_{j=1}^{m} T_{ij} y_{ij} \leq D$

$$\sum_{j=0}^{m} y_{ij} = 1 \qquad \text{for } i \in [1, n]$$

$$y_{ij} \in (0, 1) \qquad \text{for } i \in [1, n]$$

$$TF_i < TS_{i+1} \qquad \text{for } i \in [1, n] \tag{3}$$

where C_{ij} and T_{ij} are the total cost and time for resource j processing the operation i and C and T are the overall production cost and time for carrying out the entire job. D represents the due date of the job order. y_{ij} is equal to 1 if resource j is selected to perform operation i and it is 0 if it is not selected. For each operation i, there is only one resource can be selected. TS_i and TF_i represent the starting time and the finishing time of operation I respectively. Since all operations need to be carried out following a pre-designed sequence, the staring time of an operation i+1 should be later than the finishing time of its preceding operation i.

According to the formulation of the problem stated above, the cost and time for carrying out an operation is not only dependent on which resource is selected to perform this operation, but also based on which resource is selected to carry out the preceding operation. Local decisions for allocating individual operations are correlated with each other. For example, the start time of an operation on a resource relies on when the job arrives at this resource, which is related to the finishing time of the previous operation and the time to transport the job between resources. Consequently, the change of the local decision leads to the changes of performances of following operations on resources and leads to the change of the overall performance of the entire job production. As a result, different combination of resource selection and different scheduling options for operations on resources generate alternative allocation plans with different production performances. Hence, the maximal number of possible allocation plans can be approximately represented in (4),

$$P = (\sum_{j=1}^{m} S_j)^n \tag{4}$$

where *P* is the approximate maximal number of all possible allocation plans. S_j is the scheduling options assumed to be processed by the resource *j* for an operation, which depends on how many unprocessed jobs are in the resource buffer. *n* and m represent the total number of operations and resources respectively. This equation shows that the number of possible solutions for allocation plans grows exponentially with the increase of the number of job operations. This problem belongs to NP-hard (Non-deterministic polynomial-time hard) problems and it is not practicable to list all possible allocation plans and select the best one as the problem becomes more and more complex with increasingly larger number of operations. In addition, the constraint of the operational sequence, i.e., following operations have to be performed after their preceding operations finished, has not been considered in this equation, which makes this problem more complex. Therefore, it is necessary to find an efficient method to accelerate the searching speed and reduce the computational complexity.

3.3 Agent Bidding Mechanism

In this study, we propose an optimisation approach consisting of two parts for solving the job-shop scheduling problem. The first part is an iterative agent based bidding process with a weight function to search for optimal allocation and there is a Genetic Algorithm process for adjusting weight parameters. The second part is an agent self-adjusting mechanism for individual resource agent to change its own resource buffer conditions in terms of the operational priorities of the unprocessed job.

3.3.1 Agent Based Iterative Bidding Mechanism

3.3.1.1 Agent Based Bidding Process

As introduced above, a job of a customer order includes a job manager agent and n operations (i = 1, 2, …, n). The manufacturing system consists of a shop agent with m resource agents (j = 1, 2, …, m). When the order enters into the system, the job manager agent decomposes the order into sub-jobs, i.e. operations. Then it announces the operations to the shop agent which is responsible for managing the bidding operations one by one, according to the pre-designed operational sequence. After receiving the call from the job agent, the shop agent then initialises the values of weight parameter set $\{g_i\}$ randomly. g_i is introduced as cost-time preference, which determines the importance of total processing cost and lead time respectively in the bidding round for operation i. All resource agents with technical capability calculate their total processing cost C_{ij} and lead time T_{ij} according to (1) and (2) . Note that in the beginning of each iteration, all new entered jobs are expected to be arranged at the end of the buffer queue for all resources, i.e., based on the First-In-First-Out rule, where the rescheduling cost C_{rij} is 0. After receiving all bids from resources, the shop agent finds the bid with the maximal value of production cost C_{max} and the bid with maximal value of lead time T_{max}. The value of the weight function f_{ij} for resource j performing operation i can be acquired according to (5).

$$f_{ij} = g_i \frac{C_{ij}}{C_{\max}} + (1 - g_i)\frac{T_{ij}}{T_{\max}} \qquad (0 < g_i < 1) \tag{5}$$

The bid with the minimal f_{ij} value is selected as the winner in this bidding round, and the corresponding resource *j* wins the chance to carry out the operation *i*. This process repeats *n* rounds until all operations are allocated to the winner resources, and a potential allocation route is then obtained. The total production cost and lead time for the entire job can be calculated as shown in (6),

$$C_{job} = \sum_{i=1}^{n} C_i^{win}\ , \quad T_{job} = \sum_{i=1}^{n} T_i^{win} \tag{6}$$

where C_i^{win} and T_i^{win} represent the production cost and lead time of the winner bid j_{win} for processing operation *i*. C_{job} and T_{job} are the total cost and time for completing the entire job with the selected winner resources. Each allocation route needs to be examined after it is generated. The processing time of the entire job is compared with the order due time. If the expected delivery due date is not satisfied, or if the total processing cost is not low enough, the optimal solution is not achieved. Thus, the set of weight parameters $\{g_i\}$ needs to be adjusted to generate a new route. As shown in (5), the bid with higher cost and shorter time tends to be selected by the shop agent with lower value of g_i, which leads to the shorter lead time in the entire allocation plan. On the other side, the higher value of g_i results in lower production cost since the bid with lower cost and longer time is more plausible to be selected as a winner. The iterative bidding process stops when the minimum or low enough cost is achieved. The steps of the bidding process are shown in Figure 3.1.

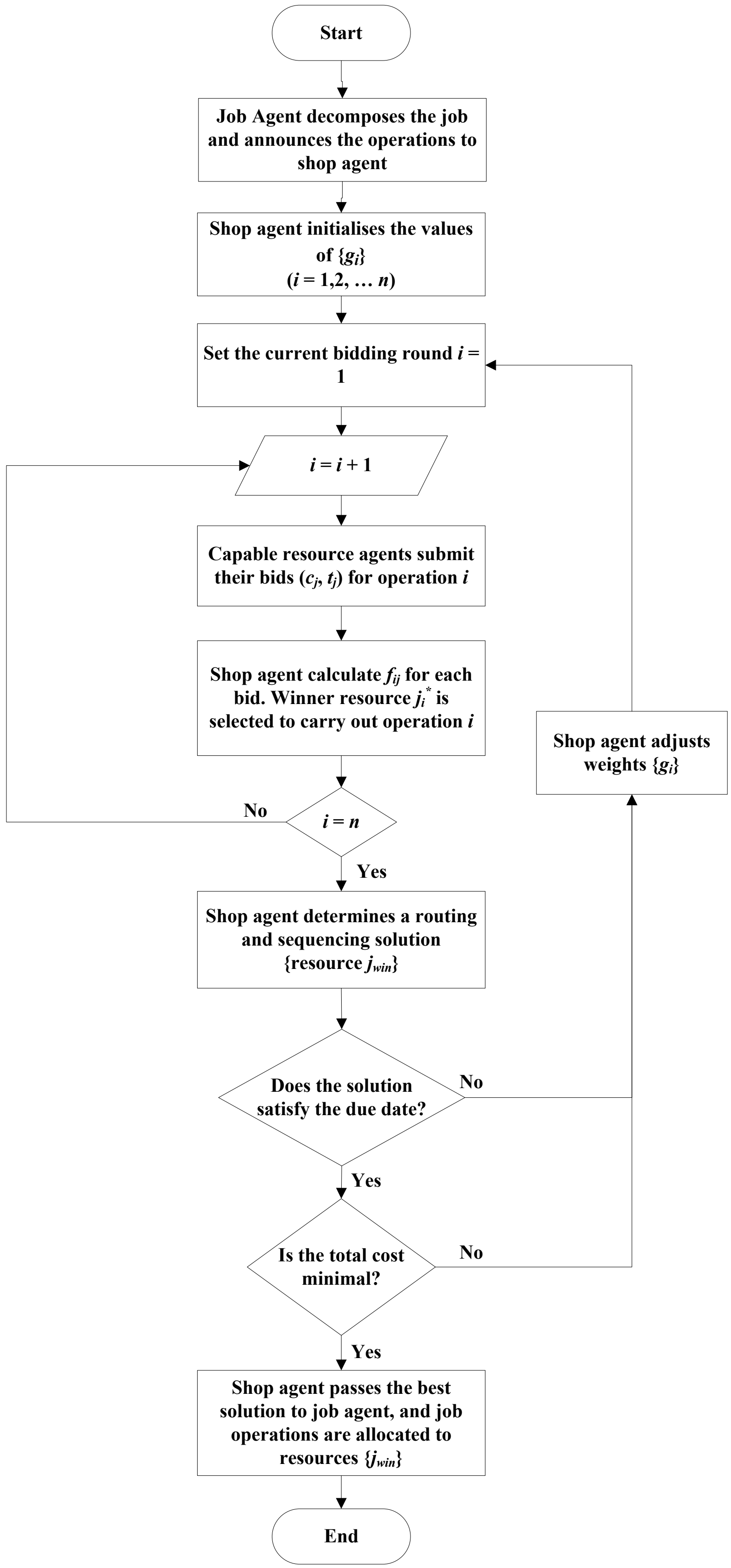


Figure 3.1 Agent Iterative Bidding Process

3.3.1.2 Genetic Algorithm Process

As introduced above, for the purpose of achieving the optimal allocation route, an effective tuning method for adjusting weight parameters is needed. In this approach, a Genetic Algorithm based tuning method is employed. In this method, a GA chromosome represents a set of weight parameters $\{g_i\}$ and each weight parameter g_i represents each gene within the chromosome. The fitness function for evaluating each chromosome is the total production cost of the allocation plan for the entire job with the corresponding weight parameters in operation bidding rounds. The optimal allocation plan in terms of the minimal production cost and satisfied due date can be obtained after the optimal values of weight parameters are found. Detailed GA process is described as the following steps:

1. After receiving the call from job agent, the shop agent randomly initialises the population of chromosomes through a random process. In each initial chromosome, the weight parameters $\{g_i\}$ are randomly generated over the range of (0, 1), i.e., chromosome $CM_x = \{g_1, g_2\ldots, g_n\}$.

2. Once a chromosome is created, it is examined whether it is eligible in terms of the job due date constraints. The weight parameters in the new chromosome are used to coordinate the bidding process so as to investigate whether the created chromosome is able to result in an allocation plan satisfying the due date of the entire job. The chromosome will be placed into the population pool if the obtained allocation plan satisfies the due date of the job. Otherwise, it will be discarded and another initial chromosome is generated and evaluated again.

3. These two steps repeat until X chromosomes are created, where X is the predefined population size. With the set of weight parameters $\{g_{ix}\}$ in chromosome x, the winner resources for all job operations can be determined according to (5). Consequently, the total cost for processing the entire production job with selected performing resources can be calculated, which represents the fitness function for the corresponding chromosome. Note that a lower total cost for the entire job represents a higher fitness value and vice versa.

4. Two chromosomes are selected as parents based on a Roulette Wheel selection process. The possibility of each chromosome being selected is proportional to its fitness. The fitter chromosome obtains a greater chance to be selected than a weaker one. An example of Roulette Wheel selection with five chromosomes is shown in Figure 3.2. When a parent is needed to be determined, firstly a random number within the range of (0, 1) is created as a threshold. If the number is less than 0.26, CM_1 is selected; if it is between 0.26 and (0.26+0.21), CM_2 is selected, etc.

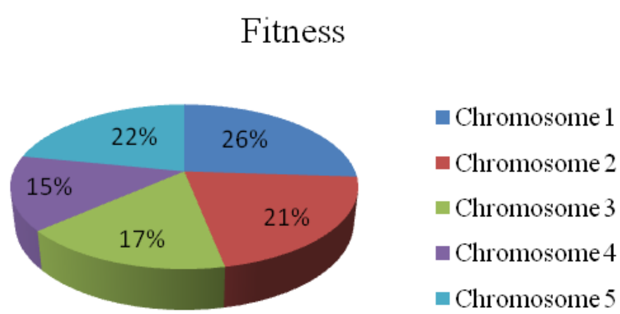


Figure 3.2 Example of Roulette Wheel Selection

5. Parent chromosomes produce offsprings based on crossover operators including point-to-point, single cutoff point or dual cutoff points crossover. If the generated two offspring chromosomes are different from their parents, they are accepted and placed in the next generation pool. Otherwise, the crossover process repeats until two new offsprings are created. Detailed mechanisms of three crossover operator types are shown below.

Point-to-Point Crossover:

$Parent\ 1 = \{C_{11}\ \boxed{C_{12}}\ C_{13}\ C_{14}\}$ $\qquad$ $Offspring\ 1 = \{C_{11}\ C_{22}\ C_{13}\ C_{14}\}$

$Parent\ 2 = \{C_{21}\ \boxed{C_{22}}\ C_{23}\ C_{24}\}$ $\qquad$ $Offspring\ 2 = \{C_{21}\ C_{12}\ C_{23}\ C_{24}\}$

Single Cutoff Crossover:

$Parent\ 1 = \{C_{11}\ C_{12} \mid C_{13}\ C_{14}\}$ $\qquad$ $Offspring\ 1 = \{C_{11}\ C_{12}\ C_{23}\ C_{24}\}$

$Parent\ 2 = \{C_{21}\ C_{22} \mid C_{23}\ C_{24}\}$ $\qquad$ $Offspring\ 2 = \{C_{21}\ C_{22}\ C_{13}\ C_{14}\}$

Dual Cutoff Crossover:

$$\text{Parent 1} = \{C_{11} \mid C_{12}\ C_{13} \mid C_{14}\} \qquad \text{Offspring 1} = \{C_{11}\ C_{22}\ C_{23}\ C_{14}\}$$

$$\text{Parent 2} = \{C_{21} \mid C_{22}\ C_{23} \mid C_{24}\} \qquad \text{Offspring 2} = \{C_{21}\ C_{12}\ C_{13}\ C_{24}\}$$

6. After crossover, new generated offspring chromosomes are mutated based on a mutation probability β ($0 < \beta < 1$). A random number α is generated and compared to β to decide whether each gene within the chromosome is mutated, i.e., it is mutated if $\alpha < \beta$. When a gene is determined to be mutated, its value is modified followed by the mutation operator μ as shown in (6):

$$g'_{ix} = (1 + \mu)\, g_{ix} \qquad (6)$$

where g'_{ix} is the new value of the selected gene after mutation, which replaces the original one. μ is a randomly generated over a predefined range of $[\mu^l, \mu^h]$. The purpose of the mutation process is to avoid local optimisation.

7. The eligibility of the created offspring chromosome is needed to be examined as aforementioned in Step 2. If the lead time of the new potential solution is not satisfied, this chromosome is discarded and a new offspring is created. (Go back to Step 4.)

8. New fitness values of the offspring chromosomes can be obtained by running the bidding process again with the new weight parameters. Fitness values of parent and offspring chromosomes are compared. A parent chromosome is replaced by an offspring if the offspring chromosome's fitness overcomes the parent's. Otherwise, the parent chromosome is directly delivered into the new population pool.

9. Go back to Step 4. The processes of selection, crossover and mutation repeat until a new population with X feasible chromosomes is generated. After a new population is created, fitness values are tested and compared with the previous generation.

10. New generations are created continuously. The iterative process stops after a certain number of generations, or until the cost of a planning route is no longer

lowered, i.e., the fitness of the chromosome is no longer improved.

This GA process provides an effective way to dynamically tune the weight parameters in order that the allocation plan is optimised over bidding iterations. Note that GA is applied here for optimising parameters in the bidding process in order to achieve balanced and optimised cost-time weight pairs. There is no centralised high-level control for optimising the routing and sequencing problem directly so the process of scheduling is flexible with resource change in the production system.

3.3.2 Agent Self-adjusting Mechanism

As stated in the last section, in the first bidding iteration for all job operations, when a new job is loaded to a resource buffer, it is arranged to the end of the buffer queue, so the rescheduling cost for the resource agent is 0. In this context, the priorities of buffer jobs are fixed based on this First-in-first-out rule, with which the objective, i.e., the minimal total cost for performing an entire job, cannot be achieved. In that case, alternative arrangements of the new job in the buffer need to be considered. As introduced in Chapter 2, some previous research such as DIMS (Zhang et al. 2007) also considered rescheduling in this routing and sequencing problem, where resource agents calculate all the possible positions for the new job in resource buffers. In the agent self-adjusting mechanism in this study, agents with more autonomy are capable of self-adjusting the priorities of the existing jobs in their buffers. This agent self-adjustment process is performed according to the sequence of the job operations. After each bidding iteration for all operations is finished, each eligible resource agent then rearranges its buffer condition and constructs a new bid for a particular operation. This process aims to reduce the computational time and enhance the efficiency of the mechanism.

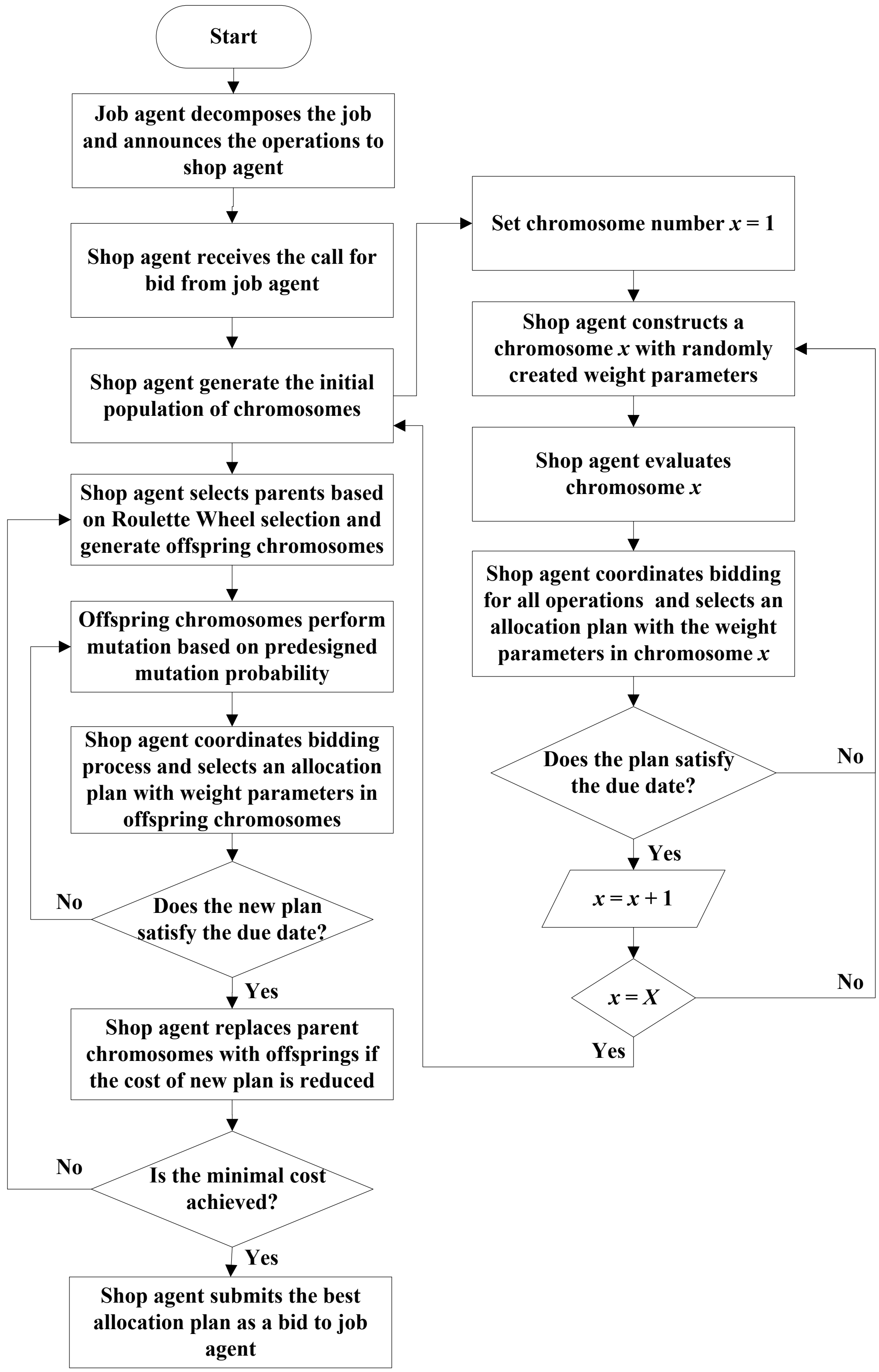


Figure 3.3 The Genetic Algorithm Process

For a particular operation *i*, after the winner bid with the cost and time pair () is selected by the shop agent in the previous bidding iteration, it is announced to all technical eligible resource agents as a *Reserve Price*. For the purpose of seeking a new potential bid, each resource agent then rearranges its buffer in terms of changing the priority of the new job in buffer queue. The rearrangement may affect other unprocessed jobs since they may wait longer time for being performed, and therefore their waiting cost and time may increase. For instance, as shown in Figure 3.4, J1, J2 and J3 are three unprocessed jobs in the buffer waiting list before the arrival of the new job, and their sequence cannot be changed. The new job was set to the end of the queue, i.e., position A, in the first bidding iteration. When the agent rearranges its buffer, there are three new possible positions which are B, C and D to add the new job in. If it is set to position C, the unprocessed jobs J2 and J3 will have to wait to be performed until the new job finished. This additional waiting cost of the affected jobs should be paid by the new job, which is C_{rij} as introduced above.

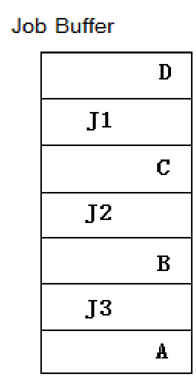


Figure 3.4 Example of job buffer list

Therefore, the rescheduling cost for the current job is the sum of the additional waiting costs of all affected unprocessed jobs. As described above, if the new job is placed at a position before that of some jobs existing in the job buffer, the waiting time of the affected jobs will be increased and the finishing time of these jobs will probably be postponed as well. In this context, if the due dates of the affected jobs can still be satisfied, this rescheduling can then be put forward for the resource agent to compose a new bid.

With received *Reserve Price* for the last bidding iteration, the resource agent *j* calculates its new total cost for performing the operation *i* after rescheduling and its own revenue per time unit, i.e., the utility of the resource agent U^j for rescheduling, followed by (7),

(7)

where C'_{ij} and T'_{ij} are the new production cost and time for resource *j* performing operation *i* after buffer condition adjustment followed by (1) and (2). A is a positive arbitrary number in order to ensure the resource agent will gain profit even if the cost of its new bid for operation *i* equals to the cost of the *Reserve Price*.

The attempts of new job rearrangement follow the order from the lowest to the highest priority position in the resource buffer list. For example, in Figure 3.4, the order of the rearrangement is from position A to D. Once the resource agent's utility is greater than *A*, the attempt stops and a new bid is submitted by resource agent *j* to the shop agent. If the agent's utility cannot be increased with any of the attempts with all the possible rearrangement positions, the agent will not submit a new bid. After all of the new bids (C'_{ij}, T'_{ij}) from resource agents are submitted to the shop agent, a new set of winner resources is then selected according to (5) and (6). This process repeats between two bidding iterations, i.e., GA generations, until no more bid from resource agents is received for all operations. Next, a new bidding iteration begins with different values of weight parameters $\{g_i\}$. The detailed steps of self-adjusting process for each resource agent are shown in Figure 3.5, where *b* represents different available position in resource's buffer list. *J* is the total number of the existing unprocessed jobs in the buffer, so there is *J*+1 possible positions in which the new job can be added.

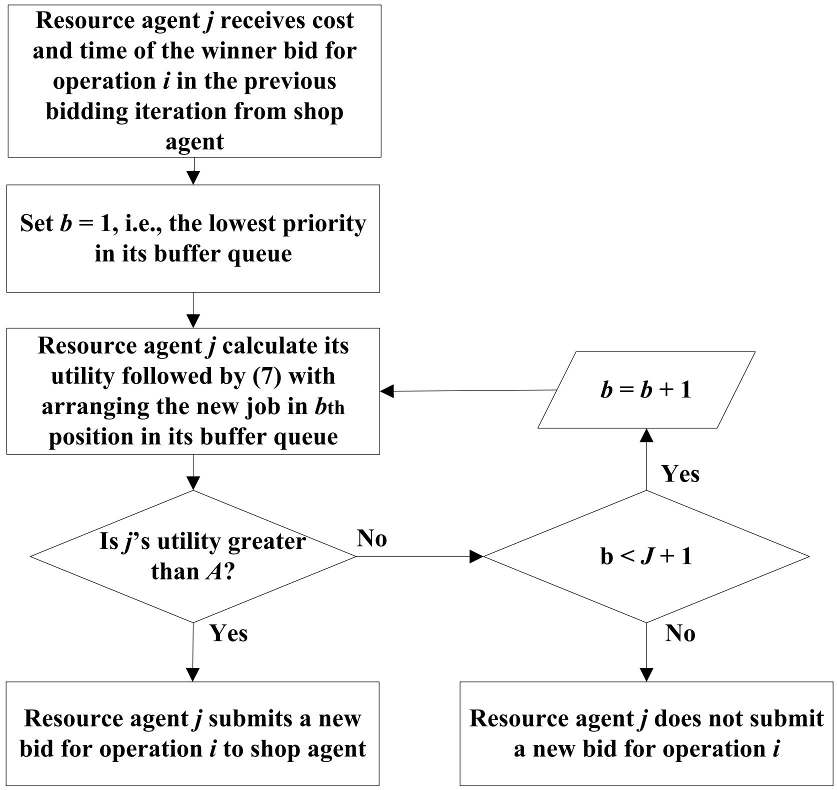


Figure 3.5 Resource agent buffer self-adjusting process

3.4 Summary

This chapter firstly formulated the basic optimisation problem of manufacturing production scheduling. It can be simplified as job operations allocation problem for achieving minimal total production cost with the constraints of lead time, resource capability and fixed operating sequence of job features . It was stated that the basic problem belongs to NP-hard problems that it is unrealistic to solve them by considering all possible solutions through enumeration. This chapter then proposed an agent bidding based approach for solving this problem. An iterative agent bidding based mechanism based on a Genetic Algorithm was introduced. Bids for performing job operations from resource agents are submitted to the shop agent and a winner resource is selected for each operation according to the weight function. The GA process provides an effective way to iteratively adjust the values of weight parameters. Through the processes of crossover and mutation, the solutions of allocation plans can be gradually optimised. An agent self-adjustment mechanism was also introduced in this chapter. After receiving the winner bids from the previous bidding iteration, agents are able to adjust the priority of the unprocessed jobs in their buffer and decide whether to submit a new bid for the announced operation based on their own utility function. With agent iterative bidding process and buffer self-adjusting process, the minimal total production cost can be achieved with satisfied lead time over iterations.

CHAPTER 4

MULTI-AGENT MODELLING ARCHITECTURE

4.1 Introduction

An agent based iterative bidding approach for optimising manufacturing production scheduling was introduced in the last chapter. In order to implement as well as investigate the effectiveness of the proposed approach, a model based on multi-agent system is built. This model belongs to the type of federated structured models, in which there is a facilitator (the shop agent) responsible for coordinating the communication among distributed machine agents. This chapter provides the detail of the model. Section 4.2 introduces the overall structure and function of the model. Different agent types and agent architectures are illustrated in Section 4.3. Section 4.4 provides a brief summary of this chapter.

4.2 Modelling for Proposed Approach Based on Agents

As introduced in Chapter 2, there are three categories of agent based approaches for scheduling based on different system structure, namely hierarchical, distributed and federated approaches. In this study, a federated multi-agent based model is built for the optimisation problem of manufacturing production scheduling. This process is performed by communication, negotiation and collaboration among different types of agent. There are four types of agents, namely a job order agent, a shop agent, several machine agents and a material handling agent. Based on the iterative bidding mechanism introduced in Chapter 3, this model is expected to be able to perform the functions described as follows:

- To search for a routing and sequencing solution for processing a required job in the manufacturing system with the lowest production cost and satisfied all the constraints including the delivery due dates of product orders.

- To allocate and schedule required job operations to certain system resourceses, i.e., machines, based on the selected allocation plans.

- To optimise the utilisation of individual manufacturing resources.

Figure 4.1 depicts the overall architecture of the model. When a job agent receives a customer order, it interprets and decomposes the order and then passes the job to shop agent. The shop agent and several machine agents work collaboratively to generate an optimal or near-optimal schedule plan while the material handling agent provides the information of material handling. The production is carried out based on the final selected plan which is returned to the job agent by the shop agent. More detailed agent information will be discussed in the following section.

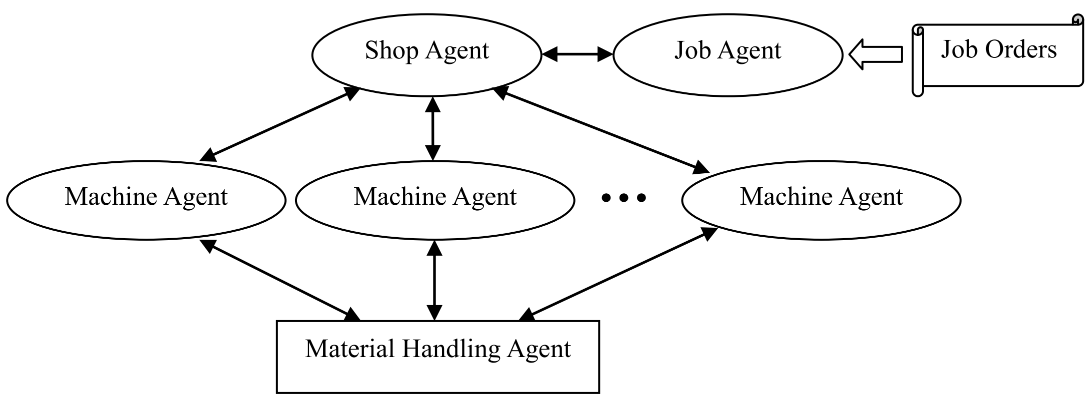


Figure 4.1 The model architecture of the multi-agent system

4.3 Agent Types and Architectures

In this part, agents in the proposed model are described in detail. Firstly, four types of agent: job agent, shop agent, machine agent and material handling agent are introduced in Section 4.3.1. The generic agent architecture is then depicted in 4.3.2.

4.3.1 Agent Types

4.3.1.1 Job Agent

For manufacturers, each customer order can be divided into several types of products, each of which consists of a number of components and each component composes of a few operations (features). In this work, since a model of a pure distributed system is aimed to be built, the job orders are then decomposed into single-layer operations.

The job agent is responsible for receiving job orders from customers. It then interprets the order and decomposes it into sub-jobs (operations). Each job agent holds the information of detailed data of its representative job order including order arriving time, due date, batch quantity and job operational sequence. The operation list of the job is announced to the shop agent who is requested to provide a satisfied optimal schedule plan for the entire job. After an optimal or near optimal schedule plan is generated by the interaction among the shop agent and machine agents, it is then passed back to the job agent. Each job operation will be awarded to the outstanding machine agent according to the selected plan to be carried out.

4.3.1.2 Shop Agent

The shop agent acts as a coordinator in the manufacturing system. It holds the rules for the process of bidding job operations among machine agents. In other words, the mechanism for selecting outstanding machines for carrying out the series of job operations is decided by the shop agent. According to the pre-designed operational sequence of the required job, the shop agent collects bids from machine agents and selects the winner ones based on the proposed algorithm stated in Chapter 3. After an optimal or near optimal schedule plan is obtained, it will then be passed to the job agent.

It should be noted that the outstanding machine selection is not controlled but only coordinated by the shop agent. Bids are submitted and adjusted by individual machine agents locally based on the change of their own utilities over bidding iterations.

4.3.1.3 Machine Agent

Each machine on the shop floor is represented by a machine agent (or resource agent) which holds the following information of its representative machine:

- Machine number
- Machine capability (operation type (types) which can be processed)
- Machine location
- Setup time for each processing type
- Setup cost for each processing type
- Machining time for each processing type
- Machining cost for each processing type
- Holding cost per time unit
- Machine buffer capacity
- Information of jobs storing in machine buffer

After receiving the announcement of bidding operation from shop agent, the machine agent firstly checks its eligibility, i.e., whether it is able to carry out this operation, and its buffer capacity. If it is capable to process that operation and simultaneously there is available space for new job in its buffer, it will then generate a bid. Since machines are allocated in a distributed manufacturing unit, the distance between two machines can be calculated. The information on material transporting cost and time per unit of distance is provided to machine agents by the material handling agent for generating their bids. On the other hand, previous jobs which are waiting in the machine buffer can be rescheduled locally by each machine agent, as long as the due dates of unprocessed jobs are not violated. With all the consideration of job loading, machine setup, processing, material transporting and buffer jobs holding, bids for the current operation can be worked out by individual machine agents and submitted to the shop agent.

4.3.1.4 Material Handling Agent

In this work, automated guided vehicle (AGV) system is considered as the material handling method to handle material flow in the manufacturing unit. It is assumed that each machine is accessible by the AGV system. The material handling agent is in charge of the material handling information including material transporting cost and time per unit of distance, which may be provided to machine agents when they generate a bid for a particular job operation.

4.3.2 Agent Architectures

As described above, there are four types of agents in the proposed model. In this multi-agent system, agents should firstly be able to communicate with each other. Hence, there must be a standard format for agent architectures, as messages can be recognised and interpreted among different agents. Secondly, there needs to be a controller for agents dealing with received messages and making intelligent decisions. Thirdly, a database is also required for agents to store their inherent data information. Finally, there is a knowledge based which contains agents' objectives, rule and algorithms for making decisions and negotiating with others.

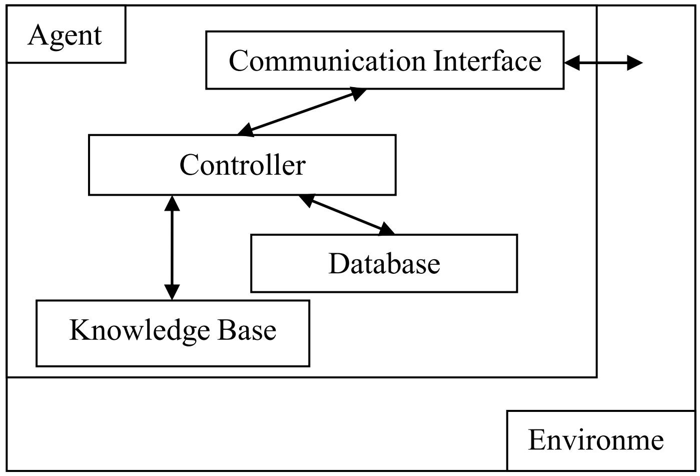


Figure 4.2 Agent architecture

Performative	Sender	Receiver	Content
Tell	Job Agent	Shop Agent	job name; operation sequence; delivery due time; max emission
Ask	Shop Agent	Resource Agent	current operation type; cost ?; time ?; emission ?
Ask	Resource Agent	Material Handling Agent	trans cost per time unit?
Reply	Material Handling Agent	Resource Agent	trans cost per time unit: $C_{tr/t}$
Reply	Resource Agent	Shop Agent	cost: C_{ij}; time: T_{ij}; emission: E_{ij}
Tell	Shop Agent	Job Agent	total cost: C; total time: T; total emission: E

Table 4.1 Examples of agent communication messages

4.3.2.1 Communication Interface

Messages are sent and received by the agent communication interface. It provides a channel for agents to communicate with each other. A message can be either a control message in which specific actions are requested to be taken, or a data message where specific information is required to be transferred. When a message is received by the communication interface, it is stored in the message input queue, and once a message is required to be sent, it is put in the message output waiting queue. These messages are handled by the agent based on the First-In-First-Out (FIFO) rule. In this study, Knowledge Query Manipulation Language KQML (Finin et al. 1994) is adopted as agent communication protocol. KQML is selected due to its simplicity and extensibility. Examples of KQML messages for agent communication are shown in Table 4.1.

4.3.2.2 Knowledge Base

The knowledge base contains different types of knowledge of the agent. Firstly, it stores the local objectives of different agents. For instance, the major task of a job agent is to fulfill the required job order with minimum cost and satisfied due date. A machine agent's objective is to maximise its own utility, i.e., to gain as much profit as possible by submitting new bids according to (7). Secondly, the knowledge base provides the rules, algorithms or procedures for the agent to solve problems. For machine agents, the production cost and time for carrying out a single job operation is formulated in the knowledge base. Meanwhile, the rules for a machine agent to adjust its buffer queue are stored in this place as well. The knowledge base of the shop agent contains the algorithm for evaluating submitted bids and the GA based tuning algorithms for updating weight parameters iteratively. Thirdly, the knowledge base of an agent also describes the rules and protocols for inter-agent negotiation and coordination. All the knowledge stored in the knowledge base can be called from the agent's controller in order to make a decision or to take an action.

4.3.2.3 Controller

The controller provides the platform where procedures and rules are carried out. The agent behavior of decisions making and actions taking are dependent on this module. It obtains rules, algorithms and mechanisms from the Knowledge Base, agent inner data from database and the external information from the Communication Interface.

For machine agents, the controller is responsible for updating machine data, calculating individual bids for bidding operations, evaluating the local utility, rescheduling the jobs in the buffer queue and composing messages which will be sent to other agents. For the shop agent, the controller makes announcement for bidding operations, initialises and adjusts the values of weight parameters, evaluates all the bids received from machine agents, selects the winner bid for each bidding round, calculate the total production cost and time for the required job and updates the schedule plans. For the job agent, its controller decomposes the job into sub-jobs, retrieves the job data from its database and passes to the shop agent, receives the result of optimised schedule plans from the shop agent and awards the outstanding machines. The material handling agent's controller is responsible for updating the AGV schedules and producing transporting cost and time function to the machine agents when required.

4.3.2.4 Database

The database consists of the agent's real-time and static information and data. It is the module for the agent to contain its constant data, to memorise its previous observations and to store its knowledge received via sensory input or from the controller. For instance, a machine agent's database keeps the static data such as machine capability, location, processing cost and time, etc, and real-time information with respect to machine current status and buffer condition. The database of a job agent contains the types of job operations, process sequence, due date, batch quantity, etc.

4.4 Summary

The multi-agent based model was introduced in this chapter, in order to implement the agent bidding based iterative mechanism aforementioned above. The system structure and functions of the model was discussed at the beginning. This chapter also described four types of agents involved in the system, namely job agent, shop agent, machine agent and material handling agent, and explained the architecture and function of each individual agent type. The multi-agent system (MAS) is able to provide a means to generate production schedules without centralised control, where distributed agents make autonomous decisions. The search for the optimised production schedule with lowest possible production cost and satisfied product due dates is performed by agent interaction and coordination.

CHAPTER 5

MANUFACTURING EMISSIONS

5.1 Introduction

In recent decades, the use of energy in all organisations has become a key issue, especially in manufacturing enterprises. The amount of carbon emissions released from manufacturing processes of a company is under the pressure of government and also affects the public opinion. Meanwhile, one of the most important issues that affect the quantity of manufacturing emissions is the energy consumption during production. Thus, decreasing energy consumed in production, to some extent, leads to the reduction of emission released. Moreover, emission reduction helps manufacturing enterprises save cost of installing and operating end of pipe pollution control devices and simultaneously increase their productivities and efficiency, since producing less waste reflects a better utilisation of inputs resulting in lower material and waste disposal costs (Schmidheiny, 1992). From this point of view, with the objective to increase the production effectiveness and flexibility, the advanced manufacturers also face the problem of how to reduce the emission released during production processes.

This chapter will firstly deliver an introduction of manufacturing emissions and discuss some existing works to reduce the amount of emission released during production. Background of manufacturing emissions and energy consumption are introduced in Section 5.2, and related policies of different countries and international organisations are also stated. Section 5.3 provides two main methods for measuring manufacturing emissions generated from manufacturing production activities. Section 5.4 presents some existing approaches for reducing manufacturing emissions. A discussion on previous research is given in Section 5.5. Section 5.6 provides two methods for involving the emission factor into the basic agent iterative bidding mechanism as introduced in the last chapter, which consider the emission as a constraint and as a second objective, respectively. A brief summary is provided in the last section.

5.2 Background of Manufacturing Energy Consumption and Emission

5.2.1 Manufacturing Energy Consumption and Carbon Emissions

With the rapid development of the worldwide industry and the increasing world population, the global warming has recently become a worldwide focus of attention. CO2 emissions have been reported as the primary negative impact on the greenhouse effects, causing approximately 55% of the global warming (Davis and Caldeira 2010). Between 2000 and 2010, the worldwide carbon emissions grew at the rate of 3.4% per year while the increase rate of previous decade was 2.4% (Mirzaesmaeeli 2010).

Manufacturing is stated as the largest source of energy related CO2 emissions in the world industrial sector, which also includes agriculture, forestry, fisheries, mining, and construction. Activities generating carbon emissions in manufacturing production domain include: burning fossil fuels in electricity generation and in transportation; releasing industrial process emissions such as in the field of chemical manufacturing, cement or steel making and machinery; and waste treatment (Schipper 2006). In different manufacturing sector, the amount of generated emissions is various. For example, Figure 5.1 depicts the amount of CO2 emissions by different sub-groups in UK manufacturing sectors in 2008, and the total amount reached were equivalent to 140 million tons of carbon (Department of Environment UK 2012). On the other hand, as mentioned before, saving production cost is one of the most important issues that manufacturers consider. Therefore, balancing the economic and environmental preferences has become a worldwide challenge in supporting the long-term sustainable development.

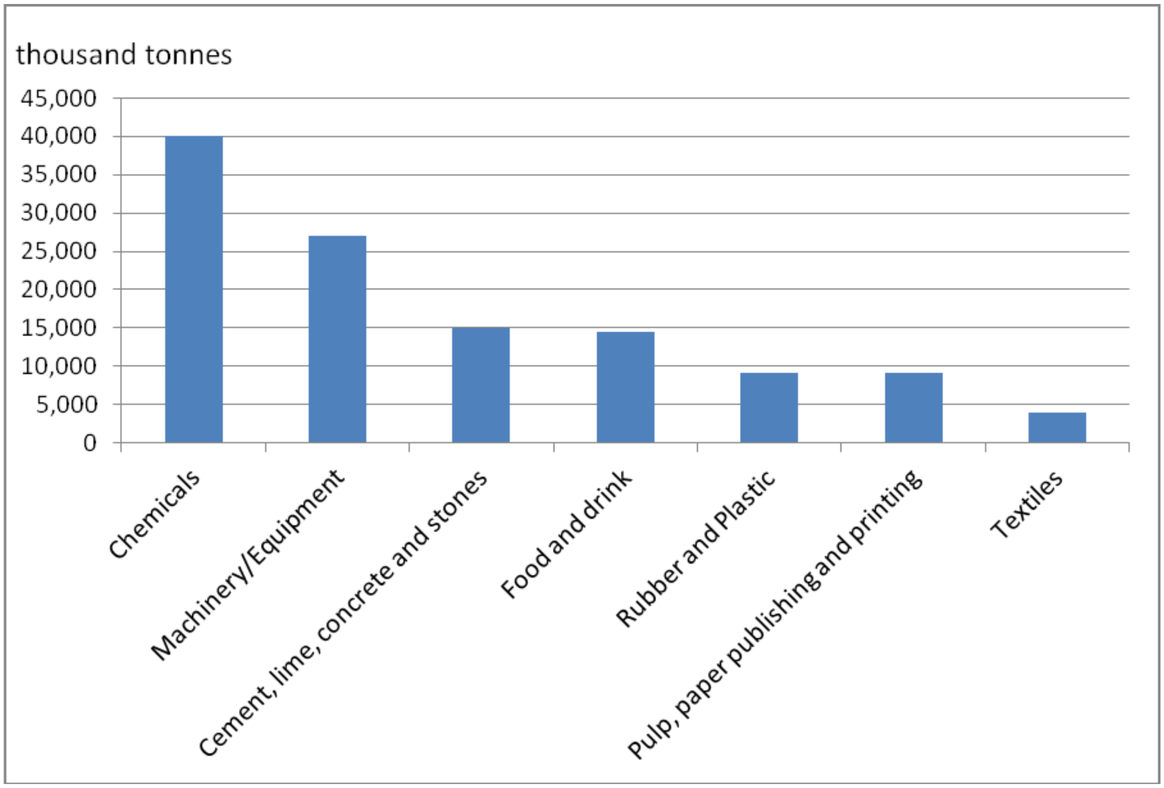


Figure 5.1 CO_2 emissions by different sub-groups in UK manufacturing sectors, (Source: Office for National Statistics, 2008).

5.2.2 Policies Related to Manufacturing Emission

With the increasing amount of carbon dioxide and other greenhouse gas emissions emitted in the earth's atmosphere, many countries and international organisations have enacted laws and policies on energy saving and emission reductions. As stated in the report by Commission of the European Communities (2007), the European Union enacted the Strategic Energy Technology Plan (SET-Plan) to involve new energy innovations and decrease the greenhouse gases. Based on the collected data from 21 EU countries in 2006, it aimed to reduce greenhouse gas emissions by 20% and realise 20% of the renewable energy sources in the EU by 2020 from the value of 2006. Legal foundations has been laid by UK government for an emissions cut of 60% by 2050 (Bows and Anderson, 2006). A new and specific Strategy was enacted by Japanese government in 2006 with the following objectives which are expected to be achieved by 2030: improving 30% of the energy efficiency; reducing the oil dependent sources in transportation to 80%;

reducing the oil dependent sources over primary energy supply to 40%; increasing 10% of the ratio of nuclear power to all power production; and increasing 10% the oil volume ratio in exploration and development by industrial companies (Itoh, 2007). U.S. Congress has passed the draft legislation of Title VII of the American Clean Energy and Security Act of 2009 for reducing the emissions of greenhouse gasses, which starts in 2012 by limiting emissions to 97% of their 2005 levels, 80% in 2020, and 58% in 2030 (Bassi and Yudken 2011). By 2011, 195 countries and organisations in the world had become members of the United Nations Framework Convention on Climate Change (UNFCCC), which had established international legal binding obligations for developed countries to reduce greenhouse emissions. It was stated that the member countries should to take actions for reducing greenhouse gas emissions to meet the goal of limiting global warming to 2^{o} C above the pre-industrial global mean temperature (King et al. 2011).

The increasing numbers of policies and legislations on energy and emission, together with the price and demand for nonrenewable fossil fuels, and the reduction in reserves of energy commodities, have resulted in greater efforts toward the minimisation of released emissions during manufacturing production. Therefore, methods for measuring CO2 emissions need to be developed, and effective and efficient approaches to reducing manufacturing emissions are needed for worldwide manufacturers.

5.3 Methods for CO2 Emissions Measurement

Firstly, in order to determine the required degree of coverage, accuracy and disaggregation for measuring carbon emissions, the factors that affect the measurement need to be stated. According to Mckinnon and Piecyk (2010), possible factors which are external to the business may include: global or local obligation or policies, which have been discussed in the previous section; social responsibility for customers and corporate responsibility for the whole supply chain. Internal motives include: examining the carbon impact of logistics decisions; measuring changes in carbon emissions through time; and improving production efficiency in terms of saving energy inputs.

Emissions from various manufacturing activities are measured in different ways. The amount of emissions from fossil fuels burning is usually estimated by multiplying the quantity of the energy used by an emission factor reflecting the typical carbon content of that type of energy. The use of such emission factors is more cost-effective and practicable than collection of air samples or use of theoretical equations. Emissions from production processes such as chemicals and cement making can be estimated by source-specific equations, for example, recording the material inputs multiplying by factors which take into account of the chemistry of those processes (Clean Energy Future 2012). This method is based on conditions of different specific facilities, but requires more time for computing than the use of emission factors, and also needs to be provided more detailed inputs. In additions, Sampling and analysis based approaches are also available to precisely and directly measure gas emissions produced in electricity industry. This type of emission estimation method can also be applied in paint and ink manufacturing units.

Due to the consumptions of raw material and energy, when the lifecycle of a product is considered as a whole system, the CO2 emission during the lifecycle of manufacturing the product refers to the amount of the total carbon emission released from the system to the external environment (Ma and Zhao 2012). In general manufacturing systems, almost all CO2 emissions are energy related, so the electricity consumption is always considered as the primary reason for generating CO2 emissions during production. Therefore, the carbon emissions in manufacturing processes can be calculated as carbon footprint of the total electricity consumption to sustain all machines and ancillary materials used in the whole production activities (Yi et al. 2012). One of the most common ways of measuring manufacturing carbon emissions is to record energy use and to convert energy values into released CO2 by employing an emission/energy ratio (Fransoo et al. 2010). In other words, the carbon emission amount can be obtained by multiplying total consumed energy by the carbon emission/energy coefficient ψ, which reflects the relationship between the amount of energy used and the emission emitted. Thus, in this study, with the consideration of the basic manufacturing scheduling problem introduced in Chapter 3, the amount emission *E* released when producing an entire job can be calculated:

(8)

where *n* is the total number of the required job operations. e_{pij} represents the quality of electricity power used per time unit when the job operation *i* is being processed in machine *j*, e_{wij} represents the electricity power used per time unit when the job is waiting in the machine *j*'s buffer, and e_t represents the electricity power used per distance unit when the job is being transported between two machines. T_{pij} is the total processing time of operation *i* in machine *j*. T_{wij} is the waiting time of operation *i* in the buffer of machine *j*, which is determined by the

l

number of unprocessed jobs whose process priorities are higher than *j*. is the distance between the locations of the machine *j** processing the previous operation and the machine *j* for carrying out the current operation. The unit of energy can be litres or grams of fuel for generating electricity used in production activities and transportation, and according to Cadmus Group (1998), one kilowatt-hour of electricity produced results in 900 grams of CO_2 emitted into the atmosphere. The value of coefficient ψ can then be assigned as 0.9 kg/kh.

On the other hand, as discussed above, manufacturing CO_2 emissions also include the emissions from production processes such as chemicals and cement or steel making. Emission factors, which are used as a tool to estimate emissions emitted to the environment, are usually expressed as the weight of a substance emitted multiplied by the unit weight, distance, volume or duration of the manufacturing activity emitting emissions (Report of Environment Australia 1998). Emission factors can be obtained by source testing, modelling, mass balance conservation or other information.

The general equation for estimating the manufacturing emissions by applying emission factors can be expressed as:

$$E = T \cdot EF \cdot Q \qquad (9)$$

where E is the emission amount, T can be time or other variables, EF represents the emission factor for the specific manufacturing activity, and Q is the amount of material spent in the activity per time unit (e.g. tonnes / yr). For example, the emission factor that reflects the kilograms of CO2 emitted per tonne of bricks produced can be applied in cement manufacturing systems.

Each emission factor is related to emission factor rating (EFR) code. There is a common rating system for similar industries such as chemical production, paper and ink manufacturing, and bricks, ceramics and clay product manufacturing (Gezerman 2012). The rating system, which was developed by the United States Environmental Protection Agency (USEPA), is an overall assessment of the degree of a factor based on the quality of the preceding tests and on the accuracy of the factor representation for the emission souce. Since emission factors can be obtained in different ways, EFR varies greatly. A higher rating of a factor is always based on numerous unbiased observations or on widely accepted experimental tests (Hodgson et al. 2011). There are five rates of the EFR system, namely from A to E. Rating A indicates the greatest degree of certainty that the given emission factor is representative of the emission source type, and on the contrary rating E reflects the worst one. For example, if the estimation of the emissions emitted in a series of manufacturing activities is only based on theoretical emission factors without any control measures, the uncertainty can be as high as 100%. If the emission estimation is based on direct measurement, the uncertainty may be reduced to 20% (Report of Environment Australia 1998).

5.4 Existing Approaches to Manufacturing Emission Reduction

In the industrial environment, there are three possible types of methods for reducing manufacturing emissions. The first one is to find the alternative sources of energy, especially renewable ones such as hydro power, wind power and solar energy sources. Farret and Simoes (2006) introduced various alternative energy sources for different types of manufacturing plants, including hydroelectric power plants, wind power plants, thermosolar power plants, photovoltaic power plants and biomass-powered microplants. Characteristics and applications of each energy source type were discussed, and a model of the integration for multiple renewable sources of energy was presented. It was also pointed out that fossil fuels are still the primary energy source for the world's industries. Wang et al. (2011) presented a low-carbon production scheduling system with the consideration of renewable energy, which aimed to minimise the total carbon emission during production processes. However, the problem of how to find such renewable energy sources was not pointed out in this approach. There is no doubt that using alternative energy resources is the most efficient way to reduce production emission. Although some renewable energy sources such as wind turbines and solar panels have been used in some industrial fields, the amount of these energy types is not stable, leading to an unreliable manufacturing environment. To date, non-renewable fossil fuels such as coal and gas are still the primary source of production energy. It was shown by Environmental Information Agency of US that the fossil fuels would be the primary choice source of energy until at least 2030 (Agha et al 2010). Hence, new energy sources development may not be the foremost consideration currently, and alternative ways for reducing manufacturing production emissions need to be considered.

Another type of the methods for reducing manufacturing emissions is to improve the technological efficiency of individual industrial equipments. For example, Tiwari et al. (2007) proposed an approach for implementing an energy efficient wireless sensor network protocol, in which the battery life of sensors was attempted to be increased by an efficient use of network and efficiently adjusting sleep times of sensors. Mouzon et al. (2007) proposed a method for minimising energy consumption of manufacturing equipment. Combining with several dispatching rules, a tradeoff between keeping running the idle machines and turning them off was discussed based a multi-objective mathematical programming model. The goal was to minimise the energy consumption and also the total completion time during manufacturing. The solutions were non-dominated, which could help the controller choose the best schedule. Dietmair et al. (2009) introduced a model for simulating the behavior of the utility of energy consumption. The process of forecasting machine production energy consumption under certain given scenarios was demonstrated in this model. A number of industrial cases were studied and results showed that this model could be applied in energy efficiency analysis and optimisation tasks.

The third type of methods for reducing manufacturing emissions is to enhance the operational efficiency of the whole system. The problem of reducing released manufacturing emissions is considered simultaneously with the overall performance of manufacturing systems. Many integrated approaches have been proposed. An integrated approach was presented by Agha et al. (2009) to deal with short-term manufacturing scheduling of a multipurpose batch plant and operational planning of utility system concurrently. A multi-objective function was used to simulate and develop scenarios which treated the overmuch emission of harmful gases as a penalty cost. The operational planning of utility system was able to minimise the energy costs (fuel and electricity purchase) and penalty cost while fulfilling the utility demands of the manufacturing unit. A multi-period mixed-integer linear programming model for power generation planning of electric systems was introduced by Mirzaesmaeeli et al. (2010). It aimed to select optimal energy supply sources and to mitigate processing pollutant that meets a specified electricity demand and CO2 emission targets as well as minimising the total cost of electricity simultaneously. Two industrial cases were studied to evaluate and analyse the model. A similar approach was adopted in General Algebraic Modelling System for balancing fuel flexibility and fuel switching, as well as minimising overall cost by Sirikitputtisak et al. (2009). Grandinetti et al. (2006) applied the niched Pareto Genetic Algorithm to optimisation of a pollutant emission reduction problem in the manufacturing industry. A multi-objective combinational linear programming model that deals with pollution emissions and production cost was proposed. The goal was to determine the Pareto set of combinations of Best Available Technologies (BATs) that allow both the maximisation of the emission reductions and the minimization of the total control cost, where the BATs include all methods, tools and technologies that could be adopted in the design and processing phrase. The computational results collected on an industrial case study were satisfactory that it could effectively address the optimal selection of best available technologies to maximise the total production emission and minimised the total cost as well.

5.5 Discussion

As introduced in the previous sections in this chapter, there has been an increasing number of research works focused on emission in manufacturing facilities. Approaches have been proposed to deal with the problems of saving production energy and reducing the amount of released emission. The methods related to searching for alternative energy sources have been shown inefficient in the near future. Some other methods regarding improvement of equipment technological efficiency are able to help manufacturers decrease the amount of production emissions, but one of the drawbacks is that as these technological methods are suitable to specific production types of manufacturing systems or under certain scenarios, they cannot provide a generic applicable approach for all manufacturers. The methods which attempt to enhance the operational efficiency of the system provide more general and flexible approaches when dealing with emission reduction problems in production processes.

In the literature, the problem of emission reduction is rarely considered simultaneously with the optimisation problem of production scheduling. As the increasing effects of global environmental deterioration and the pressure from local governments, manufacturers have to consider both economic and environmental objectives when they attempt to find optimal scheduling plans for the required jobs. In addition, agent technologies have been rarely employed in existing approaches regarding the problem of reducing manufacturing emissions. Agent and multi-agent systems have been introduced as efficient methodologies to address the improvement of the overall system performance in this optimisation problem. In this study, an extended model based on the previous mechanism introduced in Chapter

3 is proposed, where the amount of emission released during production, as well as other factors such as production cost and lead time, are all taken into account in the optimisation problem for generating optimal or near optimal production schedules.

Before developing the extended model with emission consideration, the method for measuring emission emitted during production should be determined. As introduced in Section 5.3, there are two main categories of calculating manufacturing emissions. Emissions from fossil fuels burning can be calculated by multiplying the quantity of the energy consumed by a carbon emission/energy coefficient, while emissions from production processes can be measured with the tool of emission factors and the corresponding rating codes. In this study, since there is not any single manufacturing type being specified, in general manufacturing CO2 emissions from production activities are mainly related to energy consumed, including electricity used in job processing, storing and transportation. Therefore, the method with carbon emission/energy coefficient for measuring CO2 emission during production is selected in the extended model.

5.6 Extended Model with Emission Consideration

In this study, the previous basic mechanism for solving the optimisation problem of manufacturing process planning and production scheduling is combined with the manufacturing emission factor by two methods, in which the manufacturing emission factor is considered as a constraint and as an objective respectively.

In order to involve the emission factor in the previous agent bidding based approach, one potential method is to define the production emission as a constraint the same as the production lead time, i.e., the constraints of the production process are both the total lead time and the total released emission. As introduced previously, many policies on energy saving and emission reductions have been formulated in different countries and international organisations. In this context, for local manufacturers, the released emission for completing a required job cannot exceed a certain amount during production.

On the other hand, emission reduction is able to help manufacturing enterprises save cost of installing and operating end of pipe pollution control devices and simultaneously increasing their productivities and efficiency, since producing less waste reflects a better utilisation of inputs resulting in lower material and waste disposal costs. Therefore, for the second method to involve manufacturing emissions to the basic optimisation model, maximising the emission reduction and minimising the total production cost are both considered as objectives.

5.6.1 Emission as Constraint

Based on the standard model introduced in Chapter 3, the new objective of the problem is to find a routing and sequencing solution for completing an entire job including several different operations with the total minimal production cost and both satisfied due date and the amount of emission released. The new objective function modified based on (5) is shown below:

$$Min\sum_{i=0}^{n}\sum_{j=0}^{m}C_{ij}y_{ij}$$

Subjected to $$\sum_{i=0}^{n}\sum_{j=0}^{m}T_{ij}y_{ij} \le D$$

$$\sum_{i=0}^{n}\sum_{j=0}^{m}E_{ij}y_{ij} \le L$$

$$y_{ij} \in (0, 1) \quad for\ i \in [0, n]$$

$$\sum_{j=0}^{m}y_{ij} = 1 \quad for\ i \in [0, n]$$

$$Tf_i < Ts_{i+1} \text{ for } i \in [1, n]$$

(10)

where L is the maximal emission allowed to be released when producing the entire job. E_{ij} is the total production emission for resource j performing operation i, which can be calculated by Equation (8), including the emission generated when the job is being processed, stored in the machine buffer and transported between machines in the system.

For the agent iterative bidding process, the new type of bid submitted to the shop agent by resource agent j consists of three elements: (C_{ij}, T_{ij}, E_{ij}), which represents the total production cost, lead time and emission released for carrying out the operation i. Accordingly, the weight function (5) should be modified as follow:

$$f_{ij} = g_i \frac{C_{ij}}{C_{\max}} + h_i \frac{E_{ij}}{E_{\max}} + (1 - g_i - h_i) \frac{T_{ij}}{T_{\max}} \quad (0 < g_i, h_i, < 1) \tag{11}$$

where (g_i, h_i) are two coefficients which control the weights of cost, emission and time for a resource agent j carrying out a specific operation i. All the bids are calculated and submitted by eligible resource agents and the shop agent selects the bid with the minimal f_{ij} value as the winner to perform this operation. This process repeats until all the requested operations are allocated to the winner resources.

The weight parameters {(g_i, h_i)} are used to coordinate the bidding process the same as in the standard model proposed in Chapter 3. The new type of chromosome consists of two set of weight parameters, i.e., CM_x = {g_{x1}, g_{x2}, …, g_{xn}, h_{x1}, $hx2$, …, h_{xn}}. In each chromosome of the initial population, g_{xi} and h_{xi} are both randomly generated over the range of [0, 1]. The fitness function of each chromosome is still based on the total production cost of the allocation plan for completing the entire job with the corresponding weight parameters in operation bidding rounds. Parent selection, crossover and mutation are similar with the processes in the standard model. The eligibility of each chromosome is needed to be examined in terms of total production lead time and total amount of emission released during production. If the set of weight parameters are able to result in an allocation plan which satisfies the job due date and the maximal allowed production emission, this chromosome is eligible and the corresponding allocation plan can be a potential solution. After GA iterations for adjusting the weight parameters through the bidding process, a near optimal weight pair set {(g_i, h_i)} is expected to be found.

Thus, the objective of minimising the total processing cost and with satisfied lead time and the amount of released emission can be achieved.

5.6.2 Emission as Objective

A solution of this problem is an allocation plan for a required job which minimises the total production cost and the total emission, simultaneously. When a problem has two or more objectives to optimise, it belongs to the multi-objective optimisation problems. The multi-objective optimisation is defined as the process of optimising a set of objective functions systematically and simultaneously by Marler and Arora (2004). There are two types of multi-objective problems, namely selection problems and optimisation problems. In selection problems, a solution is selected among a set of finite solutions according to the preference of the decision maker (DM). In optimisaion problems, on the other hand, the preference of DM is not required, and a single optimal solution cannot be found. Instead, there is a set of non-dominated optimal solutions which are called Pareto-optimal set, or Pareto front. DM is provided the set of solutions and chooses the preferred one according to the relative importance of different objectives (Steuer, 1986). In mathematical terms, a multi-objective optimisation problem can be represented as follows:

$$\underset{x}{\min}\{f_1(x), f_2(x), \ldots, f_m(x)\}$$

$$g_p(x) \leq 0, \qquad \forall p =$$

Subjected to

(12)

where $f_i(x)$ is the *i*th objective function among *m* objections. $g_p(x)$ and are the inequality and equality constraints. The multi-objective optimisaion problems aim to find the set of non-dominated solutions which optimise the *m* objectives over the constraints. A solution *x* is feasible if there does not exist another feasible solution *y* such that $f_i(x) \leq f_i(y)$ for all *i* ∈ (1 … m), and $f_j(x) < f_j(y)$ for at least one *j* ∈ (1 … m). The goal for a multi-objective optimisation problem is to determine the Pareto front which is formed by the set of non-dominated optimal solutions.

For solving the problem of minimising the total production cost and maximising the emission reduction during production simultaneously, the objective function (5) is modified as shown in (13).

$$Min\sum_{i=0}^{n}\sum_{j=0}^{m} C_{ij} y_{ij}$$

$$Min\sum_{i=0}^{n}\sum_{j=0}^{m} E_{ij} y_{ij}$$

Subjected to $$\sum_{i=0}^{n}\sum_{j=0}^{m} T_{ij} y_{ij} \leq D$$

$$y_{ij} \in (0, 1) \quad for\ i \in [0, n]$$

$$\sum_{j=0}^{m} y_{ij} = 1 \quad for\ i \in [0, n]$$

$$TF_i < TS_{i+1} \quad \text{for } i \in [1, n]$$

(13)

One of the common methods to solve multi-objective problems is to integrate all the objectives into a single objective function using weighted-sum aggregation (WSA) method. In the WSA method, a weighting coefficient assigned to each objective determines its particular proportion among all considered objectives, then the sum of weighted-objectives is aimed to be minimised. The generalised problem in (12) is then formulated as:

$$g_p(x) \mid 0, \qquad \forall p =$$

Subjected to

(14)

Therefore, the multi-objective optimisation problem is converted into a single-objective optimisation problem. The Pareto front is obtained by adjusting the values of $\{w_i\}$, and solving the above problem by any methods for single-objective optimisation. In this method, a weighting coefficient w_i does not necessarily correspond to the importance of the objective f_i. Kim and de Weck (2004) suggested that as long as all the weighting coefficients are positive values, the generated optimum solution is non-dominated. In this study, weighted-sum aggregation method is employed firstly due to its generality and computational simplicity, and secondly because in this extended model there are only two objectives, i.e., miminising total production cost and emission, only one independent weight is needed. This means with knowing either one, the other can be calculated simply by subtraction.

In order to add the second objective, i.e. emission reduction, in the previous agent bidding based method, weight function (11) is still adopted for shop agent to decide winner resources for all job operations. In the GA based tuning process for adjusting the two set of parameters, the fitness function for chromosome x, which represents the total production cost in the standard model, is modified as shown in (15),

$$F_x = C_x \times W + E_x \times (1-W) \qquad (0 < W < 1) \qquad (15)$$

where C_x and E_x are the total production cost and emission for performing the entire job with the coordination of agent bidding by weight parameters $\{(g_{ix}, h_{ix})\}$ in the corresponding chromosome x. The independent weight variable W determines the partial contribution of total cost and emission to the fitness function. Since the WSA is a preference based method, meaning that the values of weights are supplied by the DM, priori information on DM's preference may be needed. In this study, such information is unknown. The weight variables in this test, therefore, are assigned to distributed values with small variations over the range [0, 1]. Using similar agent iterative bidding and GA tuning processes with different values of W, the near Pareto-optimal set with different relative importance of the total production cost and released emission can be found. More details of the test simulation and experiments will be introduced in the next chapter.

5.7 Summary

This chapter firstly introduced manufacturing energy consumption and emission. Industrial companies are now under the pressure of government as well as the public option in terms of the emission exhausted in the production processes. Recent environmental strategies and policies of different countries and international organisations were presented. Methods for measuring CO2 emissions in manufacturing activities were introduced. Three types of existing manufacturing emission reduction approaches including searching for alternative energy sources, improving technological efficiency of equipment and enhancement of operational efficiency were discussed. Two methods for involving the manufacturing emission factor into the basic agent iterative bidding mechanism were described. One method is to consider the emission released from production as a constraint, in other words, the total emission with the selected allocation plan cannot exceed a certain amount. The other method is to model a multi-objective optimisation model where the total production cost and emission are considered as two independent objectives with the only constraint of the job due date. A weighted-sum aggregation method is employed to solve the multi-objective optimistion problem. In the next chapter, experimental tests and results will be described.

CHAPTER 6

NUMERICAL TESTS

6.1 Introduction

The basic model of agent based iterative bidding mechanism and the extended model with the consideration of manufacturing emissions were introduced in the previous chapters. The basic mechanism provides an efficient way for solving the production scheduling problems in manufacturing systems. A routing and sequencing plan is expected to be acquired after the agent bidding based searching algorithm, in which the total operational cost for carrying out an entire job is aimed to be minimised and simultaneously its delivery due date should be satisfied. Subsequently, manufacturing emissions are included in the proposed mechanism and two extended models are presented with the consideration of the emission factor as a constraint and as a second objective respectively. This chapter provides the implementation of the mechanisms and numerical tests to validate the models. Section 6.2 introduces the design of the tests including the machines in the manufacturing system and operation details of job orders, as well as parameters in GA searching algorithm. Results of two separate tests are provided in Section 6.3. In Test 1, the factor of manufacturing emissions is considered as a constraint, and in Test 2 the emission factor is defined as another independent objective. Section 6.4 provides a discussion based on the observed results, and a summary of this chapter is given in Section 6.5.

6.2 System Layout and Setup parameters

Firstly, in the tests for validating the proposed models, a few assumptions of the manufacturing system are made:

1. The input buffer of the system is infinite, i.e., new jobs are constantly available for loading.
2. The output buffer of the system is infinite, i.e., finished products can always be unloaded.
3. Each order of a job has its own release time and due time.
4. The order of operations for each required job is predefined and invariant.
5. Machines in the system may be capable of executing more than one type of job operation.

The job order information and data is provided in Section 6.2.1, while the data of all system machines is illustrated in Section 6.2.2.

6.2.1 Product Order Information

In the experimental tests, there are 20 job orders from customers in total. They arrive in the manufacturing unit in five different days, namely Day 0 to Day 4, with different individual due dates. 15 of the orders have four operations with fixed production sequences and the rest have 6 operations. For the sake of generalisation, job operational types are not defined explicitly in details for production jobs but follow the index from number 0 to 9, which can be milling, drilling, turning, boring, etc. The batch quantity of all jobs is 10. Table 6.1 shows the detailed data of all job orders. The volume of maximum allowed emission released when producing each job is also indicated (for the Test 1). For example, for Job 0, there are four operations to be processed sequentially. The order is released in Day 0, and it is expected to be completed in at most 61 time units. The maximum emission volume for carrying out this job is 82 emission units. Note that there are 60 time units in each simulation day.

JobNumber	Due Time	Order Set Date	Max Emission	Operation Types and Sequences					
0	61	0	82	0	1	3	5		
1	60	0	79	0	1	2	7		
2	63	0	77	0	2	0	7		
3	151	1	115	0	1	0	2	0	7
4	126	1	80	0	2	0	7		
5	131	1	86	4	5	6	8		
6	139	1	89	4	6	8	9		
7	126	1	78	0	2	0	7		
8	189	2	78	0	2	0	7		
9	201	2	87	4	5	6	8		
10	201	2	87	4	5	6	8		
11	273	3	119	3	5	7	9	3	5
12	284	3	134	3	8	0	2	0	7
13	284	3	134	3	8	0	2	0	7
14	261	3	91	4	5	7	8		
15	318	4	93	4	6	8	9		
16	318	4	93	4	6	8	9		
17	318	4	93	4	6	8	9		
18	318	4	93	4	6	8	9		
19	342	4	129	3	5	7	9	3	5

Table 6.1 Received product orders.

6.2.2 Manufacturing Resources

The manufacturing unit in the tests consists of 10 machines with different operational capabilities and different locations in the system. All machines are multi-functional, i.e., they are able to carry out more than one type of operations. For each machine, there is a buffer storing unprocessed jobs from previous orders. The capacity of each machine buffer is 10. If there are already 10 unprocessed jobs in the machine buffer, it cannot receive any more new job orders. The waiting time T_{wij} and cost C_{wij} can be calculated as follow:

$$T_{wij} = \sum_{k=1}^{K} (T_{sk} + T_{pk}) \tag{16}$$

$$C_{wij} = T_{wij} \times unitWaitCost \tag{17}$$

where K is the total number of unprocessed jobs which have already arrived before the new job is entered. T_{sk} and T_{pk} are the setup time and processing time of each unprocessed job k. *unitWaitCost* is the cost for storing an unfinished product in machine buffer per time unit, which in the tests is set to be 0.5.

Each machine has a unique coordinate position so that the distance between two individual machines in the manufacturing unit is fixed:

$$l_{jj^*} = \sqrt{(x_{ij} - x_{(i-1)j^*})^2 + (y_{ij} - y_{(i-1)j^*})^2} \tag{18}$$

The transporting cost C_{tij} can be derived from the following equations:

$$C_{tij} = l_{jj^*} \times unitTransCost \tag{19}$$

$$T_{tij} = l_{jj^*} \times unitTransTime \tag{20}$$

where j is the current machine for processing the current operation i and j^* is the winner machine for processing the preceding operation (i-1). *unitTransTime*, *unitTransCost* are the material transporting time and cost per distance unit. In the tests, the unit value of *unitTransTime* and *unitTransCost* are set to be 1.25 and 1 respectively. It should be noted that the material handling resource is assumed to be always available in this test manufacturing unit. When an operation of a job is completed by a machine, the unfinished job is immediately transported by the material handling resource to another machine which will perform the next operation of the job.

The quantity of emissions generated during production can be calculated by multiplying total consumed energy by a carbon emission/energy coefficient ψ as introduced in (8), where the electricity power used per distance unit e_t when the job is being transported between two machines is 0.75 and value of coefficient ψ is set to be 0.9.

Table 6.2 and Table 6.3 list the data of two of the ten machines in the manufacturing unit. For instance, Machine 0, which is allocated at (1, 0) in the shop floor, is able to process the types of Operation 0, Operation 2 and Operation 4. The value of e_{wij} is the energy (electricity power) used per time unit when the job operation i is waiting in the machine buffer, and e_{pij} is the quality of electricity power used per time unit when the current operation i is being processed in machine j. In the buffer of Machine 0, there are three unprocessed jobs which arrived before the test begins. The data of all the machines are introduced in Appendix A.

Machine Number	0	Position X:	1	Position Y:	0	
Operation Type	Setup Cost	Setup Time	Process Cost	Process Time	e_{wij}	e_{pij}
0	6.8	2.4	10	7.5	0.3	3.4
1						
2	7.3	2	12.5	6.6	0.3	2.9
3						
4	6.4	2.5	9.8	7	0.3	3.3
5						
6						
7						
8						
9						
Buffer Jobs					Job/Operation	
0	7.3	2	12.5	6.6	N/A (2)	
1	6.4	2.5	9.8	7	N/A (4)	
2	7.3	2	12.5	6.6	N/A (2)	
3						
4						
5						
6						
7						
8						
9						

Table 6.2 Data of Machine Number 0.

6.2.3 Simulation Parameters

In the agent iterative bidding process, the GA runs for 50 iterations and the chromosome population size is set to be 20. For chromosome crossover process, the single cutoff point crossover is adopted. The mutation probability β is 0.125 and the lower limit μ^l and upper limit μ^h are set at 0.05 and 0.1 respectively, which means the mutation operator μ is randomly generated over the range [0.05, 0.1]. When agents adjust unprocessed jobs in local buffer queues based on individual utilities as described in (7), the parameter A is assigned to be 0.5, which ensures agents to gain positive profit even if the new calculated cost is equal to the *Reserve Price*.

Machine Number	1	Position X:	2	Position Y:	3	
Operation Type	Setup Cost	Setup Time	Process Cost	Process Time	e_{wij}	e_{pij}
0	4.4	1.9	13.6	8	0.2	2.6
1	9.7	3.9	11.2	6.5		
2					0.2	3.0
3						
4	5.8	1.6	10	9.8	0.2	2.9
5						
6						
7						
8						
9						
Buffer Jobs					Job/Operation	
0	5.8	1.6	10	9.8	N/A (4)	
1	5.8	1.6	10	9.8	N/A (4)	
2	7.7	1.9	13.6	8	N/A (0)	
3	9.7	3.9	11.2	6.5	N/A (1)	
4	5.8	1.6	10	9.8	N/A (4)	
5						
6						
7						
8						
9						

Table 6.3 Data of Machine Number 1.

6.3 Tests and Results

The proposed model of the multi-agent system is implemented based on the use of Java Development Kit (JDK). Two tests are involved in this section. In the first one, the factor of manufacturing emissions is considered as a constraint as aforementioned in Section 5.2.1, while the second one attempts to solve the multi-objective problem in terms of minimising both production cost and emissions.

6.3.1 Test 1: Emission as a Constraint

6.3.1.1 Test Aim

This test aims to verify the effectiveness of the proposed agent bidding based mechanism for optimising manufacturing production scheduling, where the objective is to minimise the total production cost for completing a product order with the satisfied lead time and the amount of emission released during production. Additionally, the results of machine agent buffer self-adjustment are also examined.

6.3.1.2 Test and Results

Figure 6.1 depicts the bids received for the first operation (operation type 0) of Job 0 during self-adjustment by machine agents after the first bidding round for this operation is finished. The shop agents announce the winner bid with the best cost and time to all eligible machines. These machine agents rearrange their buffer jobs and submit new bids if they gain more profit than *A* according to (7). New bids are collected by the shop agent and a new winner bid is selected. As shown in the figure, this process repeats four times until there is no more bid received which overcomes the current best one. Therefore, among four eligible machines, Machine Number 0 is chosen as a temporary winner for carrying out this operation with 16.4 cost units.

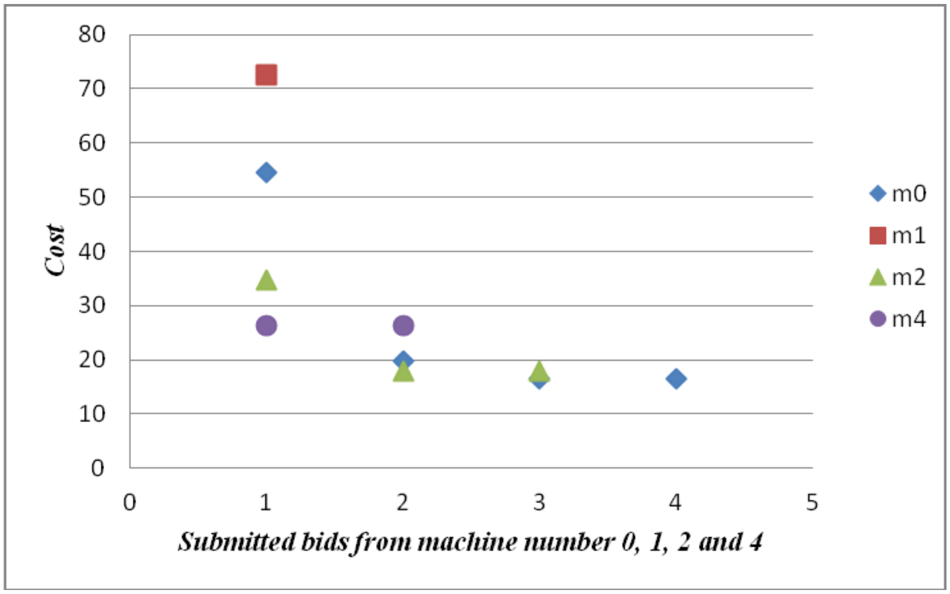


Figure 6.1 Bids received for 1st operation (Type 0) of Job Number 0 during agents adjusting the priorities of buffer job after 1st bidding iteration finished.

Results of buffer condition for machines Number 0 and 6 in five different time periods are shown in Table 6.4 and Table 6.5 respectively as typical examples for examining the performance of agent self-adjusting mechanism. Day 0 to Day 4 here refer to the beginning time of each day for production. As introduced before, in each simulation day there are 60 time units. Types in the table refer to different job operation types, and the jobs without a number are the unprocessed ones which are already stocked in the buffer before the first batch of job orders enter into the system at Day 0.

Initially, the job operations are assumed to be arranged at the end of machine buffer queues based on the FIFO rule. As shown in the tables, it can be found that after receiving the information of the previous bidding rounds, machine agents are able to adjust the priorities of the jobs stored in their buffers based on the local utility. In other words, a machine agent decides to reschedule its buffer jobs if only the profit of the new calculated bid is greater than *A* according to (7). For Machine 0, the priorities of two new entered jobs (Operation 0 of Job 0 and Operation 0 of Job 1) in the buffer are changed in Day 0, as they will be processed before two previously entered jobs. At the beginning of the last day, in the buffer of Machine 6, there are two unfinished jobs which are received one day earlier, namely Job 12 and 13. After the agent self-adjusting process, these two jobs are decided to be carried out after the new jobs (Job 15 to 18) finished, in order for the machine agent to compose a bid with higher profit. Other results of machine buffer self-adjustment are shown in Appendix B.

Buffer List	Day 0		Day 1		Day 2		Day 3		Day 4	
	Job	Type	Job	Type	Job	Type	Job	Type	Job	Type
0	N/A	4	7	0			14	4		
1	0	0	3	2			12	0		
2	1	0	N/A	2			13	0		
3	N/A	2	N/A	2						
4	N/A	2								

Table 6.4 Buffer Lists of Machine No. 0 over 5 days.

Buffer List	Day 0		Day 1		Day 2		Day 3		Day 4	
	Job	Type	Job	Type	Job	Type	Job	Type	Job	Type
0	2	2	4	2	3	7	11	7	15	6
1	N/A	7	N/A	9			12	7	16	6
2	N/A	7	N/A	7			13	7	17	6
3	N/A	9	3	7					18	6
4									12	7
5									13	7

Table 6.5 Buffer Lists of Machine No. 6 over 5 days.

Figure 6.2 depicts the results of agent iterative bidding process for performing the entire Job Number 0 with the proposed approach. The total production cost is gradually converged to the minimum (762 cost units) after 13 iterations, i.e. GA generations, of the agent iterative bidding process. Results of all other 19 job orders show similar convergent curves over bidding iterations, which are listed in Appendix C.

The obtained final schedule plans with total production cost, emission and lead time for all the twenty jobs are illustrated in Table 6.6. When comparing the resulting plans with the original data of job orders, it can be found that the constraints of job due dates and maximal production emissions allowed are satisfied.

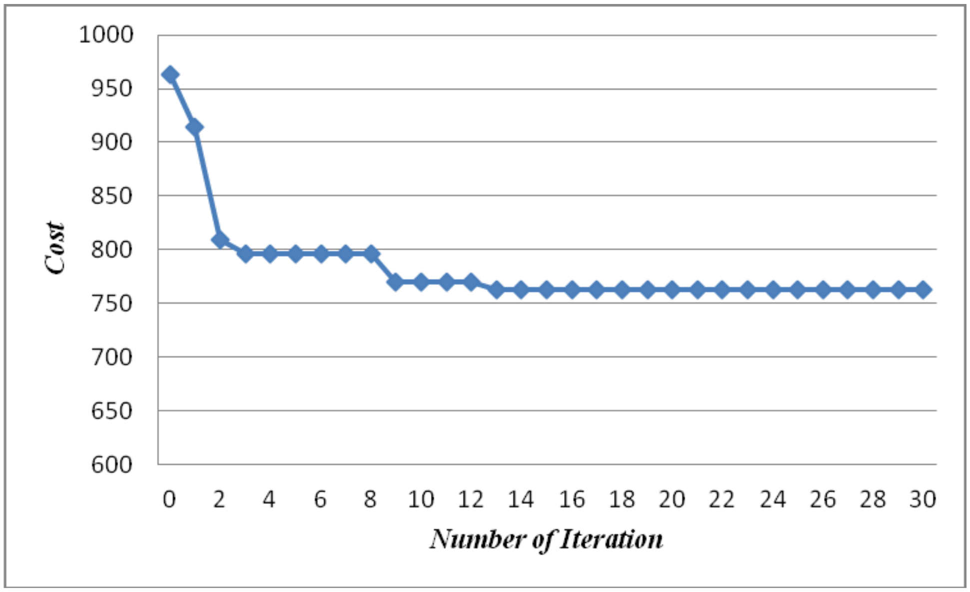


Figure 6.2 Total production cost with bidding process over iterations for Job Number 0.

Job Number	Processing route	Total Cost	Total Emission	Lead Time
0	m0 - m1 - m2 - m4	762	789	518
1	m0 - m2 - m5 - m5	754	754	489
2	m2 - m6 - m1 - m7	813	727	473
3	m0 - m1 - m1 - m0 - m4 - m6	1344	1096	765
4	m2 - m6 - m1 - m6	803	762	553
5	m8 - m8 - m5 - m5	822	832	512
6	m9 - m7 - m8 - m9	796	841	402
7	m0 - m0 - m4 - m5	774	751	431
8	m4 - m6 - m4 - m5	784	747	476
9	m1 - m4 - m5 - m5	825	833	503
10	m8 - m8 - m7 - m8	838	815	448
11	m3 - m4 - m6 - m6 - m2 - m4	1231	1123	756
12	m2 - m8 - m4 - m0 - m4 - m6	1128	1256	742
13	m2 - m8 - m4 - m0 - m4 - m6	1161	1197	781
14	m0 - m4 - m5 - m5	777	843	420
15	m9 - m6 - m5 - m9	769	877	398
16	m9 - m6 - m5 - m9	778	852	446
17	m9 - m6 - m5 - m9	766	878	483
18	m9 - m6 - m5 - m9	806	866	509
19	m3 - m3 - m7 - m7 - m3 - m4	1169	1130	817

Table 6.6 Results of final production plans for 20 job orders.

6.3.1.3 Results Evaluation

In this test, the agent bidding based approach is implemented for the production scheduling problem, where production cost is aimed to be minimised with satisfied lead time and the amount of emission released during production. Results show that the total production cost for carrying out a required job gradually converges to a minimum value. The consistency and efficiency of the algorithm is needed to be evaluated, as described in the following questions:

Question 1 – Whether the acquired solutions are consistent, or are they influenced by any of the random factors in simulation?

Question 2 – Whether the acquired solutions of scheduling are absolute optimum, or how close is the solutions to the real optima?

The Genetic Algorithm is involved in the agent iterative bidding mechanism in order to update and optimise the weight parameters $\{g_i\}$ in equation (5) over iterations. Since the weight parameters are initialised randomly and the processes of GA crossover and mutation are influenced by random factors, for the purpose of result accuracy, 10 replications are conducted under the same initial data of GA. Results show that although the initial population of chromosomes are different from each other for all replications, the same solutions of the schedule plans are achieved for each replication, which ensures that random initialisation of the parameters does not disrupt the final results. For instance, Table 6.7 illustrates the achieved solutions in the 10 replications for the first job. Additionally, in order to examine the consistency of the proposed algorithm, four situations are considered and tested with different evolutionary setup parameters: the chromosome population size in GA is assigned to 10 and 20, and the mutation probability is equal to 0.0625 and 0.125, respectively. (In the original test, the population size is 20 and the mutation rate is 0.125.) Results shown in Table 6.8 show that the same schedule plans are found in the four situations, although the convergence speed is different from each other.

Number of Replication	Processing Route	Total Cost	Total Emission	Lead Time
1	m0-m1-m2-m4	762	789	518
2	m0-m1-m2-m4	762	789	518
3	m0-m1-m2-m4	762	789	518
4	m0-m1-m2-m4	762	789	518
5	m0-m1-m2-m4	762	789	518
6	m0-m1-m2-m4	762	789	518
7	m0-m1-m2-m4	762	789	518
8	m0-m1-m2-m4	762	789	518
9	m0-m1-m2-m4	762	789	518
10	m0-m1-m2-m4	762	789	518

Table 6.7 Schedule plans with 10 different replications for Job Number 0.

Situation	Population Size	Mutation Rate	Processing Route	Terminal Generation
1	10	0.0625	m0-m1-m2-m4	16
2	10	0.125	m0-m1-m2-m4	16
3	20	0.0625	m0-m1-m2-m4	13
4	20	0.125	m0-m1-m2-m4	13

Table 6.8 Results obtained with different simulation setup parameters for Job Number 0 in Test 1.

To answer Question 2, three scenarios are designed with different size of job orders. In Scenario A, there are 5 job orders from customers in total, namely from Job 0 to job 4 in Table 6.1. In Scenario B, the problem is constructed with 10 job orders including Job 0 to Job 9. In Scenario C, the number of job orders increases to 20, which is the same as in the experimental test described above.

Enumeration Method	Job Number	Processing route	Total Cost	Total Emission	Lead Time
Computational Time	0	m0 - m1 - m2 - m4	762	789	518
	1	m0 - m2 - m5 - m5	754	754	489
0.22 ($\times 10^4$)	2	m2 - m6 - m1 - m7	813	727	473
	3	m0 - m1 - m1 - m0 - m4 - m6	1344	1096	765
	4	m2 - m6 - m1 - m6	803	762	553

		Total	4476	4128	2798
Proposed Approach	Job Number	Processing route	Total Cost	Total Emission	Lead Time
Computational Time	0	m0 - m1 - m2 - m4	762	789	518
	1	m0 - m2 - m5 - m5	754	754	489
0.09 ($\times 10^4$)	2	m2 - m6 - m1 - m7	813	727	473
	3	m0 - m1 - m1 - m0 - m4 - m6	1344	1096	765
	4	m2 - m6 - m1 - m6	803	762	553
		Total	4476	4128	2798

Table 6.9 Experimental results of Scenario A with enumeration method and proposed agent based approach.

Enumeration Method	Job Number	Processing route	Total Cost	Total Emission	Lead Time
Computational Time	0	m0 - m1 - m2 - m4	762	789	518
	1	m0 - m2 - m5 - m5	754	754	489
0.97 ($\times 10^4$)	2	m2 - m6 - m1 - m7	813	727	473
	3	m0 - m1 - m1 - m0 - m4 - m6	1344	1096	765
	4	m2 - m6 - m1 - m6	803	762	553
	5	m8 - m8 - m5 - m5	822	832	512
	6	m9 - m7 - m8 - m9	796	841	402
	7	m0 - m0 - m4 - m5	774	751	431
	8	m2 - m6 - m1 - m5	771	752	469
	9	m9 - m4 - m5 - m5	818	809	507
		Total	8457	8113	5119
Proposed Approach	Job Number	Processing route	Total Cost	Total Emission	Lead Time
Computational Time	0	m0 - m1 - m2 - m4	762	789	518
	1	m0 - m2 - m5 - m5	754	754	489
0.26 ($\times 10^4$)	2	m2 - m6 - m1 - m7	813	727	473
	3	m0 - m1 - m1 - m0 - m4 - m6	1344	1096	765
	4	m2 - m6 - m1 - m6	803	762	553
	5	m8 - m8 - m5 - m5	822	832	512
	6	m9 - m7 - m8 - m9	796	841	402
	7	m0 - m0 - m4 - m5	774	751	431
	8	m4 - m6 - m4 - m5	784	747	476
	9	m1 - m4 - m5 - m5	825	833	503
		Total	8477	8132	5122

Table 6.10 Experimental results of Scenario B with enumeration method and proposed agent based approach.

Enumeration	Job Number	Processing route	Total Cost	Total	Lead Time

Method				Emission	
Computational Time	0	m0 - m1 - m2 - m4	762	789	518
	1	m0 - m2 - m5 - m5	754	754	489
10.6 ($\times 10^4$)	2	m2 - m6 - m1 - m7	813	727	473
	3	m0 - m1 - m1 - m0 - m4 - m6	1344	1096	765
	4	m2 - m6 - m1 - m6	803	762	553
	5	m8 - m8 - m5 - m5	822	832	512
	6	m9 - m7 - m8 - m9	796	841	402
	7	m0 - m0 - m4 - m5	774	751	431
	8	m2 - m6 - m1 - m5	771	752	469
	9	m9 - m4 - m5 - m5	818	809	507
	10	m9 - m8 - m7 - m5	827	819	451
	11	m3 - m3 - m6 - m7 - m2 - m4	1230	1119	760
	12	m2 - m8 - m4 - m0 - m4 - m6	1131	1269	743
	13	m2 - m8 - m4 - m0 - m4 - m6	1172	1201	779
	14	m0 - m4 - m5 - m5	773	849	428
	15	m9 - m6 - m7 - m7	765	881	387
	16	m9 - m6 - m7 - m7	780	849	432
	17	m9 - m5 - m5 - m7	749	892	473
	18	m9 - m5 - m5 - m7	786	840	527
	19	m3 - m3 - m6 - m7 - m3 - m4	1153	1159	828
		Total	17823	17991	10927
Proposed Approach	Job Number	Processing route	Total Cost	Total Emission	Lead Time
Computational Time	0	m0 - m1 - m2 - m4	762	789	518
	1	m0 - m2 - m5 - m5	754	754	489
0.61 ($\times 10^4$)	2	m2 - m6 - m1 - m7	813	727	473
	3	m0 - m1 - m1 - m0 - m4 - m6	1344	1096	765
	4	m2 - m6 - m1 - m6	803	762	553
	5	m8 - m8 - m5 - m5	822	832	512
	6	m9 - m7 - m8 - m9	796	841	402
	7	m0 - m0 - m4 - m5	774	751	431
	8	m4 - m6 - m4 - m5	784	747	476
	9	m1 - m4 - m5 - m5	825	833	503
	10	m8 - m8 - m7 - m8	838	815	448
	11	m3 - m4 - m6 - m6 - m2 - m4	1231	1123	756
	12	m2 - m8 - m4 - m0 - m4 - m6	1128	1256	742
	13	m2 - m8 - m4 - m0 - m4 - m6	1161	1197	781
	14	m0 - m4 - m5 - m5	777	843	420
	15	m9 - m6 - m5 - m9	769	877	398
	16	m9 - m6 - m5 - m9	778	852	446
	17	m9 - m6 - m5 - m9	766	878	483
	18	m9 - m6 - m5 - m9	806	866	509
	19	m3 - m3 - m7 - m7 - m3 - m4	1169	1130	817
		Total	17900	17969	10922

Table 6.11 Experimental results of Scenario C with enumeration method and proposed agent based approach.

In Table 6.9, Table 6.10 and Table 6.11, the results from the proposed agent bidding based approach are compared with real optimum solutions obtained by the enumeration method with different problem sizes. To achieve the minimal the total production cost for a required job in the enumeration method, firstly all the possible allocation plans for all job operations are listed, where each potential unprocessed job sequence in each machine buffer is considered. Then all the combinations of machines for performing the entire job are enumerated. The lead time and total emission for carrying out the required job for all solutions are examined. Eligible solutions are obtained as the final candidates, among which the routing and sequencing solution with the minimal total production cost can be selected.

In Scenario A, when the problem size is small, the real optima are able to be found with the proposed new mechanism in this study. After the number of job orders increased to a certain amount, the proposed algorithm cannot find the real optimal allocation plans, but near optimal solutions with satisfied total production cost can be reached. The different of total cost between the proposed method and enumeration method is 0.24% in Scenario B and 0.43% in Scenario C. It is shown that the main objective of this problem, the total production cost for carrying out a required job, is highly close to that in the absolute optimum solution. The computational time, on the other hand, indicates the difference between the efficiency of the two methods. As shown in Figure 6.3, the computational intensity of the proposed algorithm grows approximately linearly, while in the enumeration method, it takes much more time to reach the optima and the computational time grows exponentially with increase of the size of the problem. It can be concluded that listing all the possible solutions is not practicable when the problem becomes increasingly complex with large number of job orders and processing operations. These results reveal that the proposed agent based searching approach provides an effective and efficient method to reach near optimal allocation plans in relatively complex and large scale scheduling problems.

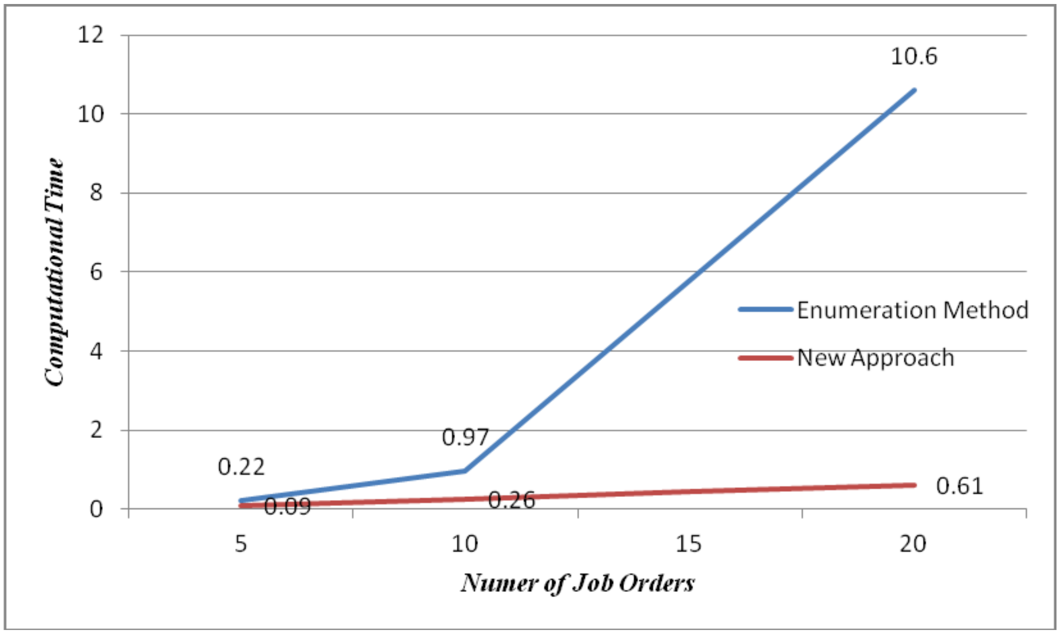


Figure 6.3 Computational times of the simulations with enumeration method and proposed agent-based approach. ($\times 10^4$ seconds)

6.3.2 Test 2: Emission as an Objective

6.3.2.1 Test Aim

As introduced in Section 5.5.2, the other method to include manufacturing emission factor into the basic model is to consider the minimisation of total production cost and reduction of emission released simultaneously. This test aims to examine the effectiveness of the multi-objective mechanism, where minimising total cost and emission are set to be two independent objectives and the lead time is the constraint which must satisfy the job due date.

6.3.2.2 Test and Results

Equation (16) introduces the weighted-sum aggregation method to integrate the two objectives into a single objective function with the variable *W* determining the proportion of considered objectives. In this test, the weight variable is assigned to be 21 different values from 0 to 1 with the interval 0.05. Table 6.12 illustrates the results of the detailed plans for Job 0. Solutions converge into the approximation set of the Pareto Front and can no longer be improved after 17 GA generations. Since some points (representing total cost and emission) are overlapping, there are 7 different schedule plans obtained. Such solutions with a good trade-off between objective values are provided to the system decision maker for choosing a particular one. For example, when the weight variable is between 0.85 and 1, the proportion of total production cost in the aggregate objective function (fitness function) is from 85% to 100% and that of total emission released during production is from 15% to 0%. Minimum total cost can be reached with comparatively higher emission released amount, and machines Number 0, 1, 2 and 4 are selected to carry out the four operations of Job 0. Such a solution may be chosen by the decision maker if the primary objective for production is to keep the cost as low as possible while production emission is not taken into consideration. On the contrary, the routing combination of machine Number 2 for processing the first two operations of the job and machines 9 and 8 for processing the rest two operations may be selected as the final solution if the emission factor is considered as the most important one by the decision maker. All the seven scheduling solutions with different weight variable values derived by the multi-objective optimisation approach can be analysed and compared. Then a particular one can be chosen according to different situations. Figure 6.4 shows all the potential solutions generated during the optimising process (represented by red X) as well as the near Pareto-optimal set (represented by blue diamond).

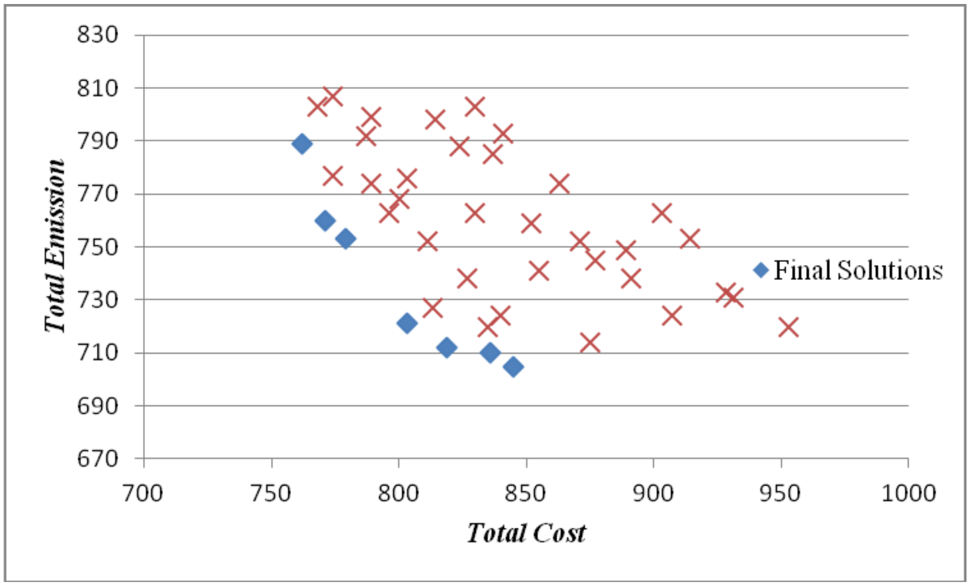


Figure 6.4 Near Pareto-optimal solutions for Job Number 0.

	W	Processing Route	Total Cost	Total Emission	Lead Time
1	0	m2 - m2 - m9 - m8	845	705	537
2	0.05	m2 - m2 - m9 - m8	845	705	537
3	0.1	m2 - m2 - m9 - m8	845	705	537
4	0.15	m2 - m2 - m9 - m8	845	705	537
5	0.2	m2 - m2 - m9 - m8	845	705	537
6	0.25	m0 - m1 - m3 - m8	836	710	513
7	0.3	m0 - m1 - m3 - m8	836	710	513
8	0.35	m0 - m1 - m3 - m8	836	710	513
9	0.4	m0 - m2 - m3 - m4	819	712	504
10	0.45	m0 - m2 - m3 - m4	819	712	504
11	0.5	m0 - m2 - m2 - m4	803	721	466
12	0.55	m0 - m2 - m2 - m4	803	721	466
13	0.6	m0 - m2 - m2 - m4	803	721	466
14	0.65	m0 - m1 - m3 - m3	779	753	497
15	0.7	m0 - m2 - m2 - m8	771	760	506
16	0.75	m0 - m2 - m2 - m8	771	760	506
17	0.8	m0 - m2 - m2 - m8	771	760	506
18	0.85	m0 - m1 - m2 - m4	762	789	518
19	0.9	m0 - m1 - m2 - m4	762	789	518
20	0.95	m0 - m1 - m2 - m4	762	789	518
21	1	m0 - m1 - m2 - m4	762	789	518

Table 6.12 Obtained schedule plans with different *W* values for Job Number 0.

Figure 6.5 shows the obtained non-dominated solutions for all twenty job orders after the agent based searching algorithm, where each point in this figure represents a whole schedule plan for all jobs. When the weight variable *W* is assigned to be 0, according to (14), the fitness function for GA tuning process is just the total emission released during production, leading to the result of total cost-emission pair (20131 cost units, 14765 emission units). When *W* is set to be 1, on the contrary, the fitness function covers only the total production cost, which leads to the result (17435 cost units, 19499 emission units). More detailed results of the multi-objective optimisation problem are listed in Appendix D. Compared to the final solution points shown in Figure 6.4, there are not any overlapping points for the collective plans for all jobs. This is because one schedule plan for the preceding job, with different values of weight variable, may result in several different plans for the following one. Therefore, the probability of achieving the same final schedule plan with two different values of weight variable is quite low. All these non-dominated solutions can be provided to the decision maker in the manufacturer who may choose a preferred one according to the relative importance of two different objectives.

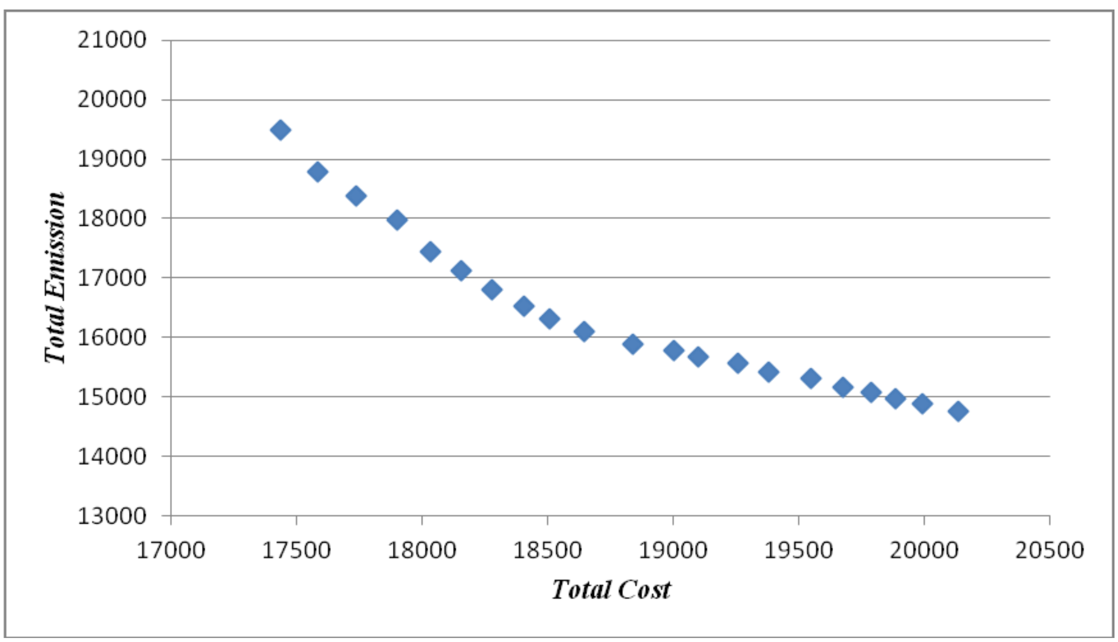


Figure 6.5 Near Pareto-optimal solutions for 20 Jobs.

6.3.2.3 Performance Measure

As discussed in Section 6.3.1.3, the coordination parameters are initialised randomly and updated by a GA tuning method where random factors in GA crossover and mutation may influence the final results. Therefore, it is necessary to qualify the consistency of the proposed algorithm. The simulation in Test 2 also runs for 10 replications to examine whether random factors influence the final solutions. Same results of final schedule plans are obtained. Also, four situations with different GA population size (10 and 20) and mutation probability (0.0625 and 0.125) are tested and compared. The allocation plans found in the final Pareto set are the same in the four situations, although the convergent speed is different from each other. Since the GA process runs repeatedly with 21 different values of *W*, the maximum GA generations before the solutions converge to the near Pareto front are compared, as shown in Table 6.13.

Situation	Population Size	Mutation Rate	Number of solutions in the Pareto set	Maximum GA generations
1	10	0.0625	6	31
2	10	0.125	6	19
3	20	0.0625	6	26
4	20	0.125	6	17

Table 6.13 Results obtained with different simulation setup parameters for Job Number 0 in Test 2.

In order to examine the solution coverage, convergence and quality of the proposed approach as well as to investigate the influence of the evolutionary setup parameters in the searching process, three standard metrics to measure the performance of multi-objective algorithms from literature are used. These measurements also examine and select the most appropriate GA setup parameters including population size and mutation rate. Job Number 0 is selected as a sample case for the measurements.

1. Progress Measure (P):

Progress measure is used to assess the convergence velocity for a single-objective algorithm (Bäck, 1996). This metric is selected because this proposed algorithm employs a weighted-sum aggregation method to convert the bi-objective problem into a single-objective problem. The fitness function presented in (15) can be used to qualify the relative convergence improvement between GA generations. In mathematical terms, the progress measure is defined as:

$$P = \ln \sqrt{\frac{f_{\min(0)}}{f_{\min(T)}}}$$

(21)

where f_{min} (x) is the best objective function value in the parent population at generation x.

In this test, the values of weighted fitness function at 10th GA generation and 20th GA generation are used to calculate P in each situation with different evolutionary setup parameters, namely the population size and mutation probability, in order to test the convergence velocity of the proposed approach. Since the values of the fitness function depend on the weight variable values, f_{min} (10) and f_{min} (20) are calculated as mean values among all the fitness functions with different W. As shown in Table 6.14, the best value of the mean fitness among chromosomes converges gradually over GA generations, and the highest P values representing the highest convergent speed which occurs when the population size is 20 and the mutation rate is 0.125 among all situations at both 10th and 20th generations. As listed in the table, f_{min} (20) is always greater than that of f_{min} (10), which suggests that increasing number of new chromosomes with higher fitness are found over the GA process. In situations 1 and 4 the same value of the f_{min} (20) is derived, because in these two situations the objective function, namely the total production cost, converges to a minimal value before 20th GA generation. For the other two situations with different population size and mutation rate, on the other hand, the minimal total cost has not been found before 20th GA generation.

Situation	Population Size	Mutation Rate	Progress Measure (P)		Two-Set Coverage (TSC)			Spacing (SP)
			P(10)	*P* (20)	1^{st}-5^{th}	5^{th}-10^{th}	10^{th}-20^{th}	SP(20^{th})
1	10	0.0625	0.0210	0.0556	0.93	0.54	0.39	0.507
2	10	0.125	0.0604	0.1574	0.68	0.43	0.17	0.156
3	20	0.0625	0.0373	0.0954	0.84	0.49	0.30	0.469
4	20	0.125	0.0769	0.1574	0.68	0.32	0.14	0.156

Table 6.14 Results obtained by the proposed approach with standard metrics

2. Two-set coverage (TSC):

This metric was introduced by Zitzler et al (2000), in which the relative coverage of obtained solutions in two Pareto sets was compared. TSC is calculated as the percentage of solutions in a Pareto set which are non-dominated as compared to another Pareto set or a previous set. In mathematical terms, consider *X1* and *X2* are two sets of solutions. TSC is defined as the mapping of the pair (*X1*, *X2*) to the internal [0, 1]:

(22)

where *a1* and *a2* are two arbitrary solutions in sets *X1* and *X2* respectively. If all solutions in the Pareto set *X1* dominate or are equal to all solutions in *X2*, TSC is equal to 1, and TSC is 0 implying the opposite.

In this measurement, for each situation with different GA population size and mutation probability, the near Pareto set obtained at the 5^{th} GA generation is compared with that at the 1^{st} generation. Similarly, the solution set acquired at the 10^{th} GA generation is compared with that at the 5^{th} generation and the solutions found at the 20^{th} GA generation are compared with those at the 10^{th} generation. As shown in Table 6.14, in Situation 1, for example, when the setup parameters are valued as in the original test, namely the population size is 20 and the mutation rate is 0.125, the TSC is equal to 0.68, 0.32 and 0.14 respectively with the corresponding comparisons, meaning that 68% of the solutions found at the 5th generation are non-dominated compared to those at the first generation, 32% of the solutions obtained at the 10th generation are non-dominated compared to those at the 5th generation, and only 14% of the solutions acquired at the 20th GA generation, which are the solutions in the final near optimal solution set, are non-dominated compared to those at the 10th generation.

These results appear to suggest that, with the proposed approach new solutions with higher performance are generated more frequently in the early stage of the GA process. After the solutions with higher performance are found, the corresponding chromosomes have more possibility to produce offsprings and less possibility to be replaced compared to other chromosomes in the population. Since the GA process adjusts and optimises the parameters gradually, the probability of achieving better chromosomes decreases with the chromosome reproduction, crossover and mutation. Using different GA population size and mutation probability, the TSC values are different. The comparative results are also listed in Table 6.14, and the best performance in terms of the solution coverage between two solution sets in two generations is found in Situation 1. In addition, TSC is also applicable for comparing the coverage of solution sets obtained by two different multi-objective algorithms, which will be introduced in Chapter 7.

3. Spacing (SP):

This metric was presented by Schott (1995) as a way of measuring the range variances of solutions in Pareto sets. Spacing metric is defined as:

(23)

where , (*i*, *j* = 1, …, *n*) represents the minimum distance from solution S_i to any other solution S_j in the Pareto set. $\bar{d}$ is the average of all d_i. m is the number of objectives and n is the number of solutions in the Pareto set being evaluated. The value of SP of zero means all the non-dominated solutions found in the evaluated Pareto set are equidistantly spaced.

In this test, the SP value of the Pareto set generated at the 20^{th} GA generation is calculated for each situation as listed in Table 6.14. As examined before, in Situation 2 and 4, since the solution points converge to the Pareto-optimal set before 20 GA generations, the same SP value 0.156 is obtained. This result indicates that this optimisation approach achieved a good distribution of final solutions. Meanwhile, in Situation 1 and 3, higher SP value is derived, which suggests that during the GA process, before the convergence of the solution points to the Pareto-optimal set, the range variance of solutions is greater. In other words, these solution points are not distributed as evenly as those in the final Pareto-optimal set. For testing the comparative solution performance in terms of range variance obtained by the proposed approach, spacing metric is also applied for mechanism comparison in Chapter 7.

6.4 Discussion on Test Results

Test 1 and Test 2 examined the effectiveness of the proposed iterative agent bidding based mechanism with the consideration of manufacturing emissions as a constraint and as an objective respectively. In doing so, 20 job orders from customers were allocated in the test manufacturing unit with several different multi-functional machines.

In test 1, the experimental results showed that with the proposed mechanism in this study, a set of optimal or near-optimal plans for scheduling production jobs could be achieved after a certain GA generations. Furthermore, it was also shown that machine agents were able to autonomously adjust the priorities of their buffer jobs according to the information received from the previous bidding rounds. The searching scheme replicated for 10 times, which ensured that the final results were not influenced by simulation random factors. Additionally, different evolutionary setup parameters were tested separately and results show the consistency of the mechanism. Furthermore, in order to evaluate the acquired results, the real optimal solutions were found by an enumeration method. Comparison showed that the real optima could be found with the proposed mechanism if the problem size was small. Near optimal solutions with satisfied objective values could be achieved for more complex problems, and the computational time growed approximately linearly. These results revealed that the proposed agent based searching approach was able to find the near optimal routing and sequencing plans efficiently in a relatively complex and large scale scheduling problem.

Test 2 aimed to investigate the approach for solving the multi-objective problem. Minimising the total processing cost and reducing the emission released from production for each job were defined as two independent objectives. Same scenarios were examined with different evolutionary setup parameters to test the consistency of the result solutions. Three standard metrics were used to measure the performance of the multi-objective algorithm in terms of the quality and efficiency. It could be observed from the results that with the propose algorithm, a near Pareto-optimal set was obtained gradually over the genetic generations with a consistent group of solutions. It was proved that when the GA population size is 20 and the mutation rate is 0.125, the proposed approach performed most efficiently. Note that because the values of weight variables $\{W\}$ were generated in a uniform distribution with small variations, the whole Pareto Front for all 20 jobs' schedules was not able to be found. Nevertheless, such non-dominated solutions obtained by this algorithm could be satisfied from the decision maker's point of view. DM may choose a preferred one according to the relative importance between two factors in different situations.

In the single objective optimisation problem, real optimal solutions were enumerated for comparison. For testing the solution coverage, convergence and quality, standard metrics were used to measure the results obtained by the multi-objective optimisation approach. Relative performance of the proposed algorithms, on the other hand, needs to be examined and discussed. In the next chapter, comparisons between the proposed approaches and two others from the literature are presented to analyse further on the efficiency of the proposed searching mechanism.

6.5 Summary

This chapter discussed the numerical tests to validate the agent based mechanism described in the previous chapters. Information and data for the testing job orders and manufacturing system resources were firstly introduced. The first test verified the capability of the agent based mechanism for finding near optimal schedule plans with the objective of minimising the total production cost subject to the constraints of lead time and released emission during production. The second test aimed to minimise the total production cost and emission simultaneously with the lead time as a constraint. Near Pareto-optimal solutions with different preference of cost and emission were found, which may be provided to the system decision maker. Experimental results were evaluated under different scenarios with standard multi-objective performance measures. In the next chapter, the tests and results will be further discussed and two other approaches from literature will be implemented to investigate the comparative efficiency of the proposed algorithms.

CHAPTER 7

COMPARATIVE EXPERIMENTATION

7.1. Introduction

In the last chapter, the proposed agent bidding based mechanism for optimising production scheduling was implemented. Two models were built with emission factor as a constraint and as a second objective respectively. In this chapter, the results from the tests are further discussed by comparing them with those of two other mechanisms from literature. Two models are rebuilt and the simulation results are compared in order to investigate the relative performance of the proposed mechanisms. In Section 7.2, Dynamically Integrated Manufacturing Systems (DIMS) and a dispatching rule based mechanism (D.R. for abbreviation) are described. Modification of these two mechanisms are also introduced for the purpose of ratinal comparison. DIMS is used in the comparison of the results acquired from Test 1, and D.R. is involved to compare both of the experimental results presented in the last chapter. Section 7.3 compares the results obtained from the single-objective mechanism, and in Section 7.4 the comparison between two multi-objective optimisation methods based on standard performance measuring metrics is described. A discussion is provided based on the results of comparison in Section 7.5. Section 7.6 gives a brief summery to this chapter.

7.2 Introduction and Modification of Two Previous Mechanisms

7.2.1 Maione and Naso (2001)

Dispatching Rule based Mechanism (D.R.)

As mentioned in Chapter 2, dispatching rules have been widely applied to the problems of optimising the performance of a manufacturing system, such as total tardiness, production lead time and system utilisation. In this test, in order to examine the comparative efficiency of the proposed mechanism, another approach based on dispatching rules for solving production planning and scheduling optimisation problems (Maione and Naso, 2001) is selected. The model is modified and rebuilt. This mechanism is selected for comparison because firstly the system in that work is in a distributed environment the same as in this study. Secondly, similar weight parameters are involved and there is an Evolutionary Algorithm (similar with GA but without crossover) tuning method for updating parameters for the purpose of gradually seeking the global optimum. Thirdly, it employs different approach for generating the schedule plans, i.e., by determining the combination of dispatching rules, rather than by agent bidding process. Therefore, by comparing the proposed mechanism with the one based on dispatching rules, solid comparative results would be reached.

In the research work of Maione and Naso, job parts and system resources are represented by part agents (PA) and workstation agents (WA). Each PA controls the decision algorithm and contains the data necessary to process the associated part. It connects with all the capable machines to obtain the necessary real-time data, and then determines the processing route of the part. The route is selected based on a fuzzy weighted combination of decision rules, including Shortest Operation Completion Time (SOCT), Shortest Distance from next station (SD) and Shortest Set-up with reference to the Last Part in queue (SSLP). Workstation agents are associated with different machines in the system, and make local decisions for job sequencing within the machine buffer based on a similar fuzzy weighted combination of dispatching rules, including First Come First Serve (FCFS), Earliest Due Date (EDD) and Shortest Set-up Time (SST).

When a given operation of a part is finished, the corresponding PA needs to choose the next destination (workstation) according to m different potential decision rules. Analogously, when an operation is completed by a workstation, its WA must select the next part to process among those waiting in its buffer based on n different pre-designed dispatching rules. The objective of this problem is to find the alternative combination for both PAs and WAs which provides the best trade-off of satisfaction of certain criteria such as total processing time and slack time.

The set of weight parameters {ω}, which are associated with each of the single PA and WA decision rules, control the preference of different rules in the combination. Weight parameters are updated based on the evolutionary algorithm tuning process. Each chromosome is formulated as shown in (24),

$$I^x = (\omega^x, \sigma^x) \tag{24}$$

where I represents an individual chromosome in xth population. ω is the set of weight parameters, each of which refers to a selected dispatching rule. σ is the standard deviation vector, which defines the amplitude of the additive perturbation applied to each of the corresponding ω^x if I is chosen for reproduction. One of the methods for adapting σ^x is illustrated in (25),

$$\sigma^{x+1} = \begin{cases} \alpha \cdot \sigma^x & \text{if} \quad \varphi < \gamma \\ \beta \cdot \sigma^x & \text{if} \quad \varphi > \gamma \\ \sigma^x & \text{if} \quad \varphi = \gamma \end{cases} \tag{25}$$

where α, β and γ represent three configuration parameters, which are assigned empirically to 0.82, 1/0.82 and 1/5 respectively. φ represents the percentage of successful mutations among all mutations in the last generation. For generating a new population, it follows the schema stated below:

1. Rank all x chromosomes according to their fitness;
2. μ best chromosomes are selected as parents;
3. λ offspring chromosomes are created, where $\mu + \lambda = x$;
4. The new population is composed by μ parents and λ offsprings.

In the mechanism of Maione and Naso, part agents and workstation agents share the same adaptation strategy, but the two EA processes are carried out separately. The objective of PAs is to minimise the total flow time for completing all parts, which is assigned as the fitness of each chromosome in the EA process. On the other hand, WAs aim to minimise job waiting time in their buffer queues and simultaneously minimise the number of set-ups.

In the mechanism of Maione and Naso, the Evolutionary processes are separately and independently performed for machine agents and part agents, the two types of agents have different objectives, i.e., to minimise the total production time, and to simultaneously minimise the number of set-ups and job waiting time in buffer. In order to make rational comparisons with the mechanism proposed in this study, the emission factor is added and the original model is modified in this test. The objective (objectives) of the modified model based on the mechanism of Maione and Naso is redefined as the same as the proposed model, i.e., to minimise the total production cost with satisfied total lead time and released emission, or to minimise the total cost and emission simultaneously with the satisfied lead time, for both part agents and workstation agents.

In the EA tuning process, the length of each chromosome is equal to the number of total dispatching rules for PAs multiplied by the number of total rules for WAs. For instance, there are m alternative rules for PAs and n rules for WAs, each chromosome is formulated as $\{w_{11}, w_{12}, \ldots, w_{1n}, w_{21}, \ldots, w_{2n}, \ldots, w_{mn}\}$, where w_{ij} represents the weight for using the ith rule for PAs and jth rule for WAs. As the same as in the original work, the weight parameters associated to different dispatching rules are pre-designed and fixed for each agent. For the single objective model, after each chromosome is created, the total production cost with the corresponding $\{w_{ij}\}$ is calculated as the fitness of the chromosome, and the total lead time and emission should also be calculated so as to investigate whether the this chromosome is able to result in an allocation plan satisfying the due date and the maximal allowed emission amount of the entire job. If the constraints cannot be satisfied, a new chromosome is generated and evaluated again. For the multi-objective model, the weighted-sum function described in (15), which aggregates the total cost and emission into a single function, is set to be the fitness function in EA process. The total lead time for performing the job is the only constraint which needs to be examined after chromosomes initialised or reproduced. The same testing environment and data of product orders and machines are used in this comparison test.

7.2.2 Zhang et al. (2007)

DIMS

Dynamically Integrated Manufacturing Systems (DIMS) was developed by Zhang et al. (2007), which aimed to find a routing and sequencing plan for each customer order with the satisfied delivery time and near minimal operating cost, while alternative system configurations and structures were considered when the current system was not able to meet the market challenges. A hierarchical structured system was built with heterarchical autonomous agents. In the agent bidding based coordination approach, two adjustable parameters are involved, namely, the virtual price of each operation of the job: (P_1, …, P_n) and the minimal virtual profit of each resource in the shop: ($F_{min,\ 1}$, …, $F_{min,\ m}$). Within each iteration, the shop agent announces the job operations according to their operational sequence for resource agents to bid. For each operation *i*, each resource agent with the technical ability calculates the lead time and operational cost for carrying out the operation, and then works out a virtual profit if it takes part in the operation, based on the following equation:

$$F_{i,\ j} = P_i - C_{i,\ j} \tag{26}$$

where $C_{i,x,k}$ is the total operational cost for resource *j* to carry out the operation *i*. The resource agent *j* decides to submit a bid for a job, only if its profit for carrying out the operation is not less than its minimal virtual profit, namely, $F_{i,\ j} \geqq F_{min,\ j}$. Each resource agent may produce more than one bid by placing the new entered job in different positions in its buffer list. If the position of any of the existing jobs is changed, it should be examined that the due dates of the existing jobs in the buffer are not violated, and the extra waiting cost are added to the current bid cost of resource *j*. All the bids are collected by the shop agent, and the winner bid with the shortest lead time is selected. Once the winner bids for all operations of a required job are selected, a routing and sequencing plan is generated. The shop agent then evaluates the due date and the total operational cost of the plan. If the due date is not satisfied, or the total cost is not considered low enough, the two set of parameters, the virtual prices for operations and the virtual minimal profits for resources, are adjusted in the next iteration.

A tuning method based on Genetic Algorithms is involved to adjust the parameters. In the crossover process, chromosomes are selected by a crossover probability p_{cro} which is generated randomly, and three alternative crossover operators, point-to-point, single cutoff point and dual cutoff points are used on the selected chromosome pair to reproduce two offspring chromosomes. Next, offspring chromosomes are selected randomly to mutate, based on a mutation probability p_{mut} and the mutation operator $Clone_{(new)} = (1+\mu)\ Clone$, where μ is generated randomly in a given range. The parameters in the offspring chromosomes are then used to coordinate the agent bidding process. If an offspring chromosome is able to result in a plan with lower total operational cost and satisfied due date, it replaces the ordinal one. The set of "winner" resources can be found over GA iterations, and the near optimal solution with the minimum total cost and satisfied production lead time is able to be achieved.

The agent based optimisation mechanism in DIMS is selected for comparison with the proposed mechanism firstly because it involves a similar GA process for gradually updating parameters. Secondly, the objective is to minimise the total production cost with the constraint of lead time, which is the same as the objective for the single-objective problem in the proposed mechanism. Thirdly, in DIMS, each eligible resource agent submits all the possible bids for the bidding operation at the same time if the agent gains a certain amount profit when carrying out it. In the mechanism introduced in this study, on the other hand, there is another way for solving the rescheduling problem in machine buffers. Machine agents receive the knowledge of the information for previous bidding rounds, based on which agents can adjust their buffer jobs and make decisions on whether to submit a new bid. This agent self-adjustment process repeats iteratively until no more bids are received for all job operations. This comparison is expected to examine the effectiveness of the agent self-adjustment process.

For the purpose of comparing the two mechanisms rationally, the model of DIMS is rebuilt and the same simulation data as listed in Section 6.2 are used. In the GA tuning process of the modified DIMS agent bidding mechanism, the chromosome is defined as {P_1, P_2, …, P_n, $F_{min,1}$, $F_{min,2}$, …, $F_{min,m}$}, where P_i represents the virtual price for operation i and $F_{min,j}$ represents the minimal virtual profit for resource j. With the machine data of the system machines shown in Appendix A, it can be easily calculated that the total cost for any machine processing a required job operation is less than 100. Therefore, the initial values of the parameters are randomly generated in the range [0, 100] in order to ensure that the maximum value of the virtual price is greater than the total cost of any job operation processed in the shop. Otherwise, agents always obtain negative profit and consequently no bid is submitted. The emission factor is also considered in the rebuilt DIMS model. A generated chromosome is used to coordinate the agent bidding process. If it is able to result in a routing and sequencing plan with the satisfied due date and the amount of emission released during production, this chromosome is eligible. Otherwise, it will be discarded and a new one is created and evaluated again. Because the main objective of the comparison between the approach of DIMS and the proposed one in this study is to examine the efficiency of the agent self-adjustment process, the same methods and generic parameters for crossover and mutation as described in Chapter 3 are employed.

7.3 Results of Comparison for Single-objective Problem

The experimental results found by agent bidding mechanism in DIMS and the dispatching rule based searching mechanism are compared with that obtained from Test 1 described in Section 6.3.1. Table 7.1 illustrates the comparative results with single objective, i.e., to minimise total production cost. For completing all the 20 jobs the proposed mechanism, DIMS and D.R. acquire 0.43%, 4.43% and 7.82% more production cost compared with the real optimal results calculated by the enumeration method. It can be found that results obtained by the new mechanism are closer to the real optima than those by the other two mechanisms. It can be noticeably shown that for the majority of the required jobs the new mechanism outperforms the mechanism based on dispatching rules in terms of the total production cost in the final scheduling solutions. For the first six jobs, DIMS generates the same schedule plans as the new mechanism does. However, the total production cost in the rest job schedule plans is greater than the new mechanism. For all of the mechanisms in the test, the total production lead time and emission released are satisfied with the corresponding constraints. Additionally, all the three mechanisms can greatly reduce the computational time in comparison to the enumeration method.

	Enumeration Method			New Mechanism		
Runtime	10.6 ($\times 10^4$)			0.61 ($\times 10^4$)		
Job Number	Total Cost	Total Emission	Lead Time	Total Cost	Total Emission	Lead Time
0	762	789	518	762	789	518
1	754	754	489	754	754	489
2	813	727	473	813	727	473
3	1344	1096	765	1344	1096	765
4	803	762	553	803	762	553
5	822	832	512	822	832	512
6	796	841	402	796	841	402
7	774	751	431	774	751	431
8	771	752	469	784	747	476
9	818	809	507	825	833	503
10	827	819	451	838	815	448
11	1230	1119	760	1231	1123	756
12	1131	1269	743	1128	1256	742
13	1172	1201	779	1161	1197	781
14	773	849	428	777	843	420
15	765	881	387	769	877	398
16	780	849	432	778	852	446
17	749	892	473	766	878	483
18	786	840	527	806	866	509
19	1153	1159	828	1169	1130	817
total	**17823**	**17991**	**10927**	**17900**	**17969**	**10922**
	D.R.			DIMS		
Runtime	0.45 ($\times 10^4$)			1.04 ($\times 10^4$)		
Job Number	Total Cost	Total Emission	Lead Time	Total Cost	Total Emission	Lead Time
0	762	789	518	762	789	518
1	754	754	489	754	754	489
2	813	727	473	813	727	473
3	1574	934	754	1344	1096	765

4	846	737	543	803	762	553
5	875	804	508	822	832	512
6	832	817	412	796	841	402
7	814	700	425	782	746	427
8	827	773	487	801	752	456
9	863	819	496	844	805	507
10	882	801	428	859	812	427
11	1471	1089	733	1477	1107	746
12	1305	1017	727	1209	1239	768
13	1231	1175	769	1240	1267	749
14	794	833	431	836	808	428
15	807	906	401	775	870	401
16	798	872	437	791	879	457
17	819	847	448	814	907	469
18	842	839	479	852	884	493
19	1308	1096	823	1239	1155	804
total	**19217**	**17329**	**10781**	**18613**	**18032**	**10844**

Table 7.1 Comparative results for final schedule plans.

The dispatching rule based mechanism in the table performs worse than the other two mainly because it determines the processing route based on weighted combinative dispatching rules. Each local scheduling problem for a single operation or resource buffer is handled independently. Although through the EA process the best combination of different rules is found gradually, a single dispatching rule or a combination of a few rules are comparatively fixed, which cannot produce optimal results for a whole processing line. The other two mechanisms “break” the rules and attempt all potential solutions to search for the optimum. Agent bidding process is involved for determining the allocation of job operations. To decide the priorities of unprocessed jobs in machine buffers, DIMS attempts all various possibilities while the proposed mechanism in this study employs an agent self-adjusting process, with which agents make decisions according to their local utilities. The new mechanism outperforms DIMS implying that the machine agent self-adjusting process is able to contribute to the speed of whole optimum searching process and to produce better solutions to a certain extent. It can be concluded that the proposed agent bidding based mechanism with an agent self-adjusting process is able to provide near-optimal routing and sequencing plans for the required jobs, especially when the problem becomes increasingly complex.

7.4 Results of Comparison for Multi-objective Problem

The dispatching rule based algorithm introduced in Maione and Naso (2001) is used in the comparative experiment for the multi-objective problem. Same input data and setup parameters are used in the comparison tests. The final non-dominated solutions obtained by the models based on the proposed mechanism and the weighted combinative dispatching rules are presented in Figure 7.1. For both algorithms the population size of generic process is 20. The same values of weight variables for determining the preference between cost and emission in the GA/EA fitness functions are used as described in Section 6.3.2.2, namely, *W* is assigned to be different values over the range [0, 1] with the interval 0.05. It is clearly shown that the results obtained by the model of new mechanism introduced in this study dominate the ones found by the model using the dispatching rule based mechanism. In other words, for each solution in the near Pareto-optimal set obtained by the D.R. model, there exists at least one solution found by the proposed model which has better cost and emission attributes.

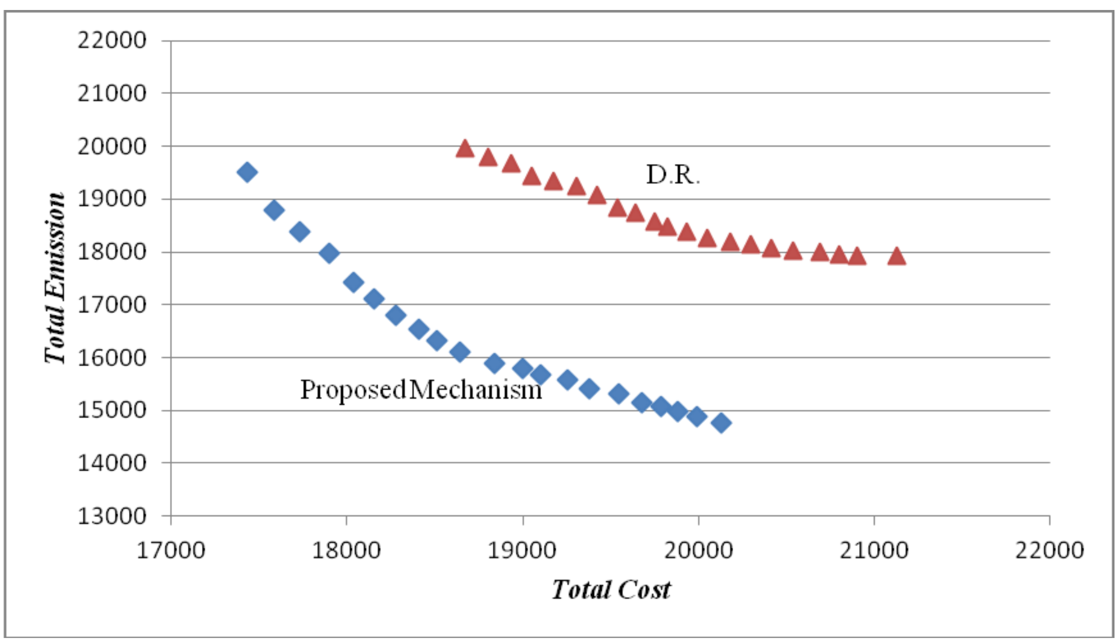


Figure 7.1 Comparative results of obtained non-dominated solutions between New Mechanism and D.R. for all 20 jobs.

Since for each job in the simulation test the searching process runs individually and the final solution points in Figure 7.1 are composed of the summation of the total production cost and total emission for all the 20 jobs, the performance of the combinative results within the searching progress are difficult to measure or compare. Therefore, solutions for the first job are discussed and analysed as an example in the rest of this section. Figure 7.2 shows the two sets of the non-dominated solutions after certain GA/EA generations. The new mechanism takes 17 generations to converge to the final solutions, while the dispatching rule based mechanism converges after 13 generations. As mentioned in the last chapter, there are many standard metrics for measuring the performance of different multi-objective algorithms. In this section, Progress measure (P), Two-set coverage (TSC) and Spacing (SP) metrics, which are introduced in the last chapter, are used to examine the comparative coverage and quality of the obtained non-dominated solutions by the two algorithms.

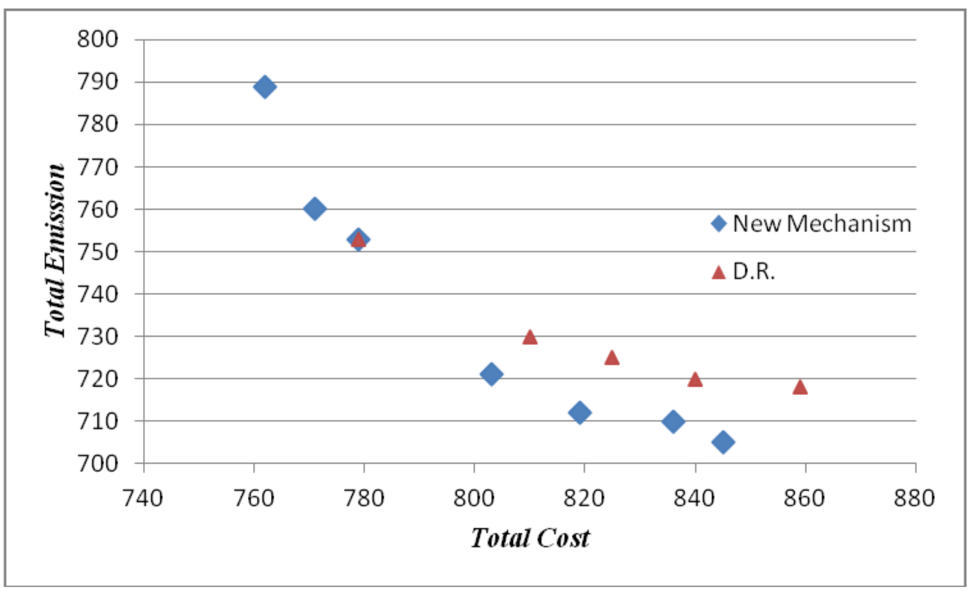


Figure 7.2 Final Set of non-dominated solutions found by the new mechanism and by D.R. for Job Number 0.

Comparative results are listed in Table 7.2. In this experimental test, these two algorithms share the same fitness function, i.e., the weighted sum of total production cost and emission for completing an entire job. Therefore, in the comparison with the Progress metric, the average fitness values at 10th and 17th

GA/EA generations are used to calculate the P values according to (21) for both mechanisms, in order to test the convergence velocity for the two mechanisms in different generic generations. After 17th generic generation, both algorithms have finished searching and solutions have converged. It can be observed from the results that solutions obtained by both algorithms converge over GA/EA generations and the proposed algorithm has a comparatively lower convergent speed. As discussed before, a dispatching rule or combination of dispatching rules are comparatively fixed. The number of possible solutions in the whole solution space obtained by the D.R. algorithm is less than that by the proposed agent bidding and self-adjusting algorithm, which attempts to examine every possible scheduling solution. Accordingly, the possibility of occurrence of new chromosomes created with higher fitness is relatively lower in the D.R. algorithm, which leads to a higher convergence speed in the searching process.

Mechanism	Computation Time (×10^4)	Progress Measure (P)		Spacing (SP)	
		P (10)	*P*(17)	SP(10th)	SP(17th)
New	0.021	0.0769	0.1574	1.318	0.156
D.R.	0.019	0.1201	0.1834	1.039	0.202

Two-Set Coverage (TSC)			
	5th	10th	17th
(New, D.R.)	0.16	0.56	0.75
(D.R., New)	0.80	0.42	0.14

Table 7.2 Comparative results obtained by the new mechanism and D.R. with performance measuring metrics

With TSC metric, the solutions at the 5th, 10th and 17th GA/EA generations found by the two algorithms are compared bilaterally to investigate the relative coverage between two sets of solutions. As the results show, at the 5th generation of the searching process, 80% of the solutions found by D.R. are non-dominated compared with those found by the new mechanism, but at the 17th generation this rate decreased to 14%. On the contrary, although the searching speed is comparatively slow, 75% of solutions obtained by the new mechanism at the 17th generation are non-dominated compared to those found by D.R.. These results suggest that in the early stage of the searching process, the D.R. algorithm applying the combination of certain predesigned dispatching rules overcomes the proposed algorithm in terms of the non-dominated solution coverage. With the increase of the GA/EA generations, however, since the proposed algorithm has a better and better performance compared to D.R. in non-dominated solution coverage until the near Pareto-optimal solutions are found for both algorithms. In other words, in most of the final solutions obtained by the proposed algorithm, the minimisation of both total production cost and released emission amount, namely the two independent objectives, reach a better achievement in comparison with the counterparts with the D.R. algorithm.

In the test with Spacing metric, the solution set at 10^{th} generation and at 17^{th} generation are recorded and compared for both the two mechanisms respectively. In the results, there is no great difference of the SP values between the two algorithms. At 10^{th} generation solutions obtained by the proposed algorithm acquires a higher range variance compared to the D.R.. In other words, the proposed algorithm achieves a poorer solution distribution. At 17^{th} generation, on the other hand, after the solution convergence for both algorithms, the SP value of the proposed algorithm becomes lower, which appears to suggest that the solution points obtained by the proposed algorithm distribute more evenly compared to those by the D.R..

7.5 Discussion on Test Results

The experimental tests and results introduced in the last chapter were further analysed by comparing them with those from two other approaches from the literature, i.e., an agent bidding mechanism in DIMS and a mechanism based on a weighted combination of dispatching rules. After comparison, it could be concluded that, firstly, the proposed iterative agent bidding mechanism and the agent self-adjusting mechanism produced better solutions under a certain testing environment in both single-objective and multi-objective scheduling problems.

When the total production cost was considered as the only objective to be minimised with the constraints of job due date and amount of released emission during production, the proposed approach was able to produce solutions closer to the real-optimal ones than the other two mechanisms with a similar computational time. The agent buffer self-adjusting process was proved to be helpful in searching for optimal or near optimal solutions when the situation in each resource buffer becomes increasingly complex, as new job orders from customers are continuously arrived at the system.

When we considered the minimisation of the total production cost and emission released simultaneously during completing a required job, the agent bidding based mechanism in this study provided near Pareto-optimal solutions with better performance than the modified model based on combinative dispatching rules. Standard performance measuring metrics were used to investigate the convergence speed, relative coverage and solution distribution variance of the obtained results. For the proposed mechanism, although the convergence speed of the optimum or near optimum searching process was not as high as that derived by the dispatching rule based mechanism, it covered a larger non-dominated solution space after certain genetic generations with a similar range variance of solutions in the near Pareto optimal set. As discussed above, the predesigned fixed dispatching rules or a combination of such rules might not be satisfied for a whole processing line in real time production. The mechanism combined with agent coordination and agent local buffer adjustment in this study provided an efficient method to search for satisfied solutions of the routing and sequencing plans in the production scheduling problem.

7.6 Summary

Based on the numerical tests and results introduced in the last chapter, this chapter provided further discussion by comparing the experimental results with two other approaches from the literature. An agent bidding based coordination mechanism in DIMS and a combinative dispatching rule based mechanism were introduced and two models were implemented with the uniform input data of the original tests. Comparative results of the single-objective problem and multi-objective problem were presented and analysed to examine the relative performance of the proposed approach in this study, where three standard metrics were used to examine the quality and convergence of the acquired solutions. Results showed that the new approaches can provide better solutions in both single-objective and multi-objective scheduling problems.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

8.1 Introduction

In this PhD research work, an agent based approach for solving manufacturing production scheduling optimisation problems with the consideration of manufacturing emissions is proposed. In this chapter, conclusions of this study are summarised. Research contributions and limitations of this work are pointed out. Finally, possible future work is suggested.

8.2 Research Conclusions

This PhD research work proposed an agent based approach for optimising manufacturing production scheduling problems. The major objective in such problems is to minimise the overall production cost with the constraints of the technical relationships between jobs or job operations, as well as the capability and capacity limitations of the shared manufacturing resources. Intelligent agent technology is applied in the distributed manufacturing environment. Job orders from customer and manufacturing resources are represented by agents in the model, and another shop agent is responsible for coordinating the interactions among agents. An agent based coordination mechanism is developed, in which Genetic Algorithms are employed to update the coordinative parameters. Based on the coordination mechanism, resource agents submit bids for certain job operations according to the job sequence. This process is coordinated by the system shop agent, who is responsible for the bid announcement, collection and selection iteratively. The performance of acquired routing and sequencing plans is gradually improved over genetic generations for updating parameters. Another agent internal process, called agent self-adjusting process, provides a method for each resource agent to adjust local buffer conditions. Based on the received information of the last bidding round for a certain operation, each capable resource agent rearranges its buffer list, i.e., changes the priorities of the unprocessed jobs waiting to be processed, based on the local utility. A new bid is constructed and submitted again if the resource agent can receive more profit through the adjustment. With agent iterative bidding process and buffer self-adjusting process, the near optimal solutions with the lowest overall production cost and satisfied production lead time can be achieved over iterations.

Manufacturing emissions are involved into the basic scheduling optimisation problems. Among all the methods for reducing manufacturing emissions, the enhancement of the operational efficiency for the whole system can provide practicable and generic approaches for general manufacturers. In this study, the reduction of emission exhausted during production is considered simultaneously with the objective of cost-efficiency in the standard model with two methods. The first method is to assume the emission factor as a constraint. In other words, when a required job is being processed in a manufacturing system, the emission released during production cannot exceed a certain amount. In this context, the main objective of the extended model is still the minimisation of the overall production cost, and solutions must satisfy not only the job delivery due time but also the maximum amount allowed for releasing emissions during production. The other method to consider emissions in the scheduling problem is to assume it as another independent objective. This problem, therefore, becomes a multi-objective optimisation problem, in which the minimisation of both production cost and emission need to be reached simultaneously. An introduced weighted-sum aggregation method, together with the agent based iterative searching algorithm, are able to produce a set of non-dominates solutions which can be selected by the system decision maker according to the relative importance of different objective in different situations.

In order to validate the proposed approaches and to examine their efficiency, numerical tests are introduced. A test case is designed and the proposed algorithms are simulated by Java based programming. Two other approaches with similar objectives from the literature are selected and compared with the new approaches in order to investigate the comparative efficiency of the new algorithms. For the single-objective problem, results show that the new algorithm succeeds in generating optimal or near optimal routing and sequencing plans in a reasonable time, especially with the increase of the problem complexity. For the multi-objective problem, on the other hand, the approximate Pareto Set can be found by the proposed algorithm, which consists of several non-dominated solutions with satisfied total production cost and emission amount.

8.3 Research Contributions and Limitations

The proposed approach represents a contribution to manufacturing production scheduling problems. Using the intelligent agents with the abilities of making autonomous decisions, interaction with others and learning, a searching mechanism is developed for seeking the optimal or near optimal cost-efficient scheduling solutions. The distributed agents are able to observe and learn from the previous system information, and thereafter update local bidding strategies to reach their individual goals, i.e., to maximise their own profit. The agent self-adjusting mechanism, together with the iterative coordination mechanism, is able to provide more efficient solution plans in the manufacturing production scheduling problem.

To the author's best knowledge, the problem studied and approach proposed in this research, for the first time in the literature, consider the manufacturing emissions together with the cost-efficient objective scheduling problems using agent based modelling techniques. A novel multi-objective optimisation algorithm is introduced to minimise the overall production cost and released emission simultaneously with the constraint of production lead time, relationships between job operations and resource capability and capacity.

As aforementioned, agent based techniques can provide a distributed modelling approach to simulate real manufacturing systems with high flexibility and adaptability to unexpected changes. In this study, unexpected changes such as machine breakdown and job order cancelation are not considered and the assumed manufacturing environment is relatively stable. Moreover, as stated in the problem identification, the assumed system is purely distributed, and the limitation of relationships between manufacturing resources is not considered. For instance, materials or jobs may not be able to be transported from one resource in the system to any other resources in the real situation. Another limitation of this research is related to the data collecting of manufacturing emissions. In the experimental tests, the emission values for processing a job operation, machine setup and transporting jobs for each resource are predesigned by the author. In real production situation, however, such data are not easy to be obtained. One potential method to deal with this problem is to record the energy consumption for each resource during production and the emission amount can thereafter be calculated. As mentioned in Chapter 5, the consumption of one kilowatt-hour of electricity leads to 900 grams of carbon dioxide relapsed into the atmosphere.

8.4 Future Work

Based on this PhD research, there are still some open issues that remain to be addressed as useful areas of the future work. The proposed approach employs a simple weighted-sum aggregation method to solve the multi-objective scheduling problem. A future research direction could be the development of a more sophisticated multi-objective optimisation algorithm and adopt it into the proposed mechanism. Multi-objective evolutionary algorithms (MOEAs) such as the Non-dominated Sorting Genetic Algorithm – II (NSGA–II) and Strength Pareto Evolutionary Algorithm 2 (SPEA–2) from the literature seem to be suited to this multi-objective optimisation problem. In these algorithms multiple Pareto-optimal solutions in a single simulation run are captured and similarities of solutions are then exploited through recombination.

Another future research direction corresponds to the extension of the proposed optimisation approach in a purely decentralised system to multi-layer hierarchical manufacturing systems. The distributed framework in this study is able to find near optimal scheduling plans but it does not consider the hierarchical structure of complex manufacturing systems. The proposed agent based approach could be applied in a hierarchical structured system and provide more generic methods to solve the manufacturing scheduling problems.

Furthermore, in this study numerical tests are employed to examine the effectiveness and efficiency of the proposed mechanisms. Real data from industry may be obtained in the future and used to investigate the practicability of the approach.

APPENDIX A

INPUT DATA OF ALL MACHINES

Machine Number	0	Position X:	1	Position Y:	0	
Operation Type	Setup Cost	Setup Time	Process Cost	Process Time	e_{wij}	e_{pij}
0	6.8	2.4	10	7.5	0.3	3.4
1						
2	7.3	2	12.5	6.6	0.3	2.9
3						
4	6.4	2.5	9.8	7	0.3	3.3
5						
6						
7						
8						
9						
Buffer Jobs					Job/Operation	
0	7.3	2	12.5	6.6	N/A (2)	
1	6.4	2.5	9.8	7	N/A (4)	
2	7.3	2	12.5	6.6	N/A (2)	
3						
4						
5						
6						
7						
8						

Table A1. Data of Machine Number 0.

Machine Number	1	Position X:	2	Position Y:	3	
Operation Type	Setup Cost	Setup Time	Process Cost	Process Time	e_{wij}	e_{pij}
0	4.4	1.9	13.6	8	0.2	2.6
1	9.7	3.9	11.2	6.5		
2					0.2	3.0
3						
4	5.8	1.6	10	9.8	0.2	2.9
5						
6						
7						
8						
9						
Buffer Jobs					Job/Operation	
0	5.8	1.6	10	9.8	N/A (4)	
1	5.8	1.6	10	9.8	N/A (4)	
2	7.7	1.9	13.6	8	N/A (0)	
3	9.7	3.9	11.2	6.5	N/A (1)	
4	5.8	1.6	10	9.8	N/A (4)	
5						
6						
7						
8						
9						

Table A2. Data of Machine Number 1.

Machine Number	2	Position X:	3	Position Y:	1	
Operation Type	Setup Cost	Setup Time	Process Cost	Process Time	e_{wij}	e_{pij}
0	8.0	2.4	11.7	7.6	0.5	2.4
1	8.9	4.0	11.8	6.3	0.5	2.4
2						
3	4.7	3.0	8.6	5.9	0.5	3.3
4						
5						
6						
7						
8						
9						
Buffer Jobs					Job/Operation	
0	8.0	2.4	11.7	7.6	N/A(0)	

Table A3. Data of Machine Number 2.

Machine Number	3	Position X:	0	Position Y:	2	
Operation Type	Setup Cost	Setup Time	Process Cost	Process Time	e_{wij}	e_{pij}
0						
1	8.8	3.9	12.7	5.0	0.3	2.8
2						
3	5.6	2.1	8.9	5.5	0.3	3.7
4						
5	7.1	1.7	13.4	7.8	0.3	2.2
6						
7						
8						
9						
Buffer Jobs					Job/Operation	
0	7.1	1.7	13.4	7.8	N/A(5)	
1	7.1	1.7	13.4	7.8	N/A(5)	

Table A4. Data of Machine Number 3.

Machine Number	4	Position X:	1	Position Y:	4	
Operation Type	Setup Cost	Setup Time	Process Cost	Process Time	e_{wij}	e_{pij}
0	6.9	3.1	12.5	9.3	0.7	2.3
1						
2						
3						
4						
5	5.4	3.3	13.0	5.9	0.7	2.2
6						
7						
8						
9						
Buffer Jobs					Job/Operation	
0						

Table A5. Data of Machine Number 4.

Machine Number	5	Position X:	4	Position Y:	6	
Operation Type	Setup Cost	Setup Time	Process Cost	Process Time	e_{wij}	e_{pij}
0						
1	8.5	4.1	12.0	6.3	0.5	2.7
2	7.2	2.4	13.7	5.4	0.5	2.5
3						
4						
5						
6	5.5	2.0	13.5	5.9	0.6	2.5
7	9.0	4.0	18.9	7.1	0.6	2.0
8	15.4	6.9	19.7	7.6	0.6	1.8
9						
Buffer Jobs					Job/Operation	
0	9.0	4.0	18.9	7.1	N/A(7)	

Table A6. Data of Machine Number 5.

Machine Number	6	Position X:	7	Position Y:		7
Operation Type	Setup Cost	Setup Time	Process Cost	Process Time	e_{wij}	e_{pij}
0						
1						
2	6.9	3.3	14.1	5.5	0.3	2.6
3						
4						
5						
6	6.0	1.7	14.6	5.2	0.3	2.6
7	10.3	5.2	19.7	4.4	0.4	2.0
8						
9	17.6	10	21.6	8.8	0.4	1.7
Buffer Jobs					Job/Operation	
0	10.3	5.2	19.7	4.4		N/A(7)
1	10.3	5.2	19.7	4.4		N/A(7)
2	17.6	10	21.6	8.8		N/A(9)

Table A7. Data of Machine Number 6.

Machine Number	7	Position X:	8	Position Y:	5	
Operation Type	Setup Cost	Setup Time	Process Cost	Process Time	e_{wij}	e_{pij}
0						
1						
2						
3						
4						
5						
6	6.0	1.8	15.7	5.1	0.2	2.4
7	10.8	4.9	20.4	4.0	0.2	1.9
8						
9	18.0	9.4	20.1	9.6	0.2	1.9
Buffer Jobs					Job/Operation	
0	6.0	1.8	15.7	5.1	N/A(6)	
1	18.0	9.4	20.1	9.6	N/A(9)	
2	10.8	4.9	20.4	4.0	N/A(7)	
3	6.0	1.8	15.7	5.1	N/A(6)	
4	6.0	1.8	15.7	5.1	N/A(6)	
5	10.8	4.9	20.4	4.0	N/A(7)	

Table A8. Data of Machine Number 7.

Machine Number	8	Position X:	3	Position Y:	9	
Operation Type	Setup Cost	Setup Time	Process Cost	Process Time	e_{wij}	e_{pij}
0						
1						
2						
3						
4	5.5	2.3	13.1	7.8	0.4	3.3
5	7.5	2.1	12.0	6.9	0.4	3.1
6						
7						
8	16.7	7.9	17.6	6.5	0.4	2.7
9						
Buffer Jobs					Job/Operation	
0	16.7	7.9	17.6	6.5	N/A(8)	
1	7.5	2.1	12.0	6.9	N/A(5)	
2	5.5	2.3	13.1	7.8	N/A(4)	
3	5.5	2.3	13.1	7.8	N/A(4)	

Table A9. Data of Machine Number 8.

Machine Number	9	Position X:	6	Position Y:	3	
Operation Type	Setup Cost	Setup Time	Process Cost	Process Time	e_{wij}	e_{pij}
0						
1						
2						
3	6.4	1.9	10.1	4.8	0.3	3.5
4	5.1	1.8	14.7	6.2	0.3	3.0
5	7.7	2.3	15.7	4.6	0.3	2.9
6						
7						
8						
9	13.7	6.1	21.0	6.6	0.2	2.4
Buffer Jobs					Job/Operation	
0	19.7	6.1	21.0	6.6	N/A(9)	
1	19.7	6.1	21.0	6.6	N/A(9)	
2	7.7	2.3	15.7	4.6	N/A(5)	

Table A10. Data of Machine Number 9.

APPENDIX B

MACHINE BUFFER LISTS OVER FIVE SIMULATION DAYS

Buffer List	Day 0		Day 1		Day 2		Day 3		Day 4	
	Job	Type	Job	Type	Job	Type	Job	Type	Job	Type
0	N/A	4	7	0			14	4		
1	0	0	3	2			12	0		
2	1	0	N/A	2			13	0		
3	N/A	2	N/A	2						
4	N/A	2								

Table B1 Buffer lists of Machine No. 0 over 5 days.

Buffer List	Day 0		Day 1		Day 2		Day 3		Day 4	
	Job	Type	Job	Type	Job	Type	Job	Type	Job	Type
0	N/A	4	4	0	9	4				
1	N/A	4	N/A	0						
2	N/A	4	3	1						
3	2	0								
4	N/A	1								
5	0	1								
6	N/A	0								

Table B2 Buffer Lists of Machine No. 1 over 5 days.

Buffer List	Day 0		Day 1		Day 2		Day 3		Day 4	
	Job	Type	Job	Type	Job	Type	Job	Type	Job	Type
0	2	0	4	0			12	3	11	3
1	N/A	0					13	3		
2	1	1					11	3		
3	0	3								

Table B3 Buffer Lists of Machine No. 2 over 5 days.

Buffer List	Day 0		Day 1		Day 2		Day 3		Day 4	
	Job	Type	Job	Type	Job	Type	Job	Type	Job	Type
0	N/A	5					11	3	19	3
1	N/A	5								

Table B4. Buffer Lists of Machine No. 3 over 5 days.

Buffer List	Day 0		Day 1		Day 2		Day 3		Day 4	
	Job	Type	Job	Type	Job	Type	Job	Type	Job	Type
0	0	5	7	0	3	0	14	5	12	0
1			3	0	8	0	12	0	13	0
2					9	5	13	0	11	5
3							11	5	19	5

Table B5. Buffer Lists of Machine No. 4 over 5 days.

Buffer List	Day 0		Day 1		Day 2		Day 3		Day 4	
	Job	Type	Job	Type	Job	Type	Job	Type	Job	Type
0	1	2	5	6	8	7	14	7	15	8
1	N/A	7	N/A	7	9	6			16	8
2			7	7					17	8
3									18	8

Table B6. Buffer Lists of Machine No. 5 over 5 days.

Buffer List	Day 0		Day 1		Day 2		Day 3		Day 4	
	Job	Type	Job	Type	Job	Type	Job	Type	Job	Type
0	2	2	4	2	3	7	11	7	15	6
1	N/A	7	N/A	9			12	7	16	6
2	N/A	7	N/A	7			13	7	17	6
3	N/A	9	3	7					18	6
4									12	7
5									13	7

Table B7. Buffer Lists of Machine No. 6 over 5 days.

Buffer List	Day 0		Day 1		Day 2		Day 3		Day 4	
	Job	Type	Job	Type	Job	Type	Job	Type	Job	Type
0	N/A	6	N/A	7	N/A	6			19	7
1	N/A	9	N/A	6	N/A	6				
2	2	7	N/A	6	10	6				
3	N/A	7	6	8	N/A	7				
4	N/A	6	N/A	7						
5	N/A	6								
6	N/A	7								

Table B8. Buffer Lists of Machine No. 7 over 5 days

Buffer List	Day 0		Day 1		Day 2		Day 3		Day 4	
	Job	Type	Job	Type	Job	Type	Job	Type	Job	Type
0	N/A	8	5	4	10	4	12	8		
1	N/A	5	N/A	4			13	8		
2	N/A	4	N/A	4						
3	N/A	4	6	4						

Table B9. Buffer Lists of Machine No. 8 over 5 days.

Buffer List	Day 0		Day 1		Day 2		Day 3		Day 4	
	Job	Type	Job	Type	Job	Type	Job	Type	Job	Type
0	N/A	9	6	4					15	4
1	N/A	9	N/A	5					16	4
2	N/A	5							17	4
3									18	4

Table B10. Buffer Lists of Machine No. 9 over 5 days.

APPENDIX C

ITERATIVE BIDDING RESULTS

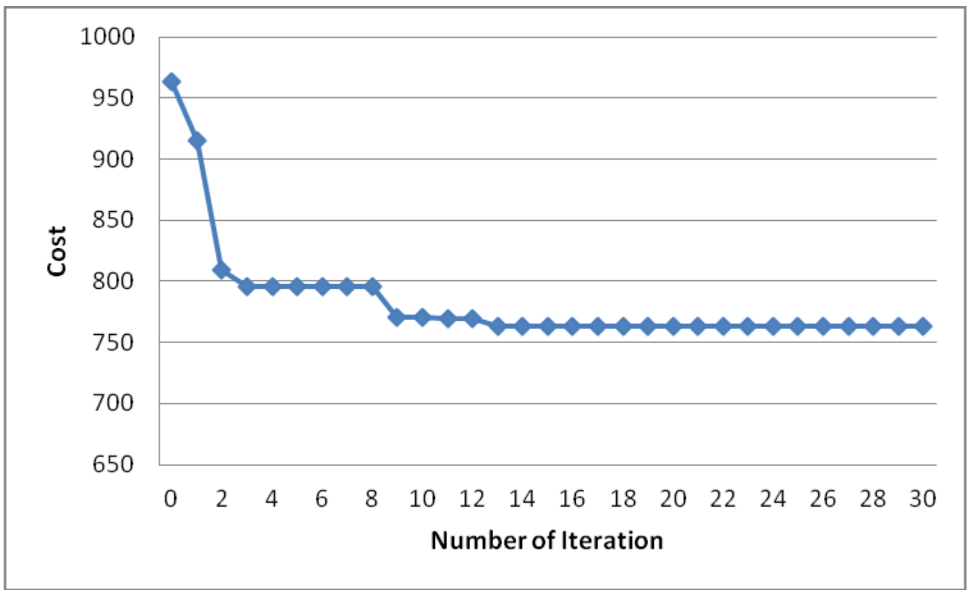


Figure C1 Total Production Cost with Bidding Process over Iterations for Job Number 0.

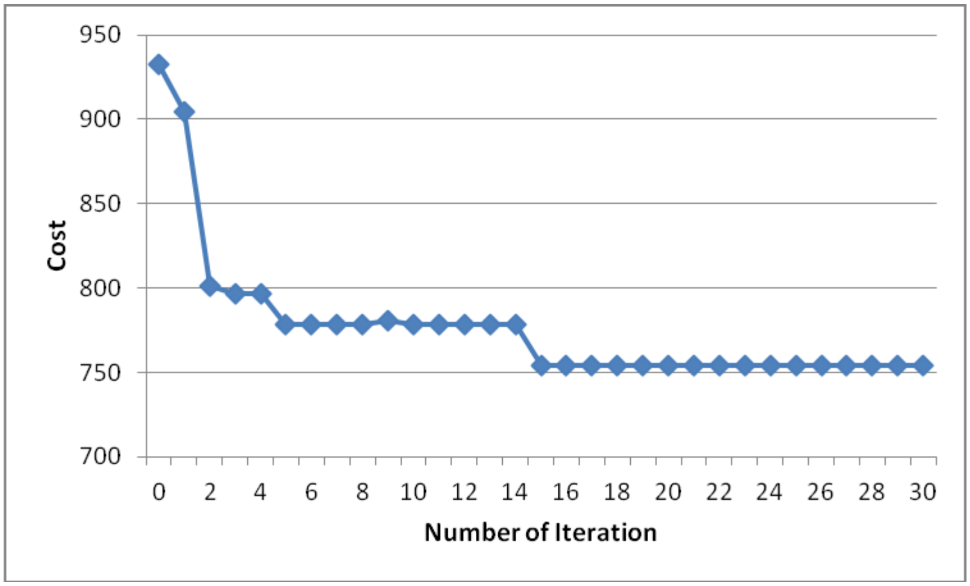


Figure C2 Total Production Cost with Bidding Process over Iterations for Job Number 1.

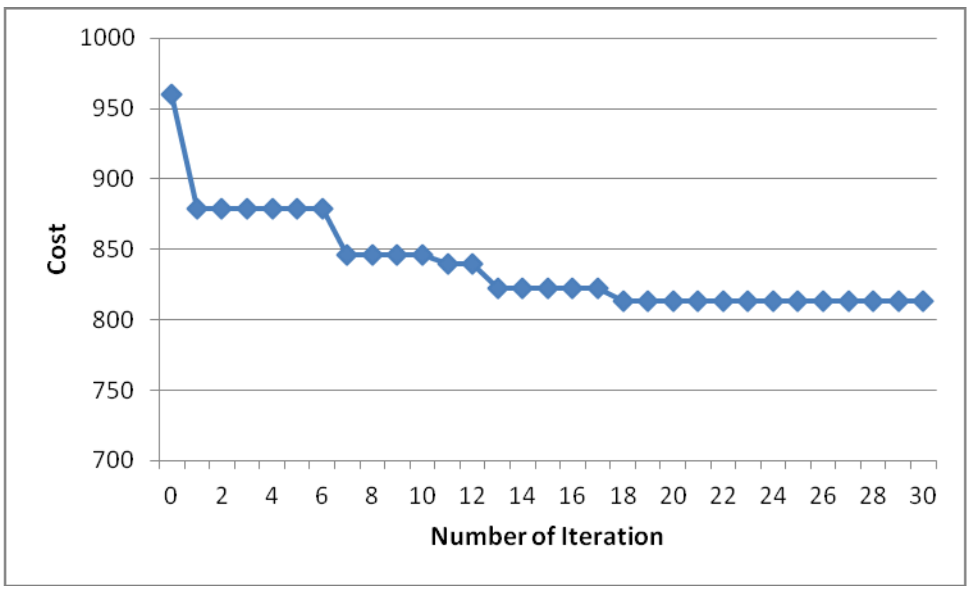


Figure C3 Total Production Cost with Bidding Process over Iterations for Job Number 2.

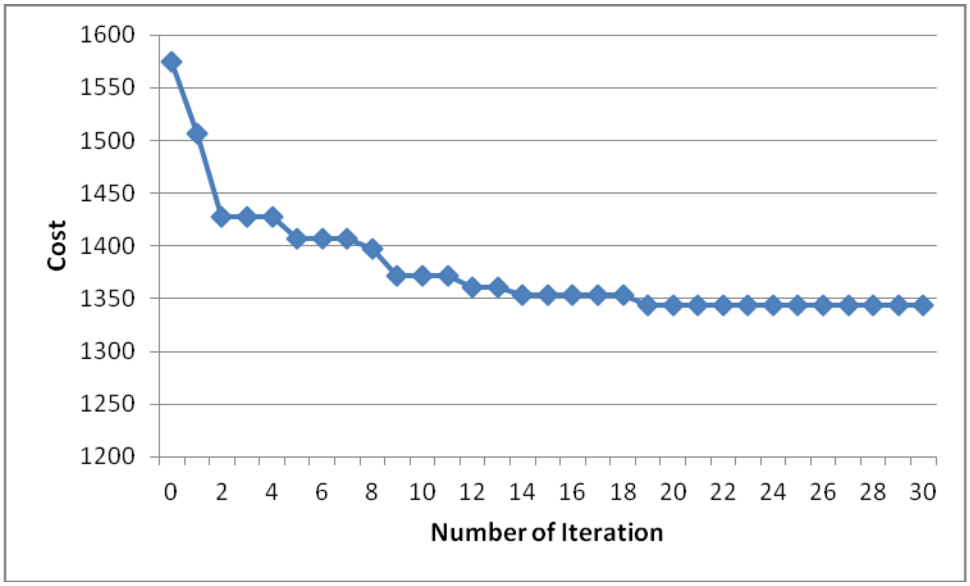


Figure C4 Total Production Cost with Bidding Process over Iterations for Job Number 3.

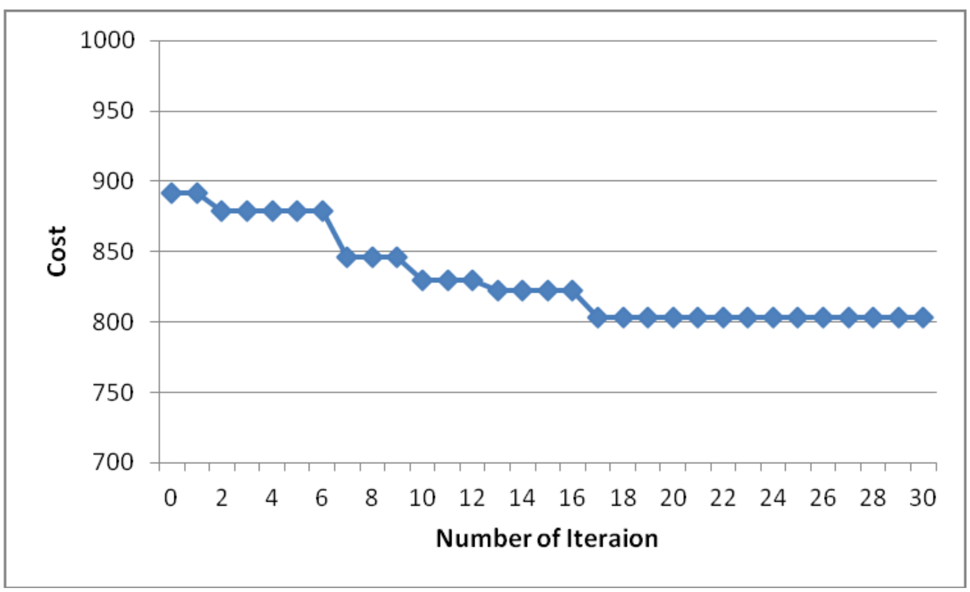


Figure C5 Total Production Cost with Bidding Process over Iterations for Job Number 4.

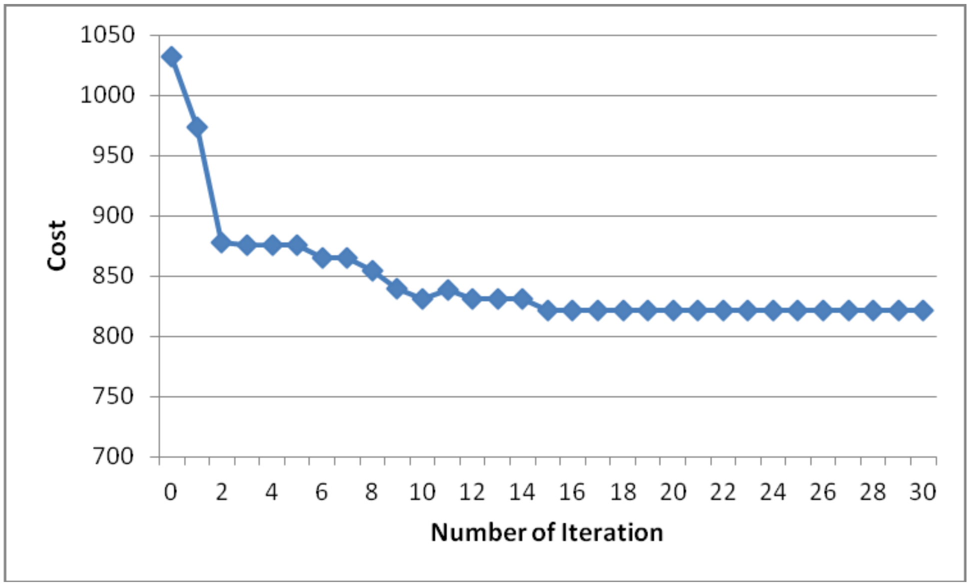


Figure C6 Total Production Cost with Bidding Process over Iterations for Job Number 5.

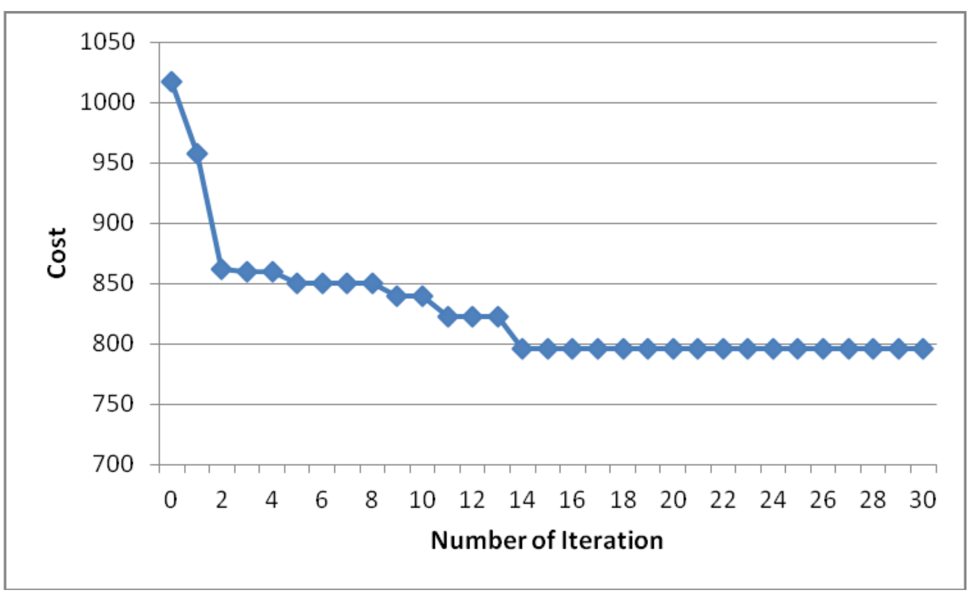


Figure C7 Total Production Cost with Bidding Process over Iterations for Job Number 6.

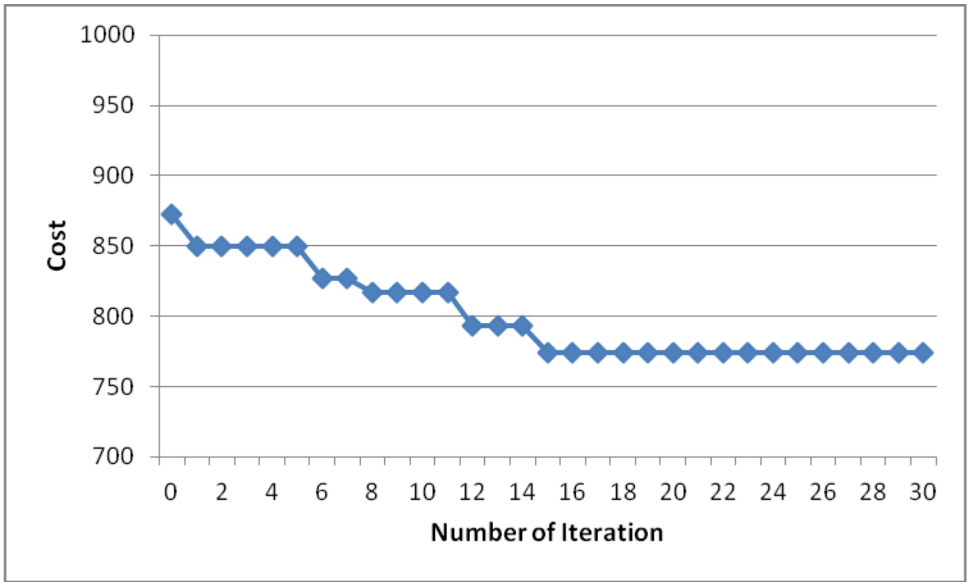


Figure C8 Total Production Cost with Bidding Process over Iterations for Job Number 7.

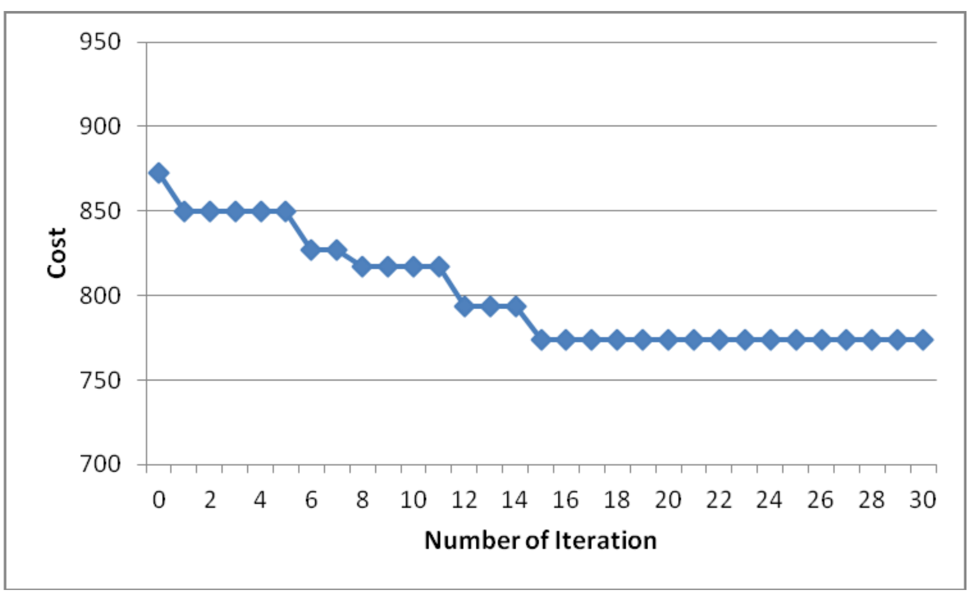


Figure C9 Total Production Cost with Bidding Process over Iterations for Job Number 8.

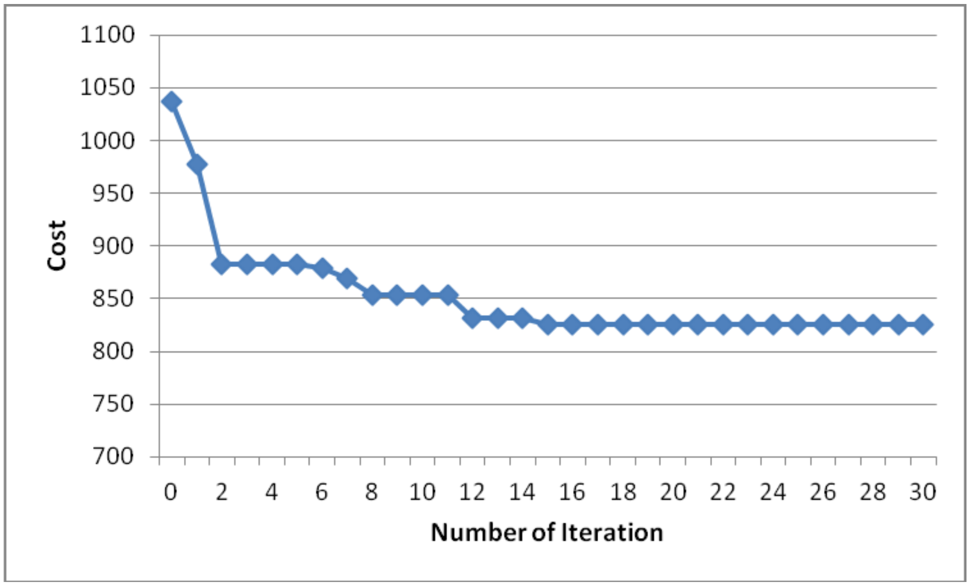


Figure C10 Total Production Cost with Bidding Process over Iterations

for Job Number 9.

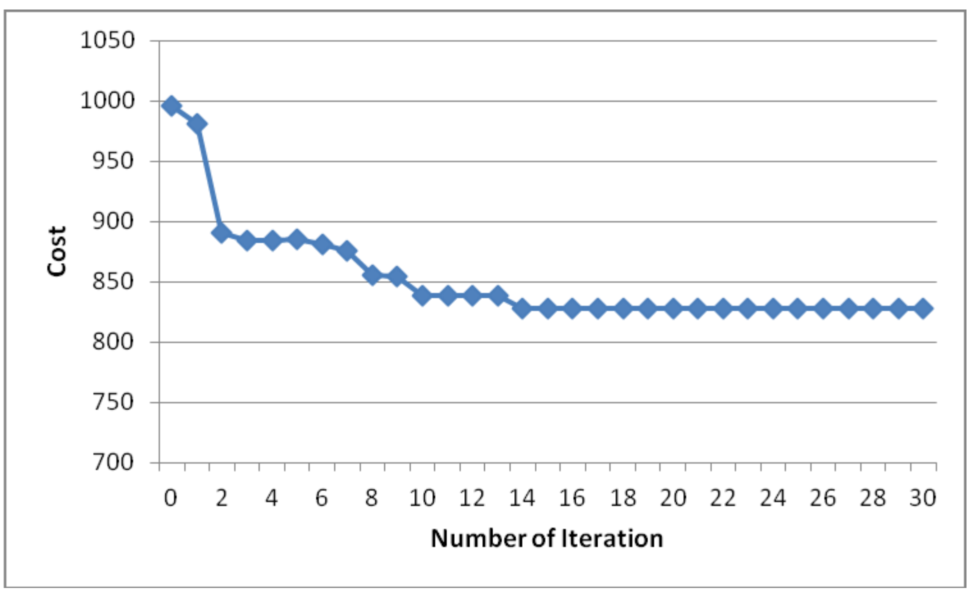


Figure C11 Total Production Cost with Bidding Process over Iterations

for Job Number 10.

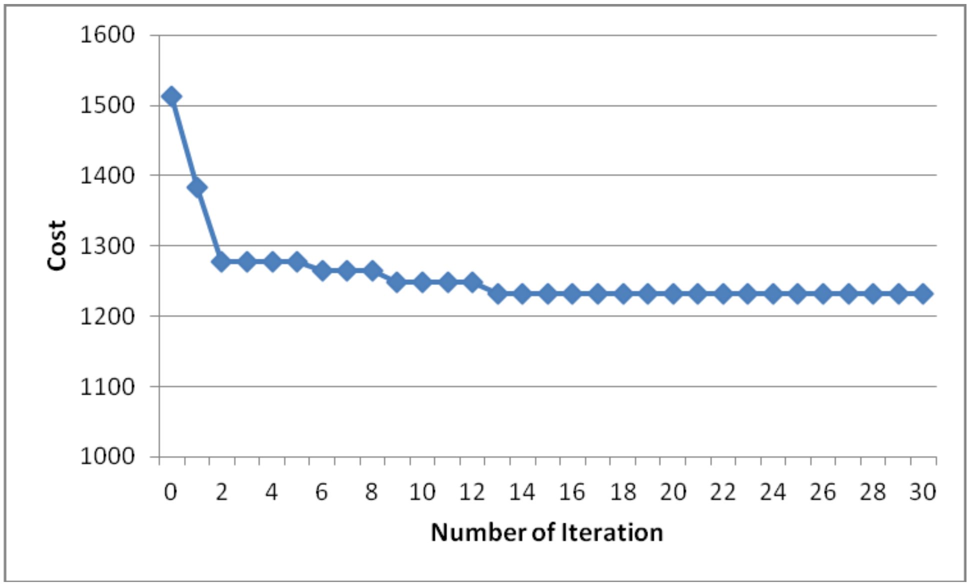


Figure C12 Total Production Cost with Bidding Process over Iterations

for Job Number 11.

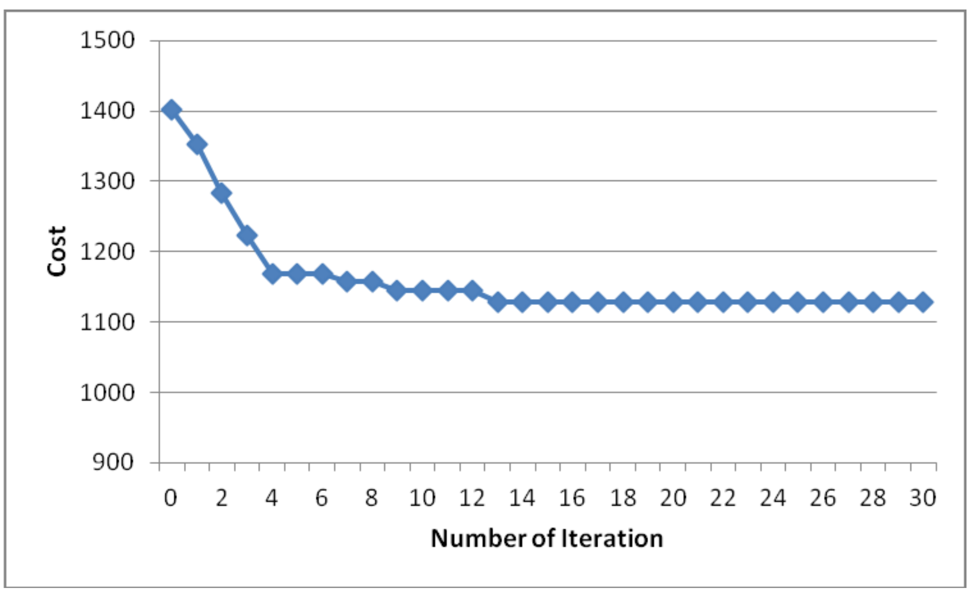


Figure C13 Total Production Cost with Bidding Process over Iterations

for Job Number 12.

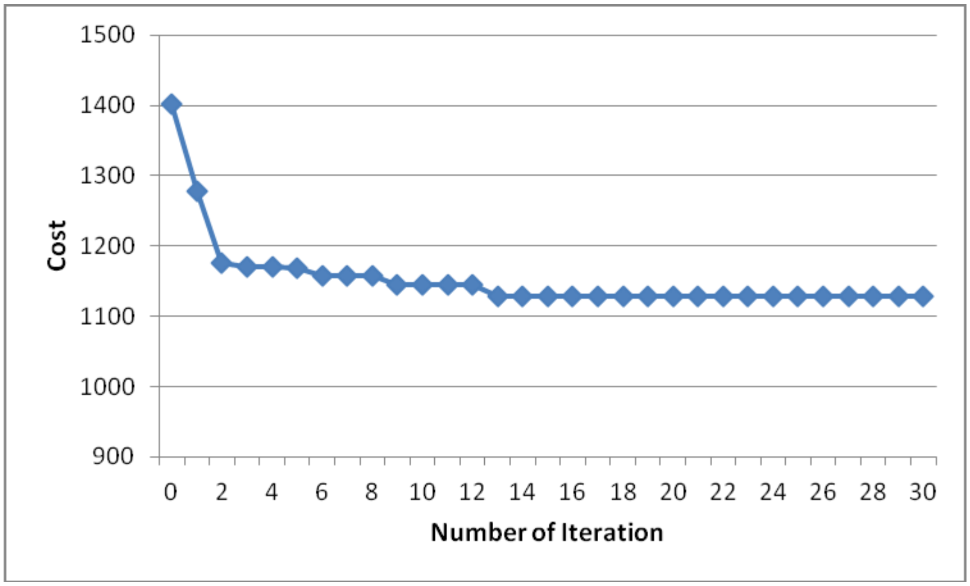


Figure C14 Total Production Cost with Bidding Process over Iterations

for Job Number 13.

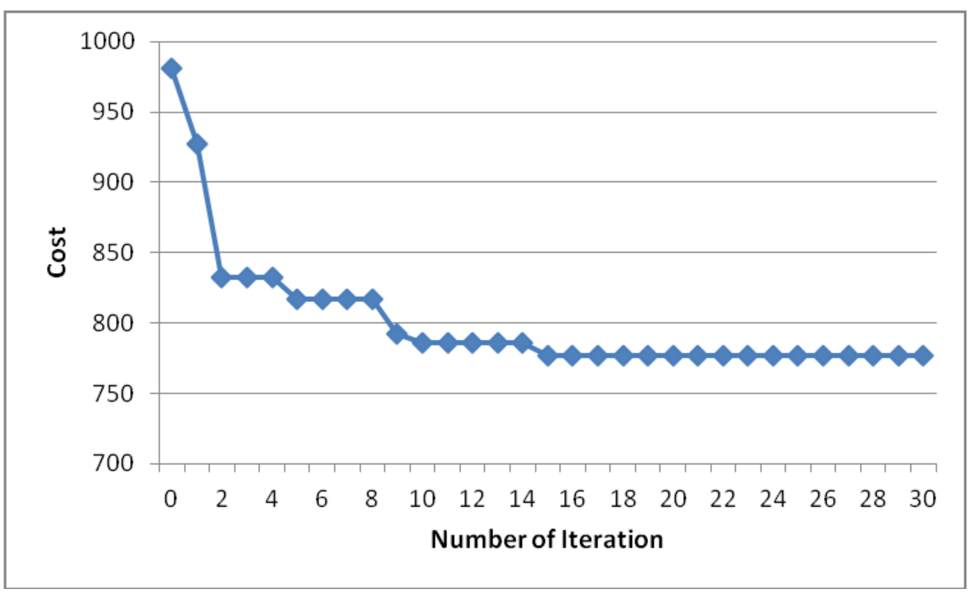


Figure C15 Total Production Cost with Bidding Process over Iterations

for Job Number 14.

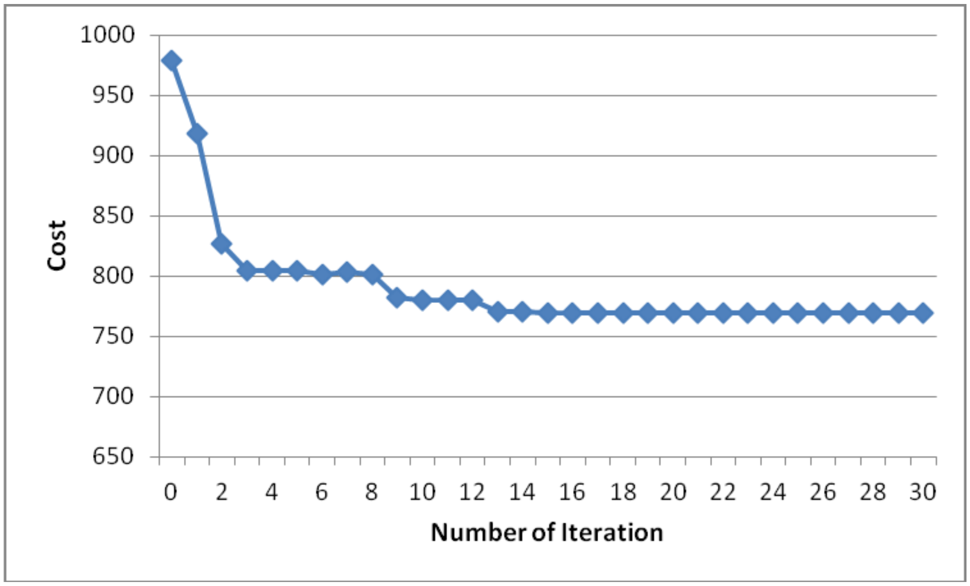


Figure C16 Total Production Cost with Bidding Process over Iterations

for Job Number 15.

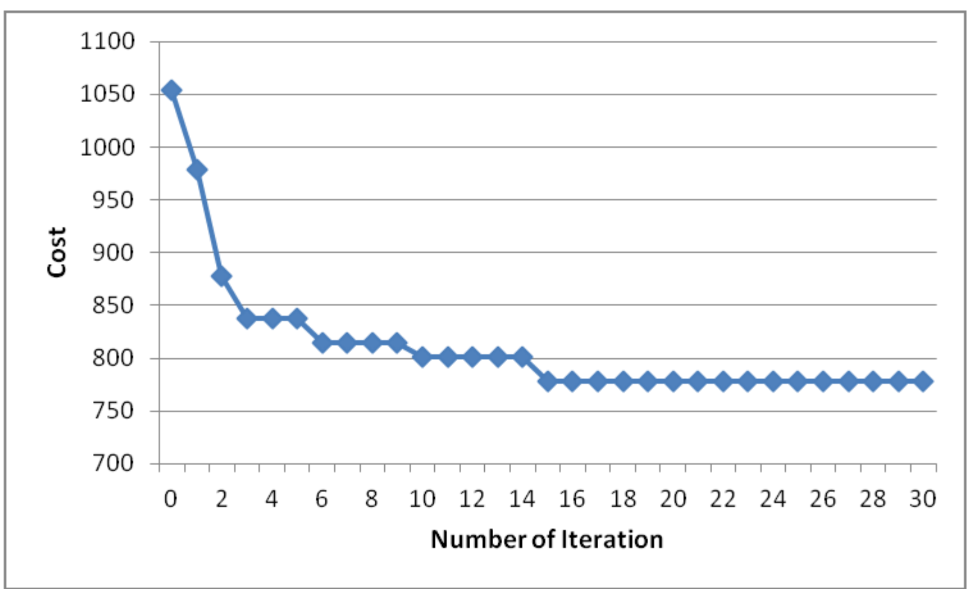


Figure C17 Total Production Cost with Bidding Process over Iterations

for Job Number 16.

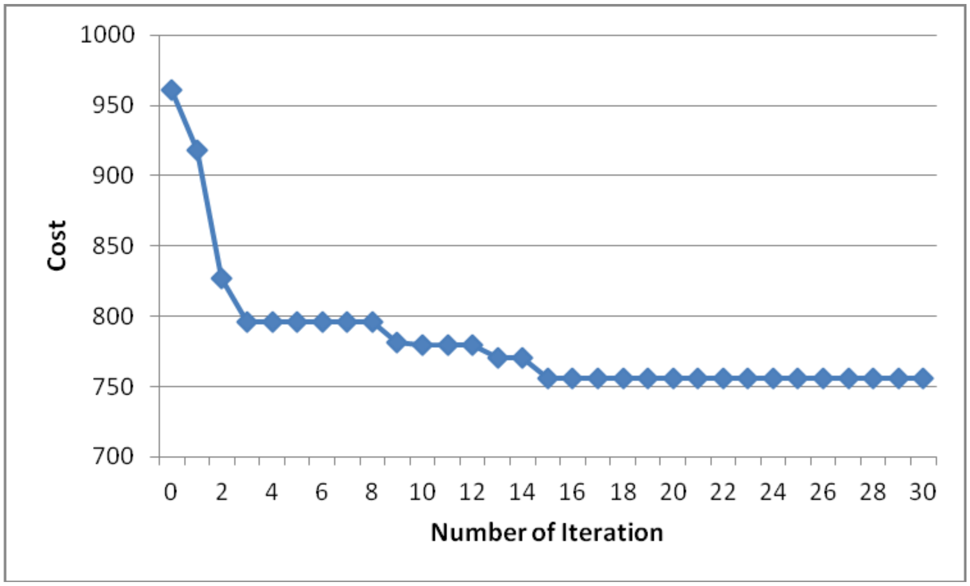


Figure C18 Total Production Cost with Bidding Process over Iterations

for Job Number 17.

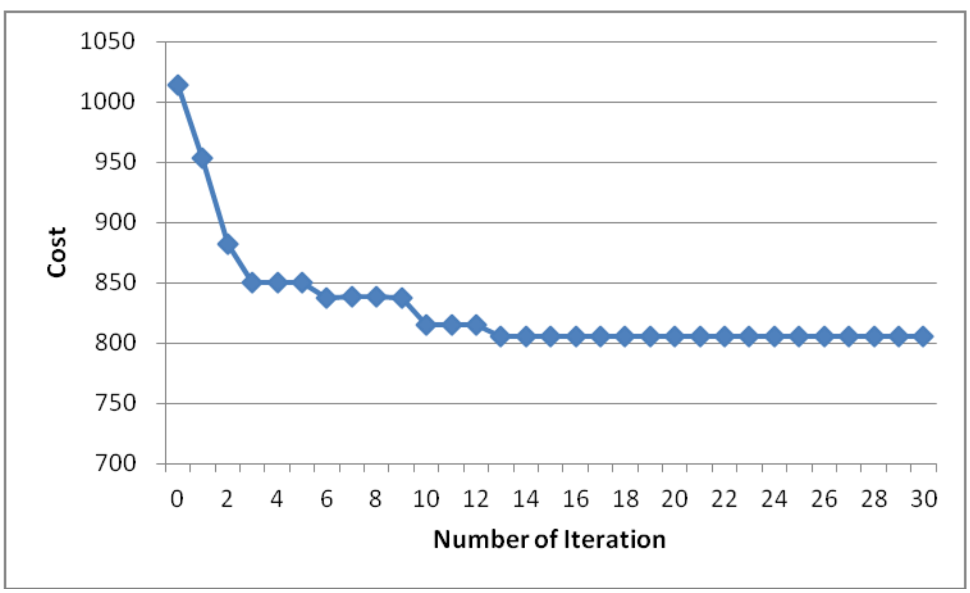


Figure C19 Total Production Cost with Bidding Process over Iterations

for Job Number 18.

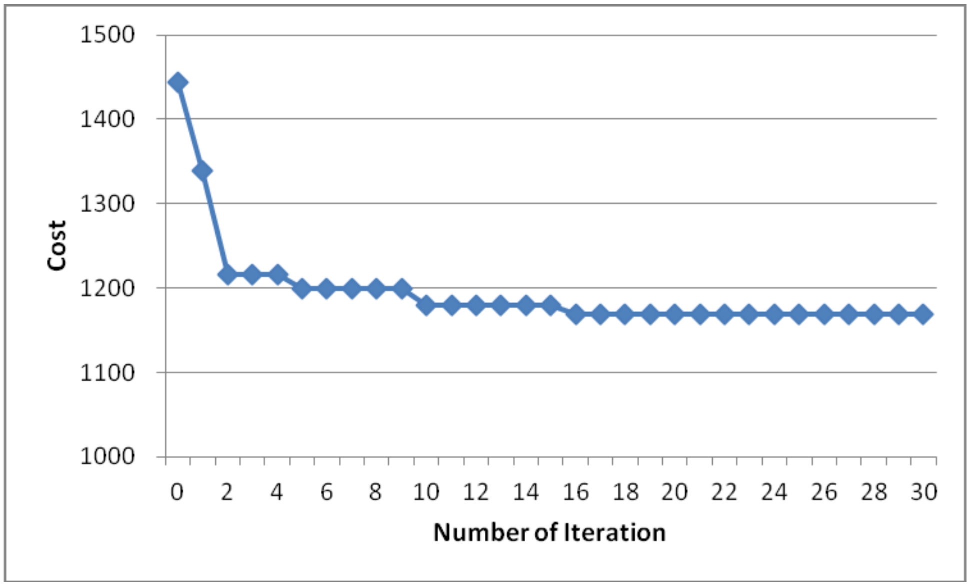


Figure C20 Total Production Cost with Bidding Process over Iterations

for Job Number 19.

APPENDIX D

RESULTS OF MULTI-OBJECTIVE OPTIMISATION

Job Number	Processing Route	Total Cost	Total Emission	Lead Time
0	m2 - m2 - m9 - m8	845	705	537
1	m1 - m1 - m5 - m5	900	589	518
2	m4 - m1 - m0 - m6	912	581	555
3	m4 - m3 - m2 - m0 - m4 - m6	1380	1069	813
4	m2 - m5 - m0 - m6	911	577	602
5	m8 - m9 - m5 - m7	929	663	535
6	m1 - m6 - m8 - m6	917	701	564
7	m2 - m0 - m2 - m5	884	609	498
8	m2 - m1 - m4 - m5	830	584	551
9	m8 - m4 - m6 - m5	918	679	580
10	m8 - m3 - m6 - m5	927	671	523
11	m2 - m3 - m7 - m9 - m3 - m3	1424	991	786
12	m2 - m5 - m4 - m0 - m0 - m6	1253	1074	801
13	m2 - m8 - m1 - m6 - m0 - m7	1296	992	842
14	m1 - m3 - m5 - m8	826	646	594
15	m1 - m7 - m8 - m9	910	653	605
16	m1 - m7 - m8 - m9	932	664	533
17	m0 - m6 - m8 - m9	899	711	514
18	m0 - m6 - m8 - m9	924	657	599
19	m3 - m3 - m5 - m7 - m2 - m4	1314	949	811

Table D1. Results of Optimised Production Plans When W = 0.

Job Number	Processing route	Total Cost	Total Emission	Lead Time
0	m2 - m2 - m9 - m8	845	705	537
1	m1 - m1 - m5 - m5	900	589	518
2	m4 - m1 - m4 - m6	901	585	538
3	m4 - m3 - m2 - m0 - m0 - m6	1368	1073	803
4	m4 - m5 - m0 - m6	905	581	580
5	m8 - m8 - m5 - m7	927	669	516
6	m9 - m6 - m8 - m7	912	707	523
7	m0 - m0 - m4 - m5	879	617	469
8	m0 - m1 - m4 - m5	818	590	570
9	m8 - m3 - m6 - m8	913	682	566
10	m8 - m3 - m6 - m8	925	688	514
11	m3 - m3 - m6 - m9 - m2 - m4	1418	996	725
12	m9 - m8 - m4 - m6 - m0 - m7	1247	1082	768
13	m9 - m8 - m1 - m6 - m0 - m7	1290	999	831
14	m1 - m3 - m5 - m5	821	650	612
15	m0 - m6 - m8 - m9	899	663	600
16	m0 - m6 - m8 - m7	903	672	539
17	m0 - m6 - m8 - m7	886	714	526
18	m0 - m6 - m8 - m7	922	663	574
19	m2 - m3 - m5 - m9 - m9 - m4	1311	966	843

Table D2. Results of Optimised Production Plans When W = 0.05.

Job Number	Processing route	Total Cost	Total Emission	Lead Time
0	m2 - m2 - m9 - m8	845	705	537
1	m1 - m1 - m0 - m7	870	593	536
2	m4 - m5 - m4 - m7	893	588	521
3	m4 - m1 - m2 - m0 - m0 - m7	1363	1077	799
4	m1 - m6 - m1 - m5	901	587	567
5	m0 - m3 - m6 - m8	912	674	541
6	m1 - m6 - m8 - m9	897	714	532
7	m0 - m5 - m4 - m7	871	625	435
8	m4 - m5 - m1 - m6	810	597	548
9	m1 - m3 - m6 - m8	899	689	560
10	m1 - m3 - m6 - m8	909	694	522
11	m9 - m3 - m5 - m9 - m3 - m4	1401	1001	716
12	m3 - m8 - m4 - m5 - m2 - m6	1239	1088	733
13	m3 - m8 - m0 - m0 - m4 - m6	1285	1002	815
14	m1 - m3 - m7 - m8	814	653	617
15	m1 - m6 - m8 - m9	892	670	605
16	m1 - m6 - m8 - m9	899	675	507
17	m0 - m6 - m8 - m6	874	718	496
18	m0 - m6 - m8 - m6	913	666	535
19	m3 - m3 - m5 - m9 - m9 - m3	1297	968	876

Table D3. Results of Optimised Production Plans When W = 0.10.

Job Number	Processing route	Total Cost	Total Emission	Lead Time
0	m2 - m2 - m9 - m8	845	705	537
1	m0 - m1 - m5 - m7	866	598	534
2	m0 - m5 - m4 - m6	889	592	503
3	m1 - m3 - m2 - m0 - m0 - m7	1361	1081	816
4	m2 - m6 - m1 - m7	894	589	545
5	m8 - m3 - m5 - m5	910	679	569
6	m8 - m6 - m5 - m9	895	717	531
7	m0 - m5 - m4 - m7	865	631	415
8	m0 - m5 - m4 - m7	837	601	570
9	m0 - m3 - m6 - m8	886	697	533
10	m1 - m3 - m6 - m8	906	698	512
11	m9 - m3 - m7 - m7 - m3 - m4	1395	1007	708
12	m2 - m8 - m0 - m0 - m4 - m6	1231	1096	714
13	m2 - m8 - m0 - m0 - m4 - m6	1274	1009	798
14	m1 - m3 - m6 - m5	901	658	569
15	m9 - m6 - m8 - m7	888	675	577
16	m9 - m6 - m8 - m7	893	677	510
17	m9 - m6 - m8 - m6	862	721	496
18	m9 - m6 - m8 - m6	901	670	573
19	m3 - m3 - m5 - m9 - m2 - m4	1286	972	892

Table D4. Results of Optimised Production Plans When W = 0.15.

Job Number	Processing route	Total Cost	Total Emission	Lead Time
0	m2 - m2 - m9 - m8	845	705	537
1	m0 - m1 - m6 - m7	861	602	521
2	m2 - m5 - m2 - m6	880	599	483
3	m1 - m3 - m2 - m0 - m4 - m6	1356	1087	796
4	m2 - m6 - m0 - m7	889	597	531
5	m8 - m4 - m6 - m8	904	684	534
6	m8 - m5 - m5 - m9	890	721	498
7	m0 - m5 - m4 - m7	861	636	455
8	m0 - m0 - m4 - m6	834	607	537
9	m1 - m3 - m6 - m8	879	699	514
10	m1 - m3 - m6 - m8	900	702	499
11	m3 - m8 - m7 - m7 - m2 - m3	1389	1011	671
12	m2 - m8 - m1 - m0 - m2 - m6	1227	1099	731
13	m2 - m8 - m0 - m6 - m4 - m6	1269	1013	770
14	m1 - m3 - m6 - m8	894	660	545
15	m0 - m6 - m8 - m9	881	679	567
16	m0 - m6 - m8 - m9	886	683	489
17	m0 - m7 - m8 - m6	857	725	473
18	m0 - m7 - m8 - m6	892	673	556
19	m3 - m4 - m5 - m9 - m3 - m3	1281	976	870

Table D5. Results of Optimised Production Plans When W = 0.20.

Job Number	Processing route	Total Cost	Total Emission	Lead Time
0	m0 - m1 - m3 - m8	836	710	513
1	m2 - m2 - m5 - m6	848	610	507
2	m2 - m0 - m4 - m7	876	609	495
3	m1 - m1 - m0 - m0 - m4 - m5	1349	1094	776
4	m3 - m5 - m2 - m6	877	601	512
5	m8 - m3 - m6 - m5	895	690	517
6	m8 - m5 - m5 - m7	882	728	473
7	m4 - m0 - m2 - m7	856	645	466
8	m4 - m0 - m2 - m7	831	614	535
9	m8 - m3 - m5 - m5	872	710	499
10	m1 - m3 - m6 - m5	892	716	534
11	m9 - m9 - m7 - m7 - m3 - m1	1385	1018	682
12	m3 - m8 - m0 - m0 - m4 - m6	1224	1109	727
13	m3 - m5 - m0 - m6 - m4 - m7	1263	1024	763
14	m0 - m3 - m5 - m8	886	667	530
15	m1 - m6 - m8 - m9	872	689	541
16	m1 - m6 - m8 - m9	880	691	466
17	m1 - m7 - m5 - m6	853	730	490
18	m1 - m7 - m5 - m6	889	682	589
19	m2 - m3 - m6 - m9 - m3 - m4	1279	983	854

Table D6. Results of Optimised Production Plans When W = 0.25.

Job Number	Processing route	Total Cost	Total Emission	Lead Time
0	m0 - m1 - m3 - m8	836	710	513
1	m2 - m2 - m5 - m5	841	619	534
2	m1 - m6 - m0 - m7	868	613	503
3	m2 - m2 - m4 - m0 - m0 - m7	1341	1097	768
4	m1 - m6 - m2 - m5	867	606	541
5	m9 - m9 - m7 - m8	890	695	520
6	m0 - m6 - m5 - m9	874	729	458
7	m4 - m0 - m2 - m6	849	651	413
8	m2 - m0 - m0 - m6	823	620	519
9	m1 - m3 - m6 - m5	864	718	506
10	m1 - m4 - m6 - m5	885	724	517
11	m3 - m4 - m7 - m9 - m3 - m4	1372	1023	732
12	m3 - m5 - m0 - m5 - m4 - m7	1218	1120	748
13	m3 - m5 - m2 - m6 - m2 - m7	1256	1029	756
14	m1 - m3 - m6 - m8	880	671	538
15	m0 - m7 - m8 - m9	866	695	509
16	m0 - m5 - m8 - m9	871	694	460
17	m1 - m7 - m5 - m6	845	737	477
18	m1 - m7 - m5 - m6	873	687	561
19	m2 - m4 - m5 - m9 - m9 - m3	1261	987	837

Table D7. Results of Optimised Production Plans When W = 0.30.

Job Number	Processing route	Total Cost	Total Emission	Lead Time
0	m0 - m1 - m3 - m8	836	710	513
1	m2 - m2 - m5 - m6	838	626	526
2	m4 - m0 - m0 - m6	861	617	487
3	m1 - m1 - m4 - m5 - m0 - m6	1339	1102	741
4	m2 - m6 - m2 - m7	851	609	559
5	m3 - m3 - m6 - m8	880	703	540
6	m1 - m6 - m8 - m9	862	738	426
7	m4 - m0 - m2 - m7	844	657	401
8	m2 - m6 - m2 - m7	820	645	498
9	m8 - m9 - m6 - m5	856	731	513
10	m8 - m4 - m6 - m5	875	735	504
11	m2 - m3 - m7 - m9 - m3 - m4	1368	1027	759
12	m3 - m8 - m1 - m0 - m4 - m7	1213	1126	735
13	m3 - m8 - m0 - m5 - m1 - m6	1248	1035	784
14	m0 - m3 - m6 - m5	877	674	496
15	m0 - m7 - m5 - m9	853	698	527
16	m0 - m6 - m5 - m9	870	701	449
17	m0 - m7 - m5 - m9	841	742	460
18	m0 - m7 - m8 - m6	869	698	535
19	m3 - m3 - m6 - m6 - m2 - m8	1254	998	849

Table D8. Results of Optimised Production Plans When W = 0.35.

Job Number	Processing route	Total Cost	Total Emission	Lead Time
0	m0 - m2 - m3 - m4	819	712	504
1	m0 - m2 - m5 - m5	831	640	531
2	m1 - m1 - m4 - m7	850	622	507
3	m0 - m3 - m4 - m5 - m0 - m7	1351	1081	760
4	m1 - m6 - m4 - m6	839	622	532
5	m9 - m8 - m6 - m5	867	710	574
6	m9 - m6 - m5 - m7	856	745	408
7	m4 - m0 - m2 - m7	830	663	441
8	m4 - m5 - m2 - m6	836	621	469
9	m9 - m9 - m6 - m8	872	719	480
10	m1 - m4 - m7 - m8	891	718	513
11	m3 - m3 - m5 - m9 - m9 - m2	1395	975	744
12	m2 - m8 - m1 - m0 - m0 - m6	1191	1135	702
13	m2 - m8 - m0 - m5 - m1 - m6	1234	1066	810
14	m9 - m4 - m7 - m5	852	698	516
15	m1 - m6 - m5 - m9	834	720	479
16	m1 - m6 - m5 - m9	843	723	437
17	m0 - m7 - m8 - m6	832	757	445
18	m0 - m7 - m8 - m7	854	735	516
19	m9 - m4 - m6 - m6 - m2 - m4	1223	1024	823

Table D9. Results of Optimised Production Plans When W = 0.40.

Job Number	Processing route	Total Cost	Total Emission	Lead Time
0	m0 - m2 - m3 - m4	819	712	504
1	m4 - m1 - m5 - m7	827	643	518
2	m0 - m5 - m1 - m6	846	628	519
3	m2 - m2 - m0 - m5 - m4 - m6	1345	1091	743
4	m1 - m0 - m4 - m7	835	631	519
5	m8 - m8 - m6 - m5	856	716	552
6	m1 - m6 - m5 - m9	851	751	436
7	m1 - m0 - m0 - m7	824	667	418
8	m2 - m5 - m2 - m6	831	623	447
9	m8 - m3 - m5 - m8	867	724	475
10	m0 - m3 - m6 - m8	889	730	501
11	m3 - m3 - m7 - m9 - m2 - m4	1392	986	752
12	m2 - m8 - m1 - m5 - m0 - m7	1189	1137	728
13	m2 - m5 - m0 - m0 - m1 - m6	1230	1069	816
14	m1 - m4 - m6 - m8	849	701	493
15	m0 - m6 - m5 - m9	831	723	465
16	m0 - m6 - m5 - m9	835	728	424
17	m0 - m7 - m5 - m6	826	761	477
18	m9 - m7 - m8 - m6	848	740	494
19	m3 - m3 - m6 - m9 - m2 - m8	1210	1027	850

Table D10. Results of Optimised Production Plans When W = 0.45.

Job Number	Processing route	Total Cost	Total Emission	Lead Time
0	m0 - m2 - m2 - m4	803	721	466
1	m4 - m5 - m5 - m6	816	663	535
2	m4 - m0 - m0 - m6	841	648	497
3	m2 - m1 - m4 - m5 - m4 - m7	1375	1085	804
4	m1 - m6 - m4 - m7	827	669	562
5	m0 - m3 - m5 - m8	849	704	548
6	m0 - m6 - m8 - m9	831	729	409
7	m1 - m6 - m2 - m7	812	687	452
8	m1 - m6 - m2 - m5	823	652	454
9	m8 - m4 - m5 - m8	859	743	498
10	m8 - m9 - m6 - m8	881	727	516
11	m3 - m9 - m7 - m9 - m3 - m4	1386	1021	749
12	m3 - m8 - m1 - m0 - m4 - m6	1179	1169	702
13	m3 - m5 - m1 - m0 - m0 - m6	1224	1070	846
14	m0 - m4 - m7 - m5	833	720	470
15	m1 - m6 - m5 - m9	812	755	419
16	m1 - m6 - m5 - m9	820	708	433
17	m0 - m7 - m8 - m7	813	741	485
18	m0 - m7 - m8 - m7	839	714	550
19	m2 - m4 - m5 - m9 - m2 - m3	1201	1036	861

Table D11. Results of Optimised Production Plans When W = 0.50.

Job Number	Processing route	Total Cost	Total Emission	Lead Time
0	m0 - m2 - m2 - m4	803	721	466
1	m1 - m2 - m5 - m6	801	703	523
2	m1 - m5 - m4 - m5	833	689	488
3	m0 - m1 - m4 - m0 - m4 - m5	1366	1140	824
4	m1 - m0 - m4 - m7	823	712	536
5	m8 - m3 - m5 - m8	841	745	521
6	m8 - m6 - m5 - m9	822	715	417
7	m4 - m5 - m2 - m7	805	679	483
8	m2 - m0 - m2 - m6	809	684	470
9	m1 - m4 - m5 - m5	844	707	516
10	m1 - m9 - m5 - m8	860	704	524
11	m9 - m9 - m7 - m7 - m3 - m4	1371	1034	736
12	m3 - m8 - m0 - m0 - m4 - m6	1162	1161	711
13	m3 - m5 - m0 - m0 - m0 - m6	1218	1103	832
14	m1 - m3 - m6 - m5	825	724	459
15	m8 - m5 - m5 - m9	803	701	413
16	m8 - m5 - m5 - m9	815	709	421
17	m0 - m7 - m8 - m7	801	716	476
18	m0 - m6 - m8 - m7	833	721	538
19	m3 - m4 - m6 - m9 - m3 - m3	1194	1053	877

Table D12. Results of Optimised Production Plans When W = 0.55.

Job Number	Processing route	Total Cost	Total Emission	Lead Time
0	m0 - m2 - m2 - m4	803	721	466
1	m4 - m2 - m5 - m7	793	683	517
2	m1 - m5 - m4 - m6	829	679	500
3	m1 - m1 - m4 - m0 - m0 - m6	1361	1120	819
4	m2 - m0 - m2 - m7	821	682	563
5	m4 - m4 - m6 - m8	837	757	514
6	m8 - m7 - m5 - m9	820	751	439
7	m1 - m5 - m2 - m6	802	674	454
8	m2 - m0 - m2 - m7	796	668	488
9	m1 - m3 - m6 - m8	837	745	523
10	m9 - m9 - m5 - m8	855	719	507
11	m2 - m4 - m7 - m7 - m3 - m3	1358	976	722
12	m3 - m8 - m1 - m0 - m4 - m5	1155	1175	740
13	m2 - m5 - m1 - m0 - m0 - m6	1207	1074	865
14	m5 - m8 - m7 - m5	814	736	436
15	m0 - m7 - m8 - m9	796	764	407
16	m0 - m6 - m8 - m9	804	735	431
17	m0 - m7 - m5 - m9	793	792	486
18	m0 - m6 - m5 - m9	826	801	519
19	m3 - m8 - m6 - m7 - m9 - m9	1186	1082	863

Table D13. Results of Optimised Production Plans When W = 0.60.

Job Number	Processing route	Total Cost	Total Emission	Lead Time
0	m0 - m1 - m3 - m3	779	753	497
1	m1 - m3 - m0 - m6	781	686	504
2	m1 - m0 - m2 - m5	822	673	507
3	m2 - m2 - m4 - m5 - m4 - m7	1359	1036	826
4	m2 - m0 - m0 - m7	824	694	553
5	m8 - m8 - m6 - m5	832	721	509
6	m1 - m6 - m5 - m9	817	726	417
7	m4 - m0 - m1 - m6	798	673	466
8	m2 - m0 - m4 - m6	791	648	490
9	m8 - m3 - m7 - m5	833	739	518
10	m8 - m4 - m5 - m5	856	704	522
11	m3 - m3 - m7 - m7 - m2 - m8	1360	1086	736
12	m2 - m8 - m0 - m0 - m4 - m6	1152	1201	747
13	m3 - m5 - m4 - m5 - m4 - m6	1203	1114	846
14	m0 - m4 - m6 - m8	811	766	433
15	m1 - m6 - m5 - m9	793	809	398
16	m1 - m6 - m5 - m9	801	786	411
17	m0 - m6 - m5 - m9	789	816	503
18	m0 - m6 - m5 - m9	821	805	518
19	m3 - m3 - m7 - m9 - m9 - m4	1185	1093	867

Table D14. Results of Optimised Production Plans When W = 0.65.

Job Number	Processing route	Total Cost	Total Emission	Lead Time
0	m0 - m2 - m2 - m8	771	725	506
1	m0 - m3 - m6 - m5	774	697	492
2	m0 - m5 - m4 - m7	816	680	517
3	m1 - m1 - m4 - m5 - m4 - m6	1352	1015	796
4	m2 - m5 - m1 - m7	811	707	574
5	m9 - m3 - m5 - m8	827	772	519
6	m0 - m6 - m5 - m9	804	761	433
7	m2 - m5 - m4 - m7	791	684	441
8	m2 - m5 - m4 - m5	789	695	478
9	m8 - m4 - m6 - m8	831	780	536
10	m8 - m3 - m5 - m5	852	749	507
11	m2 - m4 - m7 - m9 - m3 - m4	1356	1081	742
12	m3 - m8 - m1 - m0 - m4 - m6	1143	1209	729
13	m3 - m8 - m4 - m5 - m5 - m6	1197	1121	840
14	m1 - m3 - m6 - m8	806	795	423
15	m0 - m6 - m8 - m9	780	824	376
16	m0 - m6 - m5 - m9	792	806	435
17	m0 - m6 - m5 - m9	786	804	490
18	m0 - m6 - m8 - m9	819	819	527
19	m2 - m4 - m7 - m7 - m3 - m3	1178	1081	850

Table D15. Results of Optimised Production Plans When W = 0.70.

Job Number	Processing route	Total Cost	Total Emission	Lead Time
0	m0 - m2 - m2 - m8	771	740	506
1	m0 - m3 - m6 - m5	759	716	489
2	m0 - m5 - m4 - m7	814	699	503
3	m1 - m1 - m4 - m5 - m4 - m6	1348	1028	774
4	m2 - m5 - m1 - m7	807	718	558
5	m9 - m3 - m5 - m8	825	779	497
6	m0 - m6 - m5 - m9	801	806	406
7	m2 - m5 - m4 - m7	779	707	424
8	m2 - m5 - m4 - m5	784	702	501
9	m8 - m4 - m6 - m8	828	790	526
10	m8 - m3 - m5 - m5	846	771	483
11	m2 - m4 - m7 - m9 - m3 - m4	1351	1083	735
12	m3 - m8 - m1 - m0 - m4 - m6	1139	1224	741
13	m3 - m8 - m4 - m5 - m5 - m6	1182	1135	822
14	m1 - m3 - m6 - m8	791	802	413
15	m0 - m6 - m8 - m9	776	839	368
16	m0 - m6 - m5 - m9	789	807	412
17	m0 - m6 - m5 - m9	773	835	472
18	m0 - m6 - m8 - m9	815	838	519
19	m2 - m4 - m7 - m7 - m3 - m3	1176	1104	834

Table D16. Results of Optimised Production Plans When W = 0.75.

Job Number	Processing route	Total Cost	Total Emission	Lead Time
0	m0 - m2 - m2 - m8	771	750	506
1	m0 - m3 - m6 - m5	762	732	475
2	m0 - m5 - m4 - m7	817	703	483
3	m1 - m1 - m4 - m5 - m4 - m6	1350	1071	766
4	m2 - m5 - m1 - m7	811	739	571
5	m9 - m3 - m5 - m8	831	815	504
6	m0 - m6 - m5 - m9	806	819	386
7	m2 - m5 - m4 - m7	783	732	417
8	m2 - m5 - m4 - m5	789	717	498
9	m8 - m4 - m6 - m8	831	804	517
10	m8 - m3 - m5 - m5	844	796	469
11	m2 - m4 - m7 - m9 - m3 - m4	1240	1106	741
12	m3 - m8 - m1 - m0 - m4 - m6	1135	1223	736
13	m3 - m8 - m4 - m5 - m5 - m6	1176	1149	813
14	m1 - m3 - m6 - m8	782	809	400
15	m0 - m6 - m8 - m9	771	834	376
16	m0 - m6 - m5 - m9	783	822	417
17	m0 - m6 - m5 - m9	769	852	465
18	m0 - m6 - m8 - m9	811	842	517
19	m2 - m4 - m7 - m7 - m3 - m3	1172	1120	838

Table D17. Results of Optimised Production Plans When W = 0.80.

Job Number	Processing route	Total Cost	Total Emission	Lead Time
0	m0 - m1 - m2 - m4	762	789	518
1	m0 - m2 - m5 - m5	754	754	489
2	m2 - m6 - m1 - m7	813	727	473
3	m0 - m1 - m1 - m0 - m4 - m6	1344	1096	765
4	m2 - m6 - m1 - m6	803	762	553
5	m8 - m3 - m5 - m5	822	832	512
6	m9 - m7 - m8 - m9	796	841	402
7	m0 - m0 - m4 - m5	774	751	431
8	m4 - m6 - m4 - m5	784	747	476
9	m1 - m4 - m5 - m5	825	833	503
10	m8 - m8 - m7 - m8	838	815	448
11	m3 - m4 - m6 - m6 - m2 - m4	1231	1123	756
12	m2 - m8 - m4 - m0 - m4 - m6	1128	1256	742
13	m2 - m8 - m4 - m0 - m4 - m6	1161	1197	781
14	m0 - m4 - m5 - m5	777	843	420
15	m1 - m6 - m5 - m9	769	877	398
16	m1 - m6 - m5 - m9	778	852	446
17	m1 - m6 - m5 - m9	766	878	483
18	m1 - m6 - m5 - m9	806	866	509
19	m3 - m3 - m7 - m7 - m3 - m4	1169	1130	817

Table D18. Results of Optimised Production Plans When W = 0.85.

Job Number	Processing route	Total Cost	Total Emission	Lead Time
0	m0 - m1 - m2 - m4	762	789	518
1	m0 - m2 - m5 - m6	758	795	486
2	m2 - m5 - m4 - m7	810	748	469
3	m0 - m2 - m4 - m4 - m4 - m7	1321	1185	744
4	m0 - m0 - m2 - m6	806	790	573
5	m8 - m8 - m5 - m5	833	853	509
6	m8 - m6 - m8 - m6	775	892	400
7	m1 - m5 - m4 - m7	769	768	442
8	m1 - m5 - m4 - m7	779	783	481
9	m9 - m9 - m6 - m8	806	842	499
10	m9 - m9 - m6 - m5	824	818	457
11	m3 - m4 - m6 - m6 - m2 - m4	1158	1139	746
12	m3 - m8 - m1 - m5 - m4 - m7	1171	1271	738
13	m3 - m8 - m4 - m5 - m4 - m6	1165	1207	793
14	m0 - m3 - m7 - m8	763	867	417
15	m0 - m8 - m7 - m9	771	890	406
16	m0 - m8 - m7 - m9	780	851	457
17	m0 - m8 - m7 - m9	749	885	473
18	m0 - m8 - m7 - m9	774	873	518
19	m3 - m3 - m6 - m6 - m9 - m3	1161	1143	806

Table D19. Results of Optimised Production Plans When W = 0.90.

Job Number	Processing route	Total Cost	Total Emission	Lead Time
0	m0 - m1 - m2 - m4	762	829	518
1	m0 - m2 - m5 - m6	758	815	486
2	m0 - m6 - m2 - m5	807	765	468
3	m1 - m1 - m2 - m6 - m4 - m7	1315	1160	772
4	m0 - m6 - m2 - m5	814	788	534
5	m8 - m8 - m6 - m5	817	867	517
6	m0 - m6 - m8 - m9	781	887	419
7	m4 - m6 - m1 - m5	742	803	482
8	m4 - m6 - m1 - m5	775	798	449
9	m1 - m3 - m7 - m8	807	879	472
10	m1 - m3 - m6 - m8	786	869	486
11	m2 - m3 - m5 - m9 - m9 - m6	1224	1164	745
12	m3 - m5 - m0 - m2 - m0 - m7	1086	1308	759
13	m3 - m5 - m4 - m0 - m4 - m6	1143	1248	748
14	m1 - m4 - m7 - m5	754	907	478
15	m0 - m6 - m8 - m9	752	921	423
16	m0 - m6 - m8 - m9	747	872	486
17	m0 - m6 - m8 - m9	740	903	507
18	m0 - m6 - m8 - m9	795	877	517
19	m3 - m3 - m7 - m7 - m3 - m4	1180	1137	839

Table D20. Results of Optimised Production Plans When W = 0.95.

Job Number	Processing route	Total Cost	Total Emission	Lead Time
0	m0 - m1 - m2 - m4	748	841	518
1	m0 - m3 - m5 - m6	729	831	491
2	m4 - m3 - m1 - m6	837	773	465
3	m0 - m1 - m1 - m5 - m4 - m7	1319	1254	692
4	m0 - m5 - m2 - m6	767	819	541
5	m8 - m8 - m7 - m5	782	907	527
6	m1 - m7 - m8 - m9	774	843	400
7	m4 - m0 - m0 - m6	789	786	414
8	m4 - m6 - m1 - m7	745	843	468
9	m1 - m8 - m5 - m5	796	912	489
10	m1 - m8 - m7 - m8	831	884	463
11	m3 - m3 - m6 - m9 - m9 - m4	1187	1294	741
12	m9 - m5 - m4 - m0 - m0 - m7	1115	1331	753
13	m9 - m5 - m4 - m0 - m4 - m6	1147	1284	767
14	m1 - m3 - m7 - m8	748	892	419
15	m8 - m6 - m8 - m9	726	957	386
16	m8 - m6 - m8 - m9	741	926	452
17	m8 - m6 - m8 - m9	724	922	475
18	m8 - m6 - m8 - m9	784	918	501
19	m2 - m4 - m7 - m7 - m2 - m3	1146	1282	823

Table D21. Results of Optimised Production Plans When W = 1.00.

REFERENCES

Abumaizar, R.J. and Svestka, J.A. (1997), “Rescheduling job shops under random disruptions”, International Journal of Production Research, Vol. 35, No. 7, pp. 2065-2082.

Adonyi, R., Romero, J., Puigjaner, L. and Friedler, F. (2003), “Incorporating heat integration in batch process scheduling”, Applied Thermal Engineering, Vol. 23, No. 14, pp. 1743-1762.

Agha, H., Thery, R., Hetreux, G., Hait, A. and Lann, J. (2010), “Integrated production and utility system approach for optimizing industrial unit operations”, 21st International Conference, on Efficiency, Cost, Optimization, Simulation and Environmental Impact of Energy Systems, Vol. 35, No. 2. pp. 611-627.

Ahn, H.S., Chen, Y., and Moore, K.L. (2011), “Multi-agent coordination by iterative learning control: Centralized and decentralized strategies”, Intelligent Control (ISIC), 2011 IEEE International Symposium on, pp. 394-399.

Arreola-Risa, A. and DeCroix, G.A. (1998), “Make-to-order versus make-to-stock in a production – inventory system with general production times”, IIE Transactions, Vol. 30, pp. 705-713.

Bäck, T. (1996). “Evolutionary algorithms in theory and practice”, Oxford University Press, New York.

Badr, I. and Gohner, P. (2010), “Incorporating GA-based optimization into a multi-agent architecture for FMS scheduling”, In Proceedings of the 10th IFAC Workshop on Intelligent Manufacturing Systems. Lisbon, Portugal.

Bagchi, T.P. (1999), “Multiobjective scheduling by genetic algorithms”, Norwell, MA: Kluwer.

Balas, E. (1965), “An additive algorithm for solving linear programs with zero-one variables”, Operations Research, Vol. 13, pp. 517-546.

Baskak, M. and Erol, V.A. (2005), “Genetic algorithm approach for solving flexible job-shop scheduling problem”, Proceedings of 35th International Conference on Computers and Industrial Engineering, pp. 245−250.

Bassi, A.M. and Yudken, J.S. (2011), “Climate policy and energy-intensive manufacturing: A comprehensive analysis of the effectiveness of cost mitigation provisions in the American Energy and SecurityAct of 2009”, Energy Policy, Vol. 39, No. 9, pp. 4920-4931.

Bows, A. and Anderson, K.L. (2007), “Policy clash: Can projected aviation growth be reconciled with the UK Government's 60% carbon-reduction target?” Transport Policy, Vol. 14, No. 12, pp. 103-110.

Burke, P. and Prosser, P. (1994), “The distributed asynchronous scheduler”, Intelligent Scheduling, M. Zweben and M.S. Fox, San Francisco, CA: Morgan Kaufmann, pp. 309-339.

Butler, J. and Ohtsubo, H. (1992), "ADDYMS: architecture for distributed dynamic manufacturing scheduling." Artificial Intelligence Applications in Manufacturing: 199-213.

Chan, F.T.S., Zhang, J. and Li, P. (2001), “Modelling of integrated, distributed and cooperative process planning system using an agent-based approach”, in Part B – Journal of Engineering Manufacture, Vol. 215, No. B10, pp. 1437-1451.

Cheng, S.C., Shiau, D.F. and Huang, Y.M. (2009), “Dynamic hard-real-time scheduling using genetic algorithm for multiprocessor task with resource and timing constraints”, Expert System with Application, Vol. 36, No. 1, pp. 852-860.

Commission of the European Communities (2007), “A European Strategies Energy Technology Plan (SET-PLAN): Towards a low carbon future”, COM 723.

Davis, L. (1985), “Job shop scheduling with genetic algorithms”, Proceedings of an

International Conference on Genetic Algorithms and Their Applications, pp. 136-140.

Denkena, B., Tonshoff, H.K., Zwich, M. and Woelk, P.O. (2002), “Process planning and scheduling with multiagent systems”, in Knowledge and Technology Integration in Production and Services – Balancing Knowledge and Technology in Product and Service Life Cycle, Nowwell, MA: Kluwer, pp. 339-348.

Department of Environment (2012), Food and Rural Affairs, UK, Available at: http://www.defra.gov.uk/statistics/environment/green-economy/scptb11-emmbysector/ (Accessed: 28 December 2012)

Dietmair, A. and Verl, A. (2009), “Energy consumption forcasting and optimisation for tool machines”, Modern Machinery Science Journal, pp. 62-67.

Dornfeld, D., Wright, P.K., Wang, F.C., Sheng, P., Stori, J., Sundararajan, V., Krishnan, N. and Chu, C.H. (1999), “Multi-agent process planning for a networked machining service”, Technical Paper NAMRC XXVII, Vol. MS99-175, pp. 1-6.

El Khayat, G., Langevin, A. and Riopel, D. (2006), “Integrated production and material handling scheduling using mathematical programming and constraint programming”, European Journal of Operational Research. Vol. 175, No. 3, pp. 1818-1832.

Eversheim, W. and Schneewind, J. (1995), “Computer-aided process planning – state of the art and future development”, Robotics and Computer-Integrated Manufacturing, Vol. 10, No. 1-2, pp. 65-70.

Ferber, J. (1999), “Multi-agent systems: An introduction to Distributed Artificial Intelligence”, Addison-Wesley, London.

Farret, F.A. and Simoes, M.G (2006), “Integration of alternative sources of energy”, Wiley Online Library.

Finin, T., Fritzon, R., McKay, D. and McEntire, R. (1994), “KQML as an agent communication language”, In the Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94), ACM.

FIPA (1999), "Specification Part 2 – agent communication language", The test refres to the specification dated 16 April 1999.

Fransoo, J.C., Tan, T., de Kok, A.G. and van Ooijen, H. (2010), "Measuring and regulating carbon emissions in transportation", Working Paper, School of Industrial Engineering, Eindhoven University of Technology.

French, S. (1982), "Sequencing and scheduling: an introduction to the mathematics of the job-shop", New York: Wiley.

Gezerman A.O. (2012), "Emission estimation techniques for ammoniumnitrate production", Emission Estimation Techniques for Ammonium Nitrate Production, European Chemical Bulletin, Vol. 1, No. 6, pp. 202-204.

Glover, F.W. and Laguna, M. (1997), "Tabu search", Norwell, MA: Kluwer.

Grandinetti, L., Guerriero, F. Lepera, G. and Mancini, M. (2007), "A niched genetic algorithm to solve a pollutant emission reduction problem in the manufacturing industry: A case study", Computers and Operations Research, Vol. 34, No. 7, pp. 2192-2214.

Groover, M.P. (1987), "Automation, production systems, and computer integrated manufacturing", Prentice-Hall: Englewood Cliffs, N.J.

Gu, P., Balasubramanian, S. and Norrie, D.H. (1997), "Bidding-based process planning and scheduling in a multi-agent system", Computers and Industrial Engineering, Vol. 32, No. 2, pp.477-496.

Guan, Z., Lei, M., Wu, B., Wu, Y. and Yang, S. (1995), "Application of decentralized cooperative problem solving in dynamic flexible scheduling", in Proceedings of SPIE, Bellingham, WA, pp. 179-183.

Gungor, A. and Gupta, S.M. (1999), "Issues in environmentally conscious manufacturing and product recover: A survey", Computers and Industrial Engineering, Vol. 36, pp. 811-853.

Gupta, D and Benjaafar, S. (2004), “Make-to-order, make-to-stock, or delay product differentiation? A common framework of modeling and analysis”, IIE Transactions, Vol. 24, pp. 529-546.

Hajri, S., Liouane, N., Hammadi, S. and Borne, P. (2000), “A controlled geneticalgorithm by fuzzy logic and belief functions for job-shop scheduling”, IEEE Transactions on System, Man, and Cybernetics – Part B: Cybernetics, Vol. 30, No. 5, pp. 812–818.

Handfield, R.B. (1994), “Effects of concurrent engineering on make-to-order products”, IEEE Transactions on Engineering Management, Vol. 41, No. 4, pp. 384-393.

Heragu, S.S., Graves, R. J., Kim, B. I. and Onge, A. S. (2002), “Intelligent agent based framework for manufacturing system control,” IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans, Vol. 32, No. 5, pp. 560–573.

Herrmann, J.W., Eleischer, M., Lin, E. Mathur, V. and Glasser, J. (2001), “Affordable space systems manufacturing: Intelligent synthesis technology, process planning and production scheduling”, presented at the ASME DETC/CIE 2001, Pittsburgh, US, DETC2001/CIE-21281.

Hill, T. (1983), “Production/operations management”, London: Prentice-Hall.

Hitomi, K. (1996), “Manufacturing systems engineering: a unified approach to manufacturing technology, production management, and industrial economics”, London: Taylor & Francis, 2nd Edition.

Hodgson, A.T., Chen, W., Willem H. and Levin H. (2011), "Building product VOC emission testing and conformity assessment procedures in the US", Proceedings of Indoor Air, 1-6.

Holland, J. (1975), “Adaptation in natural and artificial systems”, Ann Arbor: University of Michigan Press.

Huang, S.H., Zhang, H.C. and Smith, M.L (1995), “A progressive approach for the integration of process planning and scheduling”, IIE Transactions, Vol. 27, No. 4, pp. 456-464.

Itoh, S. (2007), “Japan’s energy strategy and development of energy cooperation in the Asia-Pacific”, ERINA Report, Vol. 77, pp. 35-48.

Jacobi, S. Leon-Soto, E. Madrigal-Mora, C. and Fischer, K. (2007), “MasDISPO: A mutiagent support system for steel production and control”, AAAI, pp. 1707-1714.

Jeong, I.J. and Leon, V.J. (2002), “A distributed scheduling methodology for a two-machine flowshop using cooperative interaction via multiple coupling agents”, Journal of Manufacturing Systems, Vol. 21, No. 2, pp. 126-139.

Jeong, K.C. and Kim, Y.D. (1998), “A real-time scheduling mechanism for a flexible manufacturing system: using simulation and dispatching rules”, International Journal of Production Research, Vol. 39, No. 9, pp. 2609-2626.

Kang, K., Yang, Y.Q., Zhang, R.F. and Yang, Y.Q. (2007), “MAS equipped with ant colony applied into dynamic job shop scheduling”, Advanced intelligent computing theories and applications with aspects of artificial intelligence. Springer, Berlin, pp 823–835.

Kempenaers, J., Pinte, J., Detand, J. and Kruth, J.P. (1996), “A collaborative process planning and scheduling system”, Advances in Engineering Software, Vol. 25, No. 1, pp. 3-8.

Khoshnevis, B. and Chen, Q. (1990), “Integration of process planning and scheduling functions”, Journal of Intelligent Manufacturing, Vol. 1, pp. 165-176.

Kim, M. and Kim, Y. (1994), “Simulation based real-time scheduling in a flexible manufacturing system”, Journal of Manufacturing Management Systems, Vol. 13, No. 2, pp. 85-93.

King, D. et al. (July 2011), "Copenhagen and Cancun", International climate change negotiations: Key lessons and next steps, Smith School of Enterprise and the

Environment, University of Oxford, pp. 12.

Kiritsis, D. and Porchet, M. (1996), “A generic petri net model for dynamic process planning and sequence optimization”, Advances in Engineering Software, Vol. 25, pp. 61-71.

Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. (1983), “Optimization by simulated annealing”, Science, Vol. 220, pp. 671-680.

Kouider, A. and Bouzouia, B. (2012), “Multi-agent job shop scheduling system based on co-operative approach of idle time minimization”, International Journal of production Research, Vol. 50, No. 2, pp. 409-424.

Kutanoglu, E. and Sabuncuoglu, I. (1999), “An analysis of heuristics in a dynamic job shop with weighted tardiness objectives”, International Journal of Production Research, Vol. 37, No. 1, pp. 165-187.

Kutanoglu, E. and Sabuncuoglu, I. (2001), “Routing-based reactive scheduling policies for machine failures in dynamic job shops”, International Journal of Production Research, Vol. 39, No. 14, pp. 3141-3158.

Larsen, N.E. and Alting, L. (1990), “Simultaneous engineering within process and production planning”, Pacific Conference on Manufacturing, pp. 1024-1031.

Lee, H. and Kim, S.S. (2001), “Integration of process planning and scheduling using simulation based genetic algorithms”, International Journal of Advanced Manufacturing Technology, Vol. 18, No. 8, pp. 590-606.

Leitao, P. (2008), “Agent-based distributed manufacturing control: a state-of-the-art survey”, Engineering Applications of Artificial Intelligence, Vol. 22, pp. 979-991.

Lenstra, J.K., Rinnooy Kan, A.H.G. and Bruckner, P. (1977), “Complexity of machine scheduling problems”, Annals Discrete Math, Vol. 7, pp. 343-362.

Li, Q., Qu, D. and Du, L. (2008), “Research on hybrid-genetic algorithm for MAS based job-shop dynamic scheduling”, IEEE International Conference on Service Operations and Logistics and Informatics, 2008, IEEE/SOLI, Vol. 2, pp. 1472-1475.

Lim, M.K., Zhang, D.Z. and Goh W.T. (2009), “An iterative agent bidding mechanism for responsive manufacturing”, Engineering Applications of Artificial Intelligence, Vol. 22, No. 7, pp. 1068-1079.

Ma, A.J. and Zhao H.Z. (2012), “Studies on emissions and measures of reduction and control of greenhouse gas during lifecycle of dairy products”, 18th Biennial Conference of International Society for Ecological Modelling, Vol. 13, pp. 2310-2315.

Maione, B. and Naso, D. (2001), “Evolutionary adaptation of dispatching agents in heterarchical manufacturing systems”, International Journal of Production Research, Vol. 39, No. 7, pp. 1481-1503.

Maione, B. and Naso, D. (2003), “A genetic approach for adaptive multi-agent control in heterarchical manufacturing systems”, IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans, Vol. 33, No. 5, pp. 573-588.

Mamalis, A.G., Malagardis, I. and Kanbouris, K. (1996), “Online integration of process planning module with production scheduling”, International Journal of Advanced Manufacturing Technology, Vol. 12, No. 5, pp. 330-338.

Marler, R.T. and Arora, J.S. (2004), “Survey of multi-objective optimization methods of engineering”, Structural and Multidisciplinary Optimization, No. 26, pp. 369-395.

Maturana, F.P. and Norrie, D.H. (1996), “Multi-agent co-ordination using dynamic virtual clustering in a distributed manufacturing system”, Proceedings of the Industrial Engineering Research Conference, pp. 473-478.

Maturana, F.P. and Norrie, D.H. (1999), “MetaMorph: An adaptive agent-based architecture for intelligent manufacturing”, International Journal of Production Research, Vol. 37, No. 10, pp. 2159-2173.

McDonnell, P., Smith, G., Joshi, S. and Kumara, S.R.T. (1999), “A cascading auction protocol as a framework for integrating process planning and heterarchical shop floor control”, International Journal of Flexible Manufacturing Systems, Vol. 11, No. 1, pp. 37-62.

Miyashita, K. (1998), “CAMPS: a constraint-based architecture for multiagent planning and scheduling”, Journal of Intelligent Manufacturing, Vol. 9, No. 2, pp. 147-154.

Morad, N. and Zalzala, A.M.S. (1999), “Genetic algorithms in integrated process planning and scheduling”, Journal of Intelligent Manufacturing, Vol.10, No. 2, pp. 169–179.

Mouzon, G., Yildirim, M.B. and Twomey, J. (2007), “Operational methods for minimization of energy consumption of manufacturing equipment”, International Journal of Production Research, Vol. 45, No. 18-19, pp. 4247-4271.

Muller, J.P. (1998), “Architectures and applications of intelligent agents: a survey”, The Knowledge Engineering Review, Vol. 13, No. 4, pp. 353-380.

National Pollutant Inventory, (1998), “Emissions estimation technique manual for bricks, ceramics and clay product manufacturing”, Environment Australia.

Nejad, H.T.N., Sugimura, N., and Iwamura, K. (2011), “Agent-based dynamic integrated process planning scheduling in flexible manufacturing systems”, International Journal of production Research, Vol. 49, No. 5, pp. 1373-1389.

Nowicki, E. and Smutnicki, C. (1996), “A fast taboo search algorithm for the job shop problem”,Management Science, Vol. 42, pp. 797–813.

Nowicki, E. and Smutnicki, C. (2005), “An advanced tabu search algorithms for the job shop problem”, Journal of Scheduling, Vol. 8, pp. 145–159.

Ou-Yang, C. and Lin, J. (1998), “The development of a hybrid hierarchical/heterarchical shop floor control system applying bidding method in job dispatching”, Robotics and Computer-Integrated Manufacturing, Vol. 14, No. 3, pp. 199-217.

Ouelhadj, D. and Petrovic, S. (2009), “A survey of dynamic scheduling in manufacturing systems”, Journal of Scheduling, Vol. 12, No. 4, pp. 417-431.

Palmer, G.J. (1996), “A simulated annealing approach to integrated production scheduling", Journal of Intelligent Manufacturing, Vol. 7, No. 3, pp. 163-176.

Parunak, V.D. (1987), "Manufacturing experience with contract net", Distributed Artificial Intelligence, M.N.Huhns, Ed. New York: Pitman, pp.285-310.

Pham, D. T. and Karaboga, D. (2000), "Intelligent optimisation techniques: genetic algorithms, tabu search, simulated annealing and neural networks", Springer, London.

Rajendran, C. and Holthaus, O. (1999), "A comparative study of dispatching rules in dynamic flow shops and job shops", European Journal of Operational Research, Vol. 116, No. 1, pp 156-170.

Sabuncuoglu, I. and Gurgun, B. (1996), "A neural network model for scheduling problems", European Journal of Operational Research, Vol. 93, No. 2, pp. 288-299.

Sauser, J., Suelmann, G. and Appelrath, H.J. (1998), "Multi-site scheduling with fuzzy concepts", International Journal of Approximate Reasoning, Vol. 19, pp. 145-160.

Schmidheiny, S. (1992), "Changing course", MIT Press, Cambridge.

Schott, J.R. (1995), "Fault tolerant design using single and multicriteria genetic algorithm optimization", Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Shapiro, A. (1979), "A survey of Lagrangian techniques for discrete optimization", Annals of Discrete Mathematics, Vol. 5, pp. 113-138.

Shaw, M.J. (1988), "Dynamic scheduling in cellular manufacturing systems: A tramework for networked decision making", Journal of Manufacturing Systems, Vol. 7, No. 2, pp. 83-94.

Shen, W., Maturana, F. and Norrie, D.H. (2000), "MetaMorph II: An agent-based architecture for distributed intelligent design and manufacturing", Journal of Intelligent Manufacturing, Vol. 11, No. 3, pp. 237-251.

Shen, W. (2002), “Distributed manufacturing scheduling using intelligent agents”, IEEE Expert/Intelligent Systems. Vol. 17, No. 1, pp.88-94.

Shen, W., Wang, L. and Hao, Q. (2006), “Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey”, IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and Reviews, Vol. 36, No. 4, pp. 563-577.

Sheng, Z., Tang, C. and Lu, C. (2010), “Modeling of agile intelligent manufacturing-oriented production scheduling system”, International Journal of Automation and Computing, Vol. 7, No. 4, pp. 596-602.

Shukla, S.K., Tiwari, M.K. and Son, Y.J. (2008), “Bidding-based multi-agent system for integrated process planning and scheduling: a data-mining and hybrid tabu-SA algorithm-oriented approach”, International Journal of Advanced Manufacturing Technology, Vol. 38, pp. 163-175.

Sikora, R. and Shaw, M.J. (1997), “Coordination mechanisms for multi-agent manufacturing systems: Applications to integrated manufacturing scheduling”, IEEE Transactions on Engineering Management, Vol. 44, No. 2, pp. 175-187.

Smith, D., Frank, J. and Jonsson, A. (2000), “Bridging the gap between planning and scheduling”, Knowledge Engineering Review, Vol. 15, No. 1, pp. 47-83.

Smith, J.S (2003), “Survey on the use of simulation for manufacturing system design and operation”, Journal of Manufacturing Systems, Vol. 22, No. 2, pp. 157-171.

Smith, R.G. (1980), “The contract net protocol: High-level communication and control in a distributed problem solver”, IEEE Transactions on Computers, Vol. C-29, No. 12, pp. 1104-1113.

Song, S., Ren, J. and Fan, J. (2012), “Improved Simulated Annealing Algorithm Used for Job Shop Scheduling Problems”, Advances in Advances in Electrical Engineering and Automation, Vol. 139, pp. 17-25.

Srinivasan, V. (1971), “A hybrid algorithm for the one machine sequencing problem to minimize total tardiness”, Naval Research Logistics Quarterly, Vol. 18, pp. 317-327.

Stecke, K.E. (1985), “Design, planning, scheduling, and control problems of flexible manufacturing systems”, Annals of Operations Research. Vol. 3, No. 4 pp. 1-12.

Stone, P. and Veloso, M. (2000), “Multiagent systems: a survey from a machine learning perspective”, Autonomous Robots, Vol. 8, No. 3, pp. 345-383.

Sugimura, N., Hino, R. and Moriwaki, T. (2001), “Integrated process planning and scheduling in holonic manufacturing systems”, in Proceeding IEEE ISATP2001 – Assembly and Disassembly in the 21st Century, pp. 250-355.

Sundaram, R.M. and Fu, S. (1988), “Process planning and scheduling – a method of integration for productivity improvement”, Computers and Industrial Engineering, Vol. 15, No. 1-4, pp. 296-301.

Swaminathan, V. and Chakrabarty, K. (2003), “Energy-conscious, deterministic I/O device scheduling in hard real-time systems”, IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems, Vol. 22, pp. 847-858.

Tan, W. and B. Khoshnevis (2000), “Integration of process planning and scheduling-a review”, Journal of Intelligent Manufacturing 11(1): 51-63.

Tiwari, A., Ballal, P. and Lewis, F.L. (2007), “Energy-efficient wireless sensor network design and implementation for condition-based maintenance”, ACM Transaction on Sensor Networks, Vol. 3, No. 1, pp. 1-23.

Tonshoff, H., U. Beckendorff and Andres N. (1989), “FLEXPLAN: A concept for intelligent process planning and scheduling”, Proceedings of the CIRP International Workshop, Hanover, Germany, pp. 87-106.

Vieira, G.E., Herrmann, J.W. and Lin, E. (2003), “Rescheduling manufacturing systems: a framework of strategies, policies, and methods”, Journal of Scheduling, Vol. 6, No. 1, pp. 39-62.

Vinicius, A.A. and Debora, P.R., (2012), "Tabu search for total tardiness minimization in flowshop scheduling problems", Computers and Operation Research, Vol. 26, No. 3, pp. 219-235.

Wang, L. and Shen, W. (2003), "DPP: An agent-based approach for distributed process planning", Journal of Intelligent Manufacturing, Vol. 14, No. 5, pp. 429-440.

Wang, Y.H., Yin, C.W. and Zhang, Y. (2003), "A multi-agent and distributed ruler based approach to production scheduling of agile manufacturing systems", International Journal of Computer integrated manufacturing, Vol. 16, No. 2, pp. 81-92.

Wang, X., Ding, H., Qiu, M. and Dong, J. (2011), "A low-carbon production scheduling system considering renewable energy", 2011 IEEE International Conference on Service Operations, Logistics, and Informatics. pp. 101-106.

Wong T.N., Leung C.W. Mak, K.L. and Fung, R.Y.K. (2006), "Integrated process planning and scheduling/ rescheduling – an agent-based approach", International Journal of Production Research, Vol. 44, No. 18-19, pp. 3627-3655.

Wooldridge, M. and Jennings, N. (1995), "Intelligent agents: theory and practice", The Knowledge Engineering Review, Vol. 10, No. 2, pp. 115-152.

Wooldridge, M. (1997), "Agent-based software engineering", In IEE Proceedings on Software Engineering, Nol. 144, No. 1, pp.26-37.

Wooldridge, M. (2000), "An introduction to multiagent systems", Addison-Wesley, Reading, MA.

Wu, D. (1987), "An expert systems approach for the control and scheduling of flexible manufacturing systems", PhD Dissertation, Pennsylvania State University.

Wu, S.D., Storer, R. H. and Chang, P. C. (1993), "One machine rescheduling heuristics with efficiency and stability as criteria", Computers Operations Research, Vol. 20, No. 1, pp. 1-14.

Xiang, W. and Lee, H.P. (2008), "Ant colony intelligence in multi-agent dynamic manufacturing scheduling", Engineering Applications of Artificial Intelligence, Vol. 21, No. 1, pp. 73-85.

Xu, Z. and Randhawa, S. (1996), "Dynamic job shop scheduling in a tool shared environment", Computers and Industrial Engineering, Vol. 31, No. 1/2, pp. 197-200.

Yi, Q., Li, C., Tang, Y. and Want, Q. (2012), "A new operational framework to job shop scheduling for reducing carbon emissions", 8th IEEE International Conference on Automation Science and Engineering, August 20-24, 2012, Seoul, Korea, pp. 58-63.

Zijm, W.H.M. (1995), "The integration of process planning and shop floor scheduling in small batch part manufacturing", CIRP, Vol. 44, No. 1, pp. 429-432.

Zitzler, E., Deb, K. and Thiele. L. (2000), "Comparison of multiobjective evolutionary algorithms: empirical results", Evolutionary Computation, Vol. 8, No. 2, pp. 173–195.

Zhang, D.Z., Anosike, A. and Lim, M.K. (2007), "Dynamically integrated manufacturing systems (DIMS) – a multiagent approach", IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, Vol. 37, No. 5, pp. 824-850.

Zhang, H.C and Merchant, E. (1993), "IPPM – A prototype to integrate process planning and job shop scheduling functions", CIRP, Vol. 42, No. 1, pp. 513-518.

Zhao, C. and Wu, Z. (2001), "A genetic algorithm approach to the scheduling of FMS's with multiple routes", International Journal of Flexible Manufacturing systems, Vol. 13, No. 1, pp. 71–88.

Zhao, F.L., Tso, S.K. and Wu, P.S.Y. (2000), "A cooperative agent modeling approach for process planning", Computers in Industry, Vol. 41, No. 1, pp. 83-97.