An Adaptive Simulation-based Decision-Making Framework for Small and Medium sized Enterprises

Xin Zheng

Faculty of Computing, Engineering and Technology

Staffordshire University

A thesis submitted in partial fulfilment of the requirement of Staffordshire University for the degree of Doctor of Philosophy

May 2011

To my dear parents and sister

Declaration

I declare that this thesis was composed by myself, that the work constrained is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

The following has contributed to some aspects of my thesis:

- X. Zheng, H. Yu, and A. S. Atkins, Visual Simulation-based Hospital Theatre Management System, SKIMA2010, Paro, Bhutan, pp.25-27, August, 2010.
- X. Zheng, H. Yu, and A. S. Atkins, A Simulation-based Dispatching Optimisation Algorithm in Batch Scheduling, SKIMA2009, Fes, Morocc, pp.21-23,October, 2009.
- X. Zheng, H. Yu, and A. S. Atkins, An Overview of Simulation in Supply Chains .Advanced Design and Manufacture to Gain a Competitive Edge, Springer London, 2008, pp. 407-416, ISBN 978-1-84800-240-1.
- X. Zheng, H. Yu and A.S. Atkins, A transferring approach – from a Witness model to a Petri net, Proceedings of the 13th International Conference on Automation and Computing, Stafford, England, 15 September 2007.
- X. Zheng, H. Yu and A. S. Atkins, Inventory Management Simulation using a Novel Replenishment Algorithm, IEEE ICNSC 2007, pp.356-361, London, April, 2007, ISBN 1-4244-1076-2
- X. Zheng, H. Yu and A. S. Atkins, Modelling Supply Chains Using Timed Coloured Petri Nets, SKIMA2006, pp. 8-13,Chiang Mai Thailand, Dec.,2006, ISBN 13:9781851432417
- X. Zheng, H. Yu and A. S. Atkins, Integrated Solution to Reduce Bullwhip Effect in Supply Chain Management,12th CACSC, pp.236-240, Loughborough, Sep. 2006, ISBN 095338909x
- X. Zheng, H. Yu and A. S. Atkins, A supply chain simulation framework for SMEs, Submitted to International Journal of Production Economics, 2011.
- X. Zheng, H. Yu and A. S. **A**tkins, A two-phase optimisation algorithm for a job shop batching scheduling with unrelated parallel machines, setup times, jobs splitting, Submitted to European Journal of Operation Research, 2011.

Abstract

The rapid development of key mobile technology supporting the 'Internet of Things', such as 3G, Radio Frequency Identification (RFID), and Zigbee etc. and the advanced decision making methods have improved the Decision-Making System (DMS) significantly in the last decade. Advanced wireless technology can provide a real-time data collection to support DMS and the effective decision making techniques based on the real-time data can improve Supply Chain (SC) efficiency. However, it is difficult for Small and Medium sized Enterprises (SMEs) to effectively adopt this technology because of the complexity of technology and methods, and the limited resources of SMEs. Consequently, a suitable DMS which can support effective decision making is required in the operation of SMEs in SCs.

This thesis conducts research on developing an adaptive simulation-based DMS for SMEs in the manufacturing sector. This research is to help and support SMEs to improve their competitiveness by reducing costs, and reacting responsively, rapidly and effectively to the demands of customers. An adaptive developed framework is able to answer flexible 'what-if' questions by finding, optimising and comparing solutions under the different scenarios for supporting SME-managers to make efficient and effective decisions and more customer-driven enterprises.

The proposed framework consists of simulation blocks separated by data filter and convert layers. A simulation block may include cell simulators, optimisation blocks, and databases. A cell simulator is able to provide an initial solution under a special scenario. An optimisation block is able to output a group of optimum solutions based on the initial solution for decision makers. A two-phase optimisation algorithm integrated Conflicted Key Points Optimisation (CKPO) and Dispatching Optimisation Algorithm (DOA) is proposed for the condition of $Jm|ST_{si,b}$ with Lot-Streaming (LS). The feature of the integrated optimisation algorithm is demonstrated using a UK-based manufacture case study. Each simulation block is a relatively independent unit separated by the relevant data layers. Thus SMEs are able to design their simulation blocks according to their requirements and constraints, such as small budgets, limited professional staff, etc. A simulation block can communicate to the relative simulation block by the relevant data filter and convert layers and this constructs a communication and information network to support DMSs of Supply Chains (SCs). Two case studies have been conducted to validate the proposed simulation framework. An SME which produces gifts in a SC is adopted to validate the Make To Stock (MTS) production strategy by a developed stock-driven simulation-based DMS. A schedule-driven simulation-based DMS is implemented for a UK-based manufacturing case study using the Make To Order (MTO) production strategy. The two simulation-based DMSs are able to provide various data to support management decision making depending on different scenarios.

Acknowledgments

It is a pleasure to thank the many people who made this thesis possible.

I would like to express my gratitude to my first supervisor, Professor Hongnian Yu for giving me the opportunity to conduct this interesting and challenging research work under his supervision. I am very appreciative of his patient guidance, strong support, motivation, and invaluable advice. I wish to express my warm and sincere thanks to my second supervisor Dr. Tony Atkins for his valuable guidance, instant support, and continuous encouragement. My warm thanks are due to Professor Bernadette Sharp for her kind support and helpful advice. Special thanks go to Mrs. Lynette for her help on writing my thesis.

I also want to thank to the Faculty of Computing, Engineering and Technology of Staffordshire University to provide financial and resources support. I am indebted to my student colleagues for providing a stimulating and fun environment in which to learn and grow. I am especially grateful to Yang Liu, Po Yang, Lizong Zhang, Noppon, and Saisakul. I also like to extend my thanks to the technology support team in Lanner Group for their value support on building the simulation models. I wish to acknowledge the support of EPSRC Collaborative Training Account (CTA) grant to support my work on the UK-based case study and also thank to Dr. Dali Dong for all his contributions and ideas. I also like to extend my thanks to staff at Staffordshire University for directly or indirectly extending their help during the work.

I owe my loving thanks to my dear parents and sister. They have lost a lot due to my research abroad. Without their encouragement and understanding, it would have been impossible for me to finish this work.

Xin Zheng, May, 2011

Table of Contents

List of Figures

List of Tables

Abbreviations

1NF	First Normal Form
2NF	Second Normal Form
3NF	Third Normal Form
ACO	Ant Colony Optimisation
AJSP	Assembly JSP
ASAP	AutoSched AP
ATM	Automated Teller Machine
ATO	Assemble To Order
B&B	Branch and Bound
BBFFL	Bottleneck-Based heuristic
BCO	Bee Colony Optimisation
BIS	Department for Business Innovation &Skills
CDA	Conflicted Disjunctive Arc
CDAS	Set of Conflicted Disjunctive Arcs (definition in Section 4.4.1)
COTS	Commercial Off-The-Shelf
CKPO	Conflicted Key Points Optimisation
CR	Common Resource
CRM	Customer Relationship Management
CRP	Common Resource Pair (definition in Section 4.4.3)
DA	Disjunctive Arc
DAS	Set of Disjunctive Arcs (definition in Section 4.4.1)
DC	Distributor Center
DEDS	Discrete Event Dynamic Systems
DMS	Decision-Making System
DOA	Dispatching Optimisation Algorithm
ECR	Best expecting CRPs (definition in Section 4.4.3)

EDD	Earliest Due Date first
EOQ	Economic Order Quantity
ERD	Earliest Release Date first
ERP	Enterprise Resource Planning
E-SCM	E-Supply Chain Management
ETO	Engineering To Order
FEL	Future Event List
FFL	Flexible Flow Line
FOB	Free On Board
FSMP	Flow Shop with Multiple Processors
GA	Genetic Algorithm
GLS	Genetic Local Search
GPS	Global Positioning System
GRASP	Greedy Randomized Adaptive Search Procedure
HLA	High Level Architecture
HTSA	Hygrid Tabu Search Algorithm
IP	Integer Programming
IPA	Infinitesimal Perturbation Analysis
IRIS	Integrated Refinery In Silicon
IT	Information Technology
JIT	Just-in-Time
JSSP	Job Shop Scheduling Problem
LEs	Large Enterprises
LPT	Longest Processing Time
LR/SF	Likelihood Ratio/Score Function
LS	Lot-Streaming(job split)
MCA	Multiple Comparisons
MCB	Multiple Comparisons with the Best

MCC	Multiple Comparisons with a Control
MCPs	Multiple Comparison Procedures
MIP	Mixed Integer Programming
MES	Manufacturing Execution System
MPS	Master Production Schedule
MRPII	Material Requirements Planning II
MS	Minimum Slack
M-SC	Mobile-Supply Chains
MTO	Make To Order
MTS	Make To Stock
OCR	Occupied CRPs (definition in Section 4.4.3)
ODE	Ordinary Differential Equations
OPT	Optimized Production Technology
OOP	Only One Process (definition in Section 4.4.4)
PA	Perturbation Analysis
PLC	Programmable Logic Controller
PN	Petri Net
PSO	Particle Swarm Optimisation
RCb	Release when Completing blocking
RDA	Released Disjunctive Arc
RDAS	Set of Released Disjunctive Arcs (definition in Section 4.4.1)
RDBMS	Relational Database Management System
RFID	Radio Frequency Identification
ROP	Reorder Points
R&S	Ranking & Selection
RSb	Release when Starting blocking
SA	Stochastic Approximation
SBA	The US Small Business Administration

SBP	Shifting Bottleneck Procedure
SC	Supply Chain
SCM	Supply Chain Management
SCOR	Supply Chain Operations Reference
SCR	all Suitable CRPs (definition in Section 4.4.3)
SMEs	Small and Medium sized Enterprises
SMSC	Small-Medium sized Supply Chain
SIRO	Service in Random Order
SPT	Shortest Processing Time
TCPN	Timed Coloured Petri Net
TQM	Total Quality Management
TS	Tabu Search
TSP	Travelling Salesman Problem
UCR	all Unoccupied (available) CRPs at a time (definition in Section 4.4.3)
VBA	Visual Basic for Applications
WIP	Work In Progress
WSC	Winter Simulation Conference
WSPT	Weighted Shortest Processing Time

Notation List

σ_{di}: the standard deviation of daily usage of family i of units

σ_{Li} :the standard deviation of usage of family i of units during *lead time*

C_{max}: makespan

D: the annual demand

DV: the allowed maximum number of sub-jobs split from a job

$\overline{d_i}$: the average daily demand of family i of units

ECR_{ij}: Best expecting CRPs $\{(m_1,l_1)\ldots.(m_i,l_i)\ \ldots(m_n,l_n)\}\}$ for the operation ij

EL_{ij}: People set of ECR_{ij}

EM_{ij}: Machine set of ECR_{ij}

$E(Z)_i$: the expected number of family i of units short for each order cycle

F_{ij}: the relevant cost function of criterion j in the ith job

H: the holding cost per unit

LE_u: the latest end time of operation u

L_{max}: maximum lateness

L_i: the lead time in days of family i of units

L_{ij}: People set of CR_{ij}

M_{ij}: Machine set of CR_{ij}

Mp(ij): a predecessor operation of operation ij on a machine i task

Ms(ij): a successor operation of operation ij on a machine i task

OCR_{ij}: Occupied CRPs $\{(m_1,l_1)\ldots.(m_i,l_i)\ldots(m_n,l_n)\}$ of the operation ij

Op(ij) : a predecessor operation of operation ij on an operation j task.

$p_{ij:}$ the operation time of the operation ij

P_{order}: the order price

P_{setup}: the setup cost for every order

q_{ij}: the tail time of operation ij

qc_{ij}: the conjunctive tail time of operation ij

R_i: the reorder point of family i of units

r_{ij}: the released time of the operation ij

SCR_{ij}: all Suitable CRPs of the operation ij

T_a: the advance days of booking a order,

Tb_{ij}: the start time of the operation ij

TC: the total cost

Te_{ij}: the end time of the operation ij

Tp_i: the process time of the sub-job using the ith CRP

T_{max}: Maximum Tardiness

Ts_i: the setup time of the sub-job using the ith CRP;

u0: the start node on a critical path

UCR_{ij}: all unoccupied (available) CRPs of SCR_{ij} at the time Tb_{ij}

ui :the operation ui on a critical path

$W_{ij:}$ the weight value of the certain criteria

Z_i :the number of standard deviations for a specified service level of family i of units

Chapter 1 Introduction

1.1 Current Situation and Motivations

In today's competitive environment, enterprises are more likely to enable the integration of information on the entire Supply Chain (SC) and establish competitive advantages on the whole value chain rather than independent enterprise units (Li et al., 2006). In a SC, both Large Enterprises (LEs) and a substantial amount of SMEs are involved. In order to build a highly efficient and effective SC, SMEs are required to adopt Decision-Making Systems (DMS) using IT for responding quickly to up-to-date marketing or customers' demands. However, most vendors of management systems put more attention and effort on LEs and ignore the requirements of SMEs. A recent definition of SME from the European Commission indicates the threshold of SMEs consist of Annual Work Unit (number of employee)<250 and Annual Turnover <= €50m or Annual Balance Sheet Total <= €43m (European Commission, 2003). SMEs play an important role in the global economy (Bruque and Moyano, 2007). In the UK, 59.8% of employment and 49.0% turnover in all private sector enterprises are contributed by SMEs in 2009 (BIS, 2010). In the EU-27, 67.4% of employees are employed in SMEs which account for 58% of total turnover and 54% of the total production value in the non-financial business sector in 2008 (European Commission, 2010) . In the U.S., SMEs account for about 49.6% of employment in the private sector referring to the US' definition of SME (fewer than 500 employees) in 2008 (SBA, 2011). Therefore, the requirements for improving SMEs' core competitiveness on applying IT should not be ignored.

There are several barriers to IT adoption and implementation in SMEs, which are

outlined as follows:

1) SMEs do not have sufficient financial support to buy expensive complex management software, such as Material Requirements Planning II (MRPII), Enterprise Resource Planning (ERP), and Customer Relationship Management (CRM) etc compared to LEs, and operate on small budgets in IT (Morabito et al., 2005). Although, ERP vendors optimise their ERP systems, for example duration reduced from 9 months to 2-3 months, this will still not increase sales from SMEs (Morabito et al., 2005).
2) SMEs lack professional staff or knowledge to implement or maintain software described in 1) above (Morabito et al., 2005).
3) The complex information software is difficult to successfully implement and requires SMEs to change their organisations' structure to fit the demands from the software and gain the 'best practices' (Liang and Xue, 2004).
4) There exists a high risk in adopting information software and failures of implementation can cause financial disaster to these enterprises (Liang and Xue, 2004).
5) Most management systems lack an ability to answer 'what-if' questions.

Therefore, an economic, easy-to-use, DMS which does not need changing the core business processes using IT is required by SMEs.

1.2 Background and Challenging Issues

To design a DMS for SMEs from a SC perspective, there are several requirements which are outlined as follows:

- Limited resources including IT infrastructure, knowledge, staff, and capital need to be considered. Therefore, a framework and structure design are required to achieve the aims of low deployment, maintenance and training costs.
- The easy-to-use and rapid deployment features are critical for adopting and implementing such a system by SMEs. A complex ERP system which needs several months for deployment and altering organisations' business processes or structures could result in a high uncertainty of risks for SMEs (Aloini et al., 2007).
- A SC is dynamic and a flexible manufacturing system is necessary for responding quickly to market changes (Fayez et al., 2005).This leads to the proposed system need to have the characteristics of generalisation and agility.
- The proposed system should be able to answer 'what-if' questions based on different scenarios.

To meet the above requirements of SMEs, simulation modelling methods are required for developing the DMS. Compared to other modelling methods, simulation offers the following advantages (Robinson, 2004; Pidd, 1998):

- Simulation models can be used for repeating 'what-if' analysis depending on various inputs. Managers can set different scenarios such as adding equipments, training employees, etc. Then running a simulation model checks the performances to determine the effect of making these decisions.

- Compared to a mathematical model, a simulation model is easy-to-use and more easily understood by staff without professorial skills.

- Marketing is dynamic and historical value data is available in SMEs. This allows the system to support input data from standard distributions (dynamic marketing) as well as real data from experiences and imitates random events.

- To model and simulate the processes, and to analyse the performance of SMEs in SCs, you need relevant experience and preferably a case study which can enrich the simulation knowledge leading to reduced risks in designing the system.

- A number of commercial simulation software manufacturers support simulation methods and provide good technical support.

The literature review in Chapter 2 indicates that an adaptive simulation framework for SMEs in SCs normally has two characteristics: firstly flexibility configuration and secondly hierarchical structures. Extensive literature review of available frameworks indicated that these two characteristics are not normally integrated in a simulation framework, and a hierarchical network structure which is developed from traditional hierarchical structure is proposed to extend a representation from an internal SC to a complete SC. Thus, a novel simulation framework with flexibility configuration and hierarchical network structure is required for designing a simulation-based Decision-Making System (DMS) for an SME in SCs.

To design a simulation framework for SMEs is a challenging research project. The challenging issues are not only proposing a new methodology and developing from previous methodology based on theory, but also designing a method or logic for programming and implementing these methodologies.

Integrating flexibility configuration and hierarchical network structures into a simulation framework is a difficult challenge. The proposed framework has to be considered in terms of practical implementation. After proposing a framework, a series of methods are developed for designing a simulation-based DMS to fit the framework.

The simulation system is developed using a commercial simulation software package WITNESS which has several advantages and will be discussed in Chapter 2. However, it lacks a capacity of optimising a complicated problem. Therefore, an optimisation block is added into the simulation framework to overcome the lack of optimisations of complicated problems and this block is then able to provide a group of optimum solutions according to a special cost function for decision makers.

To design a simulation-based DMS, there are several challenging issues.

- An optimisation block is required to find a 'good enough' optimum solution for a complicated problem within a reasonable time interval. Normally, after analysing the optimum results, managers make decisions to alter scheduling, invest, or reduce orders etc to improve the performance and maximise the requirements of the customers.
- Flexibility is an important feature of this simulation system and is also a challenging issue. This feature is associated in every aspect of designing this system, such as simulation worldview, simplifying requirements of SMEs and database design techniques.
- The limited resources of SMEs are also a challenge to develop a simulation-based DMS system. Therefore, the simulation framework should consider these limits of SMEs and choose relatively common and economic hardware and software to construct a DMS.

Refining the core business process and simplifying requirements are important to design an easy-to-use flexible simulation-based DMS. As the proposed system includes all kinds of management levels of SCs, it is difficult in practice to incorporate an easy-to-use feature because SCs by nature are complicated. However, if the system is only achieving several special simple functions for an enterprise, it cannot be called an adaptive and flexible DMS. Therefore, refining and simplification are key processes in designing a system. To develop a flexible system, a simulation worldview design is a key process after defining the requirements, because it is a control mechanism of a simulation system in terms of arranging the events of the system. Designing a suitable database is a key task to achieve a flexible characteristic as the database stores the relationships among entities of the system to reduce utilisation of special programming units for an adaptive and flexible approach.

1.3 Aims, Objectives, and Contributions

This research aims at developing an adaptive and flexible simulation-based Decision-Making framework for SMEs in SCs. The proposed framework needs to meet the requirements of SMEs' certain constraints previously described in Section 1.2, such as a limited budget, lack of professional staff, and rapid deployment, etc. In order to satisfy these requirements, SMEs are allowed to design their relative independent unique/particular systems/subsystems according to their own situations. In Section 3.3.4, the concept of simulation block (subsystem) is developed to achieve the aim of designing their independent systems/subsystems. The modularisation and reusability design is introduced into the structure of the framework, which enables SMEs to easily adjust business strategies and policies with minor modifications of their DMSs. The systems/subsystems of an SME are not completely independent; they can cooperate

with each other or other simulation-based DMSs in a SC to provide relevant information for decision making under different scenarios.

SMEs are able to enhance their capability by implementing their DMSs. Firstly, DMSs can identify the bottleneck resources and provide the simulation results based on the solutions provided by employees. Section 6.5 presents this function based on data collected for a March 2008 period of the UK-based manufacturing company and compares the performance factors of two solutions-investment and order control scheduling. Secondly, companies can carry out planning and execution managements, such as inventory management, production scheduling, and order management etc, and adjust their policies to achieve the predefined aim depending on the results provided by their DMSs. A gift SC case study mainly focuses on the order and inventory management and the UK-based SME case study discusses the production scheduling and demand planning (order control) of a job shop. Thirdly, if an SME develops its optimisation block based on the structure of the proposed framework, this block can provide the optimum solution to achieve the predefined target based on the initial solution obtained from the cell simulator. Fourthly, the proposed framework provides a cooperation capability of different simulation-based DMSs implemented in the enterprises for optimising the common resource through a SC.

Different simulation software packages, such as WITNESS, Arena, SIMUL8, etc., and mathematical optimisation methods like the Conflicted Key Points Optimisation (CKPO) and Dispatching Optimisation Algorithm (DOA) are integrated and constructed. This system supports various optimised algorithms to improve the performances of SMEs. To achieve the above aims, the following objectives are identified:

1) To identify the current problems of simulation systems associated with SMEs. Today, DMS is widely applied for enterprises to establish and manage efficient and effective SCs to reduce the total cost and avoid the risk caused by uncertain and unpredicted factors. However, SMEs usually lack DMS expertise and financial support for implementation. The study starts from the current situations of IT system implementation in SMEs and identifies barriers to IT system adoption and implementation by SMEs.
2) To undertake a literature review of current simulation frameworks for SMEs in SCs. Through this review, the challenging issues and limitations of current frameworks and systems are recognised. Following a discussion and analysis of the features of the simulation frameworks, the characteristics of proposed simulation frameworks are identified.
3) To develop a novel decision-making framework to support the decision making of SMEs in SCs. The simulation framework is required to be sufficient for meeting the basic requirements of SMEs. It is also used as a test platform to evaluate and validate the proposed optimised algorithms. This framework is able to provide completed tracking datasets of products and various parameters to monitor the performances of an SME for checking and evaluating the results of optimised algorithms or policies applied. This helps managers to modify and develop algorithms and policies to achieve satisfactory results and performances in order to avoid the unnecessary risks of carrying out a new policy and rules caused by lack of completed evaluations.
4) To investigate and develop the structure of an optimisation block. The integrated CKPO and DOA algorithms in the optimisation block is proposed and developed from a UK-based SME case study. This block is

able to provide a ‘good enough’ optimum solution based on an initial solution from a cell simulator. Managers are able to make decisions depending on the valuable information by analysing the optimum solution under different scenarios.

5) To develop a cell simulator structure. The worldview of a cell simulator is required to be proposed and developed based on the *three phase* approach for meeting the requirements of the UK-based case study. This will also include two event controlling mechanisms for two identified events.

6) To evaluate the simulation framework and demonstrate the integrated method of information flow between the simulation and the optimisation module using case studies. Two simulation-based DMSs are developed based on two case studies which adopt two production strategies, Make To Order (MTO) and Make To Stock (MTS) respectively. An SME based in the UK which operates a simple ERP system was chosen as a case study for the MTO scenario. In order to validate its adaptive characteristic, the proposed MTO DMS will take a series of function testing to determine whether it meets the requirements of a flexible simulation system defined by U.S. Department of Commerce (Umeda, 2004). Data over a 6 months period was filtered, transferred, mined into the database of the simulation–based DMS. A proposed optimisation block integrated CKPO and DOA algorithms was employed to prove the possibility of a test platform and provide a group of optimum solutions for decision makers. All the functions of the simulation-based DMS were tested based on different scenarios to prove the possibility of the simulation system being applied to a real manufacturing enterprise to support decision making.

In order to develop the adaptive and flexible simulation-based DMS for SMEs, there are numbers of emerging knowledge and technologies involved to meet the requirements of SMEs. The contributions of this research work are as follows:

1) A proposed simulation framework with flexibility configuration and hierarchical network structure based on simulation blocks is proposed for accurately representing the relationship among the participants of a SC. A simulation block integrated simulation and optimisation is proposed to solve a complicated optimisation problem. Every simulation block is relatively independent and separated from the data filter and convert filters, which allow SMEs to design their blocks according to their requirements and limitations. This independent feature brings an important advantage for SMEs' users. In fact, simulation models in cell simulators coded by simulation software packages are supported by software vendors with friendly user interfaces and good technical supports for various applications. However, they lack of the ability of mathematical algorithms to implement complicated optimisations. Therefore, an optimisation block is applied to provide optimised services for a special problem based on simulation results. WITNESS is adopted to develop a cell simulator to provide an initial solution under different scenarios. If the initial solution is not good enough, an optimisation block is generated to further analyse data based on the initial solution and provide a group of optimum solutions for decision makers. This process is called simulation-based optimisation.
2) The structure of an optimisation block is investigated and developed. The proposed optimisation block consists of an optimisation algorithm, and performance analysis module. An optimisation block with the proposed

CKPO and DOA is developed to reduce the total tardiness of Jm|STsi,b with LS which is a strongly NP-Hard problem for a flexibility job shop scheduling problem of a UK-based case study. This complicated scheduling problem is divided into two sub-problems. The sub-problem in the top layer is optimised by the proposed CKPO algorithm, which assigns suitable resources for suitable jobs at suitable time points. The sub-problem in the bottom layer is handled by the proposed DOA algorithm, which splits a job and takes the suitable quantity of a job to every chosen common resource, such as machine and labour. The optimisation block integrated CKPO and DOA algorithms is approved to be able to provide the relative stable best optimum results by validations using six months' real data from a UK-based case study.

3) The entire processes of designing a cell simulator based on the adaptive structure of a cell simulator are developed. A new approach of simulation worldview is proposed to achieve a flexible purpose of the cell simulator developed from a *three phase* approach. This proposed approach also contributed to design other cell simulators.
4) Two case studies are conducted for two types of DMSs which adopt two common production strategies, namely MTO and MTS, respectively. The MTO case study is an efficient UK-based case study using six months data inputted into the database of the simulation DMS and several functions are tested after running the simulation operation. Then the optimisation block with proposed optimisation algorithms is used to evaluate the performances of optimisation under various parameter settings.
5) The proposed 4P's closed-loop and multi-layer control concepts are applied in designing the framework and structure of a simulation-based

DMS. The four concepts help the system maintain a fine tuning mechanism in a dynamic environment and scope the requirements of different groups.

6) Verifying feasibility of a scheduling of UK-based case study is investigated. A verification program is developed to verify the solution from a cell simulator or an optimisation block.

7) A flexible and adaptive simulation-based DMS is developed. The performances of SMEs are shown after running simulations based on different scenarios and are critical support data for management decisions. Then if necessary, an optimisation block can provide further analysing and optimisation services for decision makers.

1.4 Research Methodology

Figure 1.1 shows the structures of the research 'onion' consisting of different layers and research approaches which are commonly adopted for research. The following sections are outlined according to the classifications of the research 'onion'.

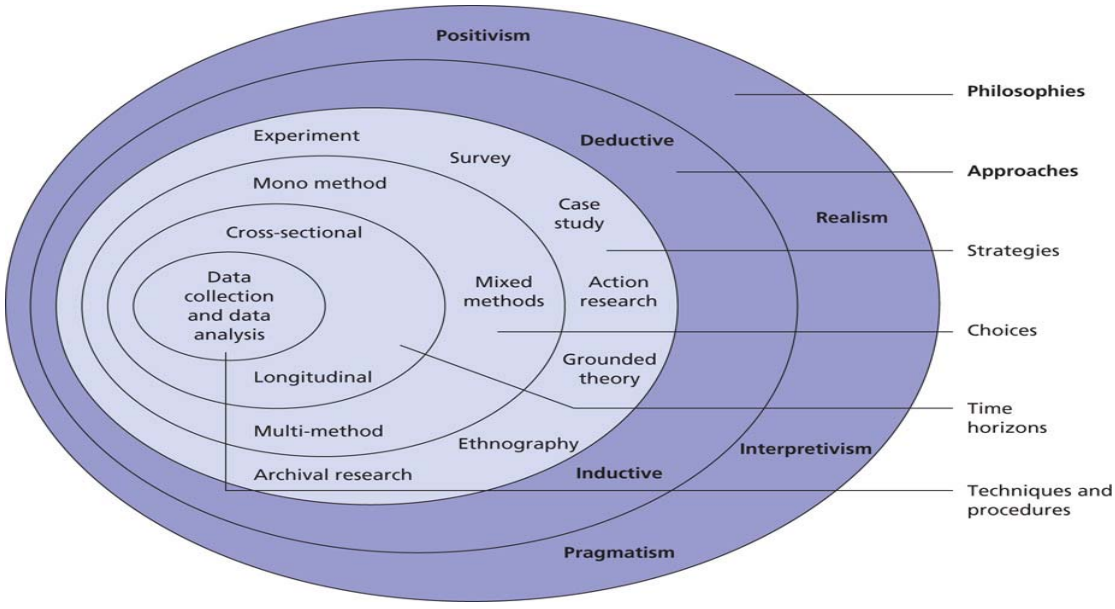


Figure 1.1 The Research 'Onion' (Saunders et al., 2009)

1.4.1 Research Philosophy

A research philosophy which generally discusses the methods of collecting and analysing data about a phenomenon should be chosen depending on a research topic. There are several research philosophies, namely positivism, realism, interpretivism, and pragmatism (Saunders et al., 2009).

- Positivism is one of the most important philosophies in the natural science field of research. It is 'working with an observable social reality and that the end product of such research can be law-like generalisations similar to those produced by the physical and natural scientists' (Remenyi et al., 1998). This philosophy enables researchers to collect data by an observation method to develop a new theory based on the existing knowledge which has been tested and validated.
- Realism mainly demonstrates what human sense is truth and is derived from the philosophy about the existing instance being independent of the human mind. Realism is widely applied in the research studies of natural scientists. It believes a scientific method is able to be used to develop knowledge and enhance the methods of collecting data and understanding data. It is normally adopted for comparing two forms of management studies.
- Interpretivism focuses on a social field based on the assumption of human considered as 'social actors'. This philosophy demonstrates different people carrying the same research may cause differences because they are the 'social actors' which can affect the studies. This philosophy is derived from phenomenology and symbolic interactions. Phenomenology emphasises on human acted as 'social actors' are able to affect the society. Symbolic interactions express human are able to change their actions when continually interpreting the involved actions. It is widely adopted for the studies of organisational behaviour, marketing

and human resource management.

- Pragmatism focuses on practical research and adopts multi-philosophies on research. Researchers should 'study what interests you and is of value to you, study in the different ways in which you deem appropriate, and use the results in ways that can being about positive consequences within your value system' (Tashakkori and Teddlie, 1998).

In the natural sciences, positivism is an important commonly research philosophy to be widely adopted. Positivism philosophy based on cause and effect normally is used firstly to build a framework and then evaluate this framework in a particular research field. Positivist researchers normally adopt a case study and other research strategies to meet the criteria of 'evidence of formal propositions, quantifiable measures of variables, hypothesis testing and the drawing of inferences about a phenomenon from the sample to a stated population' (Orlikowski and Baroudi, 1991). The current research aims at developing an adaptive information system framework for SMEs in SCs and mainly focuses on the common resources and their relationships. The proposed framework is developed based on the features from several existing frameworks and two case studies are used to validate the framework. Firstly a gift SC case study is obtained from literature as evidence of a formal proposition approach and secondly a UK-based case study from a SME manufacturing company is used in the research. Both case studies provide quantifiable data for the research. The results from the simulations assist management to understand the situation of their SCs and enable them to make 'what-if' analysis. This type of research belongs to positivist research perspective.

1.4.2 Research Approach

Two research reasoning approaches are widely applied in the management research, namely deductive and inductive approaches. The definitions of the two types

of research approaches are described as follows:

- Deduction is used to identify theories by a literature review and test these theories by real data. It is mainly used to test a theory during the development period of the theory. This approach is a 'top-down waterfall' method from theory to confirmation through testing and observation (Saunders et al., 2009).
- Inductive approach is employed to develop theories by analysing real data and review relevant theories. This approach is mainly adopted for building a theory using a method from observation to generalisation and is reverse from the deductive one. It is a 'bottom up climbing hill' approach from observation to theory' (Trochim and Donnelly, 2006).

The research aims at developing a framework from the current existing ones and then the remaining tasks are to validate this framework using case studies which adopt different product strategies. Consequently, the current research process of proposing and evaluating a framework mainly follows on deductive reasoning research approach from a positivistic view and the longitudinal studies based on a rich source.

When facing a real problem from a practical UK-based case study, the inductive approach is used to develop an optimisation algorithm to solve it by observing data, facts, and reviewing the current state-of-art relevant knowledge.

1.4.3 Research Strategies

There are mainly two distinctive clusters of research strategies, namely quantitative and qualitative research strategies. The quantitative research strategy focuses on 'the quantification of the collection and analysis of data'. The qualitative research strategy focuses on 'the words rather than quantification in the collection and analysis of data' (Bryman and Bell, 2007). In Figure 1.1, five main strategies related

to qualitative research are described as follows (Saunders et al., 2009):

- Case study: ‘Case study is an empirical inquiry that investigates a contemporary phenomenon with its real-life context, especially when the boundaries between phenomenon and context are clearly evident’ (Yin, 2003) and is widely adopted in a research field concerning information system. Case study is mostly considered as an exploratory research to collect details of a or a group of units, like company, worker, and processes, etc. over a relative long period for gaining knowledge (Collis and Hussey, 2003).
- Action research: ‘Action research aims to contribute both to the practical concerns of people in an immediate problematic situation and to the goals of social science by joint collaboration within a mutually acceptable ethical framework’ (Rapoport, 1970). This strategy is used to add knowledge of the social sciences and mainly adopted as a valid method. Its assumption is that researchers and the research projects belong to the constantly changing world and its purpose is to ‘enter into a situation, attempt to bring about change and to monitor the results’ (Collis and Hussey, 2003). Figure 1.2 shows the normal processes of action research from diagnosing, planning, taking action and evaluating.
- Grounded Theory: Grounded theory is adopted for developing the theory depending on data collection and analysis and ‘grounding the account in empirical observations or data’ (Martin and Turner, 1986). It is a classic type of the inductive approach to define the aim of a study by observations.
- Ethnography: Ethnography is derived from anthropology science, aiming

to ‘describe and explain the social world’ a time consuming method. This approach is used to interpret the patterns of activity in the human society by using ‘socially required and shared knowledge’ (Coolis and Hussery, 2003).

- Archival Research: Archival research is a method, which only uses the principal source of data, such as administrative records and documents, and does not adopt secondary source of data. This method focuses on original purpose of data collection for the research (Saunders et al., 2009).

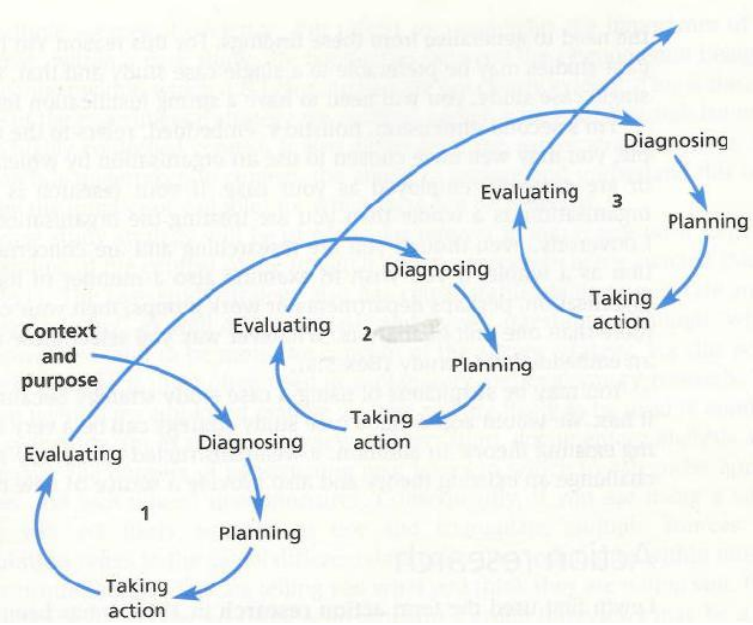


Figure 1.2 Action Research Spiral (Saunders et al., 2009)

There are two main strategies associated with quantitative research are outlined as follows (Saunders et al., 2009):

- Experiment: Experiment mainly aims to obtain the answer of a causal problem by experiments and tests. In exploratory and explanatory research, the experiment strategy is adopted to solve the ‘how’ and ‘why’ questions. Normally, it is adopted to measure the dependent variable before and after changing the independent variable and this strategy is relatively difficult to be applied in business research (Coolis and Hussery, 2003).

- Survey: The survey strategy is employed to solve the questions concerning 'who', 'what', 'where', 'how much', and 'how many'. This strategy usually adopts a large amount of data to research the 'trend, attitudes, or opinions' (Babbie, 1990). In a big population, in order to avoid consuming too time in collecting information from each individual, choosing a suitable and representative sample is a critical and necessary stage of a survey approach.

The main aim of the research is to develop an adaptive simulation-based decision making framework for SMEs. Literature review was conducted of existing frameworks to determine features associated with the adaptive framework. These features are then integrated and developed into the proposed framework. During the development of this framework, a case study approach is employed to validate the framework for achieving the research aim. Consequently, the research is involved in case study analysis based on a gift SC and a UK SME manufacturing company. In a practical scheduling problem of the case study, grounded theory is adopted to develop the optimisation algorithm to solve this practical problem. Additionally, action research is applied to design the 4P's closed-loop control method.

1.4.4 Time Horizon

Time horizon is concerned with the attribute of a research planning. Cross-sectional study is a 'snapshot' time horizon, which records are only made at a particular time point. This study is usually adopted to record 'the incidence of a phenomenon' by using a survey strategy (Saunders et al., 2009). Longitudinal study is a 'diary' time horizon, which adopts a series of time points or events (Saunders et al., 2009). To record change and development is its obvious advantage compared to a

cross-sectional study. The conduct of the research mainly adopts a longitudinal horizon to cover information and data from literature review for analysing the differences of current existing simulation frameworks and tracing the development of Tabu optimisation method.

1.4.5 Data Collection Method

Original data collection and secondary data collection are two main data collection methods. Original data means data gathered and recorded by the researchers for the current research purpose. In order to collect original data, interview, observation, and questionnaires are main collection methods (Collis and Hussey, 2003). Secondary data collection refers to the recordation of data which are made by other researchers for the certain purpose other than the current research aim (Zikmund, 2002). Normally, secondary data is collected from books, journals, and conference papers, etc. Two types of data are involved in the current research, interview, observation, literature review, and data transferring from the ERP database are all data collection methods.

1.4.6 Research Development

Figure 1.3 shows the development path of the research and the qualitative research strategy mainly consisting of documentary and case study analysis. This path clearly shows primary research methods adopted at each stage and inputs and outputs of each part/component during the research. In the *exploration phase* as shown in Figure 1.3, the simulation-based information system is chosen as a research issue by reviewing the background and challenging issues of IS in SMEs. Popular simulation-based frameworks are then compared and analysed. Secondary data sources consisting of academic journals and conference papers are reviewed and the requirements for an adaptive framework are discussed. In the *explanation phase* of the research, an adaptive

simulation-based DMS consisting of simulation blocks is proposed and developed based on these requirements. In the *testing phase* of the research, two case studies which adopt two common products strategies - Make To Order (MTO) and Make To Stock (MTS) are used to evaluate this framework. In the design of MTO case study, its optimisation block requires an optimisation algorithm to optimise scheduling and find a 'good enough' schedule, thus ground theory is introduced for building this optimisation algorithm. Published papers about optimisation algorithms of Jm/ST_{id} with LS are reviewed indicating there are not any existing optimisation algorithms to satisfy these requirements of the MTO case study. A proposed integrated CKPO and DOA optimisation algorithm based on the Tabu Search algorithm is developed and evaluated by different CKPO parameters. This proposed optimisation algorithm is used to evaluate the design of an optimisation block.

1.5 Statement of Ethics

The research design follows the University's ethical principles and the statement of the related ethical principles are discussed as follows:

- Due accreditation: A Harvard referencing method is adopted to build a referencing system for this research. All referring materials in this thesis are clearly listed in the reference section in order to track them easily and accurately.
- Anonymity and confidentiality: An anonymity concept is adopted for all data collection particularly in UK-based case study to reassure industrial collaborators that data collected from them are only used within the context of this research.

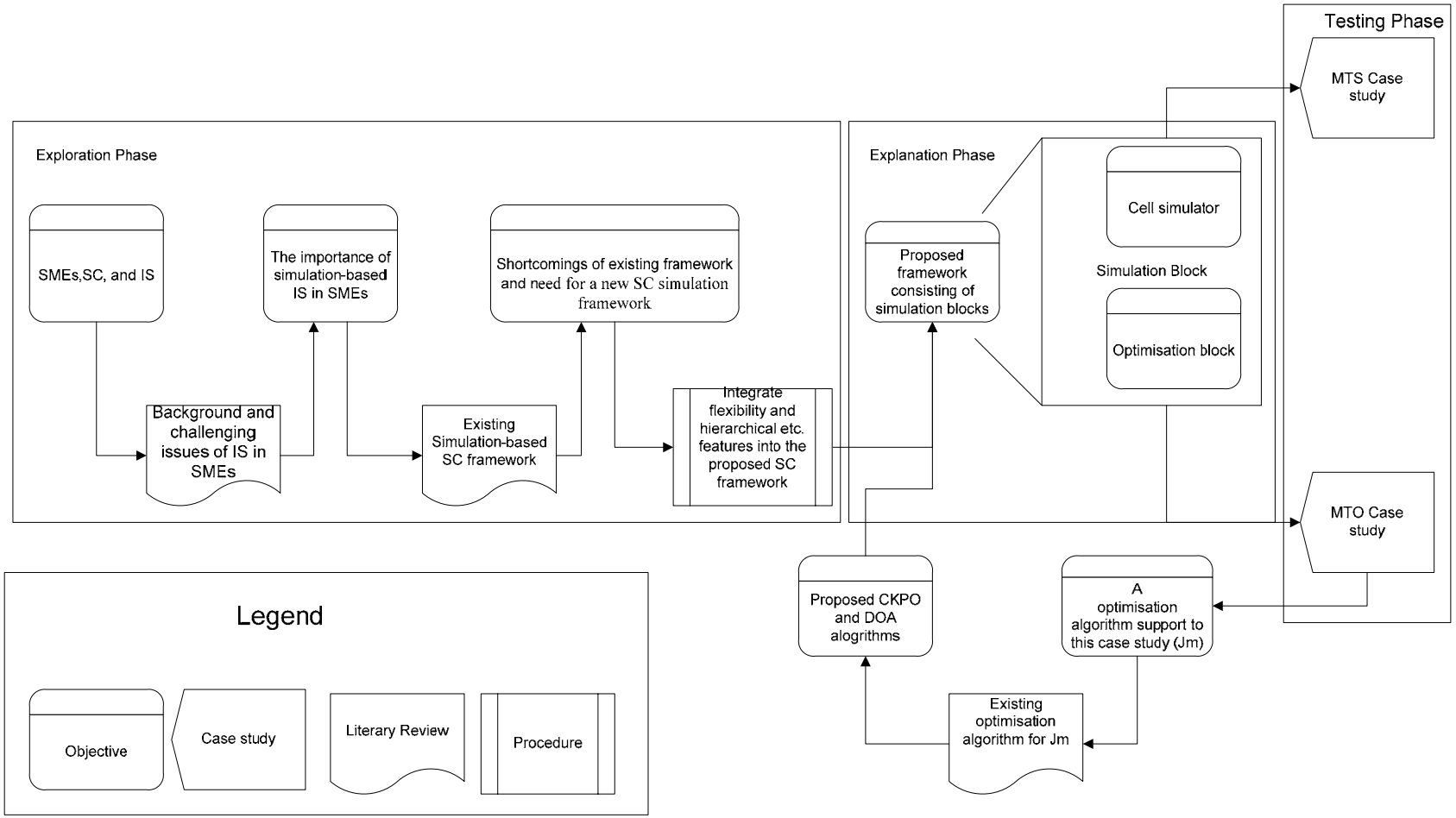


Figure 1.3 Research Development Path

- Purposeful collection of data: only data which is relevant to the research is collected and the other unnecessary information on organisations and people is not taken. All data collected are only used for the research and are not provided to the parity party.
- Informed consent and withdrawal from Investigation: the collected data is allowed and guarded by the industrial collaborators and could have been withdrawn by them at any time.

1.6 Structure

This thesis consists of seven chapters which are organised from literature review, theory exploring, framework, system design, and case studies as shown in Figure 1.4. The contents of the chapters are as follows:

Chapter 1 demonstrates the motives of the research work and lists the requirements of the information system in SMEs. The challenging issues are identified and their contributions towards the research are discussed in this chapter.

Chapter 2 reviews relevant knowledge of designing the proposed framework of a simulation-based DMS for SMEs in SC. Generally, a SC is considered as a complex discrete-event system and thus the simulation of a discrete-event system is the focused topic in this thesis. The worldview of the simulation model, especially the *three phase* approach, is firstly discussed and most methods of simulation optimisation are discussed. The current popular SC simulation frameworks are introduced and their characteristics are summarised for designing a new framework of a simulation-based DMS. Several popular simulation software packages are compared as a basis for choosing a suitable one to develop a cell simulator. Finally, the relevant knowledge of the optimisation of Job Shop Scheduling Problem (JSSP) is reviewed as a preparation for designing an optimisation block.

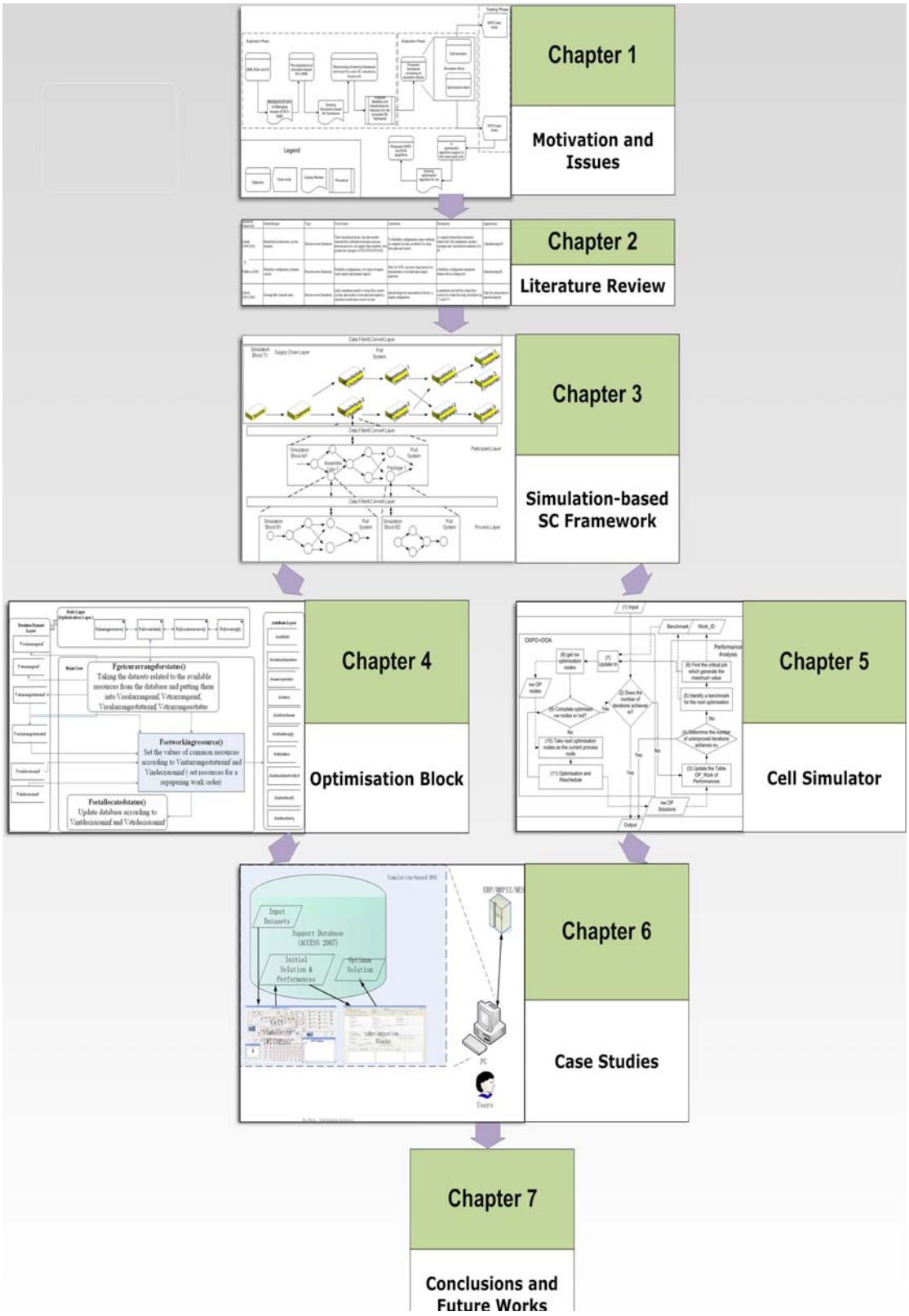


Figure 1.4 Structure of Thesis

Chapter 3 outlines the entire processes of developing the flexible simulation-based decision-making framework for SMEs from a SC perspective. The proposed concepts such as 4 Ps' closed-loop control, multi-layer control, a proposed simulation worldview, etc. are discussed. A proposed simulation framework with flexibility configuration and hierarchical network structure based on simulation blocks is developed and is able to accurately represent the relationship among the participants of a SC. The design of a simulation block and a cell simulator are briefly discussed.

Chapter 4 introduces the architecture of an optimisation block consisting of an optimisation module and a performance analysis for a special, complicated problem. A proposed optimisation module is developed for finding a 'good enough' solution of a JSSP based on the UK-based case study. Two optimisation algorithms, CKPO and DOA, are integrated into an optimisation module to improve the performances of the optimum solutions based on the initial solutions from a cell simulator.

Chapter 5 introduces the adaptive structure of a cell simulator and describes the design processes for a UK-based case study. In the simulation worldview, the design based on two types of identified events i.e. *order coming* event and *common resources released* event are explored and their workflow are discussed. A conceptual model of a cell simulator of a UK-based case study is developed and its relevant input, output, and database are discussed. Finally, a verification program which determine the feasibility of the scheduling has been developed.

Chapter 6 illustrates two case studies to validate the simulation framework. A stock-driven simulation-based DMS is developed based on a 'gift SC' from a literature review and the performances of a proposed replenishment algorithm are explored by comparing results from the cell simulator under different scenarios. A schedule-driven simulation DMS is proposed to simulate an agile job shop of a UK-based case study. Real data obtained over a six months' period of a UK-based

company linked to a simple ERP system are used in a simulation system. The functions and performances of the simulation system are tested and the optimisation's features and capability of CKPO and DOA algorithms are evaluated under different parameter settings.

Chapter 7 outlines the conclusions of the research work and describes the current stage of designing the simulation-based DMS of SMEs in SCs. Future works form the experimental work are outlined for enhancing and extending the proposed simulation systems.

Chapter 2 Literature Review

2.1 Introduction

This chapter identifies the areas of the research work and introduces the relevant knowledge of the research. The aim is to provide a knowledge support to design an adaptive and flexible simulation-based decision-making framework consisting of a cell simulator and an optimisation block outlined in Chapter 3. Current simulation frameworks and their characteristics are discussed and analysed as a preparation for designing a proposed framework for SMEs in SCs. This chapter also attempts to choose a suitable software package by comparing most of the popular simulation software packages for developing a cell simulator (outlined in Chapter 5). The relevant knowledge of the optimisation of JSSP is introduced and used as mathematical supports for designing an optimisation block (outlined in Chapter 4).

2.2 Supply Chains Overview

There are numerous definitions of supply chains depending on the different operations that they are being applied to. From a simulation view, a supply chain is a network which consists of a group of suppliers, manufactories, warehouses, distribution centres and customers. This includes the links between the groups via bi-directional material, information, order, financial, manpower and equipment flows. This definition and complexity of a supply chain and its architecture are illustrated in Figure 2.1.

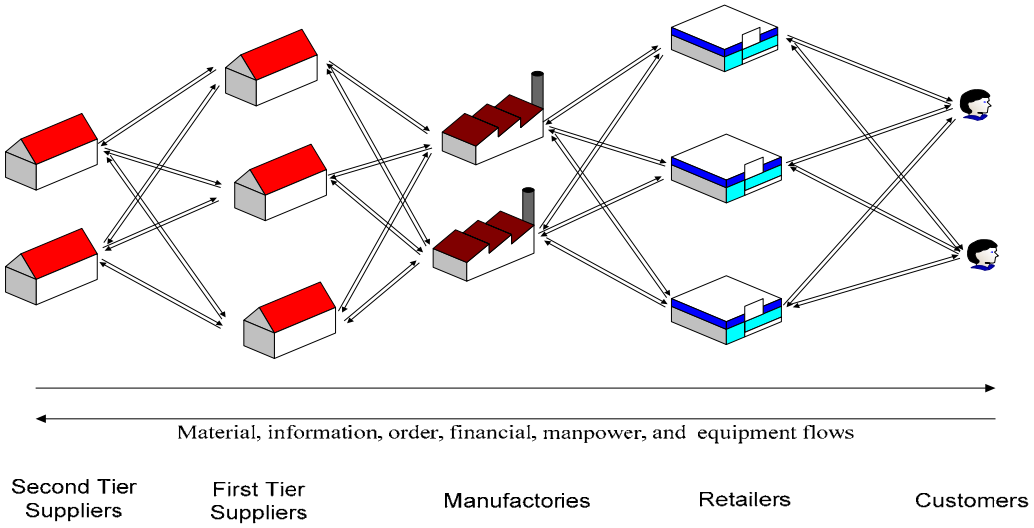


Figure 2.1 Supply Chain Network

The development of information systems in a SC is divided into the following six stages (Ross, 2003):

- Logistics decentralisation: From the late 19th century to the early 1960s, logistics was not considered by most companies to be an important factor in having a significant influence on the profitability of operations. Most companies simply divided up their departments depending on their operational functions. This led to an uncoordinated relationship between departments, and little standardisation of the management existed during this time.

- Total cost management: In the mid-1960s, companies merged the decentralised logistics departments into a single management system and applied total cost theory to the management structure.

- Integrated functions: Since the 1980s, companies have adopted Just-in-Time (JIT) and Total Quality Management (TQM) concepts because of pressures from the global market resulting in an integrated function.

- Supply chain management (SCM): In the mid-1990s, the concept of integrated functions has been extended to the entire SC using core values of competitive and collaborative relationships among trading partners. This has resulted in the development and applications of ERP software being implemented in companies.

- E-supply chain management: Since 1998, internet technology has successfully been applied in the traditional SCM to develop the E-Supply Chain Management (E-SCM) and each enterprise in the SC could share real-time information.

- M-supply chain management: Today, the traditional e-supply chains are being converted to Mobile-Supply Chains (M-SCs) via various wireless technologies applied in the SCM. For example, trucks using Global Positioning System (GPS) navigation and electronic map systems can transmit their positions to the manufacturer in real-time. The control centre can assign the optimal tasks to the trucks depending on their positions and enable the products to be delivered as soon as possible. Customers are able to check the accurate shipping time via login to the M-commerce server. The architecture of the shipping system is shown in Figure 2.2. Radio Frequency Identification (RFID) is another important wireless technique to support SCs to improve the efficiency of inventory management and reduce manual mistakes (Laudon and Laudon, 2006). These mobile techniques, such as RFID, GPS, WiMAX, ZigBee, etc., result in the rapid development of the 'Internet of Things' technology, which owns the capability of object identification, accurate data capture and real-time data exchange. These techniques are able to provide real-time accurate data for simulation-based DMSs of participants in SCs.

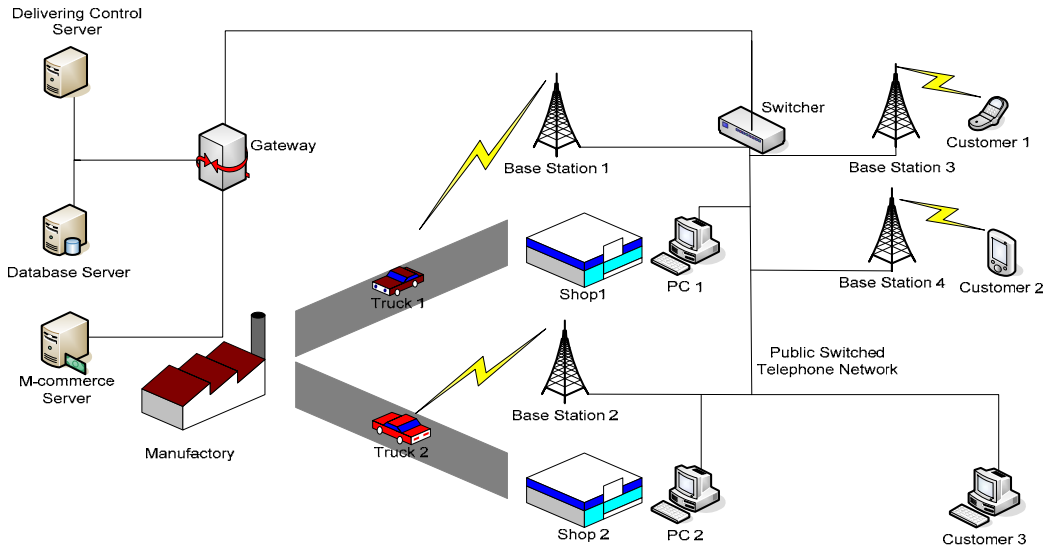


Figure 2. 2 Architecture of a Mobile SC Delivering System

2.3 Systems, Models and Simulation

In 1970, Schmidt and Taylor gave the definition of 'system'. A system presents a collection of entities, like people and machines, which act and interact together for the accomplishment of several logical ends. Most simulation experts have accepted the definition that a 'model' is expressed as a logical description of part or complete characteristics of a system. A model is studied for realising the work operations and performance of the corresponding system. To study a complex system model, simulation is employed to deal with this model by observing different simulation results and statuses depending on the varying input sets (scenarios).

A SC with multi-echelons always includes suppliers, manufacturers, delivery agencies, retailers and their relationships covering information, material, finance, etc. Thus, a model built on a SC is relatively complex so that simulation is almost the only method to explore the model. This model studied by simulation is called a 'simulation model' and its classification has three dimensions and are outlined as follows (Law and Kelton, 2000):

1) Static and dynamic simulation models

2) Deterministic and stochastic simulation models

3) Continuous and discrete simulation models

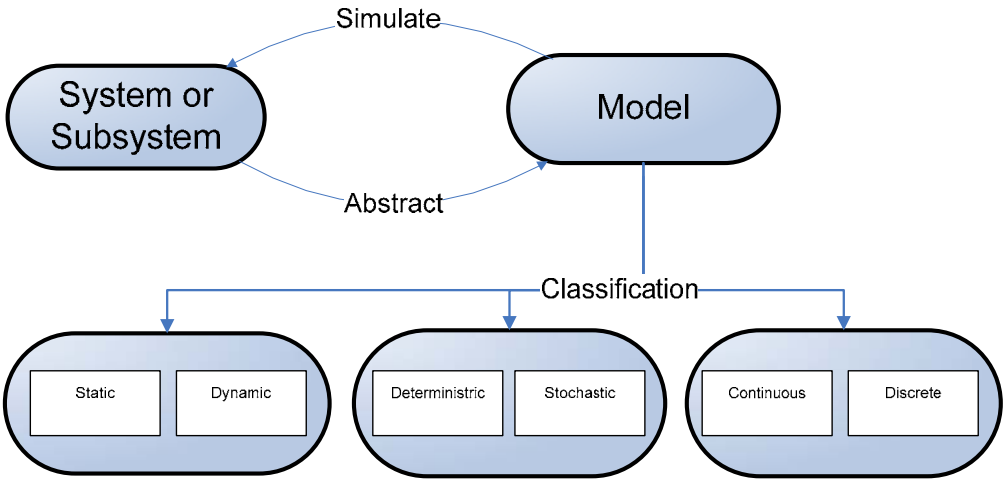


Figure 2. 3 Relationships among System, Model, Simulation and Simulation Classification

Figure 2.3 shows the relationships among system, model, simulation and the different classifications of simulation models. According to these classifications, a SC system normally belongs to Discrete Event Dynamic Systems (DEDS) and its simulation model owns dynamic and discrete features. The DEDS simulation will be the research area of this thesis.

2.4 Simulation Worldview

Before building a simulation model, designers must answer the question, 'which world view is to be adopted'. A simulation worldview describes an approach to ensure the running of the simulation model. In the past, three types of worldview, such as event scheduling, process interaction, and activity scanning, have frequently been adopted (Carson, 2004).

Event scheduling expresses all events which have been put in the schedule and is

mostly suitable for a deterministic simulation model without involving random events, because it requires that most events should be known in advance. Normally, these events are added into the Future Event List (FEL), which includes all events listed in order of their start time and duration or end time. An FEL notice indicates the imminent event which is being executed. After an execution, the FEL is updated by checking and adding new events which should be put into their proper position in the FEL. Then, the event notice will be removed and the system will move on to the next event. If random events need to be simulated by this worldview, they are expressed by statistical distribution.

Before introducing the process interaction approach, the definition of a process must be outlined. A process is a list similar to but more complex than a FEL. The list consists of not only events and activities, but also other information relating to them, such as delays, resource requirements, life cycles, etc. When an entity requires one or more limited resources for executing an activity, interactive mechanisms would take control of these resources. For example, when entity A occupies a resource which entity B wants for its activity, entity B must delay the start time of its activity until entity A releases this resource.

Activity scanning focuses on the execution conditions of activities. This mechanism would rescan all activities and check whether they are meeting their execution conditions in a small time slice. This approach is relatively simple and easy to understand, which has resulted in it being widely applied and modified by researchers. The disadvantage is that this scanning process can take up a great deal of CPU time. Based on this, the *three phase* approach has been developed and is described in Section 2.5.

2.5 Three Phase Approach

In this section, a *three phase* approach is briefly discussed. It was proposed in 1963 and developed for the UK steel industry. Pidd (2004) recommended this approach to simulation model developers. This approach defines two types of activities which change system states.

One type is called B activities fired by a timer. This means that these activities are able to be forecast and put into a schedule, hence B is an abbreviation of Book-keeping or Bound activities. These activities have the important feature of occurring on time, so that their occurrences are determined. Thus, a simulation model holds a system clock to ensure B activities happen at their start time. For example, a factory rule requires engineers to check a machine at 6pm on a working day, and this is a typical B activity.

The other is C activities, which take place in special situations. These activities occur when certain conditions are met. In fact, C activity is an abbreviation of Conditional or Cooperative activity. These activities do not need a system clock to support them, but depend on the related conditions. In simulation, achieving these conditions usually represents reaching the states of entities, resources, or other components of a system. Sometimes, there are no obvious conditions, but in fact it can be transferred to particular conditions. For instance, a traffic rule states that vehicles must give way to pedestrians on a zebra crossing. The rule seems not to be a C activity because no condition is identified. However, if you rephrase the rule to state that vehicles must stop when people are crossing, the rule has not changed its original meaning, but it now incorporates the condition of 'people crossing' and the C activity of 'vehicles stopping'.

After defining these activities, the execution of the *three phase* approach is introduced. The sequence of execution is from A to B to C as follows:

A phase: system clock turns to the time when the next B activity will occur.

B phase: B scans for B activities and releases all resources.

C phase: C scans for suitable C activities after getting all the resources from the B phase.

Then the cycle goes back to the A phase.

This cycle mechanism keeps the simulation running and avoids deadlock situations. This is a basic DEDS control mechanism and is widely adopted in simulation models.

2.6 Current SC Simulation-based Decision-making Frameworks

This section reviews the current simulation frameworks and assesses the novel issues and limitations of the frameworks from the following viewpoints in Table 2.1:

- *Methodologies* describe the main methods and technologies employed in the given framework.
- *Type* means which type of simulation models are mainly implemented in the given framework.
- *Novel issues* show the original advantages and contributions of the given framework.
- *Limitations* show the restrictive weaknesses or lacks of capacity of the given framework.
- *Description* gives a simple description of the features of the given framework.
- *Applications* show the implemented field of the industry or covering management of a SC by demonstrations or using case studies.

Through reviewing the existing frameworks, several important factors regarding the adaptive characteristics are obtained as follows:

- The adaptive framework should be suitable for the product strategies adopted

in SMEs because the product strategy can determine the business processes of an enterprise. There are four commonly product strategies, namely MTO, MTS, Engineering To Order (ETO), and Assemble To Order (ATO). Umeda and Lee (2004) proposed two types of control methods of simulation, such as stock-driven and schedule-driven, as the ground concept to design the simulation-based DMS for the enterprises which implement these four product strategies.

- The adaptive framework should meet the requirements of different product routing planning of SMEs. Thus, the routing flexibility configuration method (Wadhwa, 2008) which can provide a flexible routing directed network for a SC allows possible alternative routes to be built in the system. Therefore, an enterprise can make an adaptive product planning which allows a product to be produced and delivered according to a flexible route under the minimum restricts.
- The adaptive framework should easily be adjusted or executed with minimum modifications to suit the changes of business policies or processes in SMEs. This requires a framework which has modularisation and reusability features. The hierarchical structure technique which is similar to the Pundoor's framework (Pundoor, 2006) is able to provide a possibility for adopting the standardisation of the Supply Chain Operations Reference (SCOR) model. This technique also enhances the reusability and modular ability of the proposed framework.
- The proposed adaptive framework needs to consider how to cooperate the individual systems implemented in the enterprises of a SC and make simulation or optimum results meet all constraints of the SC. More details are given in Section 2.7.

Table 2.1 Current Simulation Frameworks I

Simulation Framework	Methodologies	Type	Novel Issues	Limitations	Description	Applications
Umeda (1998,2004,2007)	Hierarchical architecture, system dynamic	Discrete-event Simulation	Three hierarchical layers, four data models (material flow, information, business process, decision process), six supply chain members, four production strategies (ATO,MTO,MTS,ETO)	No flexibility configurations, huge workload, no complete system, no details of a shop-floor plan and control	A complete hierarchical simulation framework with standardise product strategies and classification members of a SC	Manufacturing SC
Wadhwa (2008)	Flexibility configuration, dynamic control	Discrete-event Simulation	Flexibility configurations, two types of inputs (static inputs and dynamic inputs)	Only for MTS, one layer (chain layer),two measurements (cost, lead time),simple equations	a flexibility configuration simulation framework as a learning tool	Manufacturing SC
Mönch (2003,2008)	Message Bus (transfer data)	Discrete-event Simulation	Link a simulation model to a shop floor control system, data model to store data and separate a simulation model and a control system	Special design for semiconductor factory, a simple configuration	a simulation test-bed for a shop-floor control of a wafer fab using AutoSched Ap 7.1 and C++	Main for semiconductor manufacturing lab

Table 2.1 Current Simulation Frameworks II

mulation Framework	Methodologies	Type	Novel Issues	Limitations	Description	Application
Wu (2002)	Complexity Theory	Discrete-Event simulation	Three models (operational model, scheduling model, control model), a VBA model links a Arena model to a Excel model	Only one layer (chain layer), a simple configuration, only for MTO strategic, only for stock management	A simulation framework of stock managements of a SC using VBA, Arena, and Excel	Stock management of a SC
Lee (2002)	Continuous and discrete system	Discrete-continuous combined simulation	Classify the activity of SC into discrete and continuous types	Only one layer (chain layer), a simple configuration, only order process	A simulation system to simulate the order process using continuous and discrete combined method	Order process of a SC
Pundoor (2006)	Supply Chain Operations Reference (SCOR)	Discrete-Event simulation	Reusable modules, four management processes (plan, source, make, deliver), hierarchical structure, process elements	Concept research level, no strategy considered, no flexibility configurations, now just for scheduling	A simulation framework according to SCOR using Arena and Excel	Order scheduling of a SC

2.6.1 Umeda's Framework (1998)

Umeda and Jones (1998) developed an integrated test-bed simulation system for constructing a manufacturing SCM. This system architecture consists of three subsystems including a hierarchical SC simulation system, a supply-chain management communication server, and a decision support system for suppliers' management as shown in Figure 2.4. Four data models, namely material-flow, information, business process, and decision process models support the participants of the SC simulation system to communicate with each other and define the relationships of participants. Three data drivers, such as suppliers, production, and demand data drivers take full responsibility of collecting different data from suppliers to customers.

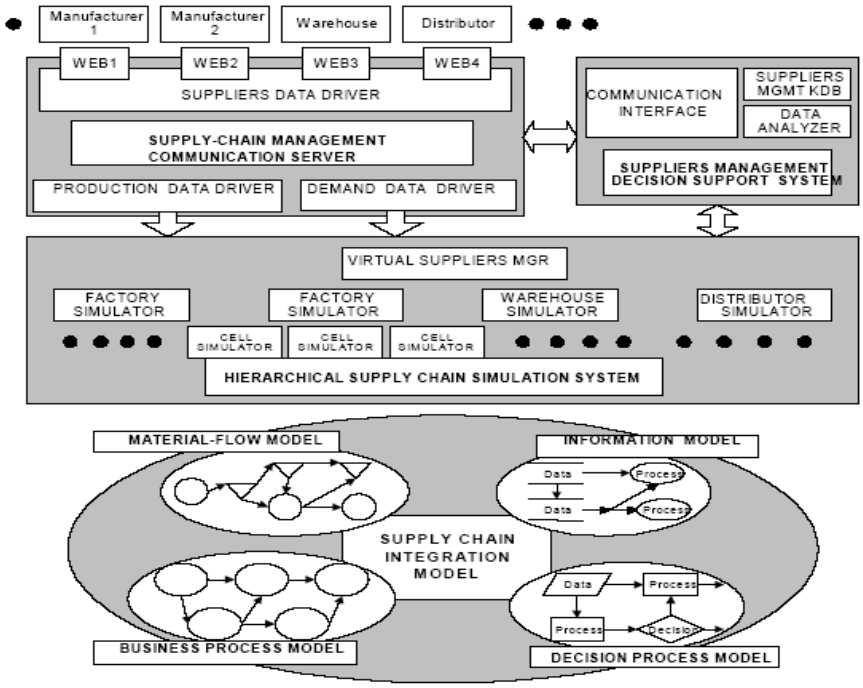


Figure 2.4 Structure of Umeda's Framework (Umeda and Jones, 1998)

This simulation system adopts a hierarchical structure with three layers. The top layer is called chain layer, which focuses on business and information process aspects of a complete SC. The middle layer is considered as individual layer which represents participants of a SC such as factories, distributors, warehouses, etc. The bottom layer

details activities in each individual. The virtual supplier's manager is a center control module to take a duty of managing the activities and providing the data of each layer.

Umeda and Lee (2004) outlined more detail definitions of a genetic simulation system based on schedule-driven and stock-driven control concepts. It reduced the previous four data models to three models, namely business process, material process, and information process models. Six supply chain members have been defined in the simulation system and two simulation control methods - Schedule-Driven and Stock-Driven were presented. Four production strategies (Umeda and Jain, 2004) require two control methods to support their implementations. Assemble To Order (ATO), Make To Order (MTO), and Make To Stock (MTS) strategies use schedule-driven control method and Engineering To Order (ETO) strategy adopts stock-driven control method. Figure 2.5 shows the operational processes of schedule-driven control. A planner generates a schedule using a Master Production Schedule (MPS) module for the members of the SC according to prediction data and related data collected from the SC members.

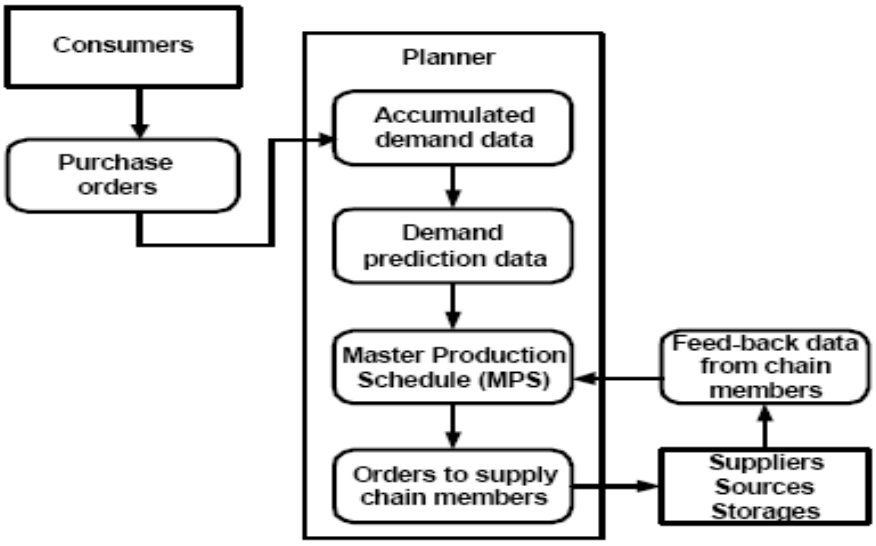


Figure 2.5 Operational Diagram of Schedule-Driven Control (Umeda and Lee, 2004)

Figure 2.6 illustrates the operational diagram of Stock-Driven control. When the stock volume drops down to the replenishment point, a planner makes an order to

replenish stock. In his study, 30 types of data need to be collected for establishing this simulation system have been described, such as supply chain, product volume data, supplier data, and source data (Umeda and Lee, 2004). Umeda (2007) developed several scenarios models using system dynamic concept of simple order planner, lead time reduction, inventory control, etc. to improve his framework.

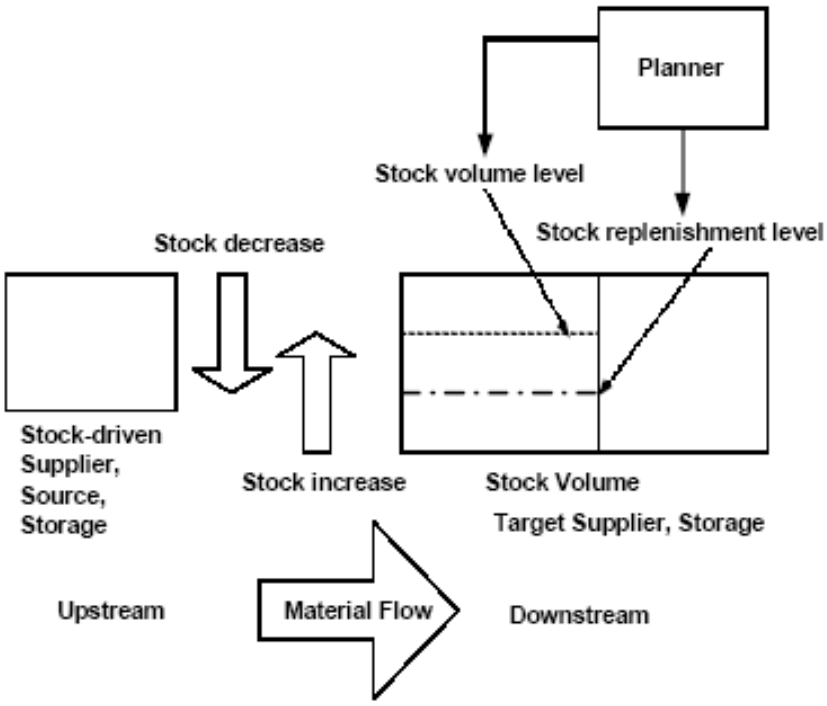


Figure 2.6 Operational Diagram of Stock-Driven Control (Umeda and Lee, 2004)

2.6.2 Wadhwa's Framework (2008)

Wadhwa et al. developed a simulation logistics system for a manufacturing SC and introduced the flexibility configuration into the simulation framework (Wadhwa et al., 2008). Three types of flexibility consisting of no flexibility, partial flexibility, and full flexibility were explored and assessed based on this framework. The framework adopts a MTS production strategy and a dynamic control in a SC, and it utilises two measurements of lead time and cost. The purpose of the simulation model is to minimise the total cost consists of lead-time, ordering cost, and transport distance. Figure 2.7 shows the operational diagram of the dynamic control. Demand Pattern makes orders according to the inventory status. The total cost is calculated by a simple

plus equation of lead time, ordering cost, and transport distance (transport cost). Transport cost consists of fixed cost, variable cost, and pipeline cost in this framework. The framework also classifies two types of inputs, namely dynamic inputs and static inputs. Dynamic inputs include product demand pattern and inventory status. Static inputs consist of ordering cost, lead time, transport distance, and transport model. The performance measure is determined by two parameters - total cost and fill rate. The demon case to validate this framework only includes four levels, manufactures, distributors, warehouses, and retailers and three stages (three manufactures, three distributors, three warehouses, three retailers).

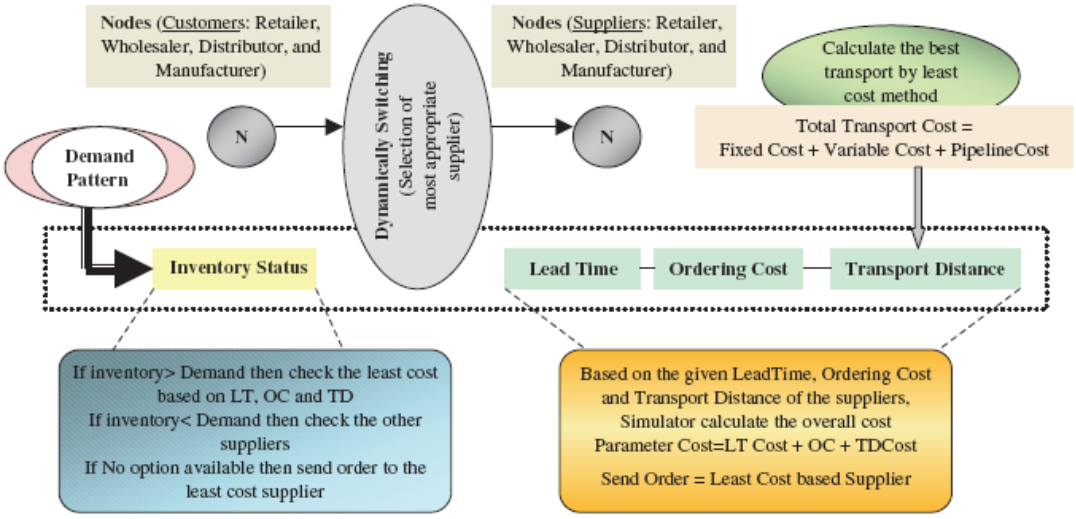


Figure 2.7 Operational Diagram of Dynamic Control (Wadhwa et al., 2008)

2.6.3 Mönch's Framework (2003)

In 2002, Mönch developed a simulation framework for assessing the performances of a shop-floor control of a semiconductor factory. Mönch proposed to integrate a simulator and shop-floor control system to service for a wafer fabrication. The trigger mechanisms of events and communication methods of the simulation model and the control system had been defined. This framework has three tasks for achieving the aim of simulating a shop-floor control. Firstly, it is able to simulate the operations of a shop floor and talk with a control system. Secondly, it needs an interface linking a

control system. Thirdly, a database is required to store the necessary data of the framework. Figure 2.8 illustrates an architecture of the communication method of a simulation model and a shop-floor control system. The result of the AutoSched AP (ASAP) simulation model is sent to the data model of a shop control system via a message bus consisting of a set of functions. The system can make a schedule according to its data model and then store the schedule into the data model for testing and comparisons. The framework chooses an ASAP 7.1 simulation language to develop a simulation model due to its abilities to link to shop-floor control by building a dynamic C++ link library, and its wide applications in semiconductor field, etc. Then a shifting bottleneck heuristic algorithm and production-scheduling algorithm were introduced to validate the framework.

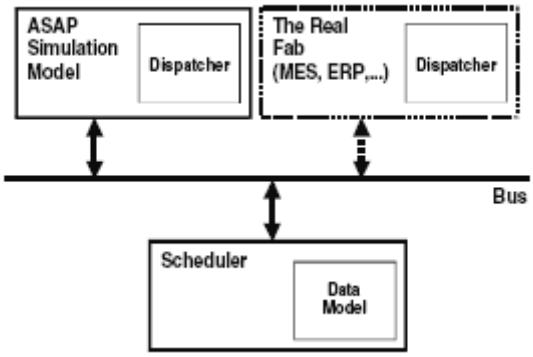


Figure 2.8 Architecture of Simulator and Shop Floor Control (Mönch et al., 2003)

In 2003, Mönch presented the operational diagram of the simulation framework and genetic algorithms in Figure 2.9. The data model is to store the dataset linking the algorithm and the simulation model. The algorithm is developed by C++ to add classes and attributes for extending the data model. The library is developed for a requirement of the AP framework.

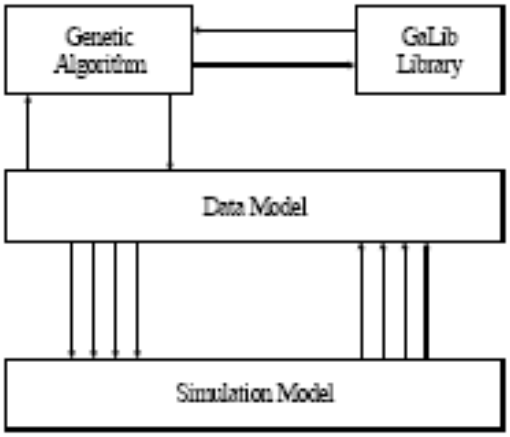


Figure 2.9 Generic Algorithm in Mönch's Framework (Mönch and Habenicht, 2003)

In 2008, Sprenger extended the Mönch's framework to a logistic aspect as shown in Figure 2.10. An *initializer* module puts the data from xml files into the blackboard layer. An order generator is used to generate the order and put the order information into the blackboard data layer. The planning system makes a schedule for the vehicles according to the order information in the blackboard layer and performance assessment's responds in calculating measurements. The simulator can mimic the behaviour of the logistic system via the blackboard layer. The blackboard can separate the simulation and the real system, and it also stores the data for re-planning.

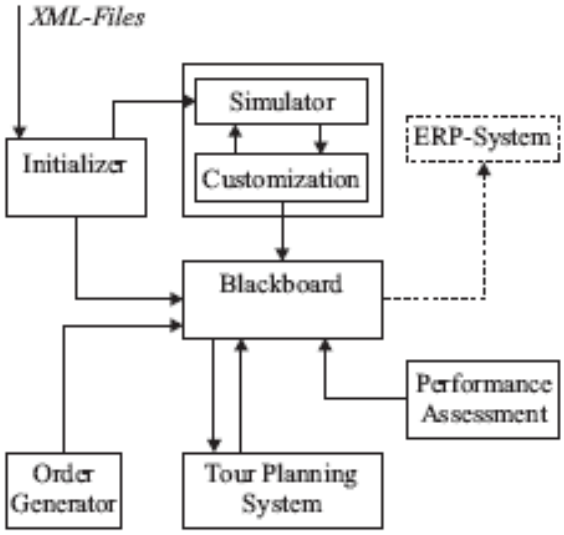


Figure 2.10 Mönch's framework in Logistic Field (Sprenger and Mönch, 2008)

2.6.4 Wu's Framework (2002)

Wu et al. (2002) proposed a simulation framework including three sub-models (control model, scheduling model, operational model) shown in Figure 2.11. A control model was developed using Visual Basic for Applications (VBA) to simulate the processes of sending orders from a customer to a supplier. A scheduling model using Microsoft Excel represents the operations for making production schedules to suppliers. An operational model programmed by an Arena simulation package is to mimic the activities of manufacturing products in suppliers. The schedule model can transfer data or get data back from the operational model via the control model. The study also researched the complexities of flow and stock. The difference between the Blue plan and the actual delivery causes the flow complexity and the variation between the scheduled stock and actual stock generates the stock complexity.

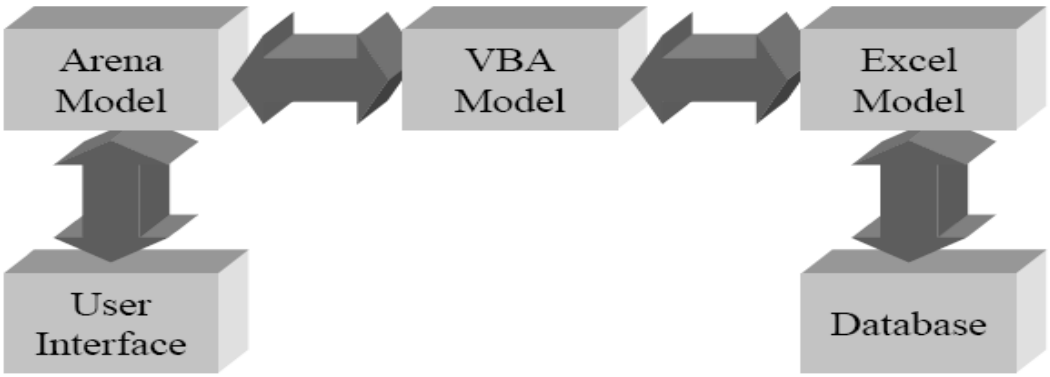


Figure 2.11 Wu's Simulation Framework (Wu et al., 2002)

2.6.5 Lee's Framework (2002)

Lee et al. (2002) developed a SC simulation system using a discrete-continuous combined method. Figure 2.12 illustrates the structure of the simulation system. The simulation system is a hierarchical system consists of retailer, distributor, factory, and supplier levels. Every level owns order information flows with two effective parameters of time delay and unfilled rate.

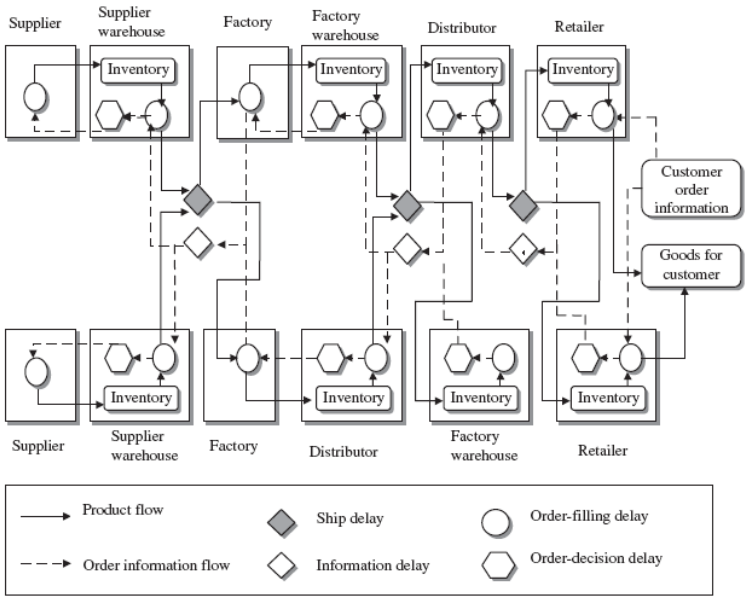


Figure 2.12 Structure of Lee's Simulation Framework (Lee et al., 2002)

2.6.6 Pundoor's Framework (2006)

Pundoor and Herrmann (2006) described the SC simulation framework based on the Supply Chain Operations Reference (SCOR) models by integrating an Arena simulation model and spreadsheets of Microsoft Excel. The purpose of this study was to emphasise the reusable components to construct a SC. Thus, standardisation and hierarchical structure are two main approaches to achieve the aim of reusability. The framework adopts four processes of plan, source, make and deliver defined by a SCOR model. Each process is labelled by planning, execution, and enable levels. Figure 2.13 shows three layers of this hierarchical framework. The top layer is a simulation model of a SC and the middle layer represents sub-models of each component such as consumers, producers, traders, etc. in the SC. The bottom layer focuses on sub-models of all operational processes of each component. This architecture gives a freedom of constructing a SC simulation model by reusable modules. The components of SC can be modelled both at a detail level and at an abstract level according to requirements of the simulation system.

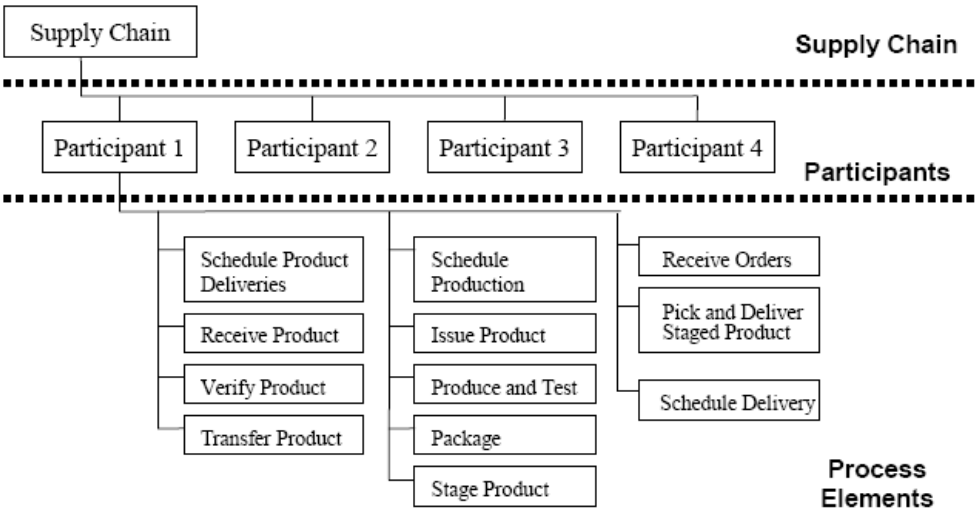


Figure 2.13 Pundoor's Simulation Framework (Pundoor and Herrmann, 2006)

2.7 The Need for a New Simulation Framework

From the previous Section 2.6, the current simulation frameworks are divided into two types-concept and practice frameworks respectively. The concept frameworks focus on how to extend simulation concepts to mimic all situations of a real world, such as Umeda's, Wadhwa's, Lees's, and Pundoor's framework (Umeda and Jones, 1998; Wadhwa et al., 2008; Lee et al., 2002; Pundoor and Herrmann 2006). The practice frameworks study show how to program and practise the simulation concepts using simulation tools, database, and programming languages etc, such as Mönch's, Wu's, and Pundoor's framework (Mönch et al., 2003; Wu et al., 2002; Pundoor and Herrmann, 2006). Another classification method is based on the fields of simulation concepts. The two most popular fields are flexibility configurations of Wadhwa's framework (Wadhwa et al., 2008) and hierarchical structure of Umeda's and Pundoor's frameworks (Umeda and Jones, 1998; Pundoor and Herrmann, 2006).

However, several limitations of the current simulation frameworks are discussed as follows:

Umeda's system design (Umeda and Jain, 2004) is still in a concept stage and a real adaptive simulation system is still under development. The reason is that the complexity of the adaptive simulation system leads to huge workloads. Now, only a simplified model has been developed and a complete simulation model has not been developed currently. Another shortcoming of this framework is the lack of considering flexibility configuration of a SC. It only supports the activities of one product in a simple SC.

Wadhwa's simulation framework (Wadhwa et al., 2008) is only considered as a learning tool and is limited by its simplicity and various assumptions. This results in the framework being unsuitable for use to design an adaptive simulation system. This simulation framework only considers a SC as one layer and defines several levels (participants) such as factories, retailers, distributors, etc. to construct a SC. Defining a factory or distributor at a detailed level is impossible in the framework by using such a simplistic method. This method is able to avoid the complexity of a SC and make a SC easy to understand. The simulation model only shows the simple attributes of the different participants on a SC layer and cannot express the detailed operational processes of these participants. Sometimes, SMEs need to design a route of new products or take an urgent order, thus, the simulation system is required to mimic the operational processes of the shop floor. Using only one layer SC simulation framework it is difficult to meet the requirements of SMEs in SCs. Furthermore, Wadhwa's simulation framework was only discussed under the condition of a MTS strategy. The other production strategies, like MTO, had not been explored. An adaptive simulation framework is required to support several production strategies. Moreover, the framework is assumed that products can be provided by only one supplier. In a real world, most companies own more than one supplier which provides the same or similar components or products. According to a standard of ISO 9001, the enterprise is required to own not less than two suppliers of one product for reducing the risk of

supplying products. Wadhwa's framework can only deal with one type of product and cannot simulate two products, which limits the applications of the simulator in a real world.

Mönch's simulation framework (Mönch et al., 2003) started from semiconductor manufacturing and enhanced the natural characteristics of the semiconductor field. Whereas, the framework has been extended by algorithms or field (logistic field), it is still a special simulation framework with partial adaptive features designed for a wafer fab. The framework has no basic characteristics of the adaptive simulation framework such as flexibility configurations and hierarchical structures.

Wu's framework (Wu et al., 2002) simulates only basic operations of a SC based on a top layer (chain layer) with a simple configuration (a supplier and a customer) and has not taken account of the flexibility configuration of a SC. Furthermore, it only studies the operations of stock management and order booking under a MTO strategy. Thus, this simple framework is not suitable for a real SME SC, because a SC is usually a complicated network including various suppliers, manufactories, retailers, and customers.

The shortcomings of Lee's simulation system (Lee et al., 2002) are as follows: Firstly, it has not taken flexibility configuration into account and only discusses a simple SC with a customer, two retailers, two distributors, two factories, and two suppliers. In a real world, most SCs are more complex than this simple SC. Secondly, it only adopts a MTO strategy and simulates order information flows. This is not sufficient for a modern SC adopting various strategies. Thirdly, the status of the order and inventory were researched, while other issues such as Work In Progress (WIP) in a shop floor and common resources assignment have not been explored by this framework.

Flexible configuration has not been explored in Pundoor's framework (Pundoor and Herrmann, 2006). This framework is still in concept design stage of describing a SC. In the operations of a production strategy to be constructed by designers, there are no exiting modules to represent the activities of the different strategies.

The frameworks outlined above indicate there is a need for an adaptive simulation framework for SMEs in SCs. An adaptive simulation framework has two characteristics firstly flexibility configurations and secondly hierarchical structures. However, there are no such frameworks existing which can take these two features into account from the literature review. Thus, a novel simulation framework with flexibility configurations and hierarchical structures characteristics is required for design SME simulation systems in SCs and is the purpose of the research conducted in this thesis.

2.8 Supply Chain Model Approaches

2.8.1 Overview

Recently, researchers have begun to pay more attention to build simulation models in SCs and a considerable number of related technical papers have been published. However, compared to a manufacturing plant, a SC is not just about more entities such as equipment, factories and people, but also their complex relationships like trust, sharing of information, and collaborative planning. These requirements increase the difficulty of simulating a SC and are therefore attractive to a researcher.

System dynamic with Ordinary Differential Equations (ODE) was modelled for a four-level SC by Forrester (Forrester, 1958). Towill developed the 'Forrester Supply Chain' by taking tiered structure and time compression into account (Towill, 1991, 1996) and applying the transfer function approach to the model (Towill, 1982). Sterman (1989) presented the stock management model with a dynamic decision-making system and simulated it via various scenarios to prove that the local rational heuristics influence the

process of decision making. Anderson et al. (2000) demonstrated demand methods using a system dynamic model of a machine tool industry and Parunak et al. (1998) extended the concept to stock and flow model.

A SC is an event-oriented system and its status changes with a set of events. Burns and Sivalian (1978) researched the SC using discrete time difference equations, and Ho (1989) illustrated the discrete event model through operations research, artificial intelligence, and systems theory knowledge. Using the discrete event model, Petri Net (PN) and agent based models have also been developed, and a special example is Moyaux and D'amours (2003) multiagent models for a SC. Vorst et al. (2000) proposed a PN for redesigning the SC and tested numerous scenarios of SCs. They also exploited the simulation of PN to choose a best scenario in decision making and adopted a food factory for their case study. Kemper (2000) demonstrated the implementation of the Generalized Stochastic PN in a small sized SC with a factory, supplier and shipper. A fluid simulation model to deal with a large number of events in the Distributor Centre has been built using Batches PN (Wang and Zhou, 2005).

A discrete event simulation model has been established for a product-distribution SC using ARENA and EXCEL (Wang and Takakuwa, 2006). An Integrated Refinery In Silicon (IRIS) was generated by Pitty in 2005 and the GA simulation optimisation based on IRIS was adopted in the refinery SC for its policies and investments (Koo et al., 2006). WITNESS's applications in simulation solution deployment have been illustrated, and a simulation model architecture wrapping WITNESS is given (Mehta and Rawles, 1999). An object-oriented simulation modelling tool with 29 classes and relationship networks for multi-echelon SCs is gradually being developed by a research group in the US (Rossetti et al., 2000; Rossetti and Chan, 2003; Rossetti et al., 2006). A parameterized simulation model has been established for discovering and confirming the relationship between information sharing and SC configurations and Arena software packages have been adopted as a tool to evaluate the performance of different SC

configures (Liu et al., 2006). Simulation techniques have been successfully applied in the SC transformation area to clearly show the business processes, inventory allocation strategy, and evaluation results to decision makers (Ding et al., 2006), and the related model SmartSCOR (SC Operations Reference) has been described (Dong et al., 2006). A real-time, in-line output analysis tool has been developed by the Extend simulation software package and a steel plant case study simulated to describe its advantages (Mullarkey et al., 2000).

Now, other techniques have been combined in simulation techniques in the SC domain. A decision-aid system based on mobile agents can provide necessary information to decision makers and protect other sensitive information (Gupta et al., 2001). A reference model of SCs has been established based on distributed simulation and business process modelling techniques (Rabe et al., 2006). An assemble-to-order SC has been simulated based on the novel Available-to-Shell management concept which IBM has adopted (Ervolina et al., 2006). It was demonstrated that IBM saved $100 million in inventory management by using the ATS method. Another management concept - Available-to-Promise was used to check its SC performances based on the simulation technique (Lee, 2006).

The supply chain approaches outlined in this section indicate there are two main approaches for modelling a SC. Firstly is Commercial-Off-The-Shell (COTS) (simulation software packages) based on discrete event simulation concepts and secondly is PN. Compared to PN，COTS has a friendly user interface and provides analysis platforms to show different performance after running the simulation models based on different scenarios. Consequently, COTS is adopted for the proposed simulation framework.

2.8.2 Simulation Software Packages

There are approximately 37 popular simulation package vendors in the world. According to the design aims and applications, simulation packages are classified as general-purpose and application-oriented (Law and Kelton, 2000). A general-purpose simulation package has a wide application and is designed for various special functions or entities according to the requirements of different areas. Its applications include transportation system design, logistics support analysis, supply chain optimisation, customer service, internal business process, six sigma project, etc. It covers an extensive range of industries including manufacturing, energy, nuclear, aviation, food and beverage, defence, financial service, call centre, and government. An application-oriented package is normally suitable for an enterprise involved in manufacturing, telecom, customer service, etc. and consists of packages such as OPNET, NS2, MATLAB, Arena and WITNESS etc (Robinson, 2004). These software packages are designed for easy-to-use, rapid modelling, and are extremely interactive and visually real.

The review of popular simulation software is based on the Winter Simulation Conference (WSC) which is the premier international conference about exploring the knowledge of simulation technology, specially in discrete-event simulation and combined discrete-continuous simulation. In the WSC, almost all popular simulation software companies for SCM applications make their representations to participants. Figure 2.14 shows a comparison from Winter Simulation Conference (WSC) 2008 conference on simulation software. The graph shows the percentage usage of vendor simulation software which are Arena, AutoMod, ProModel, Witness, Anylogic, Simul8 and Extend (Rockwell Automation, 2009). Table 2.2 shows the comparative information of the most popular simulation SCM software packages which account for almost 90% commercial usage.

Compared to these COTS, WITNESS was chosen as a simulation tool to develop the simulation models for SMEs. WITNESS is a widely used simulation tool developed

by the Lanner Group in the UK (Saker, 2007) and it can be applied in several areas, such as resources arrangement, investment planning, customer management, business process, network design, equipments layout, etc. It covers most aspects of the SCM from the view of resource and it is a suitable tool to build the models of SCs. The technical team in Lanner Group has a good relationship with our research group. The Faculty has dedicated licensees for WITNESS and consequently this simulation software was used in the research work.

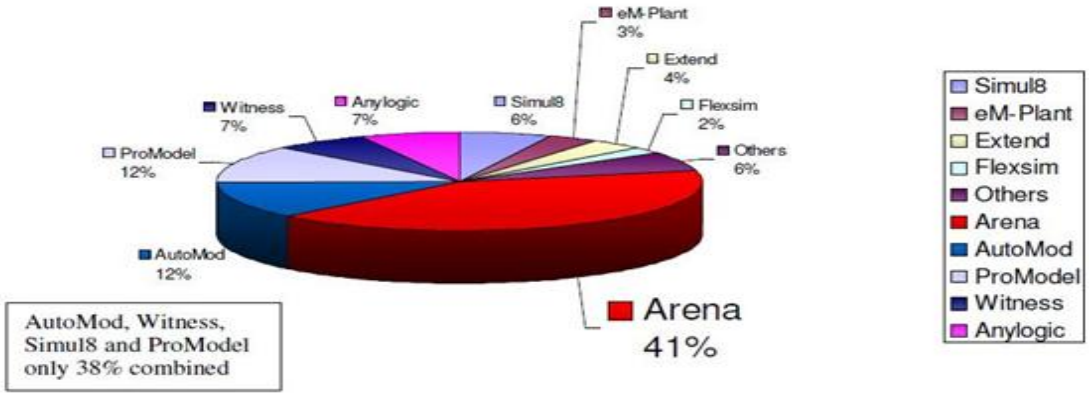


Figure 2.14 WSC ’08 Papers Referring to Simulation Software (Rockwell Automation, 2009).

2.9 Simulation Optimisation Introduction

Simulation optimisation is the approach used when seeking a set of appropriate input values to produce the desired outputs and performance of a simulation model. For the DEDS system, there are several popular optimisation approaches with different suitable areas, such as Ranking and Selection (R&S), Stochastic Approximation (SA), Ordinal Optimisation, etc. Many recent papers from the Winter Simulation Conference (WSC) have focussed on simulation optimisation (Fu, 2001; Ólafsson and Jumi, 2002; Fu et al., 2005; Swisher et al., 2000). In this section, popular simulation optimisations and their applications are briefly introduced and the later Section 2.10 will focus on the optimisations applied to one important area, Job Shop Scheduling Problem (JSSP), of DMSs for SMEs in SCs.

Table 2.2 Popular Simulation Software Packages in SCs

Simulation software	Supplier	Prices	Application Domains	Optimisation Tool	Optimisation Procedures	Website
SIMUL8	SIMUL8 Corporation	$1495(standard),$ 4995(professional)	6 sigma projects, Evaluating new equipment and processes, Sales and operations planning,Capacity analysis and improvement	OptQuest ($495)	Optimisation model by designed controls, constraints, and objective	http://www.simul8.com/
Arena	Rockwell Automation	$2,495 (for basic)	Manufacturing, Supply Chain, Warehousing, Military, Defense, Security, Packaging, Health Care, Medical, Contact Center, Process Reengineering	OptQuest ($995)		http://www.arenasimulation.com
0ProModel	ProModel Corporation	$30.00 (for student)	Transportation system design, Logistics support analysis, Supply chain optimization	SimuRunner Optimization	Simple optimisation by changing input factors	http://www.promodel.com/
WITNESS	Lanner Group, Inc	Dedicated Licensees	Inventory management, Distribution requirements planning optimization	WITNESS Optimizer	Tabu search,six sigma algorithms,hill climb (simulation annealing)	http://www.lanner.com/
AutoMod	Applied Materials, Inc.	$18,000	Material movement systems, Body and paint shops, Operator shifts, Cluster tool robots	Null	Null	http://www.brookssoftware.com/
Extend	Imagine That Inc	$3995(for academic)	Depot Process Model,Net-Centric Logistics Land Platform,Storage locations distribution,a melt iron transportation logistics analysis	Null	Null	http://www.imaginethatinc.com/
Anylogic	XJ Technologies	$485 (for Education)	Supply Chain Management	Null	Null	http://www.xjtek.com/

2.9.1 Stochastic Approximation (SA)

SA is similar to the gradient search but also considers the stochastic features of the model, as its nature is analogous to a gradient and is usually adopted for a DEDS with continuous decision variables. The most common developments of SA are Perturbation Analysis (PA), and Likelihood Ratio/Score Function (LR/SF). PA was developed to solve a DEDS problem at the FIAT Motor Company by extending the previous Infinitesimal PA (IPA), and using linearisation and variation differential equations as its main domain. IPA is usually adopted in the n-dimensional gradient vector of a DEDS and gives an unbiased estimate of the problem faced. IPA's applications include simple queuing networks and networks with multi-class customers, blocking or state-dependent routing mechanisms. Several successful IPA applications are used in SCM (Kapuscinski and Tayur, 1999). However, the IPA algorithm would fail in dealing with multi-class queuing because changing infinitesimal parameters makes finite outputs or events changing queues. Performance measures which are discontinuous would make this algorithm fail. PA enhances IPA by using probability-based extensions and its method is multi-experimental. PA considers simulation to be the mapping of different parameters and uses 'cut, insert, and pasting' techniques under the Markov clock assumption.

In recent years, SA optimisation has had many applications in SCM. The SF approach has been efficiently and successfully applied to an M/M/1 queue (Arsham, 1996). SA has also demonstrated the improved performance of a G/G/1 queue by using PA techniques to explore the gradient (Fu and Ho, 1988). It has been proposed that the gradient descent algorithm with Armijo step sizes should be applied to queuing networks where IPA determines its gradient. In the meantime, Armijo step sizes enable PA to play a role in one simulation run and IPA is adopted in a GI/G/1 queue (Fu, 1990).

2.9.2 Ranking & Selection (R&S) and Multiple Comparison Procedures (MCPs)

In the last decade, R&S and MCPs techniques have become popular and are mainly suitable for DEDS with discrete inputs and a small number of alternatives (traditional sizes<20) (Swisher et al., 2003). R&S is a statistical approach with the aim of choosing the best of a set of k alternatives, while MCPs indicate the difference between the alternatives to them by pair wise comparison. Briefly, R&S selects the best system design and MCPs provide information on the relationships between a set of completive systems (Swisher et al., 2003).

The basic methods of R&S were demonstrated by Bechhofer (1954), and these methods mainly consist of two important procedures: indifference-zone ranking and subset selection. The indifference-zone procedure selects the best solution if the number of solutions is small enough. Normally, selecting the best solution means making the correct decision. However, if the mean vectors of two alternatives are very close, the experimenter cannot determine which one is better, so both solutions are considered correct selections (indifference). The two alternatives would be considered indifference only when the minus of their means is less than the defined indifference-zone parameter δ. In 1988, its application in the configuration of European airports was studied (Gray and Goldsman, 1988). Subset selection procedures, which were introduced in 1956 (Gupta, 1956), aim to produce a random size subset including the best population. These procedures are difficult to apply to DEDS because they require equal and known variances, and so in order to extend its application areas, a system which permits unknown and unequal variances has been developed (Sullivan and Wilson, 1989). Recently, subset selection has been used in conjunction with indifference-zone to select the best system, which has improved statistical efficiency (Nelson and Banerjee, 2001).

The aim of MCPs is similar to R&S procedures, in that they both aim to choose the best system, but MCPs also identify the differences between the alternatives' performance. This procedure consists of three classes, namely all-pair wise comparisons, Multiple Comparisons with a Control (MCC), and Multiple Comparisons with the Best (MCB). All-pair wise comparisons have two approaches - the brute force approach (Fu, 1994) and all-pair wise Multiple Comparisons (MCA) (Tukey, 1953).The brute force approach is required to check the all possible k(k-1)/2 (to k alternatives) confidence intervals and it is difficult to apply to the situation with k>10 because of a large number of confidence intervals. MCA employs an analogous method, but the difference is that it adopts Bonferroni inequality to form a bound on the overall confidence. MCC is a technique which is suitable for a comparison between alternatives and a fixed control (Dunnett, 1955).

Through gradual development, MCC can employ current variance reduction methods and establish a two-stage design and analysis pattern (Bofinger and Lewis, 1992; Yang and Nelson, 1991). MCB (Hsu and Nelson, 1988) still adopts a confidence interval concept for determining the best system, and the concept of parameters can be used to compare each alternative to other systems. Today MCB is a two-stage procedure which is able to use variance reductions methods and is applied to steady-state experiments (Frank and Barry, 1993).

In recent years, it has been observed that R&S and MCPs have already been bound to optimise the simulation model (Hsu, 1984). In order to select the best system, the methods embedded in them are grouped together with relative freedom, such as indifference-zone and MCB (Frank and Barry, 1993), Procedure NM (advanced combined R&S and MCB procedures with common random numbers) (Nelson and Matejcik, 1995). In 2002, Procedure NM was successfully adopted in a clinic design (Swisher and Jacobson, 2002).

2.9.3 Ordinal Optimisation

Ordinal Optimisation aims to help the experimenter find a 'good' solution instead of the best one (Ho et al., 1992) and the goal softening concept is introduced for this aim (Lee et al., 1999). This reduces the number of solutions to an easy-to-manage set and sorts out a 'good enough' solution. 10,000 alternatives have been solved (Ho and Deng, 1994) and its convergence, which follows an exponential rate, is revealed when regenerating a system (Xie, 1997). Obviously, ordinal optimisation is designed for DEDS with a large number of alternatives. Two confidence probabilities which determine confidence levels have been demonstrated (Chen, 1996). One describes the probability of 'good' design in the region selected and the other illustrates the probability of the closeness between the 'good' design and the best design. In the meantime, computing budgets can be managed by following the confidence probabilities established.

2.9.4 Simulated Annealing

Simulated annealing is the heuristic search approach based on the annealing process of metal. This procedure can be considered a hybrid method of local search and usually employs a downhill algorithm unless probability is decreased to 0. In this situation, uphill moves would be applied for increasing probability. In 1992, simulation annealing was adopted in a DEDS (Haddock and Mittenthal, 1992) and since then the PA technique has been linked to simulation annealing (Zeng and Wu, 1993). An asynchronous assembly system has built its simulation model using artificial neural networks and its buffer sizes have been optimised by simulation annealing (Fulya et al., 2002).

2.9.5 Evolution Search

Evolution search is a technique developed from biological systems and a well-known Genetic Algorithm (GA) belongs to this category. GA takes a systematic

family viewpoint to system optimisation by producing better offspring using genetic operations. Usually, GA requires two processes to evolve its next generation. Firstly, in the current generation, individuals would be chosen according to their fitness relating to the objective function value. Keeping all individuals with a certain fitness level, and only keeping the next generation produced are two of the simplest selection methods. Secondly, a generic operator is added to the individuals in order to produce offspring. Crossover and mutation are two common generic operators. In real world application, GA has been successfully adopted in manufacturing systems (Michalewicz and Schoenauer, 2001). Integer programming and GA producers have been combined to design a SC configuration (Truong and Azadivar, 2003).

2.9.6 Tabu Search

Tabu search (Glover, 2001) is a hybrid type of local search and uses two main approaches, such as adaptive memory and responsive exploration. Its list of forbidden moves can be updated dynamically to improve its search efficiency. Usually, its short-term memory forbids searching the space visited and its long-term memory aims to push moves into the space unvisited.

Other simulation optimisations include random walk (Andradottir, 1996a) , rapid learning (Cassandras, 1993), and nested partitions (Shi and Ólafsson, 1997).

2.10 Optimisation of Job Shop Scheduling

2.10.1 Overview of Job Shop Scheduling Problem

Job Shop management is the important component of SCM and its core part, JSSP, is how to build an effective job shop schedule using available resources to achieve predefined goals, such as tardiness, lead time, line balance, etc. Normally, JSSP is one of the hardest problems (NP-Complete) in an optimisation research area and is also one of the most difficult production scheduling problems in SCM. In a UK-based case study,

a DMS mainly focuses on a job shop and is used to verify and validate the proposed a simulation-based decision-making framework. Thus, the optimisation of JSSP is one of the most important research fields in this thesis.

A scheduling of a job shop represents an assignment and timetable of resources and jobs. It can answer the questions of 'when to do which job' and 'how to do a job'. The answer to these two questions is to provide a timetable of the jobs with their resources occupied. This expresses which resources can process which job at what time. Thus temporal and resource constraints are two important issues in a scheduling problem. In a job shop, the other key issue is a precedence constraint which presents that a job should be executed by the resources in a special sequence.

Generally speaking, scheduling is divided into disjunctive scheduling and cumulative scheduling. In a disjunctive scheduling, one resource can be used by only one operation at a time. Otherwise, one resource can handle several operations at the same time in a cumulative scheduling. Another attribute of scheduling is *preemptive* or *non-preemptive*. *Non-preemptive* means a resource cannot be interrupted when it starts to run an operation until it is finished. *Preemptive* represents that an operation B is allowed to be inserted into the duration of the resource occupied by operation A and this results in operation A being stopped until operation B finishes.

The notation of a Job Shop Scheduling Problem (JSSP) is denoted by a three-field $\alpha|\beta|\gamma$ (Allahverdi et al., 2008). The α field represents the environment of the Common Resource (CR) or shop. The β denotes 'details of the processing characteristics', setup time, and other situations of the CR or shop. The γ field expresses an objective which needs to be optimised (minimised). The notation of a three-field $\alpha|\beta|\gamma$ is illustrated in Appendix A.

2.10.2 Disjunctive Graph

2.10.2.1 Disjunctive Graph Representation

A disjunctive formulation is used to model a JSSP .The basic job shop model describes as follows (Pinedo, 2008):

There exist m machines M={m1,…,mm} and n jobs J={j1,…,jn} in a job shop. An operation O_{ij} with an operation time p_{ij} denotes the operation of job j on machine i. PREC consisting of conjunctive and disjunctive constraints presents the precedence constraints of all operations. M_j denotes the machine set processes all operations of job j. If Operation ij precedes operation kj of job j, operation kj is started after finishing operation ij which denotes a conjunctive constraint using $Tb_{kj}-Tb_{ij}>=p_{ij}$ and $Te_{ij}=Tb_{ij}+p_{ij}$. The formulation of this conjunctive constraint is as follows:

$$Te_{ij} <= Tb_{kj} \quad (i,k \in M_j \text{ and } i \neq k) \qquad (2.1)$$

where Tb denotes the begin time of an operation, Te represents the end time of an operation, p denotes the operation time of an operation. A unary machine i can only handle one job j at the same time, which is a disjunctive constraint. If operation ij precedes operation il or operation il precedes operation ij on machine i (Jobs j and l are two independent jobs), it is denoted by $Tb_{ij}-Tb_{il}>=p_{il}$ or $Tb_{il}-Tb_{ij}>=p_{ij}$. Then set $Te_{ij}=Tb_{ij}+p_{ij}$ and $Te_{il}=Tb_{il}+p_{il}$, the formulation of this disjunctive constraint is as follows:

$$Te_{ij} <= Tb_{il} \text{ or } Tb_{ij} => Te_{il} \quad (i,j) \text{ and } (i,l) \in O \qquad (2.2)$$

where O denotes operation set.

A disjunctive formulation of a JSSP is described as follows (Pinedo, 2008) :

C (v1,…,vn) Cost Function (Objective Function)

Subject to

$Te_{ij}, Tb_{ij}=(0,\infty)$ $(i,j) \in O$

$Te_{ij} <= Tb_{kj}$ $(i,k \in Mj \text{ and } i \neq k)$

$Te_{ij} <= Tb_{il}$ or $Tb_{ij} => Te_{il}$ $(i,j) \text{ and } (i,l) \in O$

A disjunctive graph G=(N,CUD) is a typical graph representation for instances and solutions of a job shop scheduling problem developed by Roy and Sussman in 1964 and it is able to express a disjunctive formulation from a graph view. Most of effective optimisations of Branch and Bound (B&B) are successfully developed based on a disjunctive graph. N implies a set of nodes (vertices) which represent the operations of the jobs and there exist two additional nodes: a source U (start node) and a sink V (end node). There are two types of constraints, namely conjunctive C and disjunctive D in a disjunctive graph. All conjunctive constraints need to be achieved, but at least one and not all disjunctive constraints need to be satisfied. C expresses the set of directed arcs with a (i,j)→ (k,j) PREC constraint of job j. D represents the set of undirected arcs of (i,j) and (i,k) of resource i when jobs j and k are executed on the resource i respectively.

Figure 2.15 shows a disjunctive graph that represents all instances of a JSSP. In the graph, node (i,j) (N_{ij}) represents operation ij (machine i job j). A solid arc called a conjunctive one shows the route of a job. A conjunctive arc from node ij to node kj means job j should finish operation ij firstly, then do operation kj. Disjunctive arcs link all pairs of two operations using the same resource. A disjunctive directed arc from node ij to node ik expresses the resource (machine) i deals with job j first, then handles job k. Operation time p_{ij} of operation ij is put on the arcs which directs out from this operation. This results in the longest path from the *source* to the *sink* being able to represent the makespan of the schedule.

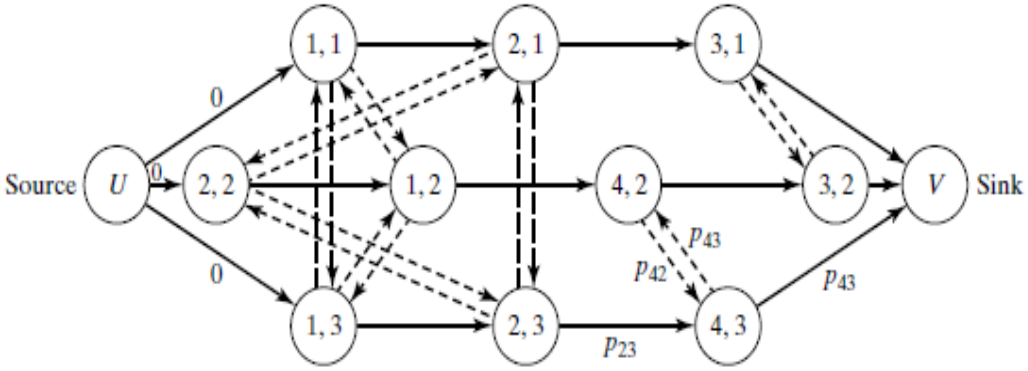


Figure 2.15 Disjunctive Graph of a JSSP with 4 Machines and 3 Jobs (Pinedo, 2008)

When all undirected disjunctive arcs are fixed to the directed disjunctive arcs in a disjunctive graph without any cycle, the graph is called a complete selection (**precedence graph**) which represents a feasible instance of JSSP, denoted as (N, CUS). S means a set of directed disjunctive arcs and S∈D. The instance can define two issues of JSSP. Firstly is the sequence of operations in any job and secondly is the sequence of operations on any resource (machine). In a complete selection, several parameters are calculated as follows (Pinedo, 2008):

r_{ij} (the release time of operation ij) = the sum of operation time on the longest path from the source to operation (node) ij

C_{max} (makespan) = the sum of operation time on the longest path from the source to the sink.

d_{ij} (local due date of operation ij) = the sum of operation time on the longest path from the operation (node) ij to the sink

r_j: the release time of operation j which represents the ready date and the earliest time the job j can be handled.

d_j: the due date of job j which expresses the shipping time of job j which has been promised to the clients.

$$r_{ij}=\max(r_{Mp(ij)}+p_{Mp(ij)},r_{Op(ij)}+p_{Op(ij)})$$

where p_{ij} represents the operation time of operation ij, $Mp(_{ij})$ denotes a predecessor operation of operation ij on a machine task, Op(x) represents a predecessor operation of operation ij on an operation task.

2.10.2.2 Critical Path

A slack job means that C_{max} is not increased by postponing this job and if C_{max} rises then the job is a critical job. A critical path is defined as the set of critical jobs of the schedule. In a complete selection (precedence graph**)** of the disjunctive graph, a critical path is its longest path and it means the length of the critical path equals to C_{max}. On a critical path, a critical block is defined as a set of consecutive operations on the same machine. The equations of a critical path are as follows (Schmidt, 2001):

$$r_{u0}=0$$

$$LE_{ui}= r_{ui}+p_{ui}$$

$$r_{ui+1}=LE_{ui}$$

where u0 expresses the start node on a critical path, ui denotes operation ui on a critical path, r_u represents the release time (earliest begin time) of operation u, LE_u denotes the latest end time of operation u, p_u denotes the operation time of operation u.

2.10.2.3 Applications of a Disjunctive Graph

The disjunctive graph representation is adopted for various optimisation algorithms for JSSP. One of optimisation algorithms is shifting bottleneck algorithms belonging to a decomposition strategy and this algorithm adopts the advantage of a disjunctive graph to easily find a critical path. Then the applications of the representation move to Tabu Search (TS) and simulated annealing optimisation fields. After that, it is implemented in the hybrid optimisation algorithms.

There are several recent applications and developments of the disjunctive graph

concept as follows:

Laarhoven et al.(1992) developed a randomised iterative optimisation algorithm depending on a simulated annealing algorithm for a JSSP and took a characteristic of a critical path of a disjunctive graph to construct a neighbourhood structure. In fact, the new neighbouring solution was obtained by swapping two successive operations on a machine of a longest path. The new graph generated by a swap method was proved to be *cycle free* and this means the new neighbouring instance is a feasible solution.

Blazewicz et al. (2000) proposed a graph matrix with three parts: a neighbourhood matrix, predecessor and successor lists, which is able to represent a disjunctive graph. A higher efficiency of the proposed graph matrix compared to the classical graph representations was proved through computational experiments.

Oey and Mason (2001) adopted a modified disjunctive graph to represent an instance of a complex JSSP in a semiconductor manufacturing field. The job shop owned a group of machines with similar processing functions, called a 'tool group'. The 'tool group' can deal with a job in parallel and one issue of sequence-dependent setup time needs to be considered in the job shop. In order to represent a solution to the complex scheduling problem in the job shop, the concepts of re-entrant flow and 'dummy' nodes were introduced to extend the capacity of a traditional disjunctive graph for the type of the complex scheduling problem.

Aiex et al. (2003) developed a hybrid parallel Greedy Randomized Adaptive Search Procedure (GRASP) enhanced by a path-relinking based on a disjunctive graph concept of a JSSP. A swap method between two consecutive operations on the critical path was adopted by Aliex as a neighbourhood strategy for his local search algorithm.

Essafi et al.(2008) developed a Genetic Local Search (GLS) algorithm from a visited local optimal algorithm based on a disjunctive graph representation with n

‘dummy’ source nodes. Reversing an arc (swap) on a critical path for minimising the makespan was modified to reverse arcs of all jobs for minimising the total weighted tardiness.

Gao et al. (2007) combined a GA and a bottleneck shifting for a flexible JSSP with three objective functions: namely minimum makespan, minimum machine workload and minimum total workload. A disjunctive graph was adopted for finding a longest path (critical path) of a feasible solution and swapping two adjacent critical operations by a bottleneck shifting algorithm.

Recently, the disjunctive graph representation has been developed as a blocking disjunctive graph for finding a ‘good’ solution of a flow shop scheduling problem with blocking. This type of scheduling problem expresses the situation of lack of buffers among machines, and a machine should be blocked when its successor machine has not completed a job without any buffer between them.

For representing the concept of blocking, a disjunctive arc with zero weight expresses a blocking constraint between two nodes in a blocking disjunctive graph (Martinez et al., 2006). This representation is suitable for two special cases of a flow shop scheduling problem with blocking, namely Release when Starting blocking (RSb) and Release when Completing blocking (RCb).

Mascis and Pacciarelli (2002) adopted an alternative graph formulation to modify a disjunctive graph for JSSP with blocking. Computational experiences showed neither a shifting bottleneck algorithm nor TS is an effectively local search method for this complex problem.

Martinez et al. (2006) developed Mascis’s concept and modified a disjunctive graph for representing a flow shop scheduling with a blocking constraint. Martinez et al. focused on analysing complex situations of the flow shop scheduling problem with

RCb cases using a blocking disjunctive graph because there were plenty of existing research works in RSb. The problem of F4|Rcb|C_{max} was completely discussed and F3|Rcb|C_{max} was only given a polynomial formulation. Martinez et al. also proved the type of scheduling problem with more than five machines is NP-hard.

However, there are only a limited number of research works of modifying the disjunctive graph representation for a complex batching JSSP.

2.10.3 Relevent Optimisation Methods

2.10.3.1 Dispatching Rules

Dispatching (priority) rules are simple purpose procedures for scheduling problems and different rules aim to optimise different objective functions in different environments. When the Common Resources (CRs) in a job shop are available, the relevant operations are ranked according to their priorities using a simple dispatching rule or a combination one consisting of several rules. Then the operations are able to be assigned to the CRs order by the priorities. In a relative simple environment, dispatching rules is an efficient approach to find a 'good' schedule with one single objective function. The definitions of popular dispatching rules are as follows (Pinedo, 2008):

- Service in Random Order (SIRO): The next job is randomly chosen for the available CRs.
- Earliest Release Date first (ERD): First coming job is firstly assigned to the available CRs.
- Shortest Processing Time (SPT): A job with the shortest operation time is the next one which is assigned to the available CRs.
- Weighted Shortest Processing Time (WSPT): A job with the shortest weighted processing time is assigned next.
- Longest Processing Time (LPT): A job with the longest operation time is

assigned next to the available CRs.

- Earliest Due Date first (EDD): A job with the earliest due date is assigned next to the available CRs.
- Minimum Slack (MS): A job with the minimum slack (maximum(d_i-p_i-t,0)) at time t is assigned next.

Table 2.3 (Pinedo, 2000) shows the objectives pursued by several popular dispatching rules.

Table 2.3 Objectives of Several Popular Dispatching Rules (Pinedo, 2000)

	RULE	DATA	OBJECTIVES
Rules Dependent on Release Dates and Due Dates	*ERD*	r_j	Variance in Throughput Times
	EDD	d_j	Maximum Lateness
	MS	d_j	Maximum Lateness
Rules Dependent on Processing Times	*LPT*	p_j	Load Balancing over Parallel Machines
	SPT	p_j	Sum of Completion Times, WIP
	WSPT	p_j, w_j	Weighted Sum of Completion Times, WIP
	CP	p_j, *prec*	Makespan
	LNS	p_j, *prec*	Makespan
Miscellaneous	*SIRO*	–	Ease of Implementation
	SST	s_{jk}	Makespan and Throughput
	LFJ	M_j	Makespan and Throughput
	SQNO	–	Machine Idleness

2.10.3.2 Tabu Search

The Tabu Search (TS) was developed by Glover (1989, 1990) based on a hill climbing heuristic. TS is a heuristics approach to search a 'good' solution by iterative moves from one solution to another solution. The move represents a transformation from a solution to the next solution. The neighbourhood structure is defined a subset of new solution moved from an old solution. TS is a guided search with a capacity of preventing cycling and searching back the solution visited in previous tn (the length of

the tabu list) steps using a tabu list. A tabu list is a class of the short–term memory which stores the recent search history to avoid visiting the recent solutions again. This list is a limited queue and is normally adopted by a First In First out (FIFO) rule to store the new move and put the old move out when the list is full. Every move taken should be put into the list. It owns two features of adaptive memory and responsive exploration. The adaptive memory enables it to earn a more efficient search than the algorithms embedded with memoryless designs, such as semigreedy, original simulated annealing, original GA, and one with rigid memory designs like original B&B. The responsive exploration can collect more useful clues than a good random search even depending on a bad strategic one. These clues are used to improve the quality of the adopted strategy and find new promising search spaces. The method of aspiration criterion aims to accept attractive solutions which are forbidden in the tabu list and is also an important strategy to find the 'good' solutions overridden by the tabu list.

Figure 2.16 (Zhang et al., 2007) shows the general workflow of TS and it begins with an initial solution (seed solution). Then a special neighbourhood strategy is adopted to construct neighbours (candidate list) based on the initial solution. The given aspiration criterion are provided to determine whether the 'best' neighbour satisfies or not. If satisfying the criterion, it is considered as a seed solution to run the next turn. If not, the move from the seed solution to the 'best' neighbour is checked whether or not existing in the tabu list. If not, the 'best' neighbour is taken as a seed solution for the next turn and updates the tabu list. If existing, the second 'best' neighbour is taken to check the aspiration criterion. The procedures are run until satisfying termination criterion. Then an optimum solution will be provided.

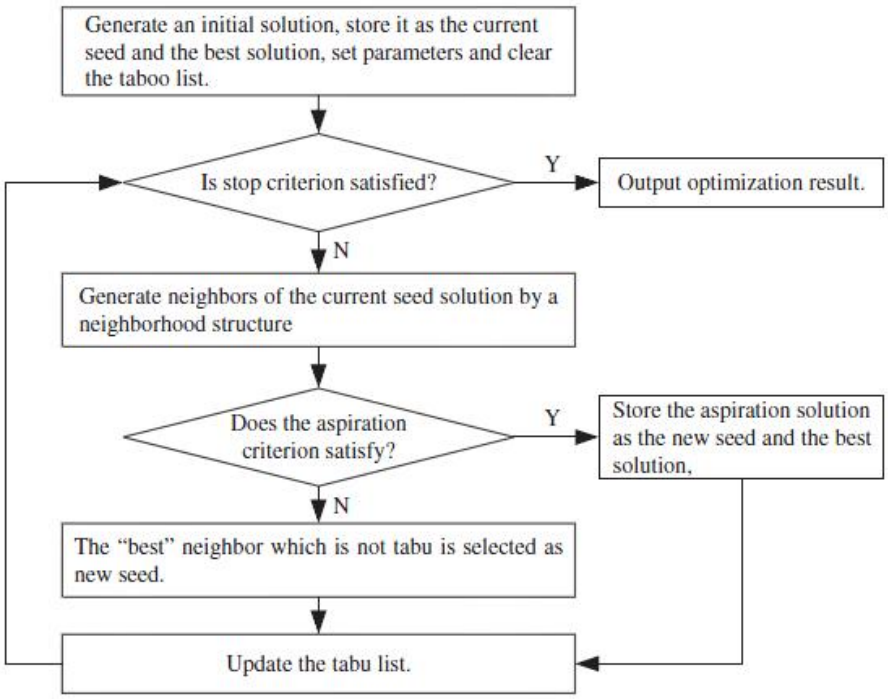


Figure 2.16 General Workflow of TS (Zhang et al., 2007)

In TS, the length of a tabu list is an important factor which affects the search efficiency. A long list causes redundant restrictions for optimisation and a short list is not able to effectively avoid revisiting recent solutions. Now, the factor setting still remains an open question. In the past, the length of the tabu list is fixed and static. Recently, the technique of the dynamic tabu list is developing and the popular methods for JSSP with n jobs and m machines are introduced as follows:

Grabowski et al. (2004) set L=6+(n/10m) and the length of the tabu list is taken L or 1.5L determined by a changing iteration number.

Zhang et al. (2007) proposed an experiment equation to calculate the length of a tabu list: L=10+n/m. Then defining Lmin=L and (Lmax=1.4L when n<=2m) or (Lmax=1.5L when n>2m). The random value between Lmin and Lmax was adopted as the length of a dynamic tabu list for improving the efficiency of avoiding a cycle.

Zhang et al. (2008) defined L=10+n/m and the number randomly chosen between L and L+2 is taken as the length of the tabu list for the TS algorithm.

2.10.3.3 Neighbourhood Structure

The key of moves of TS is to construct a neighbourhood structure. Various neighbourhood strategies of TS are developed from swap operations on a critical path in a disjunctive graph representation over the last 15-20 years.

In order to introduce these neighbourhood strategies, the definitions of several notations relating to swap operations on a critical path should be described as follows:

u: a operation on critical path

Mp(u): a predecessor operation of u on a machine task

Ms(u): a successor operation of u on a machine task

A disjunctive arc {Mp(Mp(u)),Mp(u),u,Ms(u),Ms(Ms(u))}is assumed on a critical path in a disjunctive graph. This arc links all consecutive operations to the same machine on a critical path called a critical block.

There are two criteria (Schmidt, 2001) to describe the important characteristics of the neighbourhood strategies. One is the feasibility and the other is the connectivity. The feasibility criterion defines the new neighbouring solution generated by this neighbourhood strategy is always a feasible solution. The connectivity criterion denotes a global minimal solution is able to be achieved by the neighbourhood strategy. To find a global minimal solution in a complex JSSP normally takes considerable time and costs. Consequently, a ‘good’ solution is normally used to replace the global minimal solution. Several popular state-of-the-art neighbourhood structure strategies are discussed as follows:

Nz1 (van Laarhoven et al., 1992) is a simple neighbourhood strategy and focuses on any pair of two successive operations on a critical path. It is able to construct a new neighbourhood structure by swapping the two operations (reverse the directed disjunctive arc between two operations) as shown in Figure 2.17. Nz1 can preserve the feasibility and connectivity issues.

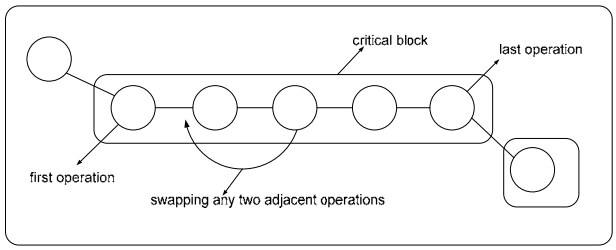


Figure 2.17 Nz1 Neighbourhood Strategy

Nz2 (van Laarhoven et al., 1992) reduces the permutations of Nz1 and only deals with two pairs of operations: one is the first operation of a critical block and its successor [Ms(u),Ms(Ms(u))] ; the other is the last operation of a critical block and its predecessor [Mp(Mp(u)),Mp(u))]. The internal pairs of operations are ignored because swapping them cannot contribute to C_{max} as shown in Figure 2.18. However, N2 only satisfy the feasibility criterion.

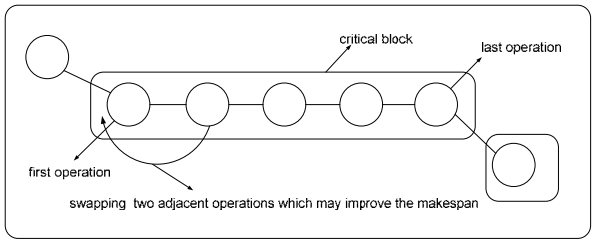


Figure 2.18 Nz2 Neighbourhood Strategy

Nz3 (Dell'Amico and Trubian, 1993) runs three iteratives of the swap method of two operations on a critical block at a time and is able to move the operation into the very beginning of a critical block or into the very end as shown in Figure 2.19.

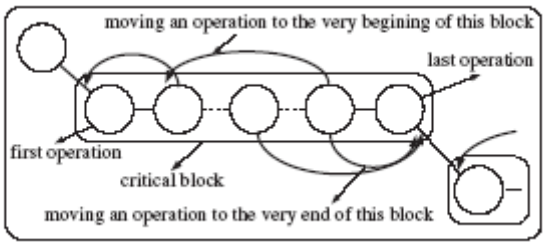


Figure 2.19 Nz3 Neighbourhood Strategy (Zhang et al., 2007)

Nz4 (Nowicki and Smutnicki, 1996) focuses on the first pair of two operations and

last pair of two operations of all critical blocks. It swaps the first pair of two operations and/or the last pair of two operations of one or several critical blocks at the same time as shown in Figure 2.20.

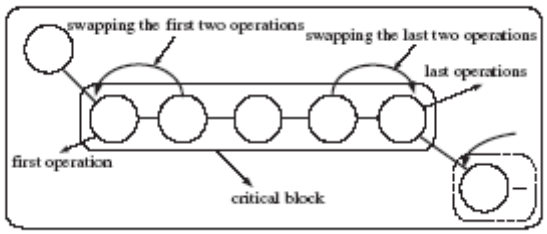


Figure 2.20 Nz4 Neighbourhood Strategy (Zhang et al., 2007)

Nz5(Balas and Vazacopoulos, 1998) still concentrates on interchanges of the operation and the first operation or the last operation of a critical block similar to Nz3 is shown in Figure 2.21. Compared to Nz3, Nz5 requires at least one first operation or last operation should be contained in a serial of interchanges.

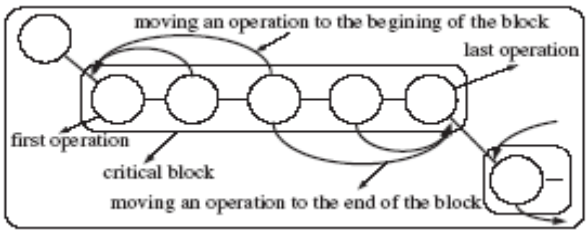


Figure 2.21 Nz5 Neighbourhood Strategy (Zhang et al., 2007)

Nz6 (Zhang et al., 2007) defines a new neighbourhood structure of TS by inserting the first or the last operation of a critical block into the block's internal space as shown in Figure 2.22. This new neighbourhood strategy was proved to be effective compared to the previous popular five neighbourhood strategy by a computational study.

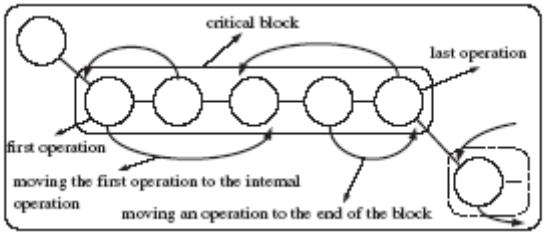


Figure 2.22 Nz6 Neighbourhood Strategy (Zhang et al., 2007)

2.10.3.4 Tabu Search for JSSP

TS is a popular and one of the most efficient optimisation methods adopted to find a 'good' solution for JSSP. Taillard (1994) first introduced TS into the JSSP field to minimise C_{max}. The techniques of a dynamic tabu list and a frequency-based memory were embedded in TS. For many 'rectangular' search problems, computational experiments showed TS is more efficient than simulated annealing, shifting bottleneck heuristic and other optimisation algorithms.

Schmidt (2001) adopted TS for $Jm|ST_{sd}$ and demonstrated TS is able to be altered to extend its application area for JSSP with sequence dependent setup time. A poor initial solution obtained by the bidirectional list scheduling algorithm was also efficiently optimised by TS.

Zhang et al. (2007) developed a new neighbourhood strategy which enables TS to solve $Jm||C_{max}$ more efficiently. A computational study proved the high efficiency of this new neighbourhood strategy compared to the other five neighbourhood strategies.

El-Bouri et al. (2007) proposed four heuristic variants combined of adaptive memory programming, simulated annealing, and TS for a classic JSSP. Compared to a traditional simulated annealing algorithm, four versions of the heuristics were evaluated by a computational study.

Zhang et al. (2008) integrated simulated annealing and TS for $Jm||C_{max}$ and a simulated annealing algorithm was adopted in a *recency-based* memory mechanism for generating an elite solution. TS was implemented to search a 'good' solution based on the elite solution.

Li et al. (2010) developed an effective Hybrid Tabu Search Algorithm (HTSA) to solve a flexible JSSP with three objectives: C_{max}, the total workload of machines, and the workload of the critical machines. A variable neighbourhood search based on a

neighbourhood strategy of one swap and two insert was introduced into TS. The computational experiments confirmed HTSA has better search efficiency than other recent proposed heuristics.

2.10.4 Job Shop batching scheduling with unrelated parallel machines, setup time, job splitting (lot-streamming)

2.10.4.1 Overview

The UK-based case study is a manufacturing enterprise with more than 20 years professional experience in the high-precision sheet metal machining field. It has more than 50 customers and processes hundreds of different types of products to satisfy its customers' specific and individual requirements. Every product owns it special processing route on more than 30 unrelated machines. Different labourers may have different setup and processing times in completing the same operation of one job on the same machine. Its JSSP belongs to a job shop batching scheduling with unrelated parallel machines, setup time, job splitting (lot-streaming) ($Jm|ST_{si,b}$). In this case study, the job shop scheduling has the characteristics of the dynamic job release. The jobs are not available at time zero for operations in the planning horizon and depend on the dynamic time of incoming orders. Based on a scenario of more than 30 machines and 100 jobs of the job shop, a traditional JSSP with this scenario is NP-hard in a very strong sense (Pinedo, 2008). In the case study, the various factors of unrelated parallel machines, sequence independent setup time and lot-streaming make this JSSP more complicated. Thus, the JSSP of the case study is strongly NP-hard. Further details of the UK-based case study will be discussed in Chapter 4 and Chapter 6.

To the best of our knowledge, no previous research work on this type of JSSP is published so far. Allahverdi et al. (2008) surveyed more than 300 research papers on scheduling problems relating to setup time since 1999 and classified the survey by single machine, parallel machines, flow shop, job shop and open shop. In these

classifications of job shop and open shop, only a small number of researches (Sotskov et al., 1999; Low et al., 2004; Aldakhilallah and Ramesh, 2001; Artigues and Roubellat, 2001) studied $J|ST_{si,b}$ or $J|SC_{si,b}$ of batching scheduling even though there are approximately 40 papers published per year on scheduling optimisation. However, the key factors of unrelated parallel machines and job splitting are still not added into their works. Gen et al. (2009) provided a comprehensive survey for optimisation algorithms in an integrated manufacturing system. In his flexible job-shop scheduling section, five representation methods: parallel machine, parallel job, kacem' approach, priority rule-based and multistage operation-based representations were introduced and none of them are suitably adopted for the scenario of JSSP with job splitting. A study of Potts and Kovalyov (2000), Lauff and Werner (2004), Gordon et al. (2002), Ribas et al. (2010), Ma et al.(2010), and Ruiz and Vázquez-Rodríguez (2010), indicates this type of complex JSSP was not apparent in these review papers. From these previous surveys and reviews, the following Sections 2.10.4.2 and 2.10.4.3 of the literature review is divided into two sub-sections consisting of lot-streaming and unrelated parallel machines.

2.10.4.2 Lot-Streaming and Batch Scheduling

Lot-Steaming (LS) called job splitting technique is a popular method from Optimized Production Technology (OPT) proposed by Goldratt in 1980 (Huang, 2010). LS is to split a job into a number of sub-jobs (sublots) for accelerating production, overlapping operation, reducing lead time and cost (Botsali, 1999; Huang, 2010). Today, LS has become an important topic in operation research field. Lin and Jeng (2004) developed a three-phases including sequencing, dispatching, and batching heuristic algorithm using the dynamic programming algorithm for solving an integrated problem of $Pm|\ ST_{si,b}|L_{max}$ and $Pm|\ ST_{si,b}|\sum U_j$ based on a job-focused heuristic approach. Chang and Chiu (2005) reviewed 100 recent research works of LS and classified these works into three main categories: single-product time-related,

multi-product time-related, and single-product cost-related problems. In the multi-product time-related section (more similar to the UK-based case study than other two categories), only four research papers of JSSP have been published, thus JSSP with LS is still a complex open question. A Mixed Integer Programming (MIP) method was adopted as a model representation and a two-level heuristic algorithm was used to determine *sublots* and their sizes in the work of Dauzere-Peres and Lasserre et al. (1993, 1997). Then a shifting bottleneck algorithm was applied to solve a JSSP with every *sublot* of each job considered as an independent job. An Integer Programming (IP) was used to build the model and an integrated backward dynamic programming with a heuristic method was applied as an optimisation algorithm for JSSP (Jin et al., 1999). Jeong et al. (1999) described a method with three steps: the first step is to gain a initial solution by Shifting Bottleneck Procedure (SBP); the second step is to split the jobs; the third step is to increase the performance of the solution. The four research works outlined on JSSP with LS have not considered unrelated parallel issues.

For parallel machine scheduling with LS, KIM et al. (2004) proposed a two-phase optimisation algorithm for minimising total tardiness with considering setup time. In the first phase, an initial solution is obtained by the available combination algorithm of two SPT and EDD dispatching rules. In the second phase, three types of rules: job selection, subjob selection, and machine selection rules are adopted for splitting job and rescheduling. Yalaoui and Chu (2003) demonstrated an optimisation algorithm for parallel machine scheduling with the objective function of makespan and two issues of LS and sequence-dependent setup time were considered. Firstly, a Travelling Salesman Problem (TSP) ,which is reduced and transferred from the original scheduling problem, is solved by Little's approach (Yalaoui and Chu, 2003). Secondly, the solution obtained in the first step is extended to a feasible solution for the original problem.

Chan et al. (2008) used GA and dispatching rules to solve Assembly JSP (AJSP) with LS. In 2009, a hybrid GA called LSGAVS with two steps for JSSP with LS was

developed (Chan et al., 2009). The first step is to split lot determined by the sub-lot number and the sub-lot size using the model LSGAVS with a varying size splitting function. The second step is to apply an optimisation procedure for JSSP after assigning sub-lots. The computational results proved the LSGAVS with varied size splitting is better than the model LSGAES with equal size splitting for JSSP with LS.

Huang (2010) proposed the Ant Colony Optimisation (ACO) algorithm for JSSP with LS considered a multi-objectives function including the factors of stock, machine idle and carrying costs. Huang analysed four different LS methods, discussed and compared the performance relating to the above factors in two scenarios of small problems and big problems.

2.10.4.3 Unrelated Parallel Machine Scheduling

The parallel machine scheduling focuses on the scenario with n independent jobs processed on m parallel machines and only considers a particular situation of m machines spend have same operation time units p_j in completing a particular job. Furthermore, p_j is a finite and fixed positive integrate number only corresponding to jobs and different jobs may have different operation time units. All m machines own the same processing speed and capacity for a job.

Unrelated parallel machines represent m different machines in parallel and these machines own different processing speeds for different jobs. It means the processing speed is determined by two factors: machines and jobs.

In the study of Low (2005), only four research reports considered setup time in a Flow Shop with Multiple Processors (FSMP), which presents operations are able to be processed in parallel in a flow shop, however, none of these researches introduced a LS technique. Low illustrated a MIP method was not able to handle FSMP with more than four jobs and a heuristic based on a simulated annealing algorithm was proposed for

FSMP. In Chen's (2009b) review paper, they found that only Low's research work was related to FSMP with unrelated parallel machines and developed a three stages approach for this type of a scheduling problem. The three stages are an upstream stage, a bottleneck stage and a downstream stage. Based on the three stages concept, a Bottleneck-Based heuristic (BBFFL) (2009a) with a bottleneck stage was proposed and demonstrated for solving a candidate Flexible Flow Line (FFL) problem. The heuristic composed of three steps: identification, generation and insertion, focuses on scheduling at the bottleneck stage to wholly optimise the scheduling. Several dispatching rules and TS were used for comparing the performance of the optimal solution.

2.10.4.4 An Integrated Optimisation for JSSP of the Case Study

Based on the literature review of optimisation of JSSP, there is no existing optimisation algorithm for Jm|STsi,b with LS. Therefore, a novel suitable search algorithm is required to be developed for the case study. Before developing a novel optimisation, a disjunctive graph is chosen as a representation tool for JSSP of the case study. The complexity of the case study indicates that the traditional disjunctive graph needs to be altered and modified for representing the JSSP. TS is one of the highest efficient search algorithms, and Taillard (1994) proved TS is the best one among the several popular optimisation algorithms. Consequently, the Conflicted Key Points Optimisation (CKPO) is developed from TS to find a 'good' solution by the proposed neighbourhood strategy and rescheduling technique, and this proposed algorithm focuses on how to assign the jobs into a CR set. Another proposed optimisation algorithm, called Dispatching Optimisation Algorithm (DOA), is able to handle the partial scheduling relating to LS of the case study. Further details of CKPO and DOA will be discussed in Chapter 4.

2.11 Conclusions

This chapter has introduced the background of developing a simulation system

from the literature review. The relevance of a discrete-event system was the focused topic because the proposed simulation system belongs to discrete-event systems. In the simulation worldview section, the popular *three phase* approach was emphasised for developing a novel worldview of the proposed system. In order to choose a suitable software for developing the simulation system, several popular simulation software packages are compared and the result shows WITNESS is one of the most suitable software for developing a simulation framework for SMEs in SCs.

This chapter also discusses JSSP and provides background knowledge to support an optimisation block for the UK-based case study. A popular representation of JSSP, namely disjunctive graph representation, is outlined and its method of determining a critical path is introduced. Tabu Search (TS) optimisation algorithm and its related knowledge are introduced particular in the neighbourhood strategy. The JSSP of the case study is a job shop batching scheduling with unrelated parallel machines, setup time, jobs splitting (lot-streaming) ($J_m|ST_{si,b}$). There appears no previous published research work on this type of JSSP from the literature review on JSSP. Therefore, the research topic on this type of JSSP is split into two subtopics, namely job splitting techniques and unrelated parallel machines scheduling method for further literature study. There still appears no optimisation algorithm is suitable for this type of JSSP from the reviews on these two subtopics. Consequently, the proposed CKPO and DOA are required to be developed for solving JSSP of a case study using this investigation.

Chapter 3 The Adaptive Simulation-Based Decision-Making Framework for a Small and Medium Sized Supply Chain

3.1 Introduction

Chapter 2 has surveyed the current popular SC simulation frameworks which are classified into two types, namely concept and practice frameworks.

The concept framework is going to two different classifications. One tries to use simulation concepts to cover all the fields of a SC, such as material-flow, information, business process, decision process, etc. This makes the system very complicated and it is difficult to develop a practical system based on this framework. Umeda's framework (Umeda and Jain, 2004) is a classical case of this type of framework, however there is not a practical system currently developed from this framework. The other framework is so simple that it lacks necessary flexibility. For example, Lee's framework (Lee et al., 2002) only describes a simple SC with one customer, two distributors, two factories, and two suppliers. This framework just considers fixed relationships among these participants on the top layer and does not explore the process relationships on the bottom layer. This framework is so abstract and simple that it is difficult to be adopted in practical situation.

The practice frameworks are normally developed from special industrial fields and own the obvious characteristics of these domains. This leads to a difficult task to extend these frameworks to support a generic field. The Mönch's framework (Mönch et al., 2003) is designed for a semiconductor manufacturing and is mainly focused on the flow shop. It is extremely difficult to extend applications of a flow shop to a job shop which is a more generic existing case in most industry. Consequently, an adaptive simulation

framework should have two features: hierarchical structure and flexibility configuration. A new adaptive simulation framework with these two characteristics is required for designing DMSs for SMEs in SCs.

This chapter introduces the entire process of developing a flexible simulation framework consisting of hierarchical network, simulation block, cell simulator, and flexibility configuration of simulation model. A 4Ps' closed-loop control and multi-layer control for designing the framework and structure are introduced in the theoretical background. The hierarchical network extends the applications of the simulation framework from an internal SC to a complete SC and accurately represents the relationship among the participants in SCs. The structures of simulation block and cell simulator are illustrated in detail. Flexibility configuration is divided into several types and design methods are given according to different types respectively.

3.2 Theoretical Background

3.2.1 4 P's Closed-loop Control

4P's Closed-loop Control is developed from a 3P theory of software development. In 3P theory, people, process, and product are the key factors in developing software. The 4P theory follows these three key factors, however new factor, performance, is added as a feature of the simulation results. In an operational cycle, people make a set of processes (rules or parameters) and run a simulation system to make visual products. A simulation system can provide performance (results) based on inputting scenarios to people and processes which can be gradually altered to achieve the aims they want, as shown in Figure 3.1. An entire operational cycle is a closed-loop control cycle, which has the advantages of closed-loop control such as maintaining evenness in a dynamic environment, accurate tuning and making results achieve to objectives, and ease in tuning against disturbances (Lipták, 2003) . The 4P's Closed-loop Control will be applied to design the architecture of a simulation block.

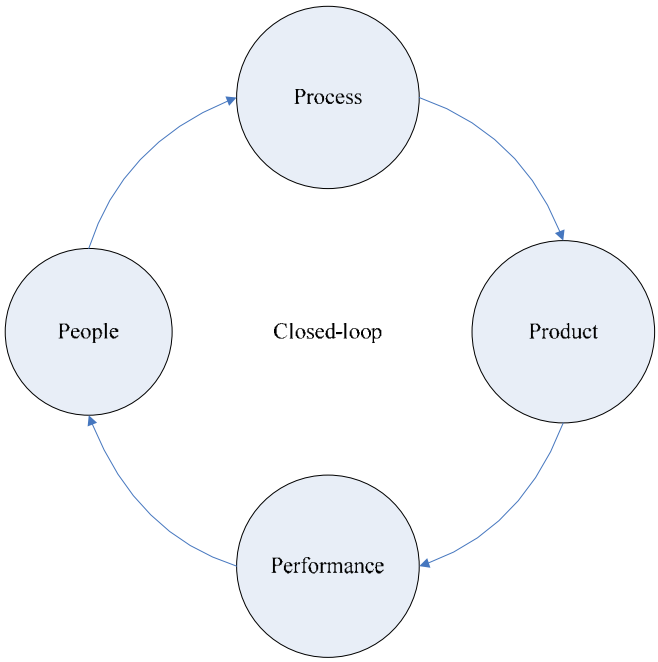


Figure 3.1 Diagram of 4 P's Closed-loop Control

3.2.2 Multi-Layer Control

The relevant decisions of a SC normally are divided into three categories: strategic, operating, and control (Gaither and Frazier, 2002). A strategic decision normally is a long-term plan made by management groups. An operating decision is usually based on a weekly or monthly decision by an engineering group guided by a management group. Work groups and engineer groups take control of the control decisions which own a real-time characteristic.

Within a business, different people want a simulation system to provide the information relevant to them. This information therefore has to be divided into different layers and this introduces the concept of multi-layer control. Figure 3.2 shows a model of multi-layer control and the different information required by staff holding different positions. A management group looks to find out the long-term profit for a company. A sales team calculates the profit of a current order (called a short-term profit) and wants to know the cost and processing time of an order, not the cost and processing time of an individual product. A technical team builds routes for machining products and is

concerned with the process time and cost of a product. A work group usually wants the time of the processes operated by them. Therefore, different layers need different information and controls, and the information would be summed from bottom to top. This gives a tree structure of relationships of objectives located in different layers. An order is able to take one of several routes and a route can occupy several common resources. Different choices result in different performance like costs, lead time, and profits.

A simulation mode is designed to provide relevant information into a work group and transfer simulation data to the database. The other groups can take the relevant data from a database by related interfaces. This concept will be applied in designing a simulation block to the functions assignment of databases and simulation models.

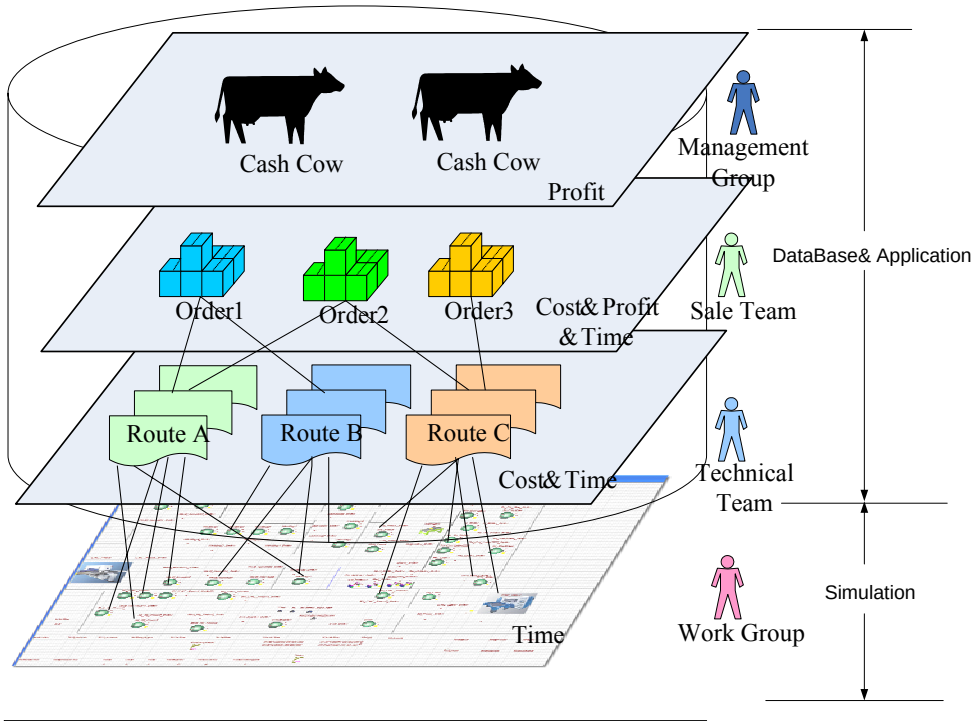


Figure 3.2 Diagram of Multi-layer Control

3.2.3 Pull Model

Machine and labour are the two main common resources of a company. In a

general simulation view, machines represent all resources operated by labourers, such as vehicles, operation rooms, even factories, etc. When a machine receives a batch of products, it matches labourers with the ability to operate these products. After completion, the machine takes a batch of products into its buffer or store room. Then products are left there until other machines from the next process pull them as shown in Figure 3.3. When a current situation (process) finishes, the next situation (process) is not determinate because at that time there are uncertainties regarding what resources are available. This pull model is designed for this situation to avoid being affected by un-determined factors in the future.

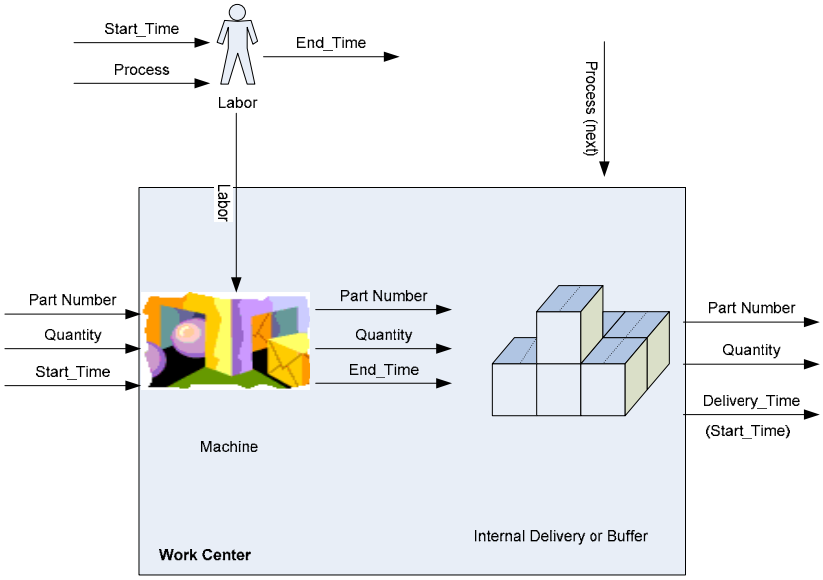


Figure 3.3 Diagram of Pull Model

3.2.4 Event Driven Model

A SC or job shop model is usually described as a discrete system, which is expressed as an event driven model. When a value of this system is changed, the system creates an event. In the simulation system, there are two situation changing classes: one relating to products and the other to common resources released. These are created in two situations. Firstly, a product comes from outside into a system, like an incoming order of materials being delivered. Secondly, a product is waiting for its next process,

because previous common resources release a product, e.g. products which have been cut by one machine and are waiting for sawing by another machine. Common resources released situations are mainly focused on labour and machines according to the specific point of the system. If labour and machines are idle, they can operate suitable products. In this system, two function groups of a system are designed and driven by two events. One is products (orders) arriving from outside, which is not relating to the releasing of the system's inner common resources. Its key factor is an arrival time (called an **order coming** event which drives a function group to change the related values of the system). As it comes from outside, a high priority is given to it. The other is related to the release of common resources including related products waiting (called a **common resources released** event). This event would be dealt with after operating all current *order coming* events.

3.3 The Proposed Supply Chain Simulation Framework

3.3.1 Direct Link and Indirect Link

In this section, two concepts of direct link and indirect link are defined in order to clearly illustrate the workflows of a SC. Before outlining the two definitions, several organisations are described as follows (Umeda and Jain, 2004):

Source: generate raw materials without anything input; Supplier: make products or components with materials input; Storage: store products or components; Planner: control material and information flows, and make schedules; Consumer: place orders to a planner; Deliverer: transfer products or components from one place to another place; Simulator Control: control a cell simulator, run simulations under predefined scenarios, analyse results of simulations and create new scenarios for simulations.

When a planner directly places orders or makes assignments to their external

suppliers, deliveries, sources, storages, etc., the relationships between a planner and the partners, such as suppliers, deliverers, storages, etc. belong to **direct links** and this calls the planner **directly links** the partners. When a planner only places orders or makes assignments to *Partner A* and *Partner A* places orders or makes assignments to *Partner B* in order to complete original orders or assignments from the planner, the relationship between the planner and *Partner B* is defined as an **indirect link** and this calls the planner **indirectly links** *Partner B*.

3.3.2 Hierarchical Structure of an Internal Supply Chian

In the proposed framework, the three main layer hierarchical structure is similar to Pundoor‘s framework. The definitions of three layers mainly refer to Pundoor’s definitions (Pundoor and Herrmann, 2006), namely supply chain, participants and process, because this three layers structure is able to balance the tradeoffs between accurately representing activity and unnecessary details. If necessary, it is flexible to model details of activity at a suitable level in order to accurately express the internal logics of the activity. Pundoor’s structure follows the Supply Chain Operations Reference (SCOR) so that this three layers structure adopted is amenable to being supported by and referred to the suitable parts of SCOR. Compared to Pundoor’s framework, the main difference is a data layer separation between the two main layers and a data layer on the top. Furthermore, a simulation block concept is introduced into the framework in order to support effective cooperation of each part and the simulation programming is based on the concept of a simulation block.

Figure 3.4 shows the architecture of an internal SC simulation framework of an enterprise with three main layers. The top layer is called the supply chain layer, which focuses on the operational processes of the internal SC and relationships of participants. The middle layer is considered as a participant layer, which represents the operational flows of participants such as manufactures, suppliers, retailers, etc. The bottom layer,

namely process layer, can define an entity (machine) at a detail level. The separate layers among the three main layers are the data filter and convert layers, which are mainly in charge of an authority and data format.

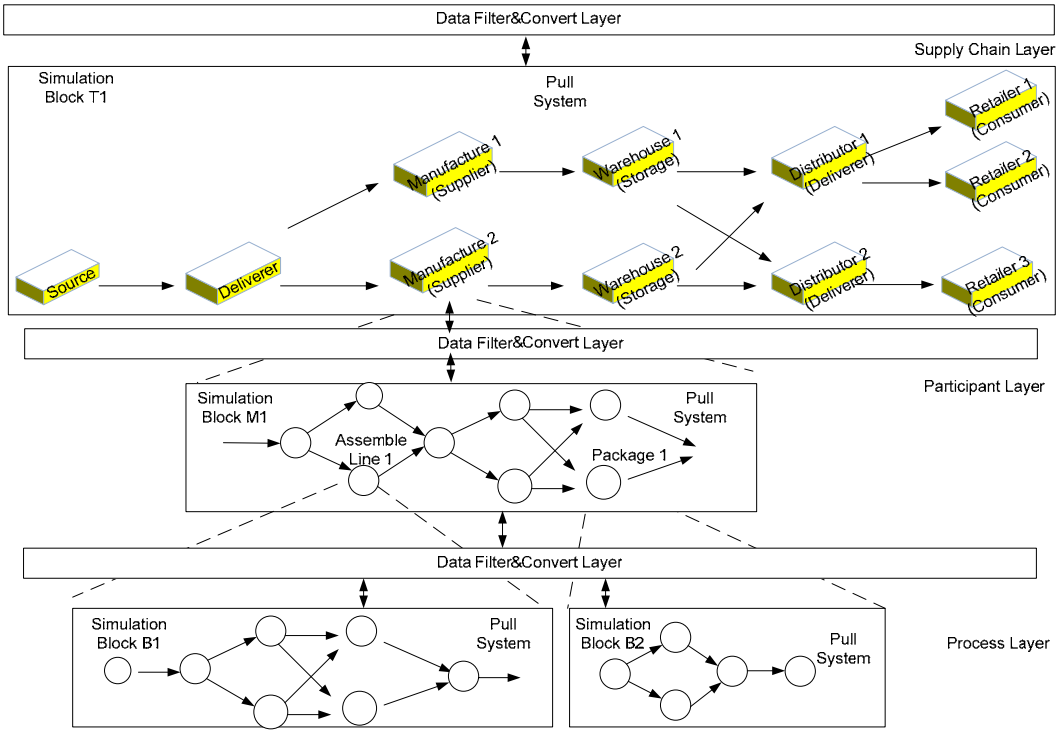


Figure 3.4 Hierarchical Structure of SC Simulation Framework

The simulation block, which represents an operational set of an individual, can exchange data with the related layers below and above. In general, when running a simulation, data (attributes) of an entity of the simulation model in a simulation block is pushed to the database at the related layer below via the data filter and convert layer. Then the required data (attributes) is pulled from the database at the related layer below into its database crossing the data filter and convert layer after finishing another simulation at the layer below. Further details of transferring communication messages and assigning a simulation sequence are discussed in the next Section 3.3.3. The workflow of a simulation block is illustrated in detail in Section 3.3.4. Thus, every layer consists of simulation blocks, which encapsulates the internal logics of relevant independent individuals in the above layer. When simple rules or distributions are not capable of representing the internal logics of an entity, a simulation block is required to

be developed for expressing the entity. A simulation block is built for the entity to mimic its operations. This is a unique advantage of the hierarchical structure to flexibly represent an entity using a concept of the simulation block.

Figure 3.4 is an example to explain how to closely cooperate with different simulation blocks in the hierarchical structure. Firstly, a simulation X1 is run in the top layer, when the entity of *Manufacture 2* (see Figure 3.4) is put the input data (attributes) and asked for output data, the input data is transferred to the database of simulation block M1 which represents *Manufacture 2* via the related data layer and simulation Y1 in block M1 is run according to the received input data. When simulation Y1 goes to *Assemble Line 1*, the related input data is pushed to the database of simulation block B1 that expresses *Assemble Line1*. Then simulation Z1 in the simulation block B1 is run. The relevant output data of block B1 is pulled back by simulation block M1 after finishing simulation Z1 (and optimisation). Then simulation Y1 continues to run until meeting *Package 1*. The similar procedures with Assemble Line 1 are adopted for getting the output data of *Package 1*. When simulation Y1 (and optimisation) is finished, the output data is pulled back by simulation block T1. Simulation X1 accepts the output data of *Manufacture 2* and continues to run. The example shown in Figure 3.4 describes the work and data flows *start* from *the up to down* and then *end* from *down to up* in this hierarchical structure.

3.3.3 Hierarchical Network of a Complete Supply Chain

In an internal SC, a hierarchical structure is sufficient to represent details and relationships of their internal participants. However, this structure cannot accurately represent the relationships among partners of an external SC, particularly partners by indirect links. An enterprise may not know the participants by indirect links so that these participants cannot be seen in the supply chain layer of the enterprise's hierarchical structure, as shown in Figure 3.7. Consequently, a hierarchical network is

proposed consisting of hierarchical structures to express the relationships among every partner of a complete SC. Every hierarchical structure represents an internal SC of an enterprise and an enterprise can communicate with the other partners via their top *data filter and convert layers* of the hierarchical structures as shown in Figure 3.4. A link between two partners expresses a *direct link* relationship which means they can directly communicate with each other. If there is not a *direct link* between two partners and the two partners need a link via other partners, their relationship is an *indirect link*. Generally speaking, two partners which link by *indirect link* are not able to directly communicate.

Figure 3.5 shows relationships of a SC composed of different external participants linking a planner by *direct links*. In a hierarchical network, the top simulation block at a supply chain layer is used to represent a hierarchical structure of an individual enterprise. When an entity in a simulation block TA links another simulation block TB, block TB represents the internal logics of this entity. If a running simulation in block TA goes to this entity, block TA will communicate with block TB. In Figure 3.5, the final plant is assumed to represent an enterprise (planner) owns a final plant and a delivery department. Four enterprises directly link the final plant (planner) which places orders to them and consequently there is a complete simulation workflow diagram of a SC at the supply chain layer of the final plant. This complete and clear workflow of a SC in the final plant will help to improve accuracy of time control at every stage in the business processes. However, workflow diagrams at the supply chain layers of external partners lack details and their SC layers only appear itself and the final plant.

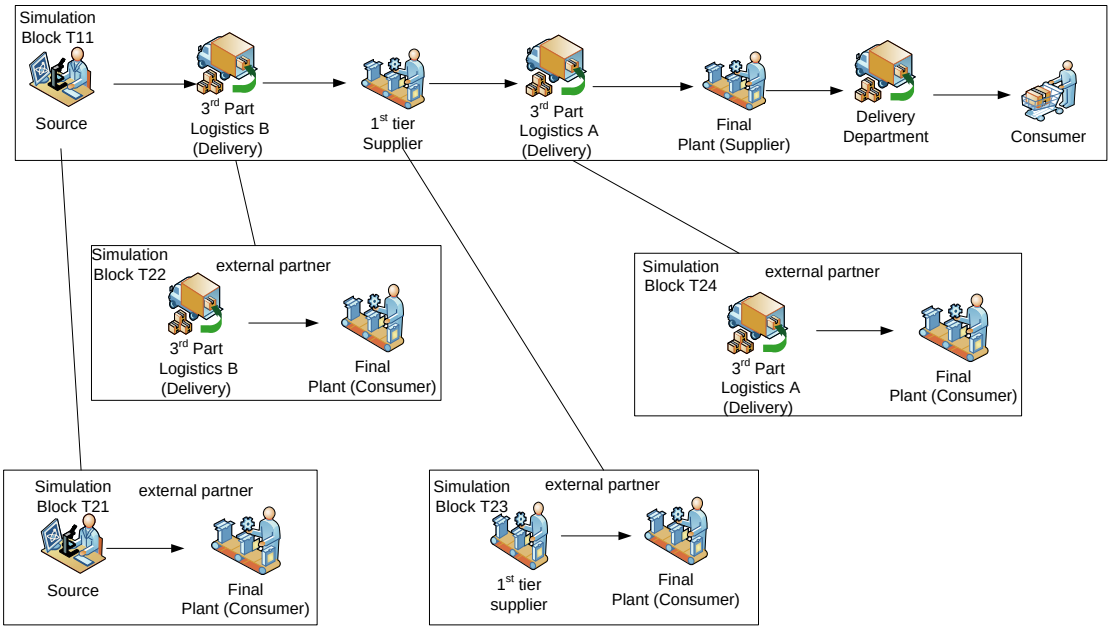


Figure 3.5 Hierarchical Network using Direct Link

Figure 3.6 shows a working sequence of simulations and communications (data flows) in the SC depicted in Figure 3.5. Every longitudinal line represents a corresponding simulation block and Y-axis is a time horizon. A label □ expresses the beginning point of a simulation (and optimisation) and a label ● shows the end point of a simulation (and optimisation). A distance from a label □ to the relevant label ● in the vertical line is the time which corresponds to the running of the simulation (and optimisation). Every horizontal line expresses a communication between two simulation blocks and the arrow of a line shows the direction of the communication.

The case of estimating the lead time of an order is taken as an example to illustrate the workflow of these simulation blocks. Block T11 of the final plant starts a simulation X11 and this simulation goes to the first entity-the source. Then, block T11 sends a message SFR1 consisting of related information of first quote with several data fields such as ‘due date’, ‘quantity’, and ‘material number’ ,etc. and a data required form with several fields like ‘estimated completion time’, to simulation block T21 of the source. A related simulation X21 is run in block T21 and a required form filled according to the

results of simulation X21 (and optimisation) is sent back to block T11 via the relevant data filter and convert layers after completing simulation X21 (and optimisation). Then, simulation X11 is able to continue to run using the data of the required form as the output of the source entity. Therefore, following a sequence of a material flow in the SC of block T11, related simulations (and optimisations) are run in block T22, block T23, block T24, and the data required forms are sent back to support the work of simulation X11. Finally, simulation X11 gets the required data from block M11 of delivery department at the participant layer of the same hierarchical structure and then simulation X11 is able to be finished.

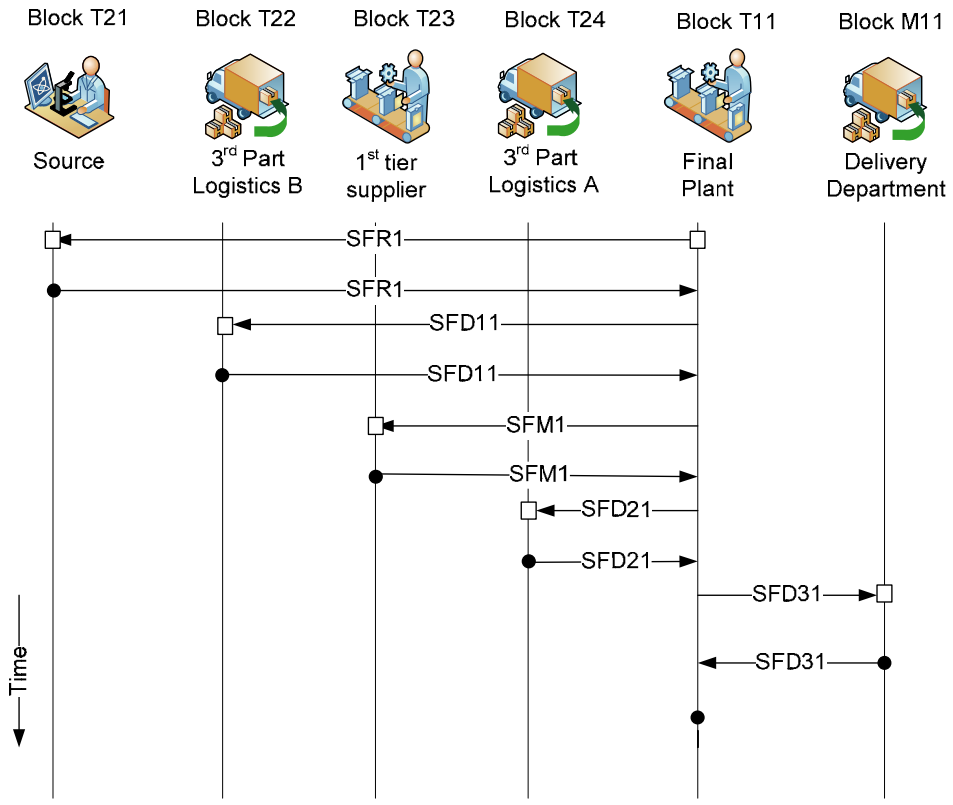


Figure 3.6 Working Sequence of Simulations and Communications using Direct Link

The lead time of the order and performance of a SC is taken from the results of simulation X11. If a simulator control creates a new scenario, the new due date of the order is placed to block T23. Thus, the message SFM1 is needed to be modified and the lead time needs to be estimated again. A simulation X12 should be run in block T11, but

this simulation is not necessary to be run from the beginning like simulation X11 running from the source entity. Simulation X12 takes the pervious intermediate data from simulation X11 at the time when simulation X11 sent a message SFM11, so that Simulation X12 starts from this time point and sends a new message to block T23. After taking the required data from block T23, block T24 and block M11, simulation X12 is finished and a new lead time is obtained. This method of re-simulate reduces the simulation time and *direct links* enable the final plant to easily create a flexible 'what-if' question, and find solutions through simulations. However, if the planner has too many direct links, this increases the complexity of the management of the planner and external partners also lack the understanding of the other related partners in the complete SC.

Figure 3.7 shows a hierarchical network of a SC composed of different external participants linking a planner by *indirect links* or *direct links*. The final plant only places orders to the 1st tier supplier and this supplier contacts the other partners about the issues of materials and deliveries. Thus, only a partial workflow of a SC is shown in simulation block T11 at the supply chain layer of the final plant. In simulation block T21, the source and the 3rd part logistics which this supplier should communicate are considered. Figure 3.7 indicates there is no complete workflow of the SC in any simulation blocks (supply chain layers) and shows a situation of separate information and decentralised management. A suitable decision is difficult to make because no relative complete data supports the decision making. However, a simulation system based on the hierarchical network is able to help enterprises of a SC to integrate information for decision making.

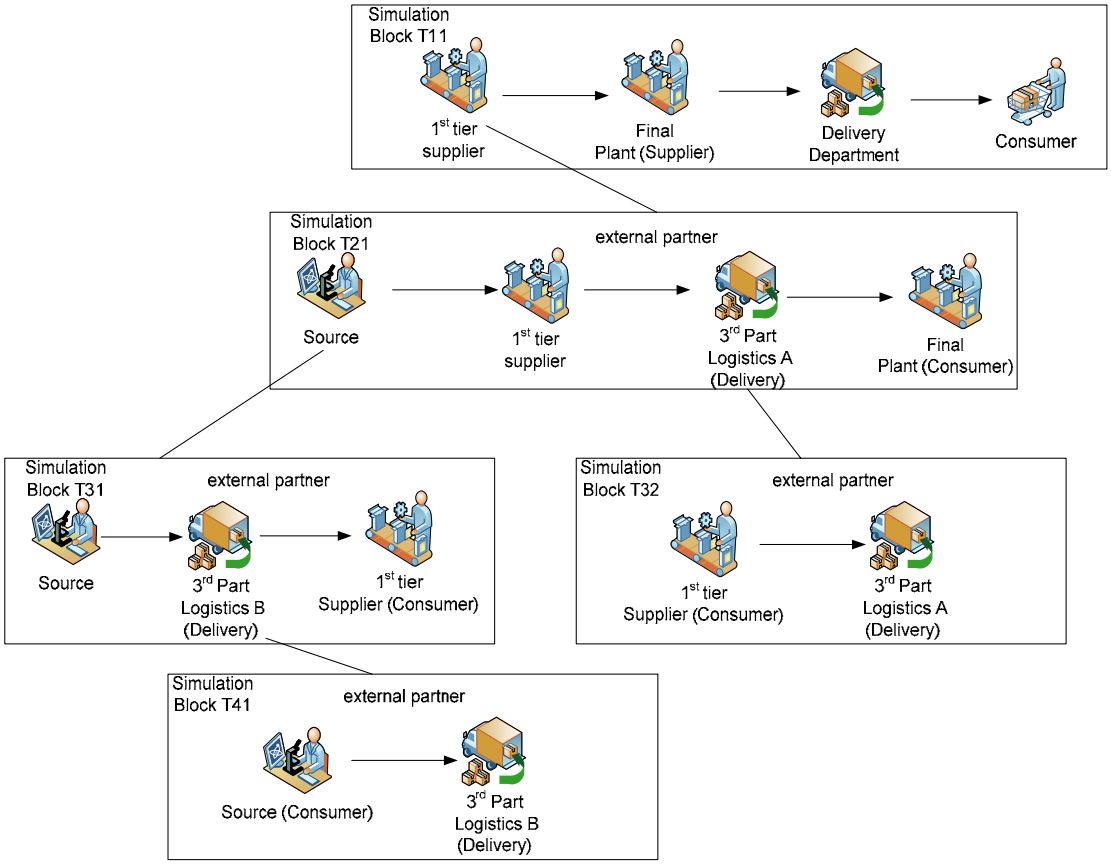


Figure 3.7 Hierarchical Network using Indirect Link

Figure 3.8 shows a working sequence of simulations and commutations for the case outlined in Figure 3.7. The illustration still takes the case of estimating the lead time of an order. A message SFM1 is sent to block T21 after a simulation X11 starts to run in block T11. Then a simulation X21 in block T21 begins to work and sends a message SMR1 to block T31. The related information of message SMR1 is made based on the relevant context of message SFM1 and a relevant Bill Of Materials (BOM) in block T21. Then block T31 answers block T21 and sends the required data back after quoting from block T41 and completing its simulation (and optimisation). After that, simulation X21 continues to run and block T21 sends the required data back to block T11 after quoting block T32 and completing simulation X21 (and optimisation). Now, simulation X11 is able to work and quotes block M11, and finally a result is able to be obtained from simulation X11.

This result is incomplete similar to Figure 3.6 and the planner cannot own details of every partner. This may affect the data analysis and creating suitable scenarios for improving the performance by the staff working in the final plant. When running a new simulation in block T11 for a new scenario, this simulation must run from the beginning. However, this result is obtained from integrating the information from all blocks and all constraints of each block are used to validate this scenario. This decentralised management also reduces the labour time of the final plant and enables its partners to improve their performance by a close cooperation with the partners linked by their *direct links*.

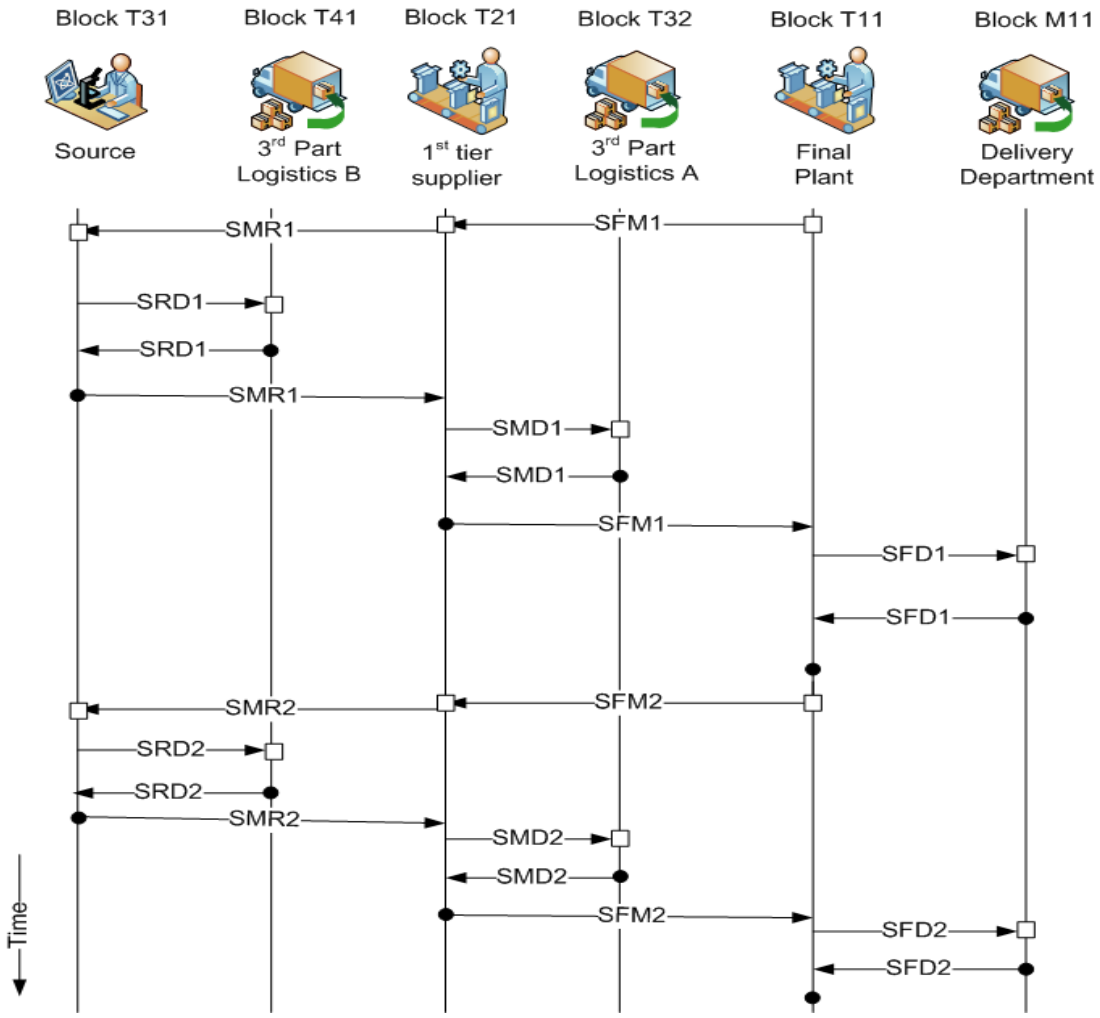


Figure 3.8 Working Sequence of Simulations and Communications using Indirect Link

3.3.4 Simulation Block Design

Figure 3.9 shows the architecture of a simulation block. The simulation block is the basic unit of the proposed simulation framework. It consists of a cell simulator, a database or dataset, a simulator control or/and an optimisation program. Inputs which come from the *data filter and convert layer* are directly sent to its database or dataset automatically or manually by a simulator control depending on the authorisation policies of the *data filter and convert layer*. According to the levels of authorisation policies, an authorised user gets different rights to operate relevant information of the database or dataset, set the scenario, control the cell simulator with a read only attribute, run the optimisation program, etc. The cell simulator can take the required data from a database (dataset) or input it by manual means. This data includes process, part number, arrive time, due time, work order (job), employee, quantity, etc. The results concerning performance of enterprises are provided after running a simulation.

A simulation control (optimisation algorithm) can check, compare or optimise the results and alter the strategic polices or management rules and then get the relevant results from the simulation model in the cell simulator. The simulator control (people), input datasets (process), simulations in the cell simulator (product), and results performance (performance) consist a 4P's close-loop control cycle. Through this cycle, performance of enterprises must be improved step by step.

In order to make a simulation block adaptive, the concept of two operational policies from control view, namely schedule-driven control and stock-driven control, is introduced into the design of a simulation block. These two policies are developed from National Institute of Standards and Technology (NIST) (Umeda and Jain, 2004) based on four operational strategies, namely MTS, MTO, ATO, and ETO. In a schedule-driven control method, 'Master Production Schedule' (MPS) generated by a planner according to the demand forecast is an important feature and factories start to

produce products according to the order received from the planner by using MPS.

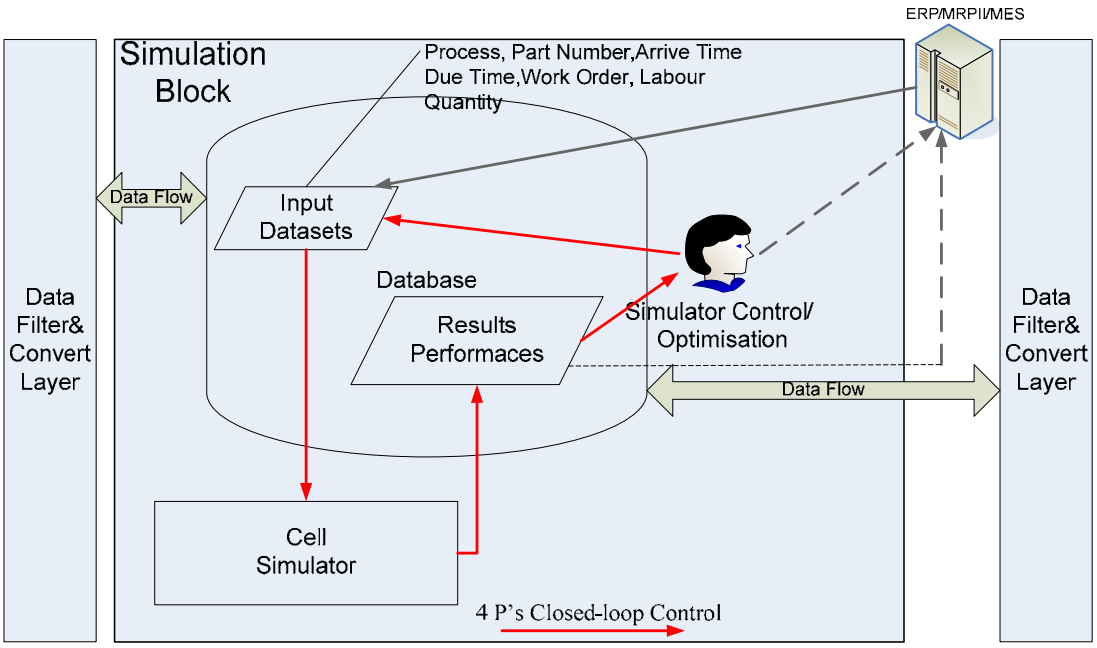


Figure 3.9 Architecture of Simulation Block for Users

Stock-driven control means factories make a schedule according to the stock volume information and normally produce a number of products after the stock reduces down to the predefined replenishment point. Depending on the aim of a simulation model, the role of planner may put into a simulation model in a cell simulator or act by a simulator control or an ERP system. The relevant information such as demand forecast, stock volume is able to store in the database (dataset) of a simulation block or an ERP system. In a schedule-driven control simulation block, MPS prefers to be stored in the database of a block so that a simulation operates it conveniently. In a stock-driven control simulation block, the planner is normally defined in a simulation model in order to place orders automatically according to the stock volume.

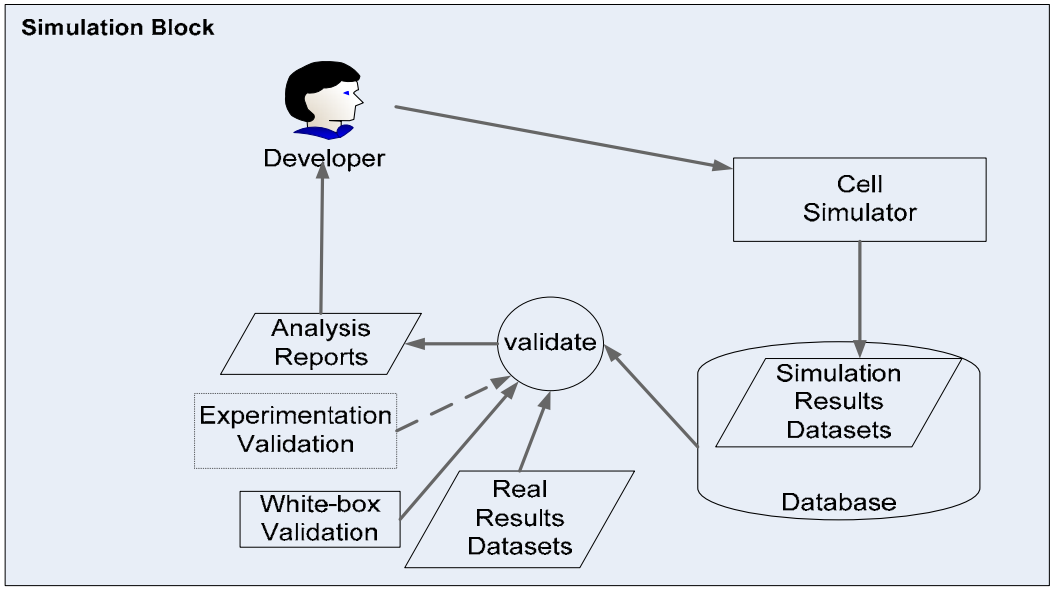


Figure 3.10 Architecture of Simulation Block for Validation

Another closed-loop control cycle is the process of the validation in Figure 3.10. A simulation model provides the results datasets that go to the stage of validation. The real results datasets are compared to the simulation results and the white-box validation method is adopted. Analysis reports based on this validation are given to developers for improving the performance of the simulation model.

3.3.5 Cell Simulator Structure

Figure 3.11 shows the structure of the cell simulator consisting of seven parts. A simulation database stores relevant data of labour, products, orders, routes, etc. The *Data Link* layer provides the bottom link functions to the database or raw datasets. The *Data Operation* layer provides reading or saving data operation functions. The *Main Core* is to manage the whole cell simulator and harmonise different tasks. The *Decision Dataset* layer offers various datasets for local optimisation. The *Rule* layer focuses on local optimisation algorithms and management rules. The *Attributes* layer manages the attributes of all entities. The *Entities* layer represents different machines, labour, products, and their fixed, simple relationships (routes), etc. Further details of their routes are discussed in Section 3.3.6.

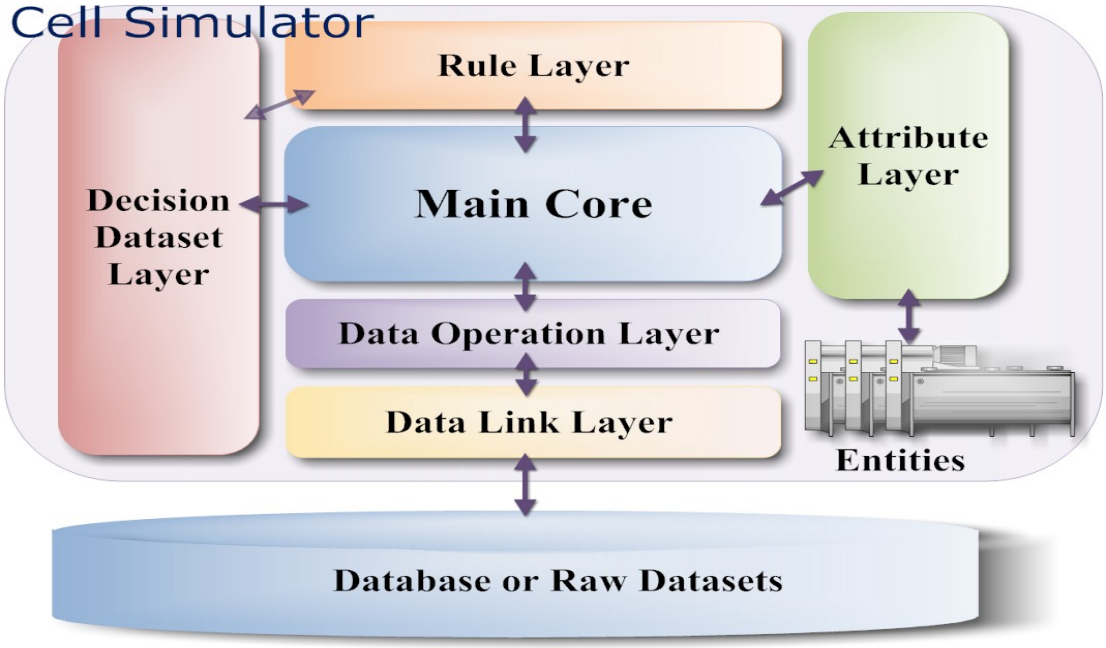


Figure 3.11 Structure of Cell Simulator with a Database or Datasets

The *Main Core* adopts the functions in the *Data Operation* layer to get the relative dataset from the simulation database or raw datasets via the data link layer. (Sometimes, the *Main Core* is able to directly operate the local raw datasets in order to simplify the structure of the cell simulator by reducing the *data operation* and *data link* layers) The *Main Core*, analyses and obtains datasets, and transfers the raw datasets into a suitable decision datasets. After that, it instructs the rule layer to optimise according to the decision datasets. After optimisation, it puts the relative values into the attributes to drive the simulation model and updates the database, accesses the *Data Operation* layer and the *Data Link* layer. Figure 3.11 is the complete structure of a cell simulator and simplifying methods are adopted in corresponding parts depending on the aim of building a simulation model. For example, if only basic rules are involved in a simulation model, decision dataset layer and rule layer may be erased.

3.3.6 Flexibility Configuration

The previous Section 3.3.5 has discussed the design of a cell simulator and this section focuses on relationships among entities in the simulation model in a cell

simulator. Generally speaking, SC or similar chain like job shop is a multi-domestic multi-stages operations network and its flexibility configuration is one of the most important issues in developing a simulation model.

The aim is to develop a simulation model with a full flexibility, as flexibility configuration is classified into three types: no flexibility, partial flexibility, and full flexibility. Figure 3.12 shows the architecture of a chain with no flexibility, which means every node can only links one node at the next level.

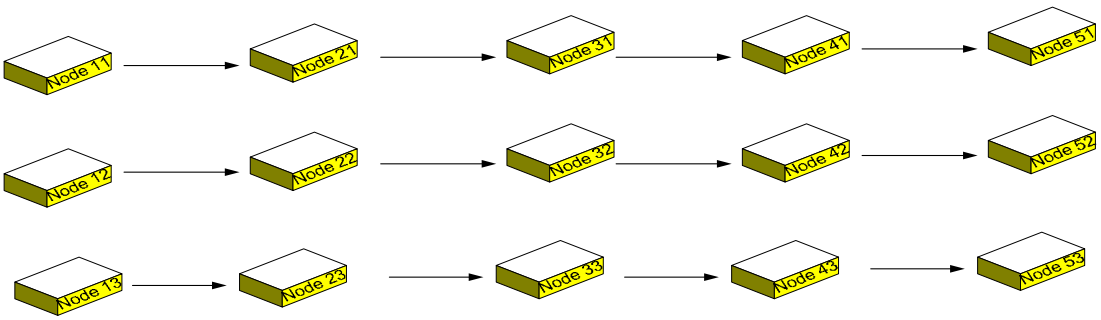


Figure 3.12 No Flexibility Configurations

Figure 3.13 illustrates a partial flexibility configuration which represents every node can link many nodes at the next level and not all nodes are linked together. Full configuration pattern which expresses one node links all nodes at next level shown in Figure 3.14.

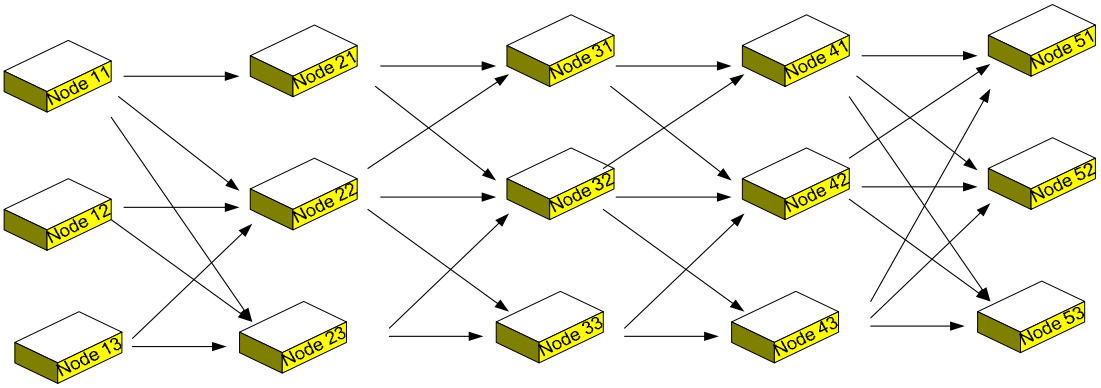


Figure 3.13 Partial Flexibility Configurations

In the proposed framework, the flexibility configuration is divided into two types, namely semi-flexibility configuration and total flexibility configuration according to

optimisation and control adaptations. No flexibility configuration is too simple to need an optimisation algorithm and a simple control can deal with this situation. The information of its fixed and simple links (routes) is able to be easily stored in the *Entities* layer of the cell simulator. Semi-flexibility configuration is a one to many and many to one network, which adopts the local optimisation algorithm (dispatching algorithm). It is a special case of partial flexibility configurations. Total flexibility configurations, which adopt advanced optimisation algorithms, represent all full flexibility configurations and part of partial flexibility configurations. Two related control methods are adopted to construct the two types of configurations.

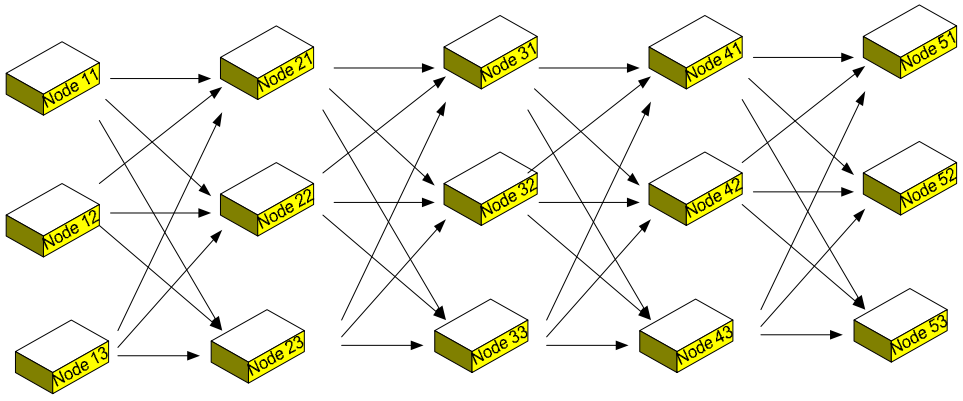


Figure 3.14 Full Flexibility Configurations

Figure 3.15 shows the diagram of Semi-flexibility configurations, which is a one to many and many to one pattern. This configuration would adopt a dispatching control method which assigns the entities (products, resources) into the related nodes at next level according to a dispatching rule, such as SPT, EDD, etc. or a local optimisation algorithm, such as Dispatching Optimisation Algorithm (DOA) controlled by several parameters (divided number, iteration number, etc.) . The dispatching is able to deal with the condition of various products and further details of DOA are discussed in Chapter 4.

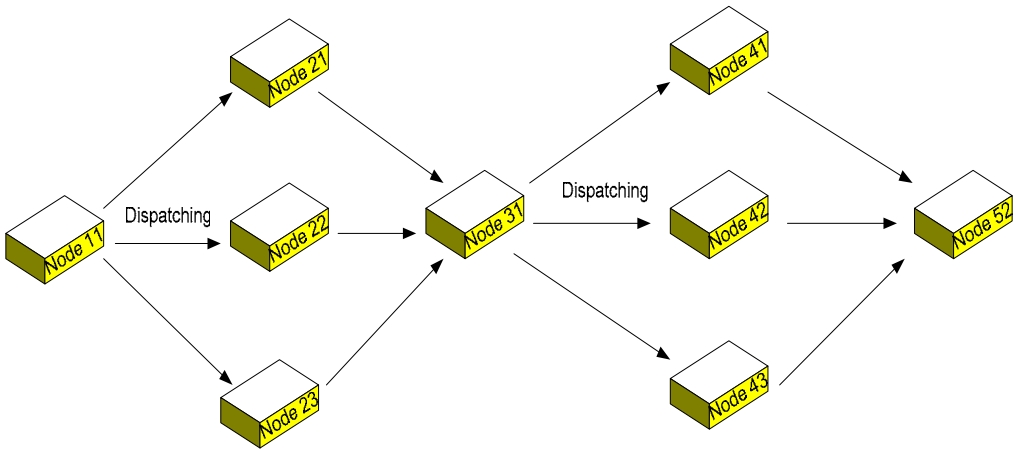


Figure 3.15 Semi-flexibility Configurations

Figure 3.16 shows the architecture of full flexibility configuration, which is a many to many pattern. The route control method which gives a defined route for each entity would be applied for this configuration. However, the optimisation of this route control needs a global optimisation algorithm which is not able to deal with control decisions (Gaither and Frazier, 2002) of a real-time control due to delaying time of implementing the global optimisation algorithm. Strategic control (a long-term plan) and operation control (based on a weekly or monthly interval) are able to adopt the global optimisation algorithm. The information of these complicated routes is usually stored in the tables of a database of the simulation block in order to easily determine the suitable routes.

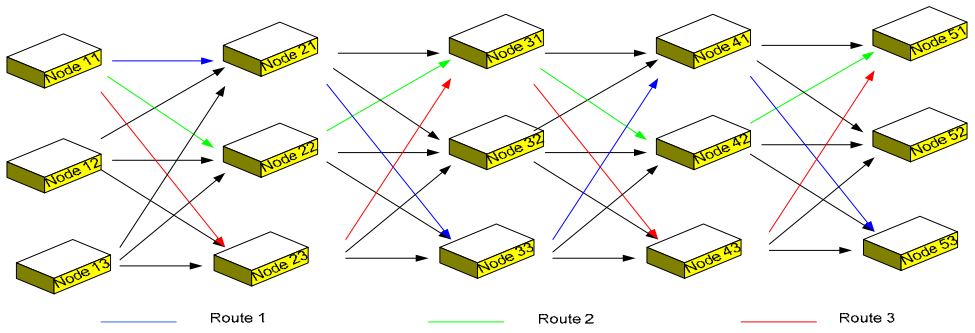


Figure 3.16 Total Flexibility Configurations

3.4 Conclusions

This chapter illustrates a simulation-based DMS framework with a hierarchical network and a flexibility configuration for SMEs in SCs. The architecture of the proposed framework is mainly divided into four layers from top to bottom, namely hierarchical network, hierarchical structure, simulation block, and cell simulator. The concept of hierarchical network is proposed to actually represent a complete SC composed of internal SC and external SC. Two examples about direct link or indirect link are used to demonstrate hierarchical networks from the communication and control views. The design of the simulation block is discussed according to two control strategies and different functions. Finally, a cell simulator structure and flexibility configuration of a simulation model are demonstrated in detail.

Chapter 4 Optimisation Block and Performance Analysis

4.1 Introduction

This chapter introduces the architecture of an optimisation block consisting of an optimisation module and a performance analysis module. A proposed optimisation module is developed for finding a 'good enough' schedule for a scheduling problem in a job shop based on UK-based case study. The optimisation module has two parts: firstly the Conflicted Key Points Optimisation (CKPO) which focuses on building a feasible and 'good enough' schedule by constructing a work sequence of CRs and jobs. Secondly, the Dispatching Optimisation Algorithm (DOA) is responsible for assigning numbers of sub-jobs split from a job (LS technique) into several available unrelated common resource pairs to minimise the makespan of an operation.

4.2 Optimisation Block

In a DMS, an optimisation mechanism is an important part, which can find a 'good' solution by searching a feasible solution region for decision makers. In the architecture of a proposed simulation block, the optimisation mechanism is able to find a 'good enough' solution in a reasonable time for a particular problem.

Normally, a decision making procedure can be divided into two stages: optimisation and decision (Glover and Kochenberger, 2003). Optimisation is to identify a value or solution to satisfy the needs of an objective function. The decision would indicate 'yes' or 'no' per solution to determine whether the related actions should be executed or not. Consequently, a decision on a complicated problem normally refers to the performance of the solution after optimisations, such as tardiness, completion times, costs, etc.

In reality, there are various optimisation problems for different SMEs in a SC. It is impossible to adopt or develop one or several optimisation methods or even combinations of these optimisation algorithms to provide optimisation solutions for all eventualities. An example of an architecture based on a particular optimisation block for a particular optimised problem is proposed, as shown in Figure 4.1.

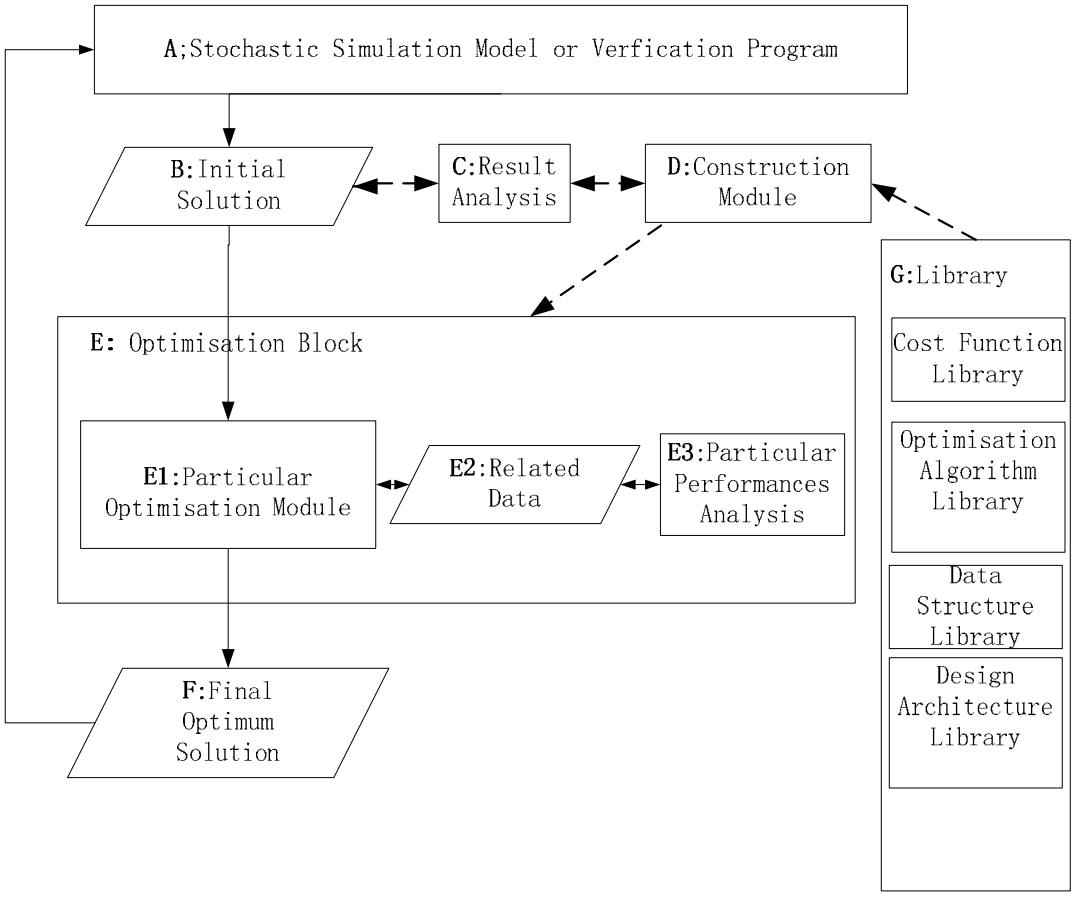


Figure 4.1 Architecture of Optimisation Block

The descriptions of each section in the architecture of an optimisation block are outlined as follows:

A: A complete deterministic or stochastic simulation model is developed by a cell simulator coded by WITNESS. It provides an initial solution like a schedule and supports a validation function for optimum solutions. A verification program is used to rapidly verify the feasibility of the optimum solutions.

B: An initial solution is provided by a cell simulator for a particular problem.

C: An analysis module or a simulator control (see in Figure 3.9) analyses the simulation results of a solution and make decisions.

D: A construction module constructs the particular optimisation block for a particular problem based on the library.

E: An optimisation block generates an optimum solution for a particular problem by the particular optimisation algorithms.

E1: A particular optimisation module is the core of an optimisation block and embeds the optimisation algorithms for a particular problem.

E2: A related data module provides data structures or datasets which the optimisation module and the performance analysis module exchange.

E3: A particular performance analysis module embeds related cost functions and is able to give the performance of a middle solution to guide the next stage optimisation.

F: An optimum solution is generated from the optimisation block and prepares to be validated by a simulation model.

G: A library module consists of a serial of knowledge libraries, such as cost function library, optimisation algorithm library, data structure library, etc., and provides a knowledge support for constructing an optimisation block.

This architecture aims to seek a particular 'good enough' solution within a limited computation time. When a particular optimised problem is required, the construction module takes suitable optimisation algorithms, cost functions, and the related data structures from the library to construct a particular optimisation block for this problem. An initial solution, which is generated from a simulation model in a cell simulator, is sent to the particular optimisation block. Then after several iterations in the

optimisation block, the final optimum solution obtained in a limited time constraint is provided for a validation in the stochastic simulation model. The optimum solution is checked and its performance is compared in a stochastic environment. A stochastic simulation model produces different performance each time the model is run. Consequently, a more accurate performance analysis is achieved by running the stochastic simulation model several times. Depending on the achieved accuracy of the requirements, the performance data of each solution is able to be stored for multi-comparison and analysis after running a stochastic simulation. When collecting data which is not sufficient for building a stochastic environment, a verification program is a possible substitution of the stochastic simulation model at the validation stage. The program is to verify and validate the optimum solution according to the mathematic constrains of a schedule. The results of the simulation model are analysed to decide whether to add Common Resources (CRs), to modify the order, alter the optimisation parameters, optimisation algorithms, or cost functions for further optimisation. Further details of structure and design of a simulation model in a cell simulator are discussed in Chapter 5.

A UK-based SME company involved in manufacturing and high-precision sheet metal machining was used as the main case study from its scheduling problems of a job shop relating to a performance analysis and an optimisation algorithm. Thus, this section of a performance analysis and optimisation algorithms emphasises on scheduling problems relating to its job shop.

4.3 Cost Function Library

In most studies, only one or two measures are taken into account for analysing the performance of a SC to determine a ‘good’ result. However, in practical applications, there are numerous parameters that affect the performance of a SC, and there are several cost functions relating to different performance requirements. This demand

needs a cost function library to store various related cost functions. In this chapter, parameters and criteria relating to scheduling are outlined. There are several parameters or criteria which are considered in the performance analysis in a scheduling problem of a job shop (Pinedo, 2008) and these are outlined as follows:

Makespan C_i: the completion time of the job i to leave the system

$C_{max}=\max(C_1,\ldots,C_n)$ is defined as the completion time of the last job in n jobs to finish all operations

d_i: the due date of job i

Lateness L_i: a violation of the due time of job i, and $L_i=C_i-d_i$

Maximum Lateness $(L_{max})=\max(L_{1,\ldots}L_n)$

Tardiness $T_i=\max(C_i-d_i,0)=\max(L_i,0)$

Maximum Tardiness $T_{max}=\max(L_{max},0)$

To attain a comprehensive performance value for multiple cost functions, a general combined objective function (cost function) is selected

$$\sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij}F_{ij} \qquad (4.1)$$

subject to $\sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij} = 1$. W_{ij} is the weight value of the certain criteria and is able to be adjusted for different aims by the decision makers (simulator control). F_{ij} represents the relevant cost function of criterion j in the ith job. It is noted for the following special cases:

- If m=1 and F= C, Equation 4.1 can be written as Σw_iC_i, which is Mean Weighted Completion Time (Mean Weighted Flow Time).
- If m=1 and F=T, Equation 4.1 becomes ΣW_iT_i, which is Total Weighted Tardiness.

4.4 CKPO Algorithm, Architecture and Performance Analysis for JSSP

There are various optimisation algorithms adopted for a particular search problem, such as Genetic Algorithm (GA), Ant Colony Optimisation (ACO), Particle Swarm Optimisation (PSO), Bee Colony Optimisation (BCO), etc. In this section, a Conflicted Key Points Optimisation (CKPO) is proposed for solving this job shop batching scheduling with unrelated parallel machines, setup time, and job splitting of the UK-based SME case study. The CKPO algorithm was developed based on the concept of a critical path widely adopted in popular heuristics including TS, simulated annealing, shifting bottleneck heuristic, etc.

To the best of our knowledge, there is no research work to solve this type of JSSP for the UK-based case study. Consequently, a proposed CKPO modified corresponding algorithms for extending applications to solve the JSSP of UK-based case study was developed. The original idea of CKPO is to determine the critical factors or jobs affecting the schedule and construct a new neighbourhood structure by a swap method on the critical path to seek a 'good enough' solution. Usually, a performance analysis module can provide which job (critical job) needs to be improved most urgently on a schedule and then swapping actions are adopted on the longest path (critical path) of a precedence graph (a complete selection of a disjunctive graph). Then rescheduling the plan, the performance analysis is adopted to assess the performance of the new plan for determining an action of keeping it or dumping it and provide new critical factors or jobs for further optimisations. Sometimes, if the objective function is simple, CKPO is considered to integrate a simple performance analysis. After running a number of iterations, the CKPO algorithm is able to provide a 'good enough' solution.

For a simple JSSP, the optimisation architecture of only integrating a CKPO

algorithm with a performance analysis is able to be adopted to solve the problem. However, a JSSP for the UK-based SME case study is a job shop batching scheduling, which has unrelated parallel machines, setup time, job splitting, and it needs to be divided into two sub-problems to be optimised because of this complexity. Consequently, it needs another optimisation algorithm in order to get a 'good' solution. The first sub-problem is a hybrid job shop batching scheduling problem about how to sequence the jobs (work orders) and choose the Common Resources (CRs) for the jobs. The second sub-problem is a parallel machine scheduling problem with unrelated parallel machines, setup time, and jobs splitting in order to assign the sub-jobs split from the job using Lot Streaming (LS) to the chosen CRs. The two sub-problems are not independent and link to each other. The solution of the first sub-problem defines a job and CRs which are input parameters for the second sub-problem stage. Consequently, the solution of the second sub-problem is obtained using output information from the first sub-problem and provides the operation time to carry out the next optimisation task.

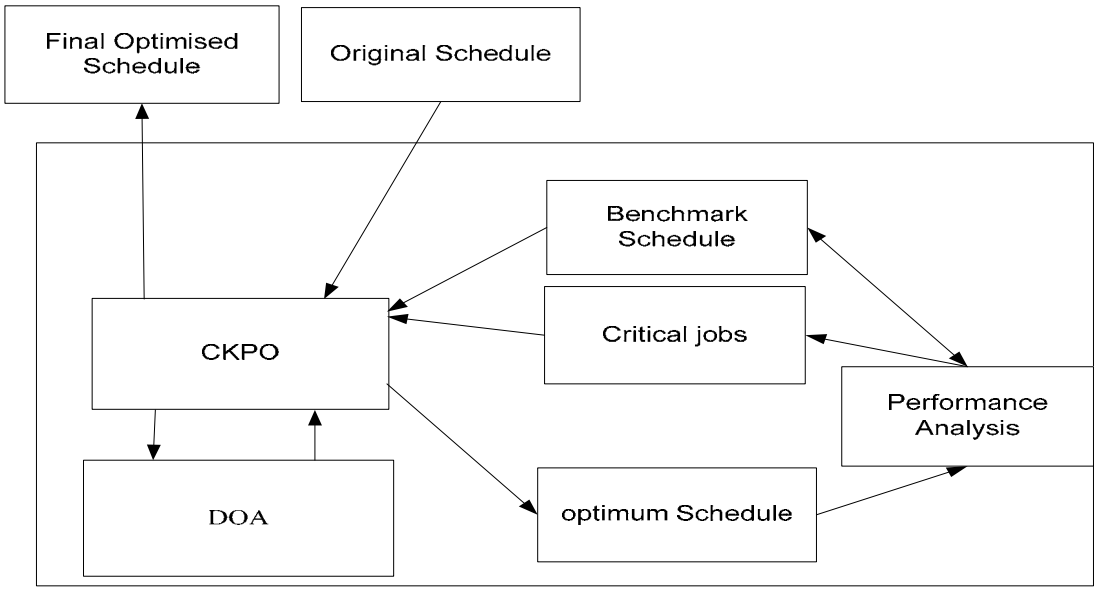


Figure 4.2 Optimisation Block with CKPO

Figure 4.2 shows the architecture of an optimisation block of a CKPO. A Dispatching Optimisation Algorithm (DOA) is adopted for assigning a batch of

components to several CRs conducting the same process in order to minimise a makespan for the second sub-problem. It allows a CKPO to focus on the first sub-problem which is the schedule optimisation based on the assumption of CR sets consdered as individual different machines. Further information is discussed in Section 4.4.3.

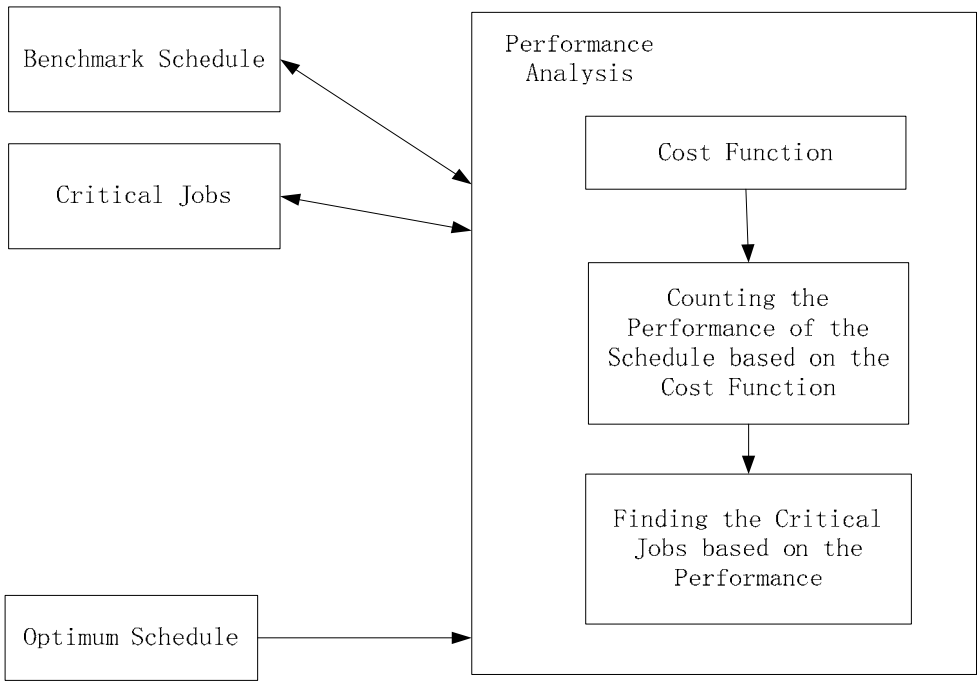


Figure 4.3 Architecture of Performance Analysis

Figure 4.3 shows the architecture of a performance analysis linked to CKPO. If the cost function is simple like min(C_{max}), the CKPO is able to contain the performance analysis with these procedures. For a relatively complicated cost function, the performance analysis module matches to a CKPO and provides the related performance and data to the CKPO for further optimisation. The performance analysis module is able to calculate the performance by the given cost functions and seek the critical factors (jobs) using a general weighted objective function as an example. The following is used to determine a rule to obtain a critical job or provide an ordered list of critical jobs:

Rule 4.0 (To determine the critical job or an ordered list of critical jobs):

If a general objective function (cost function) is chosen as $Min(\sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij}F_{ij})$, subject to $\sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij} = 1$, the critical job i is

$$\text{Max}(\sum_{j=1}^{m} W_{ij}F_{ij})$$

or the ordered list of critical jobs is

the list of the job i in a non-increasing order of their $(\sum_{j=1}^{m} W_{ij}F_{ij})$

If a general objective function (cost function) is chosen as $Max(\sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij}F_{ij})$, subject to $\sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij} = 1$, the critical job i is

$$\text{Min}(\sum_{j=1}^{m} W_{ij}F_{ij})$$

or the ordered list of critical jobs is

the list of the job i in a non-decreasing order of their $(\sum_{j=1}^{m} W_{ij}F_{ij})$

4.4.1 Disjunctive Graph Representation

The disjunctive graph representation is a common graphic formulation of JSSP. The disjunctive constraints in the graph are critical factors for constructing neighbourhood structures in optimisation. In fact, not all Disjunctive Arcs (DAs) should be emphasised at the same level. However, from Section 2.10, to the best of our knowledge, a category for disjunctive arcs don't exist. As a result, in order to classify the critical level, disjunctive arcs are proposed to be divided into two types, one is conflicted disjunctive arcs and the other is released disjunctive arcs, based on the concepts of disjunctive arcs and characteristics of scheduling. Normally, a node (i, j) (machine i, job j) is directly linked by two arcs, namely a conjunctive arc and a disjunctive arc, under the situation of one machine occupied by one operation. If the longest path from the *source* to a node passes through the disjunctive arc (directly link

to the node), this disjunctive arc is referred to as a **Conflicted Disjunctive Arc (CDA)**. If not, it is referred to as a **Released Disjunctive Arc (RDA)** (excepting a conflicted conjunctive arc with ef=0 defined in Section 4.4.3).

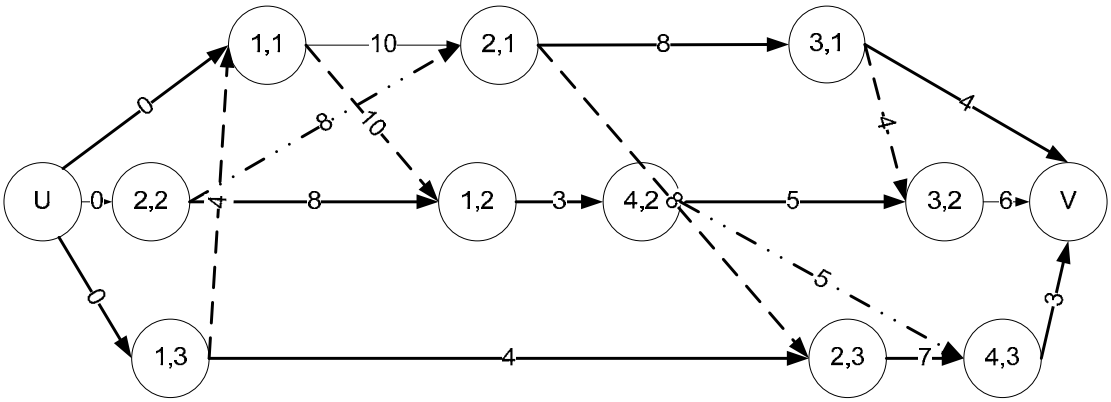


Figure 4.4 Precedence Graph with Two Types of Disjunctive Arcs

The precedence graph G shown in Figure 4.4 shows a graph with two types of disjunctive arcs. DAS, the set of disjunctive arcs (all dashed lines and dot-dashed lines in Figure 4.4) is

$$DAS=CDAS \cup RDAS$$

where CDAS is the set of *conflicted disjunctive arcs* (dashed lines in Figure 4.4), RDAS is the set of *released disjunctive arcs* (dot-dashed lines in Figure 4.4). All solid lines shown in Figure 4.4 are conjunctive arcs.

Definition 4.1: $a_o \in DAS$ is a disjunctive arc from an operation node mi (N_{mi}) to an operation node mj (N_{mj}). a_p represents the conjunctive arc from an operation node nj (N_{nj}) to an operation node mj (N_{mj})

Rule 4.1: $r_{mj}=\max(r_{nj}+p_{nj}, r_{mi}+p_{mi})$

If $r_{mi}+p_{mi} > r_{nj}+p_{nj}$

$a_o \in CDAS$

Else

$a_o \in RDAS$

Where r_{nj} is the release time of operation nj (N_{nj}) determined by the longest path in the graph G from the *source* U to $N_{nj;}$ r_{mj} is the release time of operation mj (N_{mj}) determined by the longest path in graph G from the *source* U to $N_{mj;}$ p_{nj} is the operation time of operation nj; r_{mi} is the release time of operation mi (N_{mi}) determined by the longest path in graph G from the *source* U to $N_{mi\ ;}$ p_{mi} is the operation time of operation mi; operation mi is a predecessor operation of operation mj on the resource (machine) task; operation nj is a predecessor operation of operation mj on the job task.

Table 4.1 shows the operation time and routes (machine sequence) of example of four machines and three jobs (Disjunctive Graph with CDAS and RDAS).

Table 4.1 An Example of Four Machines and Three Jobs

Jobs	Machine Sequence	Operation Time
1	1→2→3	p_{11}=10, p_{21}=8, p_{31}=4
2	2→1→4→3	p_{22}=8, p_{12}=3, p_{42}=5, p_{32}=6
3	1→2→4	p_{13}=4, p_{23}=7, p_{43}=3

Figure 4.4 shows a schedule which is obtained using SPT rule based on information from Table 4.1. In Figure 4.4, the dashed lines represent CDAS; the dot-dashed lines represent RDAS; the solid lines represent a set of conjunctive arcs.

The definition of a CDA is explained by using an example of the arc from N_{11} to N_{12}. r_{11}=4 is the release time of the longest path from the *source* (U) to N_{11} (U→N_{13}→N_{11}), r_{22}=0 is the release time of the longest path from the *source* (U) to N_{22} (U→N_{22}), p_{11}=10, p_{22}=8, and $r_{11}+p_{11} > r_{22}+ p_{22}$, thus we have proof that the arc from N_{11} to N_{12} is a CDA.

The arc from N_{22} to N_{21}, is a RDA since r_{11}=4, p_{11}=10, r_{22}=0, p_{22}=8, and $r_{22}+ p_{22}< r_{11}+p_{11}$.

In the optimisation process, a Conflicted Disjunctive Arc (CDA) plays a key role in reducing makespan in a schedule and a Released Disjunctive Arc (RDA) is considered an unimportant factor. Thus, the CKPO algorithms 4.1-4.3 are developed based on the concept of a critical CDA in the following Sections 4.4.2, 4.4.3, and 4.4.5.

The classification of DA brings a simple solution to find the longest (critical) path for a given schedule referring to **Rule 4.2**.

Rule 4.2 (To determine longest (critical) paths for Jm||C_{max})

When a schedule is provided, the last operation (node) with the completion time is taken as the **begin node**. The longest path only consists of nodes, CDAs, and conjunctive arcs. The method of determining the longest path is as follows`:

Current node = Begin node

If there are a CDA and a conjunctive arc which come to the current node (end node)

The longest path includes the CDA

Current node = New node (from the previous node along the CDA)

Else

The longest path includes the conjunctive arc

Current node = New node (from the previous node along the conjunctive arc)

This procedure above will continue in the backward manner until reaching the source node. The path obtained using **Rule 4.2** is the longest path. If there exists more than one last operation with the same completion time, these critical paths can be determined by **Rule 4.2** one by one.

Figure 4.5 shows the longest path using **Rule 4.2**. Node (4,3), which is the last operation, is considered as the begin node. There are two arcs (4,2)→(4,3) and (2,3)→(4,3) which can directly reach the current node (4,3). The longest path chooses arc (2,3)→(4,3) because arc (4,2)→(4,3) is a RDA and arc (2,3)→(4,3) is a

conjunctive arc. Then the current node becomes node (2,3) by following arc (2,3)→(4,3) from node (4,3). Now, two arcs (2,1)→(2,3) and (1,3)→(2,3) can directly reach the current node (2,3). The longest path takes arc (2,1)→(2,3) because arc (2,1)→(2,3) is a CDA. Using **Rule 4.2**, the longest path is U→(1,1)→(2,1)→(2,3)→(4,3)→V.

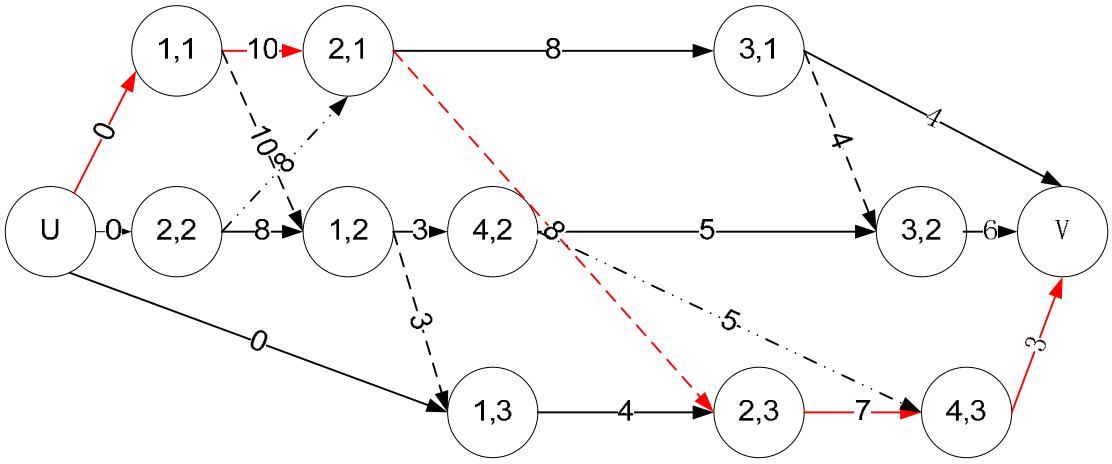


Figure 4.5 Critical Path using Rule 4.2 in Precedence Graph

4.4.2 CKPO Algorithm for Jm||C_{max}

Jm||C_{max} is a classical JSSP of n jobs to be executed on m machines without preemption. A machine can only process one job at a time and when the operation starts, it cannot be interrupted. The machine is only available after finishing the current job. The objective is to minimise C_{max} which can be obtained from a precedence graph. The performance analysis module is embedded in the CKPO algorithm for simplicity. This section discusses the basic methods and concepts of CKPO. A CKPO algorithm is proposed, which is an extension and modification from TS. It is required to construct the six components for a classical TS algorithm. The six components are described as follows.

1. **A disjunctive graph representation** is modified for a solution.
2. **An initial solution** is the starting solution for *move* and this solution is

obtained from the simulation model.

3. **An objective (cost) function** needs to be achieved. In this case, a simple objective function of minimising C_{max} is considered.
4. **Neighbourhood strategy** describes how to generate a new solution from the current solution via one or several *move* actions.
5. ***A Move issue*** discusses which *moves* are forbidden and which *moves* are allowed. This component is related to tabu list and aspiration criterion.
6. **Stopping criteria** define several situations of finishing an optimisation.

Components 4 to 6 will be further described in this section.

For component 4, there are several strategies to define a neighbourhood, for example Nz1, Nz2, Nz3, Nz4, Nz5 and Nz6 introduced in Section 2.10.3.3. Among them, Nz2 is a relative generic strategy and the others are its modifications. Consequently, a neighbourhood structure strategy for the CKPO in $Jm||C_{max}$ is developed from the concept of swapping first or last operation with another operation of Nz2 together with a hybrid control method. This strategy is able to lead to a feasible solution but may have a poor connectivity property. Therefore, the global minimum may not be obtained by this strategy. If a global minimum solution is required, Nz1 should be adopted to replace this strategy. This research aims to find a 'good enough' solution in a reasonable time, thus this strategy is a suitable method to achieve the objective**.** The details of different neighbourhood strategies are reviewed in Section 2.10.3.3.

In CKPO, a hybrid control method composed of a rank rule and OP (the number of the neighbours) increases the possibility of generating 'good' neighbourhood structures of each iteration. The OP is used to limit the number of the neighbourhood structures of each iteration to reduce computation time, but it does not include the number of the neighbourhood structures generated by *moves* in the tabu list. The rank rule focuses on

the objective function and gives the priority of each *move*. In the neighbourhood structure generated by a *move* with a high priority, a 'better solution' can be found with a higher possibility than one generated by a move with a low priority. Consequently, *moves* should be ordered by their priority. Different objectives should adopt different rank rules and the rank rule developed in this section is concentrated on minimising C_{max} . The rank rule for C_{max} in Jm||C_{max} is shown as follows:

1. Counting a tail time difference (q_{ij}- qc_{ij}) for each start node of the disjunctive arcs which are first or last disjunctive arcs in critical blocks on the critical path
2. Ranking the disjunctive arcs in a non-increasing order of their start nodes' tail time difference
3. Executing the swap (move) action in the pre-rank sequence

where q_{ij} is the tail time of operation ij (the length of the longest path from operation ij to the *sink*), qc_{ij} is the conjunctive tail time of operation ij (the length of the shortest path from operation ij to the *sink*), p_{ij} is the operation time of operation ij.

For component 5, a *move* (also called a swap in CKPO) enables a solution to be forwarded to a neighbouring solution. If a *move* currently exists in the tabu list, this *move* is forbidden. Therefore, the length of the tabu list is an important number because it can affect the quality and efficiency of the search. If the length is too large, it brings many restrictions. On the other hand, for a short tabu list the situations of revisiting the recent solution cannot be avoided. To define an effective length of a tabu list is still an open research issue (Zhang et al., 2007). In this thesis, the length of the tabu list is determined by Zhang's (2007) method for n jobs and m machines

```
If n<=2m
  TB=int(random(L,1.4L))
else
```

TB=int(random(L,1.5L))

where TB is the Tabu Tenure (a length of tabu list) and L=10+n/m defined from experience.

Another issue relating to *move* is *aspiration criterion* and the *move* on a tabu list is still permitted if it satisfies the *aspiration criterion*. The *aspiration criterion* aims to avoid this situation if a better solution cannot be found because this *move* in the tabu list is forbidden. Thus, the *move* in the tabu list is still allowed if the solution is moved to the better solution than before.

For component 6, three common stopping criteria (Glover and Kochenberger, 2003) are adopted for CKPO, which are outlined as follows:

- After running OT iteration
- After NU iterations without an improvement of C_{max}
- When C_{max}<=VO (a threshold value)

Algorithm 4.1 presented as follow is to find the 'good' schedule under the constraints defined by the users.

n: the number of jobs

m: the number of machines

C_{max}(S) : C_{max} of a schedule S

L=10+n/m % L is a variable of n jobs and m machines %

If n<=2m Then

M=1.4L % M is a variable related to L%

else

M=1.5L

Endif

Algorithm 4.1 (CKPO for Jm||C_{max})

Input: L, M,

OP, % the number of the neighbours every iteration %

NU, % the number on unimproved iterations%

VO, % a threshold value of C_{max} %

TB, % the number of iterations %

S_0 % an initial schedule%

Output: an optimum schedule S_g

Step 1 (Initial Condition)

k=0 % k is a temporary variable %

c=0 % c is a temporary variable %

S= S_0 % S is a benchmark schedule given initially %

S_g= S_0

TL= Ø %TL is a tabu list %

Step 2 (Determining Conflicted DA)

TB = int(random(L,M)) % Tabu Tenure (a length of tabu list)%

Change the length of TL according to TB

SN= Ø % SN is a temporary array of start nodes%

ca_t= Ø % ca_t denotes a temporary arc %

LCDAS= Ø

Build G of S using **Rule 4.1** % G is a precedence graph of S %

Build an array L(S) using **Rule 4.2**

% L(S) represent critical paths of S %

For each $ca_i \in L(S)$ ∪ ca_i is the fist or last CDA in the critical block

% ca_i denotes an CDA%

Add a ca_i into the array LCDAS

Step 3 (Building a Sequence for Optimisation)

d=0

ca_t=Ø

S=Ø

For each $ca_i \in$ LCDAS

Count its tail time difference (q_{ij}- qc_{ij}) for sn_{ij} (start node) of ca_i

For each $ca_i \in$ LCDAS order by non-increasing its tail time differences

If $ca_i \notin$ TL Then

d=d+1

Add a sn_{ij} into array SN

If d>=OP Then

go to Step 4

Endfor

Step 4 (Rescheduling)

For each $sn_{ij} \in$ SN order by non-increasing its tail time difference

% a rank rule%

Swap the two nodes of ca_i with a start node sn_{ij}

Reschedule to generate a new schedule S* % using the SPT method described in Section 4.4.4%

If S= Ø $\cup$ $ca_i \notin$ TL Then

S=S*

ca_t=ca_i

Else If $ca_i \notin$ TL $\cup$ $C_{max}(S) > C_{max}(S^*)$ Then

S=S*

$ca_t=ca_i$

Endif

If $C_{max}(S_g)>C_{max}(S^*)$ Then

$S_g=S^*$

$S=S^*$

$ca_t=ca_i$

$c=0$

Endif

Endfor

Step 5 (Stopping Criteria)

If $ca_t= Ø$ Then

Exit

Reverse ca_t and add it into TL

$k=k+1$

If k>=OT Then

Exit

If OV>=$C_{max}(S_g)$ Then

Exit

If c>=NU Then

Exit

$c=c+1$

go to Step 2

It is noted that OT, OP, NU, VO are inputted by the users based on the problems to be solved.

Example 4.1 (Application of the CKPO Algorithm for Jm||C_{max})

Using an example of Table 4.1 to explain the CKPO algorithm. The initial solution is obtained using a SPT algorithm (Pinedo,2000). The CKPO algorithm is adopted to find a 'good enough' solution. In this example we choose OP=1, OT=4, NU=2, VO=25 and

L=10+n/m=10+3/4≈11

When n<=2m

TB=int(random(L,1.4L))=int(random(11,15))

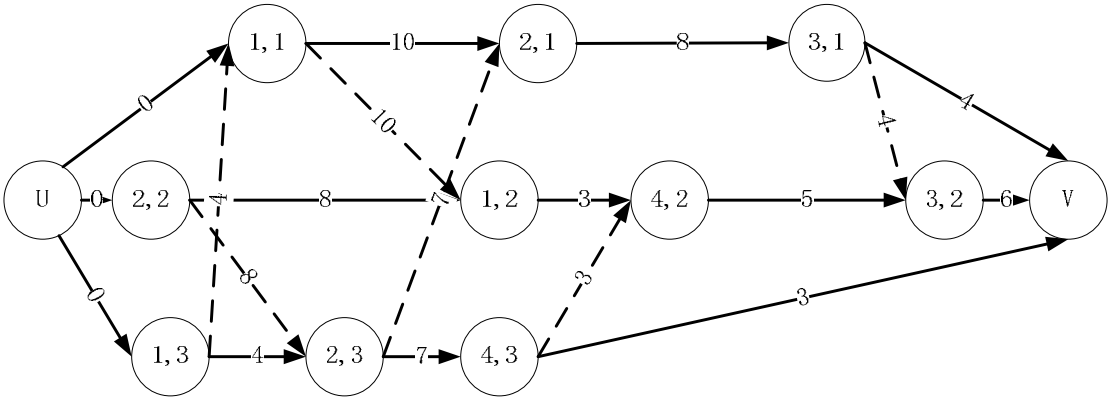


Figure 4.6 An Initial Solution for Application of CKPO for Jm||C_{max}

Figure 4.6 shows an initial solution S_0 by the SPT algorithm described in Section 2.10.3.1. The longest path from U to V using **Rule 4.2** is found: U→(2,2)→(2,3)→(2,1)→(3,1)→(3,2)→V with $C_{max}(S_0)$=33. S=S_0 and S_g=S_0. LCDAS is {(2,2)→(2,3), (2,3)→(2,1), (3,1)→(3,2)}. The *move* is (2,3)→(2,1) because the tail time difference (q_{ij}- qc_{ij}) 15 is the largest. TL={(2,1) → (2,3)}. Consequently, the rank results are obtained and shown in Table 4.2.

Table 4.2 Rank Results of an Initial Solution

Machine i, Project j	2,2	2,3	3,1
p_{ij}	8	7	4
q_{ij}	33	25	10
qc_{ij}	22	10	4
q_{ij}- qc_{ij}	11	15	6

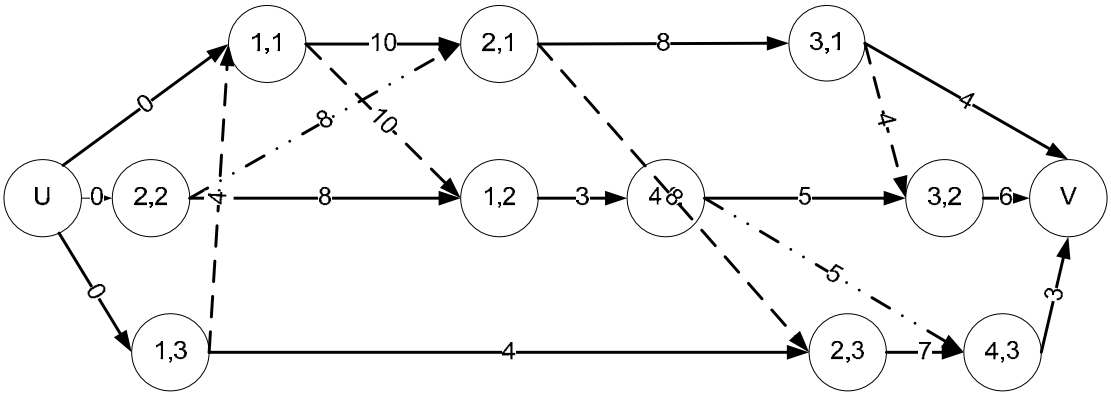


Figure 4.7 First Iteration of CKPO for Jm||C_{max}

Figure 4.7 shows a new schedule S* after swapping node (2,3) and node (2,1). Now, the longest paths are U→(1,3)→(1,1)→(2,1)→(3,1)→(3,2)→V and U→(1,3)→(1,1)→(2,1)→(2,3)→(4,3)→V with C_{max}(S*)=32. $C_{max}(S_g)$> C_{max}(S*), thus S_g=S* and S=S*. The *move* is (1,3) →(1,1) because of 18 and TL={(1,1) →(1,3), (2,1) →(2,3)}. This leads the rank results shown in Table 4.3.

Table 4.3 Rank Results of the First Iteration

Machine i, Project j	1,3	2,1	3,1
p_{ij}	4	8	4
q_{ij}	32	18	10
qc_{ij}	14	12	4
q_{ij}- qc_{ij}	18	6	6

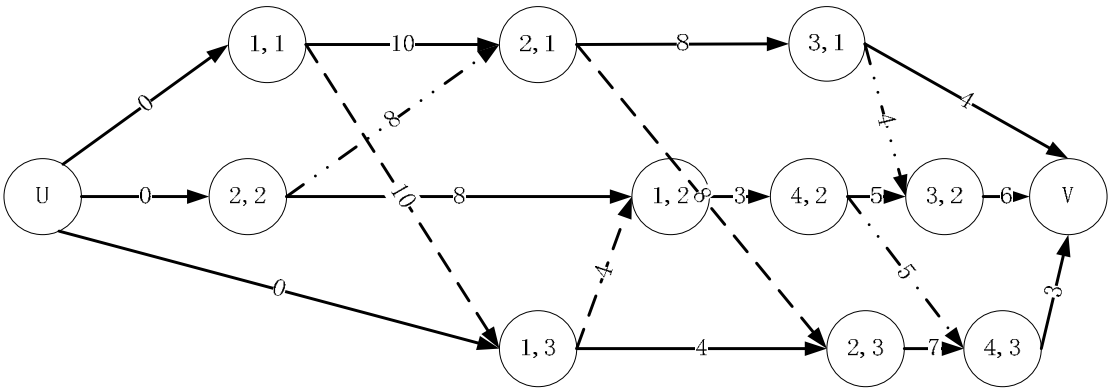


Figure 4.8 Second Iteration of CKPO for Jm||C_{max}

Figure 4.8 shows a new schedule S* when reversing a line (1,3) →(1,1). Now, the longest paths are U→(1,1)→(1,3)→(1,2)→(4,2)→(3,2)→V and U→(1,1)→(2,1)→(2,3)→(4,3)→V with $C_{max}(S^*)=28$. $C_{max}(S_g)> C_{max}(S^*)$, thus $S_g=S^*$ and S=S*. The *move* is (1,3)→(1,2) because Lines (1,1)→(1,3) and (2,1)→(2,3) are in the tabu list and the neighbourhood structure after the *moves* of Lines (1,1)→(1,3) and (2,1)→(2,3) can not satisfy aspiration criterion (<28 (C_{max})). TL=={(1,2) →(1,3),(1,1) →(1,3), (2,1) →(2,3)}.

Table 4.4 Rank Results of the Second Iteration

Machine i, Project j	1,1	1,3	2,1
p_{ij}	10	4	8
q_{ij}	28	18	18
qc_{ij}	22	14	12
q_{ij}- qc_{ij}	6	4	6

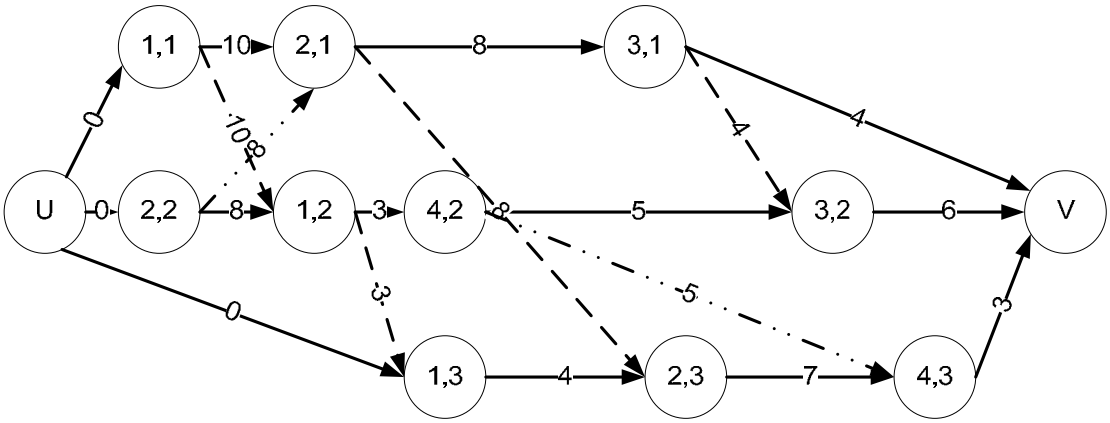


Figure 4.9 Third Iteration of CKPO for Jm||C_{max}

Figure 4.9 is a new schedule S* when reversing the arc (1,3)→(1,2). The longest paths are U→(1,1)→(2,1)→(3,1)→(3,2)→V and U→(1,1)→(2,1)→(2,3)→(4,3)→V with $C_{max}(S^*)=28$. $C_{max}(S_g)=C_{max}(S^*)$ and S=S*. The *move* is (3,1)→(3,2) because *move* of Line (2,1)→(2,3) in the tabu list can not satisfy aspiration criterion (<28(C_{max})) and TL=={(3,2)→(3,1), (1,2)→(1,3), (1,1)→(1,3), (2,1)→(2,3)}.

Table 4.5 Rank Results of the Third Iteration

Machine i, Project j	3,1	2,1
p_{ij}	4	8
q_{ij}	10	18
qc_{ij}	4	12
q_{ij}- qc_{ij}	6	6

Figure 4.10 shows a new schedule S* when reversing a line (3,1)→(3,2). The longest paths are U→(1,1)→(1,2)→(4,2)→(3,2)→(3,1)→V and U→(1,1)→(2,1)→(2,3)→(4,3)→V with $C_{max}(S^*)=28$. $C_{max}(S_g)=C_{max}(S^*)$ and S=S*. The reversing the arc (1,3)→(1,2) and the arc (3,1) →(3,2) cannot reduce C_{max}. Consequently, NU goes up to 2 which causes a stop. The output optimum result is S_g.

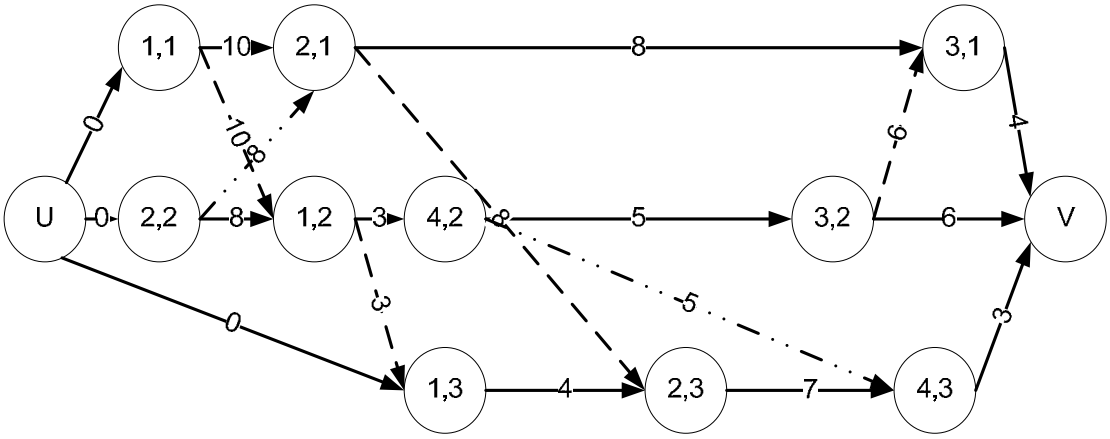


Figure 4.10 Fourth Iteration of CKPO for Jm||C_{max}

Figures 4.11 and 4.12 show the Gantt charts of the initial and output optimum schedules respectively.

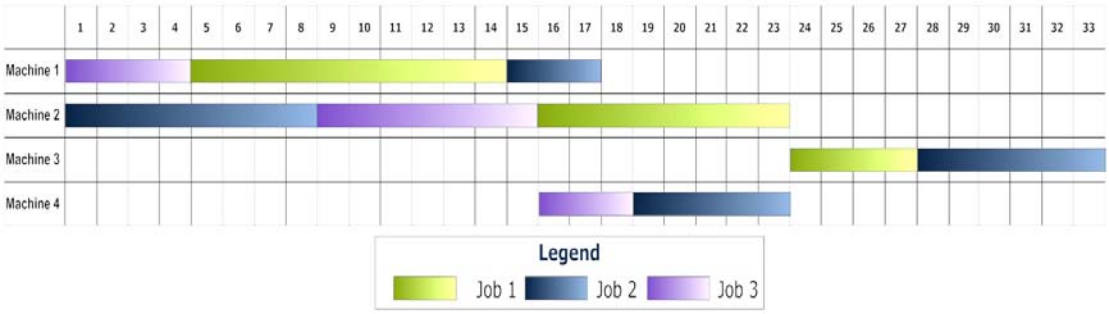


Figure 4.11 Gantt Chart of the Initial Schedule

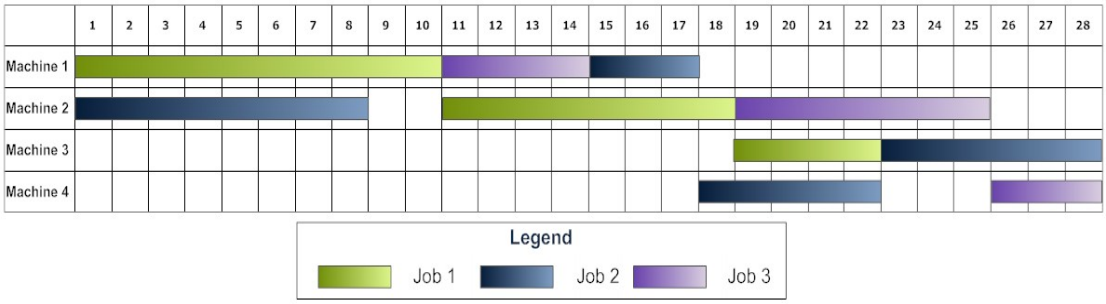


Figure 4.12 Gantt Chart of the Output Optimum Schedule

Example 4.1 is a classical case applied by the shifting bottleneck heuristics algorithm (Pinedo, 2008) and shows the advantages and characteristics of **Algorithm**

4.1. The shifting bottleneck heuristics algorithm (Pinedo, 2008) is used to divide Jm| $r_j|L_{max}$ into lots of an instance of $1|r_j|L_{max}$ and use an EDD rule to schedule the instance. The machine with a maximum (L_{max}) is determined as a bottleneck and is scheduled first by putting the related disjunctive arcs of the bottleneck into the disjunctive graph and then an optimisation for the bottleneck machine is implemented. The same procedures are applied for the remaining machines. Running the above optimisation process 10 iterations leads to the best solution C_{max}=28 of the four machines problem.

Compared to the application of the shifting bottleneck heuristics algorithm (Pinedo, 2008), the proposed CKPO algorithm only is run for four iterations based on an initial solution. In fact, a solution of C_{max}=28 is found at the second time. Consequently, it reduces the number of scenarios for finding the best solution. In the meantime, CKPO algorithm generates a feasible solution of each iteration and decision makers can set their own stopping criteria to satisfy their different requirements. Otherwise, the shifting bottleneck heuristics algorithm only provides a solution after running all iterations.

4.4.3 CKPO Algorithm for $Jm|ST_{id,b}|C_{max}$ with LS (job splitting)

In a CR field, we need to consider utilisation of both machines and labourers and their combinations, which may own different process time and set up time. This data relating to these combinations have been obtained for the UK-based case study. A combination including one machine and one person is denoted as a **CR Pair** (CRP) which expresses one labour using one machine to make one operation. The Suitable CR set (SCR) includes all CRPs that can be used for the operation. In this section, the proposed CKPO algorithm still includes the performance analysis module of C_{max}. In the later Section 4.4.5, a general cost function $\sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij}F_{ij}$ is used to replace the simple objective C_{max} and the proposed CKPO Algorithm for $Jm|ST_{id,b}|$ $\sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij}F_{ij}$ with LS is outlined.

In order to provide a possible representation of a schedule for optimisations under the complicated situation of the case study, a modified disjunctive graphic representation is proposed.

The parameters for defining (OCR, J, ECR) of operation (node) ij are described as follows:

J: the operation identification including the job number and the sequence number of operation (i,j)

OCR_{ij}: the occupied CRPs $\{(m_1,l_1)\ldots.(m_i,l_i)\ldots(m_n,l_n)\}$ of operation ij

ECR_{ij}: the best expected CRPs $\{(m_1,l_1)\ldots.(m_i,l_i)\ \ldots(m_n,l_n)\}\}$ for operation ij

i: the sequence number of the operation in job j

j: the job number (Work ID is used to represent the job number in the database)

m: machine

DV: the allowed maximum number of sub-jobs split from a job (Further details of the job splitting are given in Section 4.5)

l: people (labour)

The following notations will be used in the propose algorithm:

$$Tb_{ij}=r_{ij}$$

$$Te_{ij}=r_{ij}+p_{ij}$$

where Tb_{ij} is the start time of operation ij, Te_{ij} is the end time of operation ij, r_{ij} is the release time of operation ij, p_{ij} is the operation time of operation ij.

The following shows the relationships of SCR_{ij}, UCR_{ij}, OCR_{ij}, ECR_{ij}.

$$OCR_{ij} \in UCR_{ij} \in SCR_{ij}$$

$$ECR_{ij} \in SCR_{ij}$$

where SCR_{ij} is the total set of CPRs which operation ij may occupy. UCR_{ij} is the set of unoccupied (available) CPRs which operation ij can use at its begin time. OCR_{ij} is the

set of CRPs which operation ij actually occupies and may be several or total CRPs of UCR_{ij}, ECR_{ij} is the number of the best CRPs. ECR_{ij} is assumed to be able to process a batch of products at the shortest time. If operation ij takes all CRPs of ECR_{ij}, it usually achieves the shortest operation time for a given condition.

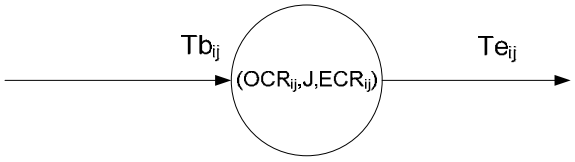


Figure 4.13 Graphic Representation of Operation ij

Figure 4.13 shows operation ij (node) with two time attributes (Begin time Tb_{ij} and End time Te_{ij}). In order to simplify the combinations' complexity, the simple Shortest Processing Time First (SPT) method is adopted for combining the machines and the labourers to construct an ECR for an operation and is defined in **Rule 4.3**.

Rule 4.3 (To construct an ECR_{ij})

To an operation ij of a job (batch) with Q (number of items in the batch), the CRP_x with the lowest operation time $p_{ij}= (T_{p*} Q/DV +T_s)$ (T_p is process time, T_s is set up time) in SCR_{ij} is chosen as the first CRP of ECR_{ij}. CRP_x with the first lowest operation time is removed from SCR_{ij}. This concept can be used iteratively to select the best CRPs for ECR_{ij} with the lowest operation time.

The DOA algorithm is adopted to get the set of CRPs with the lowest operation time among the chosen CRPs. The number of the CRPs should not be larger than DV (maximum number of the splitting job). In this case study, a database table is used to store T_p and T_s for all related CRPs classified by jobs and operations. Consequently, ECR is able to be obtained from this database table using **Rule 4.3**. Further details of CRPs are discussed in Section 4.5.

For a given job, the occupied CRPs of an operation are considered as a machine of

the operation in the original disjunctive graph representation. Machines or labourers of CR_{ij} may be changed after taking a swap action using the CKPO algorithm. This change may lead to the change of the operation time. To address the changes, a series of modifications of the classic JSSP are required. Thus, a modified CKPO algorithm is proposed for $Jm|ST_{id,b}|C_{max}$ with LS. Prior to introducing the modified algorithm, the followings rules, which are used in this algorithm, are proposed.

Rule 4.4 (To determine CDA,RDA, and DA)

There are two operation nodes N_{ij}, N_{kt}, and N_{ft} is the predecessor operation of operation N_{kt} on job t. If between r_{kt} and Te_{ft}, N_{ij} occupies a CR or several CRs of ECR_{kt} and OCR_{kt}, there exists a CDA directed from N_{ij} to N_{kt}. If N_{ij} is a predecessor operation of operation N_{kt} on a CR task of OCR_{kt} before r_{kt}, there exists a RDA directed from N_{ij} to N_{kt}.

$$DA= CDA \cup RDA$$

Rule 4.5 (To determine ef)

To describe the relationship of two operations (nodes) N_{ij} and N_{kt} linked by a CDA, a conflicted effective parameter ef is introduced to evaluate the relationship level of a CDA ($N_{ij} \rightarrow N_{kt}$).

$CM_n= M_{ij} \cap M_{kt}$ and $CL_n= L_{ij} \cap L_{kt}$

If ($M_{kt}= CM_n$ and $L_{kt}= CL_n$) % means ($M_{kt} \in M_{ij}$ and $L_{kt} \in L_{ij}$) %

ef=1

p_{ij} = operation time of N_{ij} on DA

Else

ef=0

p_{ij} =0 on DA

where M_{ij} is the machine set of OCR_{ij}, M_{kt} is the machine set of OCR_{kt}, L_{ij} is the labour set of OCR_{ij}, L_{kt} is the labour set of OCR_{kt}.

Figure 4.14 shows the changes before and after swapping N_{ij} and N_{kt} (if there only exits a CDA directed to N_{kt}) when ef=1.

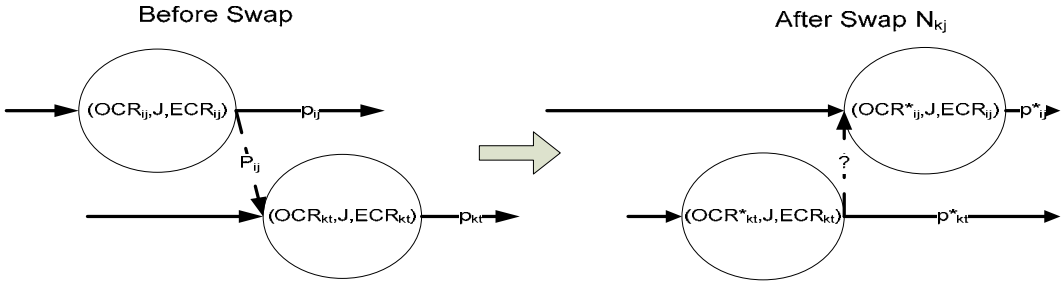


Figure 4.14 Different between before and after Swapping

There are several changing points to be considered and outlined as follows:

OCRs of the two nodes and p (operation time) of their related conjunctive arcs are changed. p_{ij} and p_{kt} are changed to p^*_{ij} and p^*_{kt} because operations ij and kt may occupy the different CRs after swapping, where operation kt is set to the highest priority. The part of ECR can be obtained from operation ij to reduce its operation time. Consequently, OCR_{kt} is changed to OCR^*_{kt}, in the mean time OCR_{ij} of operation ij are changed responsively (details given in Section 4.4.4). Due to the changes of the occupied CRs of operations ij and kt, p_{kt} of operation ij may be reduced and p_{ij} of operation kt may be increased. This is defined as follows:

$$p_{ij} <= p^*_{ij} \text{ and } p_{kt} => p^*_{kt}$$

When ef=0, their OCRs and the operation time of both operations are changed.

The method of finding out the longest path of a particular schedule is altered because the changes of the disjunctive graph representation are made to suit for the multi CRs occupied by one operation. **Rule 4.6** is proposed to find the critical path.

Rule 4.6 (To determine the critical path for $Jm|ST_{id,b}$ with LS)

The begin node is the last operation of the critical job i using **Rule 4.0**. The critical

path only consists of nodes, CDAs with ef=1, and conjunctive arcs. The method of finding the longest path is as follows:

Current node = Begin node

If there is a CDA with ef=1 which comes to the current node (end node)

The critical path includes the CDA with ef=1

Current node= New node (from the previous node along the CDA)

Else

The critical path includes the conjunctive arc

Current node = New node (from the previous node along the conjunctive arc)

This will continue in the backward manner until reaching the source node. The path obtained using **Rule 4.6** is the critical path. If there exists more than one last operation with the same completion time, these critical paths can be determined by **Rule 4.6** one by one.

In the modified disjunctive graph representation, one node may be connected by several disjunctive arcs. However, from Nz1 to Nz6 neighbourhood strategies, all neighbourhood strategies are based on a two-two lines network, which means a node at most can own two arcs in and two arcs out. When generating a new neighbourhood structure by Nz1 to Nz6 methods, a certain CR is assigned to the nodes at different time and the sequence of the CR occupied by these nodes is different. However, the modified disjunctive graph representation which expresses the job shop of $Jm|ST_{id,b}|*$ requires a new neighbourhood structure generated by not only altering the sequence of occupying CRs but also changing occupied CRs. $OCR_{ij} \in UCR_{ij}$ of an operation node means sometimes the numbers of CRs are redundant for this node. Consequently, when some CRs are occupied by the other operation node, the operation node may still start at the same time just using the different CRs. These original neighbourhood strategies are not suitable for the modified disjunctive graph representation, and a new neighbourhood

strategy is required.

To develop an alternative neighbourhood strategy, the key factor is to determine which nodes need to be swapped which can bring the largest contribution to the objective. For a complicated schedule, a rescheduling process may spend too much time and thus only swapping two nodes is not sufficient every iteration. As a rescheduling process will happen every iteration, more iterations cause more rescheduling processes which result in excessive CPU time. Therefore, a neighbourhood strategy based on an end node is proposed. This end node belongs to a critical path and directly linked by a or several CDAs. **Swapping** an end node (an optimisation node) means this node takes all CRs what it wants after the end time of its predecessor operation on a job task. The optimisation node and these nodes linked by CDAs that direct to the optimisation node are defined as a **critical block.** In the block, other nodes are swapped with the optimisation node to generate a new neighbourhood structure. Figure 4.15 shows the precedence graph of an initial solution.

In Figure 4.15, B_y is chosen as an optimisation node and a new neighbourhood structure is formed based on this scenario. In a new neighbourhood structure, the node B_y should occupy its ECR_B_y after Te_B_{y-1} (the end time of node B_{y-1}). This leads to $Te\_B_{y-1}= Tb^*\_B_y$ after a *swap*. The original concept of a critical block is modified and now it represents operation nodes directly affected by node B_y from Te_B_{y-1} to Tb_B_y. Taking node B_y as an example, its critical block is (C_z, A_x, $D_{u+2,}$ B_y) ordered by their begin time. The proposed method of generating a new neighbourhood structure is to move the optimisation node to the begin position of its critical block and let the optimisation operation node choose CRPs before the other nodes of its critical block. If no other optimisation nodes affect, its OCR may equal to its ECR and the end time of its predecessor operation on a job task equals to its start time. If there exist other optimisation nodes, the *swap* method follows an Only One Process (OOP) method demonstrated in reschedule Section 4.4.4. After the *swap* optimisation, the affected

nodes are processed step by step. The following reschedule section gives the detailed illustration on assigning the affective nodes.

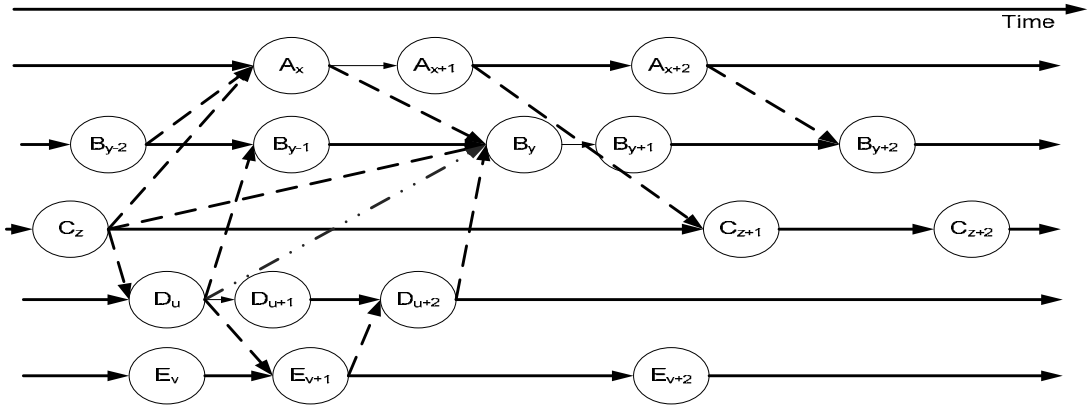


Figure 4.15 An Initial Solution for a Swap in Jm|$ST_{id,b}$|Cmax with LS

Figure 4.16 shows a possible solution after a *swap* and there are several possible changes of relationships between the optimisation node and its relative nodes. The *swap* may reverse their disjunctive arcs, such as nodes C_z and D_{u+2}, and delete their disjunctive arcs, like A_x. This action may delay the relative operation node, like D_{u+2}, and may not delay their start time, such as C_z and A_x, as these nodes are able to take other CRs to replace the CRs occupied by the optimisation node. However, the replacing method of CRs mostly may increase operation times of the relative nodes and delay their successor operation nodes like A_{x+1} and C_{z+1}.

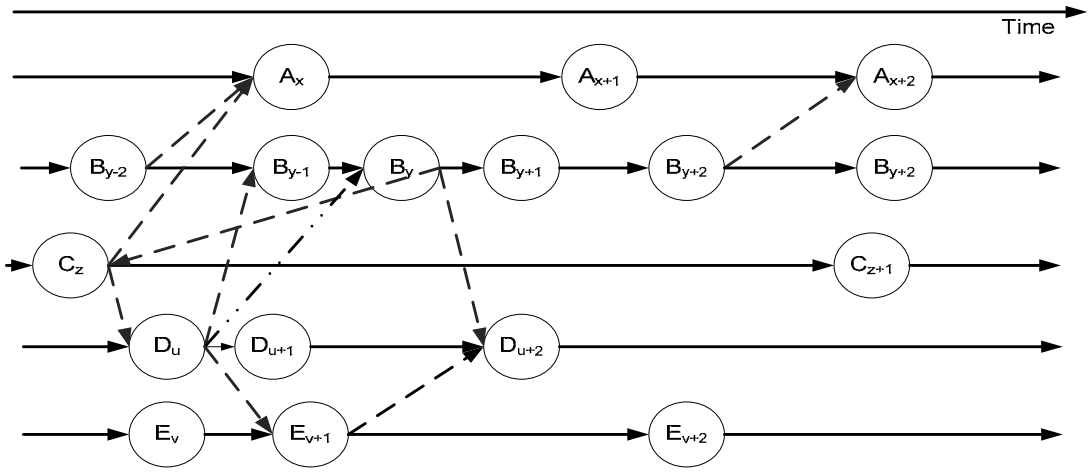


Figure 4.16 Optimum Solution after a Swap

The *tabu move* needs to give a new definition because a *swap* process does not normally happen in two nodes and OCR of the affective node is changeable at a rescheduling process. **Rule 4.7** shows how to determine whether the *swap* is tabu or not.

Rule 4.7 (To determine the tabu move)

The TL is considered as the list of the optimisation nodes and only TB recently optimisation nodes own a tabu status. There are several proposed methods of combinations of four parameters to define a tabu move. The notations of the four parameters are described as follows:

ACR: If a *swap* moves all CRs in the ECR of an optimisation node with a tabu status to the other node, this *swap* should be forbidden except when meeting aspiration criterion.

ACRP: If a *swap* moves all CRPs in the ECR of an optimisation node with a tabu status to the other node, this *swap* should be forbidden except when meeting aspiration criterion.

AO: allowing reusing optimisation nodes in the tabu list

FO: forbidding reusing optimisation nodes in the tabu list

Every tabu design combination needs to choose two parameters, one from ACR and ACRP and the other from AO and FO. The combination (ACRP, FO) expresses if a *swap* is determined a tabu move by the ACRP method or by the FO method, this *swap* is a tabu move. A combination (ACRP, FO) means if a *swap* moves all CRPs in the ECR of an optimisation node with a tabu status to the other node or this current optimisation node owns a tabu status, this *swap* should be forbidden except when meeting aspiration criterion.

The Zhang's et al. method of replacing m machines with m CRPs is adopted to count the length of the tabu list (Zhang et al., 2007). Stopping criterion and aspiration

criterion applied are given in Section 4.4.2.

Algorithm 4.2 presented below is to find the optimum schedule under the constraints defined by the users. The following parameters are used to find the inputs of Algorithm 4.2.

n is number of jobs, m is number of CRPs, and $C_{max}(S)$ is the C_{max} of a schedule S. Consequently, these are defined as follows:

L=10+n/m % L is a variable of n jobs and m machines %

If n<=2m Then

M=1.4L % M is a variable related to L%

else

M=1.5L

Endif

Algorithm 4.2 (CKPO for $J_m|ST_{id,b}|C_{max}$)

Input: L, M,

OP, % the number of the neighbours every iteration %

NU, % the number on unimproved iterations%

VO, % a threshold value of C_{max} %

OT, % the number of iterations %

S_0, % the initial schedule%,

TB combination % the tabu design combination, such as (ACRP,FO)%

Output: S_g % an optimum schedule %

Step 1 (Initial Condition)

k=0 % k is a temporary variable %

c=0 % c is a temporary variable %

$S = S_0$ % S is a benchmark schedule %

$S_g = S_0$

TL=Ø %TL is a tabu list %

Step 2 (Determining CDAS)

TB=int(random(L,M)) % Tabu Tenure (a length of tabu list)%

Change the length of TL according to TB

EN=Ø % EN is an array of end nodes%

Build G of S using **Rules 4.4 and 4.5** % G is a precedence graph of S %

Build an array L(S) using **Rule 4.6** % L(S) is a critical path%

Step 3 (Using a backward method for an Optimisation)

d=0

$en_t = Ø$ % en_t is temporary node %

S=Ø

For each $en_{ij} \in L(S)$ order by non-increasing its end time

% a backward method, en_{ij} denotes a end node %

If any member of CDAS directs to the end node en_{ij} Then

Put en_{ij} into EN

Endfor

Step 4 (Rescheduling)

For each $en_{ij} \in EN$

Swap en_{ij} % assign the resources in other nodes of the critical block to en_{ij} %

Reschedule to generate a new schedule S*

% presented in Section 4.4.4%

Determine the tabu status of the swap according the rule of TB combination using **Rule 4.7**

If S= Ø $\cup en_{ij}$'s swap is not tabu Then

$S=S^*$

$en_t=en_{ij}$

Else if en_{ij}'s swap is not tabu $\cup$ $C_{max}(S)> C_{max}(S^*)$ Then

$S=S^*$

$en_t=en_{ij}$

Endif

If $C_{max}(S_g)> C_{max}(S^*)$

$S_g=S^*$

$S=S^*$

c=0

$en_t= en_{ij}$

Endif

If en_{ij}'s swap is not tabu

d=d+1

Endif

If d>=OP Then

go to Step 5

Endfor

Step 5 (Stopping Criteria)

If en_t= Ø Then

Exit

If en_t's swap is tabu Then

Delete en_t from TL

put en_t into TL

k=k+1

If k>=OT Then

Exit

If OV>=C_{max} (S_g) Then

    Exit

If c>=NU Then

    Exit

c=c+1

go to Step 2

4.4.4 Rescheduling Process

After a *swap* process, the benchmark schedule S will be replaced by S*. The following discusses how to reschedule the solution effectively.

According to the characteristics of a disjunctive graph representation, the arcs show the relationships among operation nodes, and only operations directly linked to the changed operations are referred to as an **affected node** and need to be checked whether their status are changed or not. The partial schedule made by operations which are not directly linked by the changed operations still keeps unchanged at the current stage, and this means the attributes of these operations and their related arcs, such as OCR, J, ECR, p, Tb, Te and ef, do not needed to be modified. The other operations related to the changed operations should be calculated their several attributes: OCR, p, T_s and T_e. The arcs directly link to the changed operations may disappear. If these arcs still exist, they need to be checked their two attributes: released or conflicted and ef.

Figure 4.17 shows an original schedule and a new schedule after swapping N22 and N21. '*' shows the changed operations and operation time. '?' expresses the attributes of the arcs need to be determined.

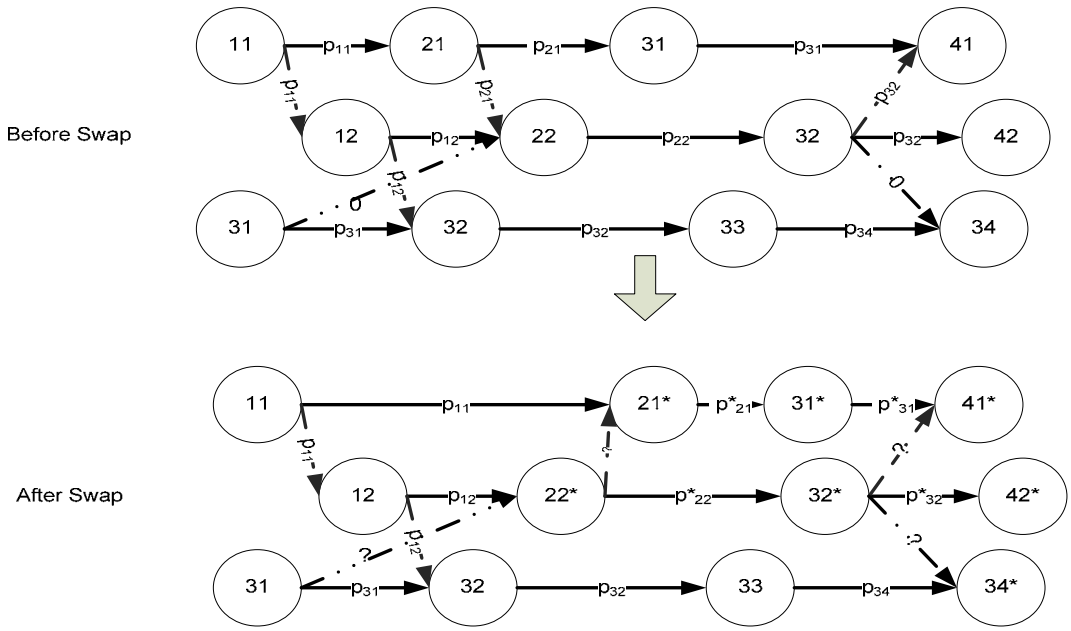


Figure 4.17 Before a *Swap* and after a *Swap* (and rescheduling)

A basic Priority Rule (PR) to assign CRs to the operations during a rescheduling process is described as follows:

Rule 4.8 (priority rule of assigning CRs to operations)

Firstly the status of the node, if the node is an optimisation node in the tabu list, it is processed before the other nodes.

Secondly the priority of a job, the high priority job gets the CRs prior to the next higher one.

Thirdly the due date, the early node gets the CRs before the later node.

Fourthly the arrival date, the early node are assigned CRs before the later node.

The system gives a choice to define a PR for comparing results under various scenarios for adding knowledge to build a serial of rules embedded in the system to choose suitable PRs for different situations.

Normally, a rescheduling process follows the time horizon to go forward and not backward, and a process pointer is used to identify a time stamp of the current operation node to be handled. From early to later on a timetable, the affected operation nodes are

recalculated one by one. The normal nodes only seek their related nodes behind them on the time horizon. If every node satisfies this rule, seeking the nodes after them, a cycle, which may result in a deadlock problem, can be avoided. However, there may exist the optimisation nodes in the tabu list and these nodes can take expected resources from their related operation nodes before it on the time horizon. Thus, when meeting the optimisation nodes, the process pointer only going forward must be broken. Under this situation, the time stamp should go back to the earliest affected operation nodes then go forward again. This may lead to a cycle which causes a deadlock. Thus, the rescheduling process should consider a method to effectively avoid the cycles (deadlock).

To avoid a cycle (deadlock), the special mechanism, called **Only One Procession** (OOP) is proposed. When any node affects an optimisation node, this optimisation node follows a *swap* method mentioned in the previous Section 4.4.3, namely taking their expected resources from the related operation nodes. This *swap* process cannot be changed because it is an important and necessary part to an optimisation process. Therefore, the *swap* process should remain and the method of avoiding the cycle (deadlock) is defined on how to avoid the optimised node, which has been *swapped* at the current iteration, being affected again by other nodes, and its *swap* is processed once in the rescheduling process. The basic idea is to generate a **protection netting** for this optimised node after *swapping* it once. This protection netting is able to forbid any node out of the netting affects the nodes inside the netting including the optimised node. Based on the concept of a protection netting, the OOP rule is developed to achieve the aim of avoiding a deadlock.

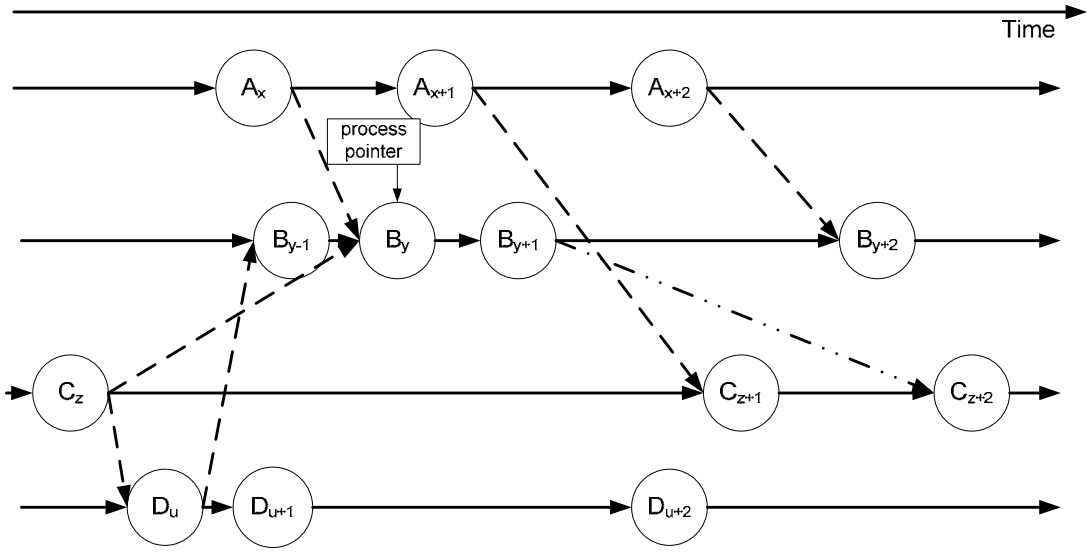


Figure 4.18 An Initial Stage before a Rescheduling Process

Figure 4.18 shows an OOP method as an example where the optimisation node B_y is prepared to apply the *swap* process and other nodes are all normal nodes. A process pointer, pointed to node B_y, means this node will be processed at the next step. At the initial stage, nodes A_x and C_z connected with node B_y (optimisation node) by CDAs. After the *swap* process, the status of nodes A_x, C_z, and B_{y+1} linked to node B_y are recalculated. Node C_z occurs at the earliest time point compared with the other two nodes A_x and B_{y+1}. Thus, the process pointer goes backward to node C_z, which means the next processing node is node C_z. Now, a protection netting is generated to protect the current optimisation node B_y and its predecessor operation nodes on a job task (conjunctive arcs) which their end time is bigger than the start time of earliest affected node C_z (normal node) as shown in Figure 4.19. If node C_z is an optimisation node, a protection netting should cover node B_y and its predecessor operation nodes on a job task which their end time is bigger than the end time of the predecessor operation node of node C_z on a job task. The netting avoids any other outside nodes occupying CRs belonging to the inside nodes until completing the operation B_y. OCRs of all nodes in the protection netting cannot be changed during this rescheduling process. Consequently, both of the original CDAs from A_x, C_z to B_y are removed. Any outside node is not able to affect the inside nodes and this represents a particular schedule in the

netting fixed during this rescheduling process. The optimisation node B_y cannot make the process pointer go backward on the time horizon again. Every optimisation node can only take a *swap* during every rescheduling process and it is called an **OOP rule**. If an optimisation node is being processed and meets another protection netting B which has already been generated by another optimisation node, this optimisation node follows the OOP rule and is not allowed to take any CRs occupied by the nodes in the protecting netting B. This rule may negatively affect the optimum schedule and 'block' the search space where a 'better' solution may exist. However, it is an effective method to remove a cycle and avoid a deadlock.

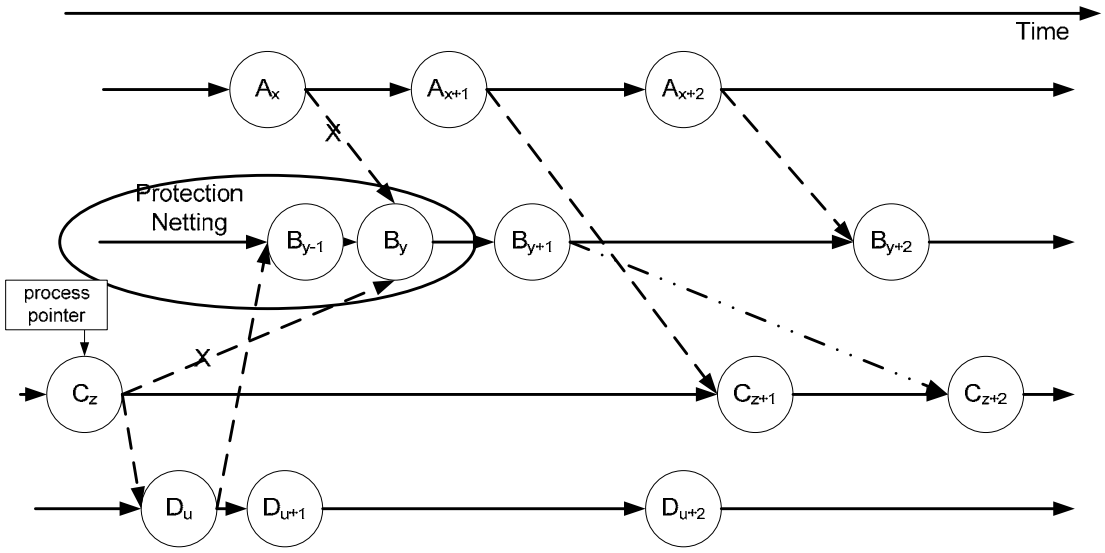


Figure 4.19 Protection Netting during a Rescheduling Process

The rescheduling process of the example shown in Figure 4.19 is discussed as follows:

1. Firstly, $OCR\_B_y=ECR\_B_y$ after a *swap* and $Te\_B_{y-1}=Tb\_B_y$ ($r_{By}=r_{By-1}+p_{By-1}$). A protection netting is generated to protect two nodes By and By-1 ($Tb\_C_z<Te\_B_{y-1}$ and $Tb\_C_z>Te\_B_{y-2}$). Then, the process pointer goes backward to its earliest affected node C_z.
2. Secondly, node C_z checks all available suitable CRPs for calculating its operation time, which satisfies two rules: $M\_C_z \notin M\_B_{y-1} U\ M\_B_y$ and

$L\_C_z \notin L\_B_{y-1} \cup L\_B_y$ before Te_By ($r_{By} + p_{By}$). (Where M_B_y, M_B_{y-1}, M_C_z represent the sets of occupied machines by nodes B_y, B_{y-1} and C_z respectively. L_B_y, L_B_{y-1}, L_C_z represent the sets of occupied labourers by node B_y, B_{y-1} and C_z respectively.) Then the other nodes are processed one by one, such as A_x, D_u, etc., and should stick with the two rules.

In the program, the function of protection nettings is achieved by two temporary tables: DTForbidlist and DTForRelist. DTForbidlist owns three coloumns: WorkID , Seq , EndTime. DTForRelist owns two coloumns: Ma_ID and Em_ID. DTForbidlist represents the optimised nodes which have been *swapped* at the current iterative using two coloumns: WorkID and Seq (represents J, a job idfentification of an operation in Section 4.4.3). EndTime expresses the complete time of the optimised nodes. DTForRelist expresses what CRPs (Ma_ID identifies machines and Em_ID represents labourers) are not able to be choosen for the current process.

Figure 4.20 shows the flow chart of the OOP method (solid lines show the work flows and dash lines denote the data flows). Before introducing the flow chart, the **current process node** is defined as the operation node which the process pointer currently points to. The flows of the OOP are described as follows:

1. To empty the table DTForRelist and a dataset CRs set which store UCR of a node at a time.
2. To take the optimisation nodes from DTForbidlist which their EndTime is larger than the time stamp of process pointer for the normal nodes. If current process node is an optimisation node, their EndTime is larger than the end time of the predecessor operation node of the current node on the job task. The CRs ,which are occupied by the defined predecessor nodes of these taken optimised nodes on the job task and the optimised nodes, are put into the table DTForRelist.
3. To put the related CRs of the current process node into the CRs set. If the

current process node is an optimisation node, take its suitable CRs into the CRs set. Otherwise, take its aviable CRs at the time stamp of process pointer and put them into the CRs set.

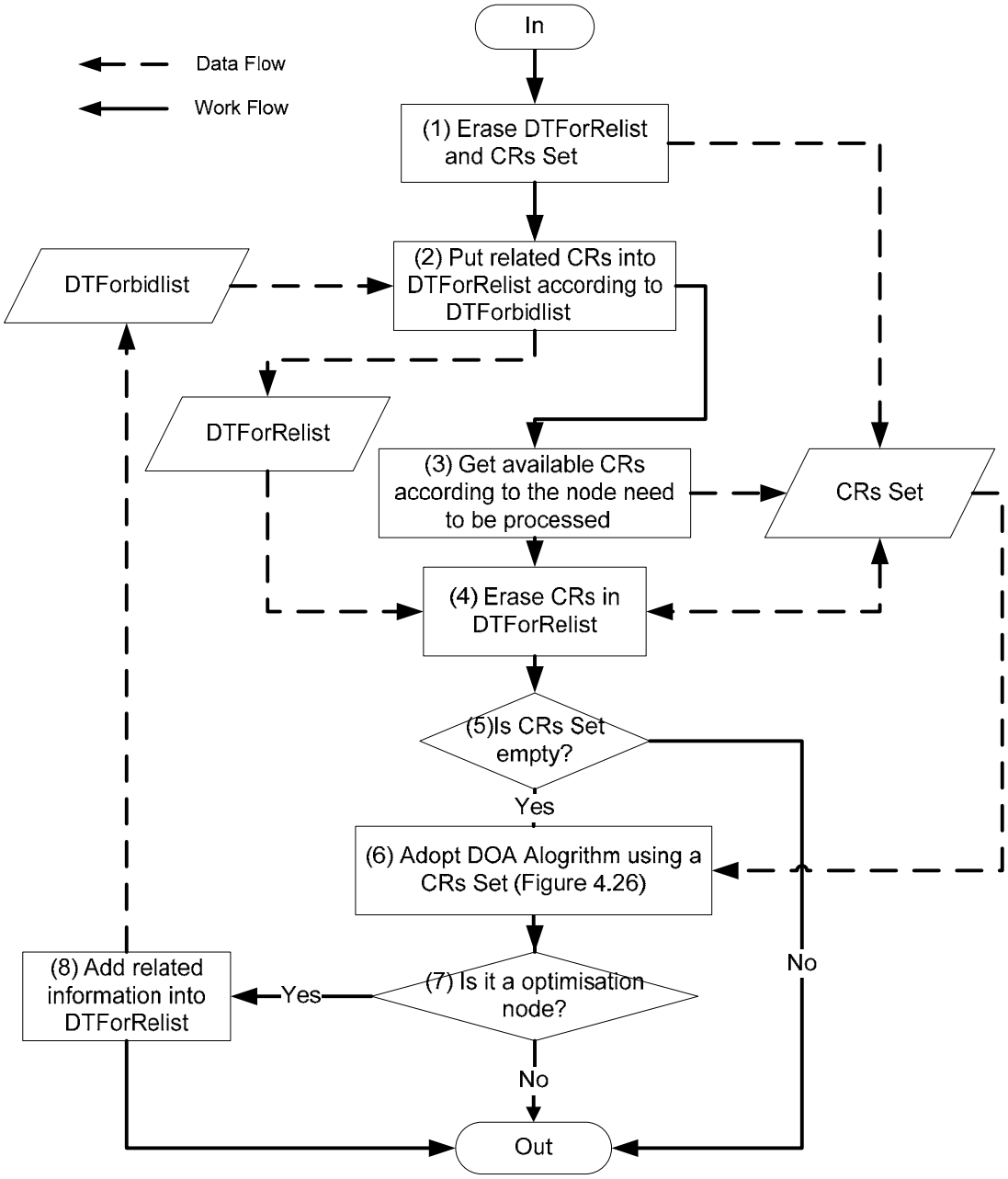


Figure 4.20 Flow Chart of an OOP Method

4. To delete the existing CRs in DTForRelist from the CRs set, and then the CRs in the CRs set are outside CRs of the protection nettings after the delete action.

5. To determine whether the CRs set is empty or not. If Yes, go to step 6. If No, go to Out.
6. To put the CRs set into the DOA algorithm (discussion in Section 4.5) to get what CRPs are occupied by the current process node and relative information such as quantity of the every sub-job assiged to every CRP and operation time.
7. To determine whether the current process node is or is not an optimisation node using the tabu list. If Yes, go to step 8. If No, directly go to Out.
8. To fill the relative information of the current process node into the DTForbidlist, and then go to Out.

Figure 4.21 briefly demonstrates the flow chart of main logic structure of the rescheduling process (solid lines show the work flows and dash lines denote the data flows). The temporary table DTctrlist stores the relative information of the node affected by the processed nodes. The flows of the rescheduling process are described as follows:

1. To get the corresponding information of the affected nodes of the current process node and put the information into the DTctrlist.
2. To adopt the OOP method and the DOA algorithm to get which CRPs the current process node should occupy and how long it takes.
3. To update the relative tables using corresponding values. For example, if the current process node is able to be assigned CRPs by DOA, the job of this step is to modify the relative information such as quantity, CRPs, operation time in the relative tables.
4. To check the status of the current process node before or after process if it is assigned CRPs by the DOA and delete the related nodes whose statuses are unchanged from the table DTctrlist.
5. To delete the current process node and relative information from DTctrlist

if the node is processed by DOA.

6. To determine whether the DTctrlist is empty or not. If Yes, go to Out. If No, go to step 7.
7. To put the process pointer to the next node at the same time stamp or the next earliest time stamp and take the node as the current process node go back to step 1. If more than one nodes need to be processed at a time stamp, the processing sequence refers to **Rule 4.8.**

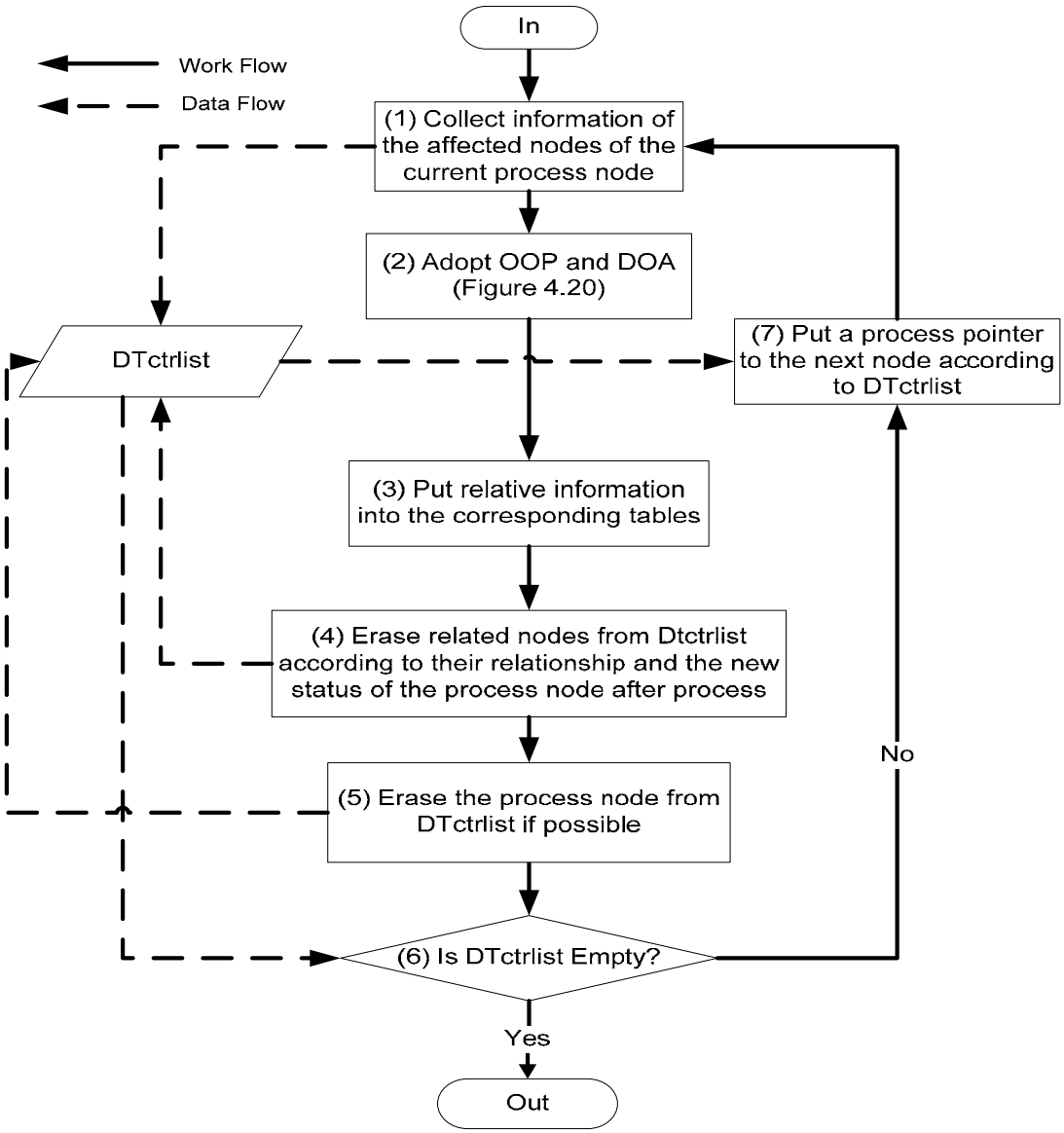


Figure 4.21 Flow Chart of Main Logic Structure of Reschedule

4.4.5 CKPO Algorithm for Jm|$ST_{id,b}$| $\sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij}F_{ij}$ with LS (job splitting)

In this section, the cost function of Algorithm 4.2 (CKPO for Jm|$ST_{id,b}$|Cmax) is extended to the generic cost function $\sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij}F_{ij}$ (F_{ij} represents the relevant cost function of criteria j in the ith job and this equation is outlined in Section 4.3). In industrial context, decision makers focus more on the tardiness than makespan (Cmax). Therefore taking Mean Weighed Tardiness ($\sum_{i=1}^{n} W_iT_i$) as a example to introduce a CKPO for Jm|$ST_{id,b}$|$\sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij}F_{ij}$. Figure 4.22 illustrates the flow chart of an optimisation block with Algorithm 4.3.

The notations of the parameters of Algorithm 4.3 are defined as follows:

Input:

O(S) is the cost function of a schedule S; OT is the number of iterations; OP is the number of the neighbours of each iteration; the method of calculating TB (Tabu Tenure); NU is the number on unimproved iterations; VO is the threshold value of $\sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij}F_{ij}$.

Output:

This algorithm may produce OP×OT solutions (every solution owns five database tables) and it outputs the performance analysis.

An Optimisation Block with Algorithm 4.3 (CKPO for $J_m|ST_{id,b}|\sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij}F_{ij}$)

1. To set all parameters of inputs, including OT, OP, TB, NU, VO, and an initial solution S, take a initial process such as emptying TL (Tabu list), etc.
2. To determine whether current number of iterations goes up to OT or not. If Yes, go to Output. If No, go to step 3.
3. To add relative results of the related solutions into the performance table

OP_Work based on the cost function. The initial solution is chosen as the related solution at first time. Otherwise, the related solutions are the OP (optimum) solutions of the current iteration. Now, the cost function is assumed as min($\sum_{i=1}^{n} W_i T_i$) (mean Weighted Tardiness).

4. To determine whether the number of unimproved iterations achieves NU or the tardiness value achieves VO. If Yes, go to Output. If No, go to step 5.

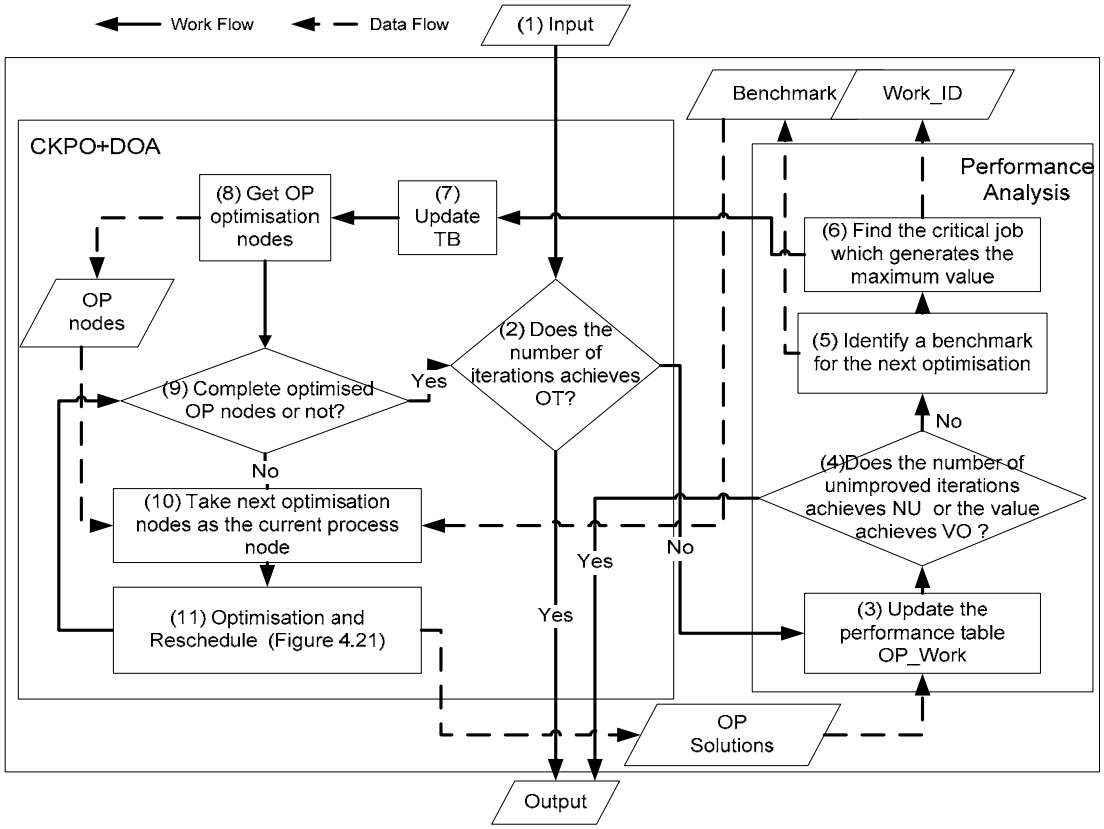


Figure 4.22 Optimisation Block with Algorithm 4.3

5. To compare the weighted tardiness of every OP solution of the current iteration and choose the solution with a minimum Weighted Tardiness as the benchmark. The initial solution is chosen as the benchmark at the first time.
6. To find the critical job (using Work_ID to identify a job) with maximum weighted tardiness (T_{max}) or an ordered list of critical jobs using **Rule 4.0**.

7. To calculate the Tabu Tenure TB and update the relative tables according to the calculated TB.
8. To get OP suitable optimisation nodes according to **Rule 4.9** and Work_ID, the benchmark and TB.
9. To determine whether the OP optimisation nodes of the current iteration are all processed or not. If Yes, go to step 2. If No, go to step 10.
10. To put the process pointer to the next optimisation node of the OP optimisation nodes and prepare to optimise the node.
11. To *swap* the current process node and reschedule the solution according to Figure 4.21 then put the solution into the optimum solution set. After that, go back to step 9.

Rule 4.9 (To determine optimisation nodes on a critical path)

A critical path of the job is generated referring to **Rule 4.6**. In order to reduce the calculated time of the reschedule, a back forward method is adopted which means considering the later node first, and then go backward to consider the earlier node on the time horizon. The node directly linked by CDAs is taken as a suitable optimisation node. If the node is only linked by RDAs or conjunctive arcs, the node is ignored because it has already occupied its ECR and *swapping* (optimising) it is unnecessary.

4.5 Dispatching Optimisation Algorithm

4.5.1 Problem Description

The batch scheduling with LS is used to assign a batch of components to several unrelated machines and labourers (CRPs) conducting the same operation in order to minimise a makespan. A batch of components needs to be taken using a defined operation, while there are UCR, including machines, labourers, etc. with the ability to handle this operation. Different CRPs take different lengths of time to deal with this operation. Therefore, ‘how to dispatch the optimum number of components to related

CRPs for minimising the maximum completion time of the operation' is a challenging issue and this dispatching problem is a sub-problem in the job-shop scheduling problem of the UK-based case study.

Figure 4.23 illustrates the batch dispatching problem in which there is a batch of components (Q is quantity of the batch) and n sets of available CRPs. The aim of the optimisation is to dispatch the batch of components into the available CRPs and minimize the maximum value of the operation time. For instance, when n=3 and Q=4, there are 12 alternatives to dispatch 4 components to 3 CRPs. A solution can be expressed as (Q1, Q2, Q3) where Q1 expresses the number of components which is dispatched into the first CRP, Q2 for the second CRP and Q3 for the third CRP respectively. The 12 solutions are recorded as (4,0,0), (0,4,0), (0,0,4), (3,1,0), (3,0,1), (1,3,0), (0,3,1), (1,0,3), (0,1,3), (2,2,0), (0,2,2), and (2,0,2).

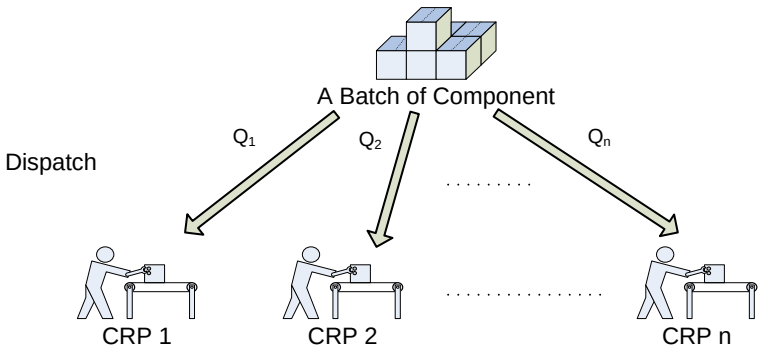


Figure 4.23 Batch Scheduling Problem

The best solution can be calculated by comparisons of all alternatives. If there are only two or three sets of CRPs, the best solution of this problem is easy to determine. However, if the number of sets of CRPs increases more than 3, its search space is enlarged so that the best solution is difficult to be found using a simple process. A proposed optimisation algorithm is used to handle this problem. The following section shows a mathematical description of the problem for developing a new algorithm.

Mathematical equations:

Minimise the maximum p_{max} among p_i ($max(p_i)$) and $max(p_i)$ is called operation time of the operation

Subject to

$$p_i = Ts_i + Tp_i Q_i \quad (Q_i>=1) \tag{4.2}$$

$$p_i = 0 \quad (Q_i=0) \tag{4.3}$$

$$Q = \sum_{i=1}^{n} Qi \tag{4.4}$$

Where Ts_i is the setup time of the sub-job using the ith CRP; Tp_i is the process time of the sub-job using the ith CRP; p_i is the total time (operation time) of the sub-job using the ith CRP; Q_i is the number of components of the sub-job allocated for the ith CRP; Q is the total number of components of a job.

The aim is to seek an optimum solution of Q_i to minimise $max(p_i)$.

Figure 4.24 shows $p_i = Ts_i + Tp_iQ_i$ is a line with an increasing slope Tp_i, and p_i increases with increasing Q_i. The following are three equations of p:

$p_1=250+5Q_1$

$p_2=20+40Q_2$

$p_3=80+17Q_3$

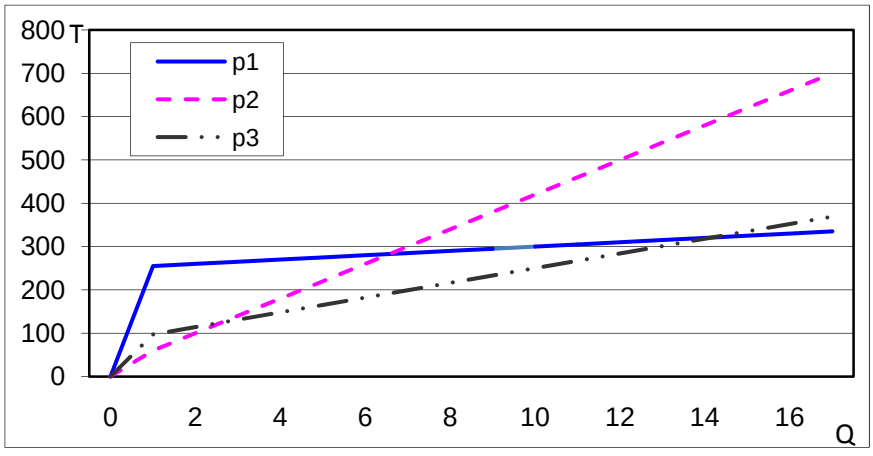


Figure 4.24 Diagram of the Linear Equation

For example, If Q=31, the optimum solution is shown in Figure 4.25.

First set: $Q_1=11$ $p_1=305$

Second Set: $Q_2=7$ $p_2=300$

Third Set: $Q_3=13$ $p_3=301$

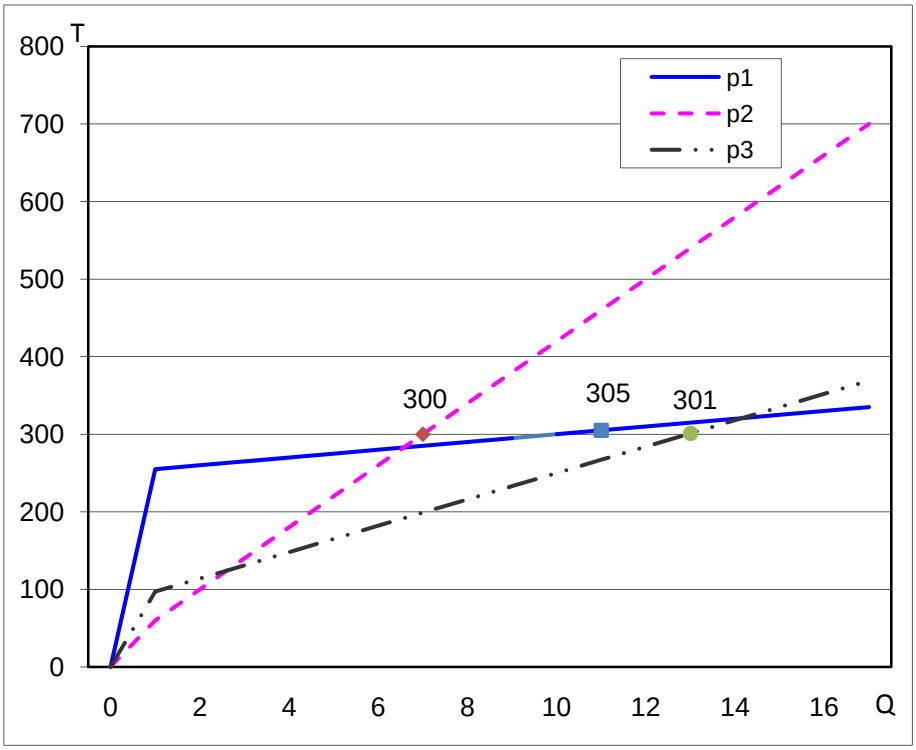


Figure 4.25 The Optimum Solution of the Linear Equation

This solution ($Q_1=11$, $Q_2=7$, $Q_3=13$) is the best solution on the scenario with max(p_i) is 305. If there are any other solutions which are better than this solution, their max(p_i) should be less than 305. If a solution is assumed to be better than this solution, its max(p_i) is less than 305 and p_1 should be decreased. This leads to Q_1 should be decreased because of a linear increasing characteristic of Equation 4.2. If Q_1 is assumed to be decreased by 1 and Q_2 or Q_3 should be increased by 1 subject to the condition of Q=31. If $Q_2=8$, then $p_2=340$. If $Q_3=14$, then $p_3=318$. Both p_2 and p_3 is more than 305, therefore the assumption is wrong and the solution ($Q_1=11$, $Q_2=7$, $Q_3=13$) is the best solution on the scenario.

4.5.2 Proposed Dispatching Optimisation Algorithm

The proposed dispatching search is a heuristic approach to seek a 'good enough' solution with the purpose of reducing the seeking procedures for saving the search time in feasible solution sets. In a scheduling problem, search space is only subject to three equations and it means all values which satisfy these equation constraints are possible solutions to this problem. Neighbourhood structure gradually reduces the boundary of the search space by local optimised transformations. After every transformation, the neighbouring space is a subset of current space and finally the current solution achieves a 'good enough' solution. The aim is to search a 'good' solution to minimise the maximum value of the complete time (operation time).

The procedures of the proposed optimised algorithm are listed as follows:

This proposed algorithm is divided into three stages: pre-dispatched process, optimised process, terminated criteria.

At the initial stage, a search space is the solution with an integer number Q_i. After the pre-dispatched process, two parameters p_{max} and Q_{minj} are calculated in order to narrow the search space. This means the best solution is located at the field with $p_i <= p_{max}$ and $Q_j >= Q_{minj}$ and the other search fields are pruned. The argument outlined in this section discusses the reasons. A variable XP, which is obtained from a series of calculations and comparisons, is used to change relevant Q_i, and then p_i is recalculated. If the termination criteria are satisfied, this optimisation procedure would be terminated and the 'good enough' solution is given. The following describes the dispatching algorithm in detail:

First stage: Pre-dispatched Process

The basic method is applied in the first stage and its procedures are as follows:

If there are n CRPs, components would be allocated into each CRP with a quantity of Q/n. (if the remainder of Q/n is not zero, the remainder is dispatched into CRPs order

by Tp one by one)

Condition 1:

If Q/n>=1, then go to an optimised process.

Condition 2:

If Q/n<1, allocating one component into the first Q CRPs, then go to an optimised process. After an optimised process, there are several situations as follows:

If x of Q CRPs have been thrown away, current x CRPs are pruned and the next x CRPs are added into the CRP pool. Then go back to a pre-dispatched process and then taking a new optimised process.

If no (of Q) CRP has been thrown away, the CRP_x with maximum operation time ($\max(p_i)$) would be replaced with the next new CRP for a comparison of whether getting the better solution. The optimum quantity of the CRP_x with maximum time is put into Equation 4.2 with Ts_i and Tp_i of the new CRP, the p_i can be calculated. If p_i is more than maximum time, the new CRP is pruned. The next new CRP would go to the above comparison procedure until completing the comparison procedures of all CRPs in UCR. If p_i is less than the maximum time, the CRP with maximum time is pruned. Then the next procedure would go to a pre-dispatched process with this new CRP until finishing by comparing all CRPs.

Second stage: Optimised Process

Step 1 Finding out the maximum p_{max} (in kth set), minimum p_{min} (in jth set) among p_i and their related Q_{max} and Q_{min}, after calculating p_i. Recording p_{max} as the maximum time of the whole CRPs and putting Q_{min} as a minimum quantity of the jth set called Q_{minj} (k, j in n). Setting three variables X1, X2, X3, X4, XS, XP=0.

Step 2 Getting X4 .

$X2= (p_{max} - Ts_j)/ Tp_j$, then $X4 = X2 - Q_j$

Step 3 Getting X3.

$X1= (p_{min} – Ts_k)/ Tp_k$

If X1> Q_{mink}, X3= Q_{max} – X1

else X3= Q_{max} – Q_{mink} .

Step 4 Getting XP

Taking the smaller one between X4 and X3 as XS,

XP= trunc(XS/2) . (The *trunc* function converts a real number to an integer number and truncates any fraction)

If XP<1 Then XP=1

Step 5 Re-calculating

Xk= Xk– XP

Xj = Xj + XP

Then calculating p_k and p_j using Equation 4.2.

Third Stage: Quitting Process (Termination Criteria)

1. (QP1 see Figure 4.27) After the process in Step 2, If X4<1, then try the secondary minimum CRP, ….., until getting X4>1, replacing the minimum set with this current suitable CRP. If not, this is the best solution for the current CRPs.
2. (QP2 see Figure 4.27) The number of iteration (OI)
3. (QP3 see Figure 4.27) The minimum value between p_{max} and p_{min} (MV)

In order to validate the proposed algorithm, several points need to be demonstrated as follows:

Argument:

1. Putting p_{max} as a maximum time, because the aim is to seek the minimised max(p_i). Any smaller value than p_{max} will hopefully be obtained gradually by the proposed algorithm.
2. The CRP with p_{min} which should take no less than Q_{min} components is a feasible or necessary resource for the best solution in the current CRPs. If this CRP cannot take Q_{min} components, p_{min} must be larger than max(p_i) of

the best solution. This means all other p_i should be smaller than p_{min}, but this is an impossible condition. If allowing p_1 to be decreased so it is smaller than p_{min}, Q_1 should be reduced according to Equation 4.2 and this must lead to a rise in the allocated quantity of the x resources set according to Equation 4.4, and an increase in its operation time p_x. This results in p_x becoming much bigger than p_{min}. Consequently, the validation is impossible to achieve and this CRP should take less than Q_{min} components.

3. The jth CRP with p_j in the best solution is not smaller than its p_{minj}. If it is smaller than p_{minj}, then this would lead to an increase in the operation time of another CRP and this is not necessary to minimise max(p_i).

Figure 4.26 and 4.27 show the workflow of all stages of DOA illustrated before.

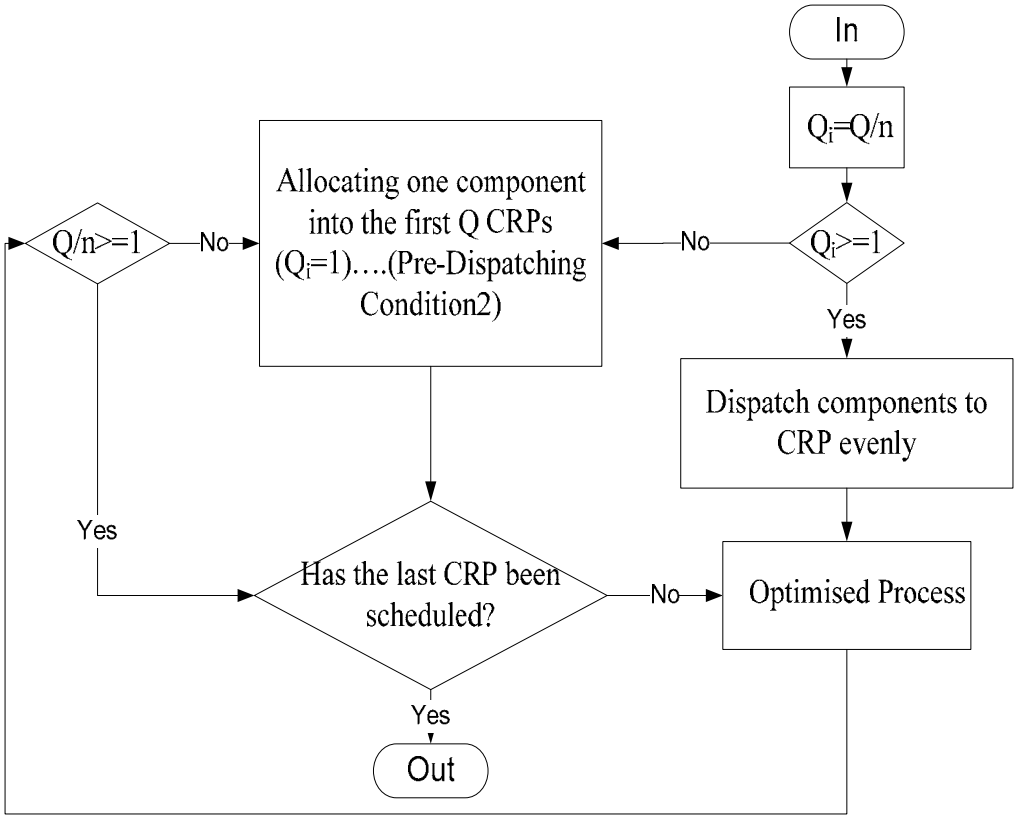


Figure 4.26 Work Flow of DOA (1)

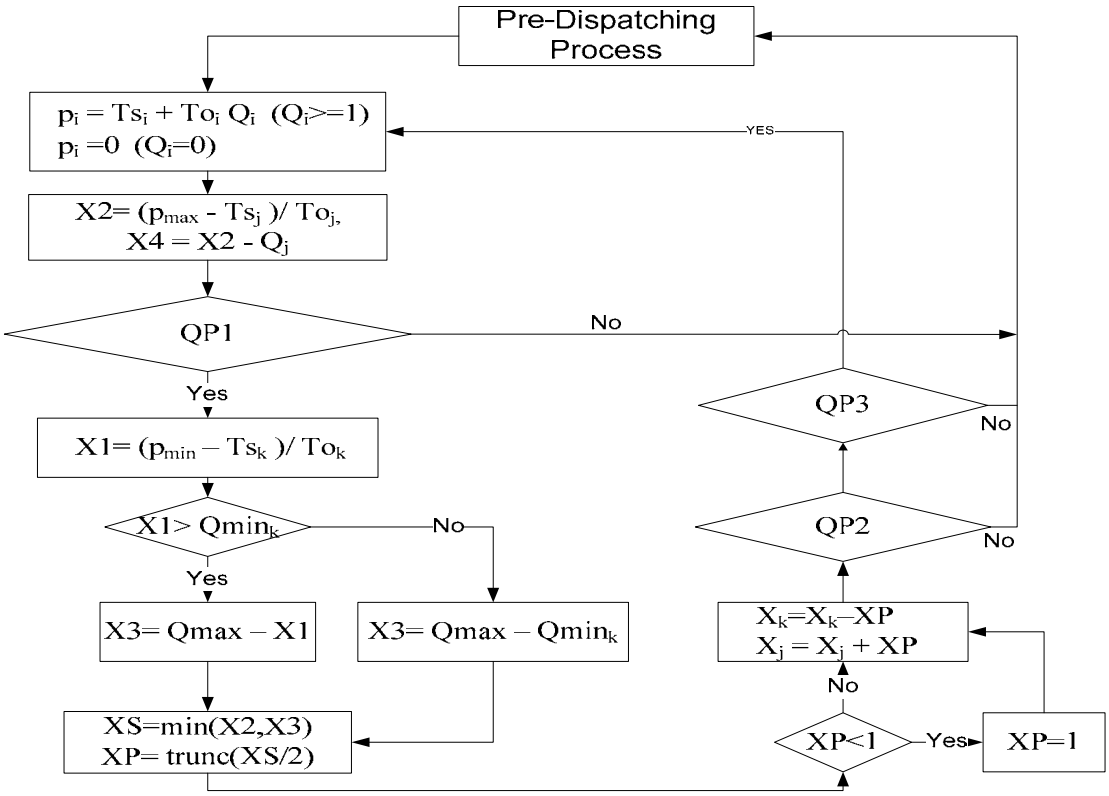


Figure 4.27 Work Flow of DOA (2)

The dispatching algorithm would be optimised in the example in the graphical description to clearly explain the whole optimisation processes.

1) Trun(31/3)=10 Mod(31/3)=1

After a pre-dispatching Stage

Q1=11 p1=305 Q2=10 p2=420 Q3=10 p3=250

2) X2=(420-80)/17=20 ,Q3 =10, X4=20-10=10

3) X1=(250-20)/40=5.75, Qmin2 =0

X1> Qmin2 So X3=10-5.75=4.25

4) XS=4.25 (smaller one between X3 and X4)

XP= trunc(XS/2)=2

5) Recalculating p2, p3,When Q2=10-XP=8, Q3=10+XP=12

p2=340, p3=284, p1=305

6) After calculating X3=0.4 X4=3.29, XS=0.4 XP=1

7) Recalculating p2, p3 When Q2=8-XP=7, Q3=12+XP=13

p2=300, p3=301, p1=305

8) X4=0.125 When taking 2nd resource, X4=0.24 When taking 3st resource X4<1.This is the best solution and the optimised procedures are terminated.

In the UK-based case study, DV (maximum number of splitting a job) is introduced into the DOA algorithm. The DOA algorithm has been modified slightly when n>DV. In the situation of when Q/DV>=1, the first DV CRPs are assigned for the first optimisation. Then the replacing method of Condition 2 in the first stage is adopted to complete the comparisons of all n CRPS. When Q/DV<1, DV this can be ignored and the optimisation will use the original procedures.

4.6 Conclusions

An architecture of an optimisation block exchanging data with a cell simulator is proposed and this block consists of two main components: a particular optimisation module and a particular performance analysis. To different optimisation problems, the related module subprograms and information of the two components can be taken from the library for constructing a particular optimisation block to solve a particular problem. To a JSSP of the UK-based case study, a two-phase method algorithm-CKPO integrated with a DOA is proposed based on the characteristics of this problem. The related concepts and general flows of a CKPO algorithm are introduced and its performance is evaluated to the shifting bottleneck heuristic. Then, for Jm|$ST_{id,b}$|Cmax with LS, disjunctive graph representation is improved for denoting an instance of this type of JSSP and a suitable and effective rescheduling process using the proposed OOP method is developed for the CKPO. After that, a generic CKPO algorithm for Jm|$ST_{id,b}$| $\sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij}F_{ij}$ with LS is developed for optimising the schedule. Finally, a DOA is proposed for an unrelated parallel machines problem with LS to minimise max(p_i) by splitting a job into sub-jobs and assigning them into suitable CRPs for supporting CKPO.

Chapter 5 Cell Simulator Design and Verification Program Coding

5.1 Introduction

This chapter introduces the entire processes in the design of a cell simulator based on a proposed generic structure of a cell simulator outlined in Chapter 3. Firstly, two identified events mechanisms are discussed. Secondly, a generic manufacture with an agile job shop is taken as a case study to develop a conceptual simulation model in the cell simulator. Input, output, and database of the concept model are developed in this chapter. Thirdly, the coding of a verification program is explored and analysed.

5.2 Simulation Worldview, Main Logical Design and Coding

5.2.1 Events Mechanisms

Prior to building a simulation model of a cell simulator, a suitable simulation worldview (which is an approach to ensure the running of the simulation model) should be investigated to determine the control mechanism for this simulation model. There are two types of events outlined in Section 3.2.4. Firstly is the *order coming* event, and secondly is the *common resource released* event. A popular *three phase* approach will be modified to design a new simulation worldview to control the two events outlined. In Section 5.2.2, 5.2.3, and 5.2.4, a description of the mechanism of a simulation worldview is outlined.

5.2.2 Cycle of Event

Figure 5.1 shows a simple cycle of relative events of a cell simulator. There are

two types of events defined in the simulator described in Section 3.2.4 and they are the key items to drive the cell simulator. This section illustrates the entire cycle of two events and it begins with the Timer module. The Timer checks the time of the next or current (when first time running) order and sets a time interval variable for it - Vrealcretime. This means it can be executed automatically after Vrealcretime seconds.

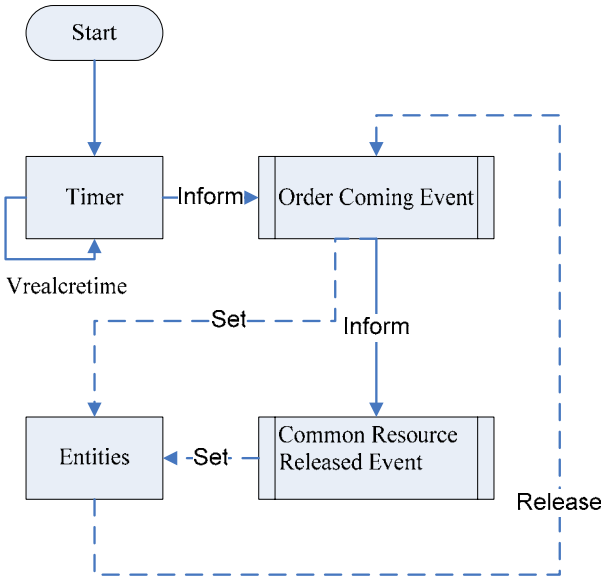


Figure 5.1 Cycle of Event

In WITNESS, a 'part' entity is able to run it by itself according to its schedule. Therefore, a 'part' represents the Timer in the cell simulator and its attribute-*Inter Arrival Time* is set to Vrealcretime. When the simulation time reaches the time of the next order, the Timer informs the *order coming* event control mechanism which deals with the *order coming* event. This mechanism drives the entities, such as machines and labourers, by setting their attributes. It then informs the *common resource released* event control mechanism to check all available resources regarding actions taken and this mechanism sets the relative attributes for the entities. When attributes relating to a *common resource released* event are changed, such as changing the status of machines and labourers from busy to idle, the relative entities give the *control handle* to the *order coming* event mechanism. These processes make a cycle of the events dealt with.

Order coming events are usually infrequently based on the UK case study regarding the number of orders which was approximately 20 per day. The previous *order coming* event mechanism only uses the function Farrangeforcomingorder, which takes the relative data from the database. The developed local datasets (Vrealcomingorderinf and Vstrcomingorderinf) allow temporal storage of information relating to orders and help to reduce 'thrashing' of the database. Consequently, the *order coming* event mechanism always checks the relative local datasets firstly and then checks the database. Based on this concept, the Timer is divided into two: Timer01 and Timer02 as shown in Figure 5.2.

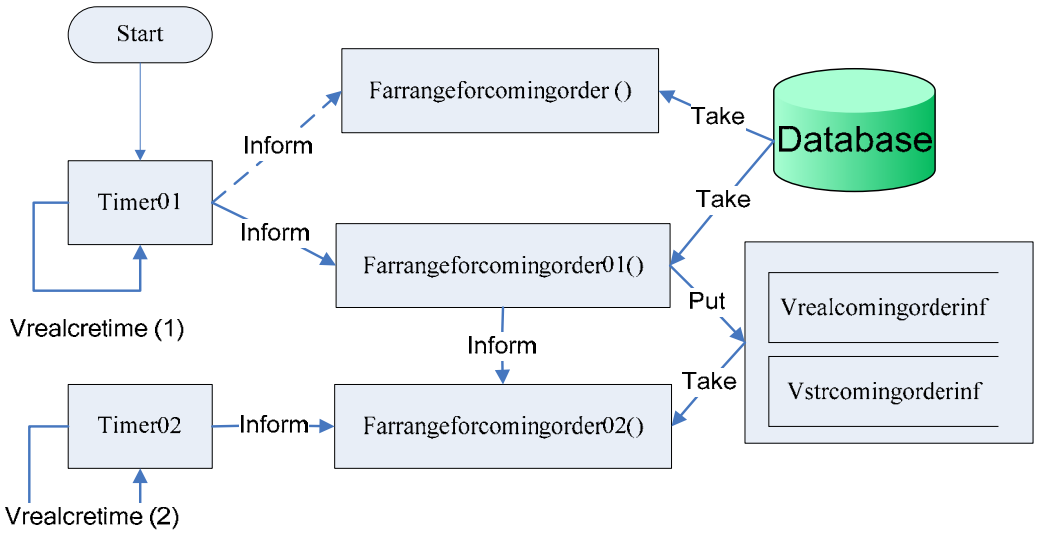


Figure 5. 2 Modified Mechanism of Ordering Coming Event

Figure 5.2 introduces the mechanism of how to 'take' and 'put' the local datasets relating to the *order coming* event. It starts from Timer01, which has similar functions to Timer in Figure 5.1, and adopts the relative function at the time of the next incoming order. However, it should choose one of two functions according to the identified situation. The first function - Farrangeforcomingorder - is to take the relative data of available incoming orders with the arrival time less than or equal to the current simulation time directly from the database. The second one -

Farrangeforcomingorder01 - just takes data of available incoming orders with the arrival time less than or equal to the current simulation time plus a constant such as 24 hours from the database and puts it into the local datasets consisting of two arrays: Vrealcomingorderinf and Vstrcomingorderinf. This function enables the relative data to be taken from the database 24 hours in advance. Another function - Farrangeforcomingorder02 - is a similar function to Farrangeforcomingorder but the difference is that it takes data from the local datasets. The further details of Farrangeforcomingorder02 and Farrangeforcomingorder will be described in the Section 5.2.3 regarding *order coming* event. Normally, Timer01 tries to adopt Farrangeforcomingorder01 and lets it fill data into the local dataset. Then Farrangeforcomingorder01 informs Farrangeforcomingorder02 to take the data. The Vrealcretime(1) is set as the value of the time of the last order in the local dataset minus the current simulation time. Sometimes, the data taken from the database is more than the size of the local dataset, so the Farrangeforcomingorder is adopted and repeated to deal with this data. Vrealcretime(1) is set as the value of the time of the next order in the dataset minus the current simulation time. Timer02 is a timer for adopting Farrangeforcomingorder02 and it can find the time of the next order in the local dataset (minus the current simulation time) for setting Vrealcretime(2).

5.2.3 Order Coming Event

Figure 5.3 illustrates the mechanism to deal with an *order coming* event. Farrangeforcomingorder or Farrangeforcomingorder02 in the *Main Core* takes the relative data and puts it into the two arrays, Vstrarrangeinf and Vrealarrangeinf in the *Decision Dataset Layer*, and informs Rollocatedata to optimise. The optimum results are put into the arrays of Vrealdecisioninf and Vstrdecisioninf.

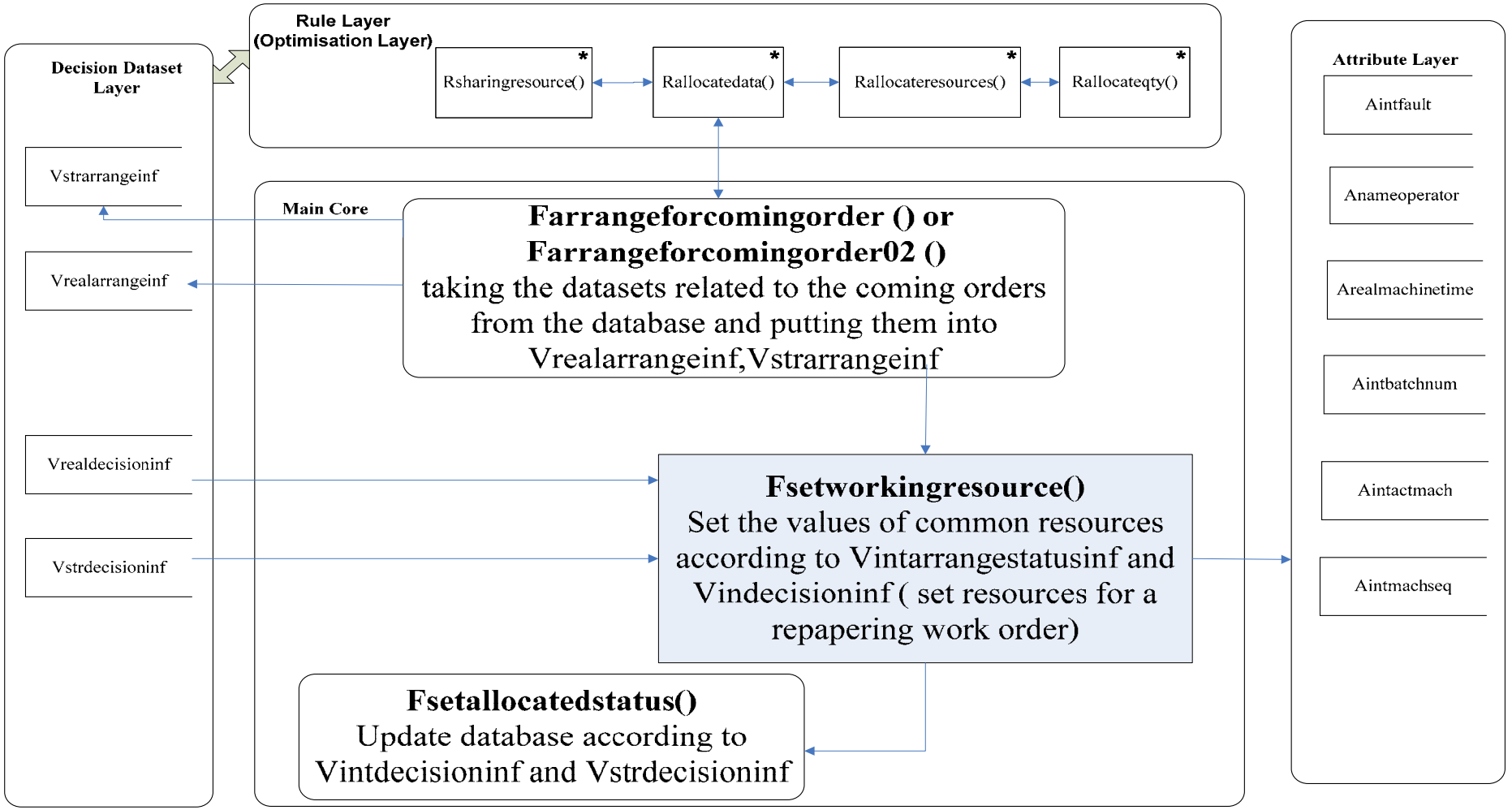


Figure 5.3 Mechanism of Ordering Coming Event

The Fetworkingresource gets an instruction when optimisation completing and sets the attributes according to these relative arrays to drive the entities. Then Fsetallocatedstatus updates the data in the relative table of the simulation database before running the entities. The meaning of relative key attributes of the related machines is outlined as follows:

Aintfault: the number of the faulty parts of the machine in this process

Anameoperator: the relative name of the labourer entities relating to the machine in this process

Arealmachinetime: the relative time of the machine in this process

Aintbatchnum: the number of the operating parts (products) of the machine in this process

Aintactmach: the type of the current process of the machine (first process is 1, the other process is 2)

Aintmachseq: the sequence of the current process of the machine

5.2.4 Common Resource Released Event

Figure 5.4 shows the mechanism to deal with a *common resource released event*. This is similar to the mechanism of an *order coming* event outlined in Section 5.2.3. However, it is more complex because the parts are not always from outside and are taken from the buffer of the previous process.

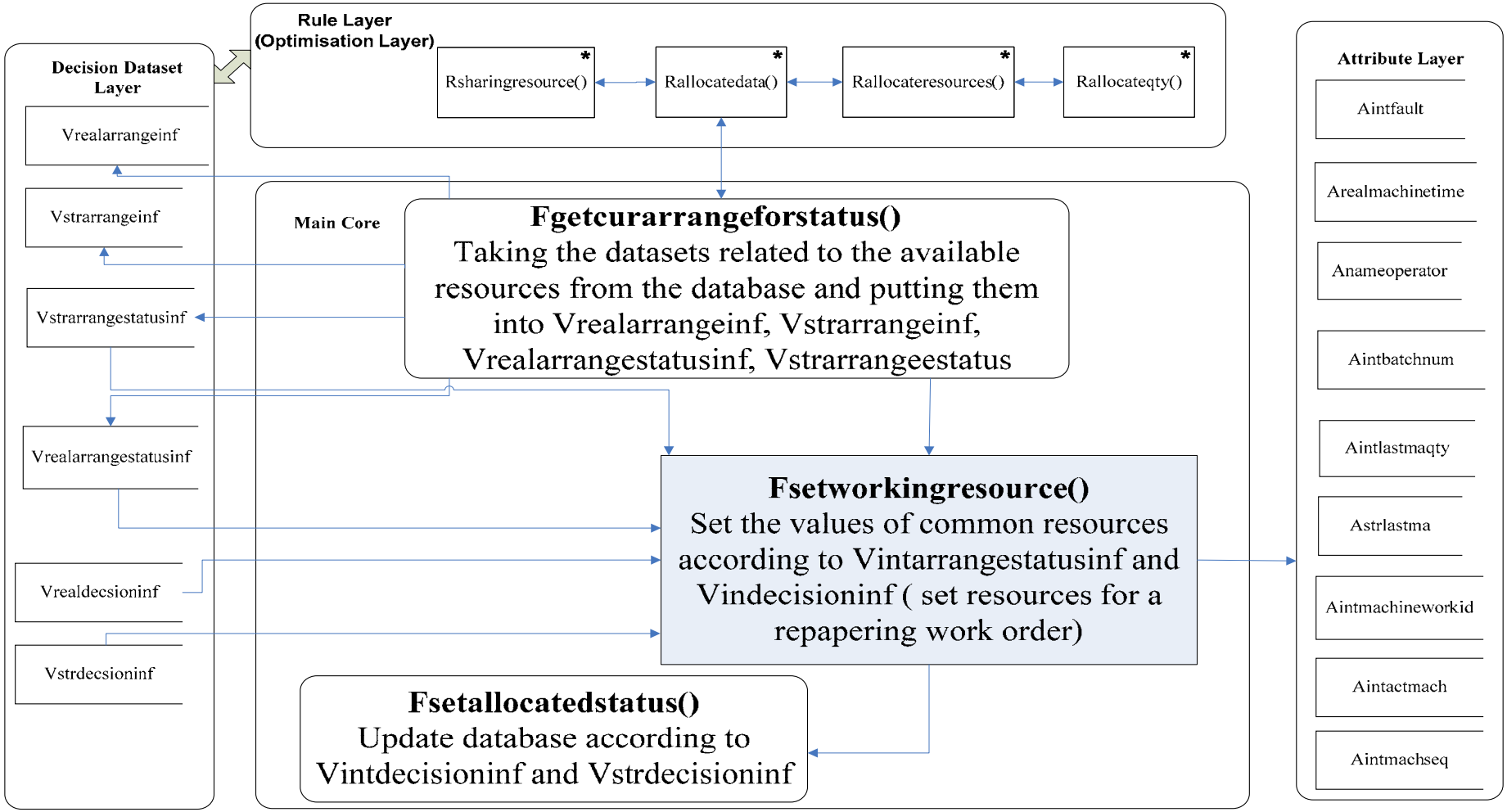


Figure 5.4 Mechanism of Common Resources Released Event

This requires that the status of the parts (products) known to be stored into the arrays: Vstrarrangestatusinf and Vrealarrangestatusinf. Fgetcurrangeforstatus takes the data and puts it into four arrays according to certain rules. Optimum results using two arrays: Vrealdecisioninf and Vstrdecisioninf. However, Fsetworkingresource sets more attributes according to information from Vrealdecisioninf, Vstrdecisioninf, Vstrarrangestatusinf, and Vrealarrangestatusinf. The definitions of the additional attributes of the related machines as shown in Figure 5.4 are given as follows:

Aintlastmaqty: the number of the parts (products) taken from the buffer of the last machines in this process

Astrlastma: the name of the last machines in this process

Aintmachineworkid: the part number of the parts (products) needed to be processed in this process

5.3 Simulation Model for an SME Manufacturing Company

This case study is a manufacturing SME with an agile job shop and processes for a variety of products for different customers. A simulation model is developed for this job shop to obtain results and performance under different scenarios to generate an initial solution for further processes such as optimisation outlined in Chapter 4.

5.3.1.Conceptual Model for a Simulation Model of a Shop Floor Control

5.3.1.1 Problem Situation

Most small and medium manufacturers make products according to various requirements of different customers. This requires a flexible simulation system to satisfy the requirements of many thousands of products for hundreds of customers. Consequently, there could be a large number of routes because every type of product needs its own process route. This situation indicates that routes should be flexible and routes are not directly written into the simulation model.

Most orders consist of small batches ranging from several to hundreds of items. Consequently, SMEs require a flexible job shop approach because of the varying products and this is difficult to achieve by a flow shop approach.

5.3.1.2 Modelling Objectives

A simulation model was developed to simulate a job shop, in order to optimise the process flows, reduce waiting time of resources (machine and labour), balance the line and provide sufficient information and results for decision makers for different scenarios.

5.3.2 Inputs and Outputs of a Shop Floor Control

5.3.2.1 Inputs

There are five types of input data: arrival order data, labour data, machine data, route data, and data related to CRPs.

Information about arrival orders provides the data related to orders, such as order number, quantity, arrival time, due time, etc. Quantity and arrival time are key items and the simulation model cannot be run without them. All information about arrival orders is stored in Table Work_Order_Tab.

Information concerning employee involves the basic relative data of persons, such as employee ID, last name, first name, hire date, group, status, ‘used’, etc. Only employee ID and status are necessary for running the simulation model. Other information is used for further statistical analysis and stored in Table Employee_Tab.

Information about machines introduces the basic data of machines, such as machine name, machine type, status and ‘used’. The necessary data are machine name and status and Table Ma_tab includes this type of data.

Information about routes shows all relative data of routes which define and choose

which route the products or orders should take, such as route no, work centre, sequence, ‘used’, etc. This type of information is complex and the structure is in three layers based on several items, such as route, work centre, machine, ‘used’, etc., This information is stored in several tables, such as Route_Tab, Route_WC,WC_Tab, WC_Ma, etc.

Information relating to CRPs mainly focuses on the data of time and cost between machines and labourers. This data represents time and costs which employees spend on this process (machine) when taking a certain route. The relationship information among labourers, machines, routes is also important and displayed in Table Time_Cost_Re.

5.3.2.2 Output

Output data includes two types of data: real-time data and statistical data. Real-time data is classified as real-time support data to record relative data about products to support further processes and real-time statistical data.

Real-time support data includes machine, labour, quantity, different time and status etc and a work-recorded form is used to provide information on time, personnel, doing or done, and processes for different products. This information is stored in Table Work_Order_Status. Real-time statistical data provides several statistical results and analysis from real-time data. In this case, only cost is to be considered and real-time set-up cost, operation cost, and failing cost are stored in Table Status_Cost.

Statistical data shows performance and provides diagrams, analysis, and reports to decision makers and this data is usually calculated after a completed simulation to satisfy different requirements for decision makers. Database tables with names that begin with ‘sum’ store this data and give the different resource utilisations to clients.

5.3.3 Model Content of a Shop Floor Control

5.3.3.1 Experimental Factors and Responses

The link between the experimental factors and responses is provided by the scope of the model. The experimental factors and responses in a shop floor control are as follows:

Experimental Factors:

Arrival Orders (information of orders every day)

Machines (machine's ability and status)

Labourers (employee's skill and status)

Routes (process flows of products)

Responses:

Completed time of orders

Delay time of orders

Resources utilisations

Waiting time for certain process (bottle neck)

Working efficiency of labourers

5.3.3.2 Model Scope

Table 5.1 shows the proposed scope of the SME's shop floor simulation model. Office, delivery, and warehouse are excluded from the model.

Table 5.1 Model Scope of the SME's Simulation Model

Component	Include/exclude	Justification
Machines	Include	Process the products
Labour-Shop Floor	Include	Experimental factor, required for simulation running
Office	Exclude	Office works affects production less
Delivery	Exclude	Delivery Schedules affect production less
Warehouse	Exclude	Not related to the objectives

Table 5.2 provides information for the level of detail of the SME's shop floor simulation model. The details of four components, namely machine, labour, buffer, and part (product) are discussed as follows:

Table 5.2 Model Level of Detail

Component	Detail	Include/exclude	Comment
Machine	Operation Time	Include	Experimental factor
	Set up Time	Include	Experimental factor
	Broken Down	Exclude	Rarely happens and no data support
	Repairing Time	Exclude	Rarely happens and no data support
Labour	Service Time	Include	Experimental factor
	Delivery Time	Exclude	Small proportion compared to Service Time
Buffer	Capacity	Include	Pre-design for stocking space
	Behaviour (take or put)	Exclude	Not effective
Part	Material delay	Exclude	Rarely happen and no data support
	Attributes	Include	Involved part number

5.3.3.3 Activity Cycle Diagram

Figure 5.5 illustrates an activity cycle of a product to explain a route concept. The control mechanism can check in real-time and inform the related machines to take part (product) entities from the Generate Part module at the relevant order's arrival time. In a typical route, Machine1 is the first process and it pulls the given number of parts from

the Generate Part module. Employee1 goes to machine1 to do the work arranged by the control mechanism. Operation time (process time and set up time) of machine1 to deal with this process for these parts is set by the control mechanism. After this process, these parts are pushed into the buffer1 to wait for the next process. The control mechanism checks whether the resources (which are suitable for the next process) are available by a *common resources released* event. Machine2 and machine3 with the suitable people - employee1 and employee2 - are available and prepare for the next process. Thus, machine2 and machine 3 take the given number of parts from buffer1 respectively. After the second process, they push their parts into buffer2 and buffer3 respectively. Now, machine1 is available to do the last process and pull these parts from buffer2 and buffer3, then push to shipment after completing the operation. If any machine makes faulty parts, these are pushed into the rubbish bin. The cleaner can empty the bin in time. The paths of a part from the Generate Part module via machine1 and buffer1 to machine2, buffer2 or machine3, buffer3, finally reaching machine1 to shipment are called the routes of the part.

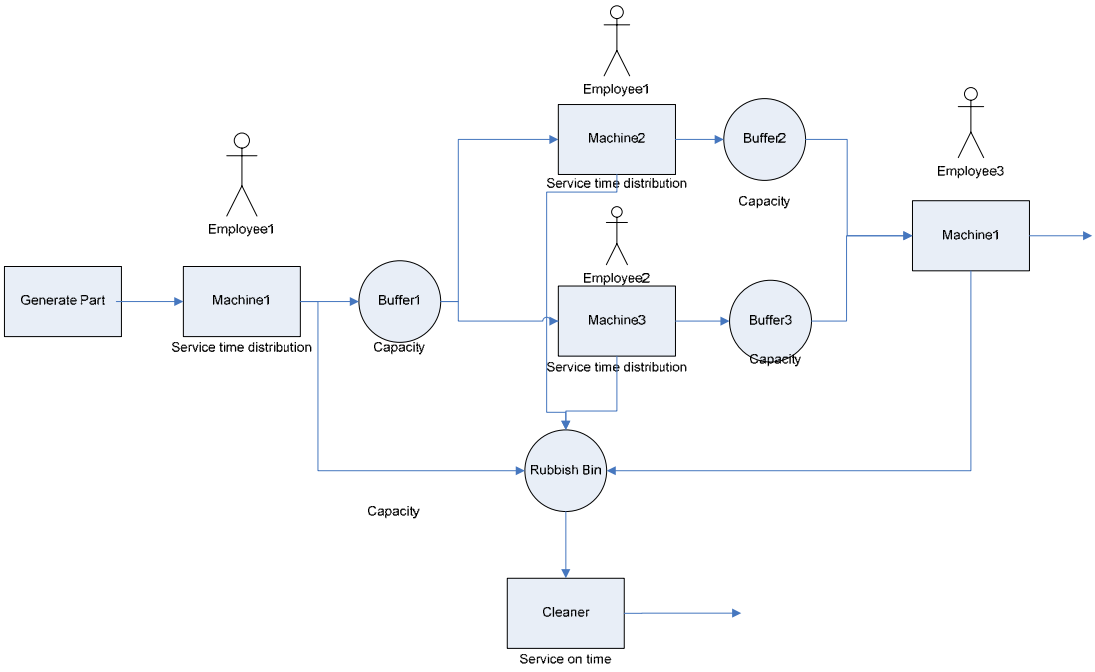


Figure 5.5 Activity Cycle Diagram

5.3.3.4 Model Simplification

In relation to machines, breakdown and repairing times have been not considered because they rarely happen, and there is no related data or information from the SME. However, the cell simulator allows using distributions to mimic breakdown situation to be in cooperated into the model.

With regard to employees, they are normally a stable team and no one leaving over several years. Normally, they choose the idle time to take the holidays thus these situations are not the main issue considered for the simulation. These two issues are considered for future work.

5.4 Verification Program

5.4.1 Theoretical Foundation

There are two types of verification methods for the optimum solutions, outlined in Chapter 4. One is a stochastic simulation model to validate a solution, and the other is to adopt a verification program to verify feasibility of the solution when designing and developing a new optimisation algorithm. Usually, the processes involving determining the distributions for a stochastic model require numerous amounts of real data to support them. Due to the lack of sufficient data or saving verification time, software was developed to achieve verification. In the UK-based case study, a verification program is adopted to check the feasibility of the optimum solutions.

According to the mathematic constraints of JSSP, two types of constraints are used to be verified for a solution. The first type is CRPs' constraints based on the concept which every operation node occupies at least one CRP and a CRP cannot be occupied by more than one operation node at the same time. In the UK-based case study, a CRP consists of a machine and labour and the constraints of CRPs are divided into two sections: machines and labour. In Chapter 2, Equation 2.2 expresses that every CR in

this case study is able to handle only one operation at a time. Therefore, using Equation 2.2 with the generic logics, the constraints of CRPs are identified as follows: (The notations of variables are described in Section 5.4.2)

- Tb<Te (steps 4 and 11 in Section 5.4.2 signify that the end time of an operation which occupies a CR is bigger than the begin time of the same operation which occupies the same CR)
- Tel<=Tb (steps 4 and 11 in Section 5.4.2 refer to Equation 2.2 and signify that the begin time of an operation which occupies a CR is no less than the end time of the predecessor one on the CR task (disjunctive constraint))

The second type is constraints of the operation nodes. The main verification equation is the Equation 2.1 in Chapter 2 which means every operation which adopts CRPs owns a certain operation time. The verification from an operation node is described as follows:

- S_l=1 (step 34 in Section 5.4.2 means the start sequence of the route of every job is 1)
- Sl=Sc-1 (step 26 in Section 5.4.2 expresses the sequence of the route is a natural integer which is a route constraint)
- CRP ∈SCR (steps 20, 29, and 40 in Section 5.4.2 represent the CRP occupied by an operation is suitable and this is the machine eligibility constraint and workforce constraint)
- Tel<=Tb (step 26 in Section 5.4.2 based on Equation 2.1 expresses the begin time of an operation is bigger than the end time of the predecessor one on the job task (conjunctive constraint)).
- Intminusqty=0 (means ΣQ_i=Q) (step 26 in Section 5.4.2 based on Equation 4.4 signifies the sum of quantities of products in sub-jobs at an operation

node equals to the total quantity of products in the job and there is a constraint of LS)

- Tb>=Ta (step 37 in Section 5.4.2 expresses the begin time of the first operation for a job is bigger than the predefined arrival time of the job)
- |Intmax|<=2 (step 26 in Section 5.4.2) There may exist a difference between the values calculated according to Equation 4.2 in different program environments, given a 2 seconds difference which is the result of different integer methods dealing with fractions by various program languages of C# and WITNESS.

5.4.2 Work Flow of Verification Program

The notations of the verification program are described as follows:

Ta: the arrival time of a job

Te: the end time of the current operation

Tb: the begin time of the current operation

Sij: the sequence number of operation ij on a job j task (starting from 1)

Ma_ID: machine ID

Em_ID: labour ID

Tel: the end time of the predecessor operation on the CR task from the beginning to step 11. From step 12 to the end it expresses end time of the predecessor operation on the job task.

Qc: the quantity of the products assigned to the current CRP

Sc: the sequence number in the route of the current operation

Inttotalqty: the total quantity of the current job

SCR: suitable common resource pairs of the current operation

CRP: common resource pair

Intqty: a variable of quantity

Sl: the sequence number in the route of the predecessor operation on the job

task

Intminusqty: the minus value concerning quantity

To: the operation time of the CRPs assigned for the current operation

Intmax: the maximum minus value between the operation time of the current operation existed in the current solution and To (To is calculated according to Equation 4.2 in C# program environment)

Figures 5.6 show the work flow chart of the verification program related to the constraints of CRPs and the following describes this verification processes.

1. To initialise all variables at the initial Stage
2. To get next Ma_ID (Machine ID) from Table Ma_Tab and set Tel=0
3. To take Te and Tb of the next operation which occupied the Ma_ID from relevant tables sorting by Tb
4. To check two verification equations: one is Te>Tb, the other is Tel<=Tb. If failing any one or both of the two equations, go to step 5, If Yes, go to step 6
5. To print a 'fail' information of Ma_ID
6. To set Tel=Te
7. To check whether all operations which occupy the Ma_ID are processed or not? If No, go back to step 3. If Yes, go to step 8.
8. To check whether all Ma_ID are processed or not? If No, go back to step 2. If Yes, go to step 9.
9. To get next Em_ID (Labour ID) from Table Employee_Tab, and set Tel=0
10. To take Te and Tb of the next operation occupied the Em_ID from relevant tables sorting by Tb
11. To check two verification equations, one is Te>Tb, the other is Tel<=Tb. If failing any one or both of the two equations, go to step 12, If Yes, go to step 13

12. To print a ‘fail’ information of Em_ID
13. To set Tel=Te
14. To check whether all operations which occupy the Em_ID are processed or not? If No, go back to step 10, If Yes, go to step 15
15. To check whether all Em_ID are processed? If No, go back to step 9, If Yes, go to the module of checking the constraints of operation nodes (start from step 16).

Figures 5.7 show the work flow chart of the verification program related to the constraints of operation nodes and the following describes this verification processes.

16. To get several variables, such as Te, Tb, Qc, Sc, CRP of the next record from the ‘query’ concerning two database tables OP_CurRe and OP_Main, (see in Section 6.3.8) of the solution sorting by job number (Work_ID), Sc and CRP
17. To check whether all operation nodes are processed? If Yes, go to step 44, If No, go to step 18.
18. To check whether the current job number is the same with the last one or not? If Yes, go to step 19. If No, go to step 34.
19. To check whether the current operation number is the same with the last one or not? If Yes, go to step 20. If No, go to step 26.
20. To check whether the CRP belongs to the SCR of the current operation? If Yes, go to step 22. If No, go to step 21.
21. To print a ‘fail’ information of SCR, the job number, and the operation number
22. To get the operation time To of the current sub-job which occupies the CRP according to Qc by Equation 4.2 and set Tel=max(Tel,Te).
23. To set Intminusqty=intminuqty-Qc and Intminustime=|Te-Tb-To|

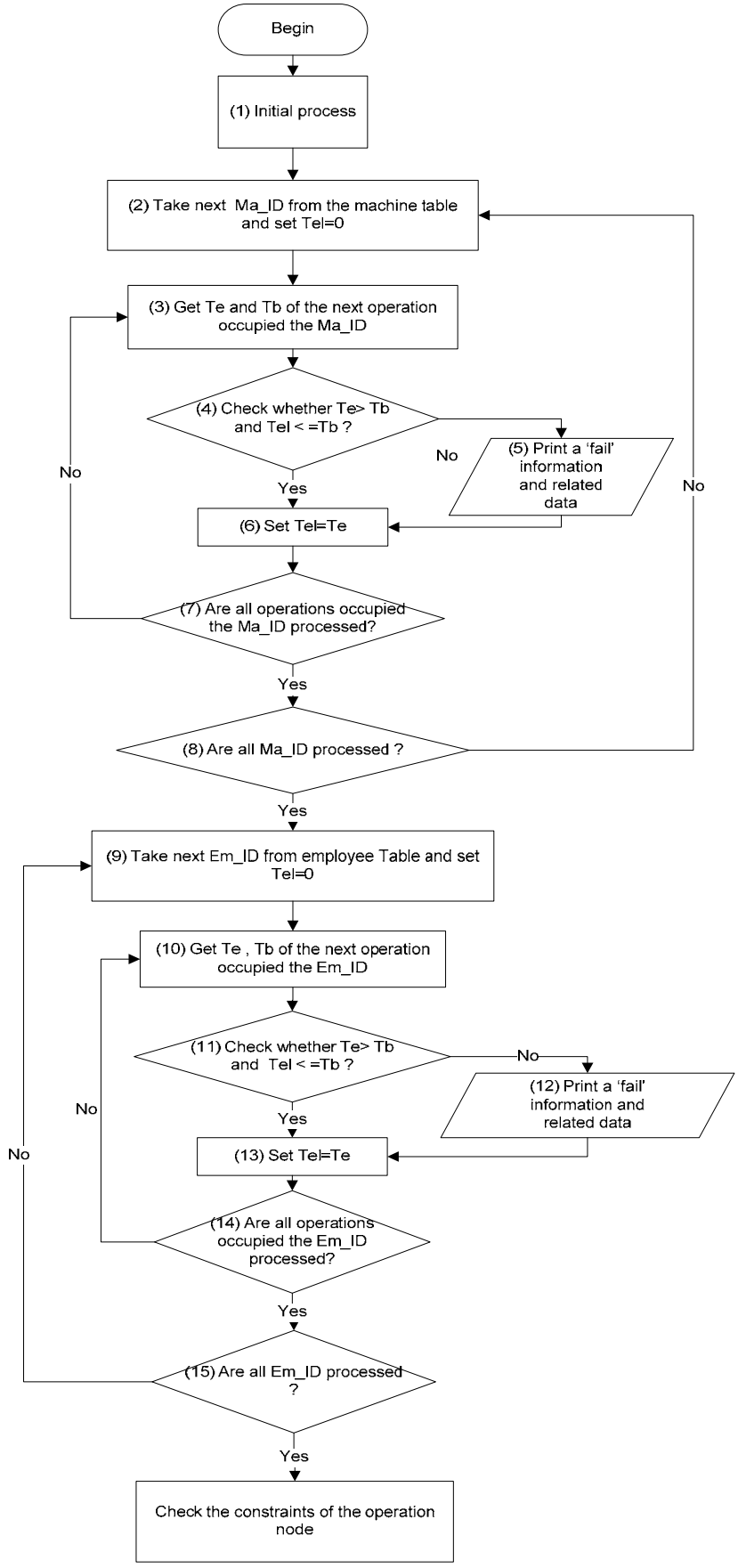


Figure 5.6 Workflow of Verification related to the Constraints of CRPs

24. To check whether Intmax<intminustime? If Yes, go to step 25. If No, go to step 16.
25. To set Intmax=intminustime
26. To check four verification equations. The first one is Intminusqty=0 and the second one is Intmax<=2. The third one is Tb>=Tel and the fourth one is Sl=Sc-1. If any one or all of the equations fail, go to step 27. If Yes, go to step 28.
27. To print a 'fail' information of the job number and the operation number
28. To set Tel=Te and Sl=Sc
29. To check whether the CRP belongs to the SCR? If Yes, go to step 31. If No, go to step 30
30. To print a 'fail' information of SCR, the job number, and the operation number
31. To get the operation time To of the current sub-job which occupies the CRP according to Qc by Equation 4.2
32. To set Intminusqty=inttotalqty-Qc
33. To set Intmax=|Te-Tb-To|, then go back to step 16
34. To check whether Sc=1? If Yes, go to step 36. If No, go to step 35
35. To print a 'fail' information of the job number, and the operation number
36. To get intotalqty (total quantity) and Ta (arrival time) of the current job number from the relevant tables
37. To check whether Tb>=Ta? If Yes, go to step 39. If No, go to step 38
38. To print a 'fail' information of the job number, and the operation number
39. To set Intminusqty=inttotalqty-Qc and Tel=Te
40. To check whether the CRP belongs to the SCR? If Yes, go to step 42. If No, go to step 41

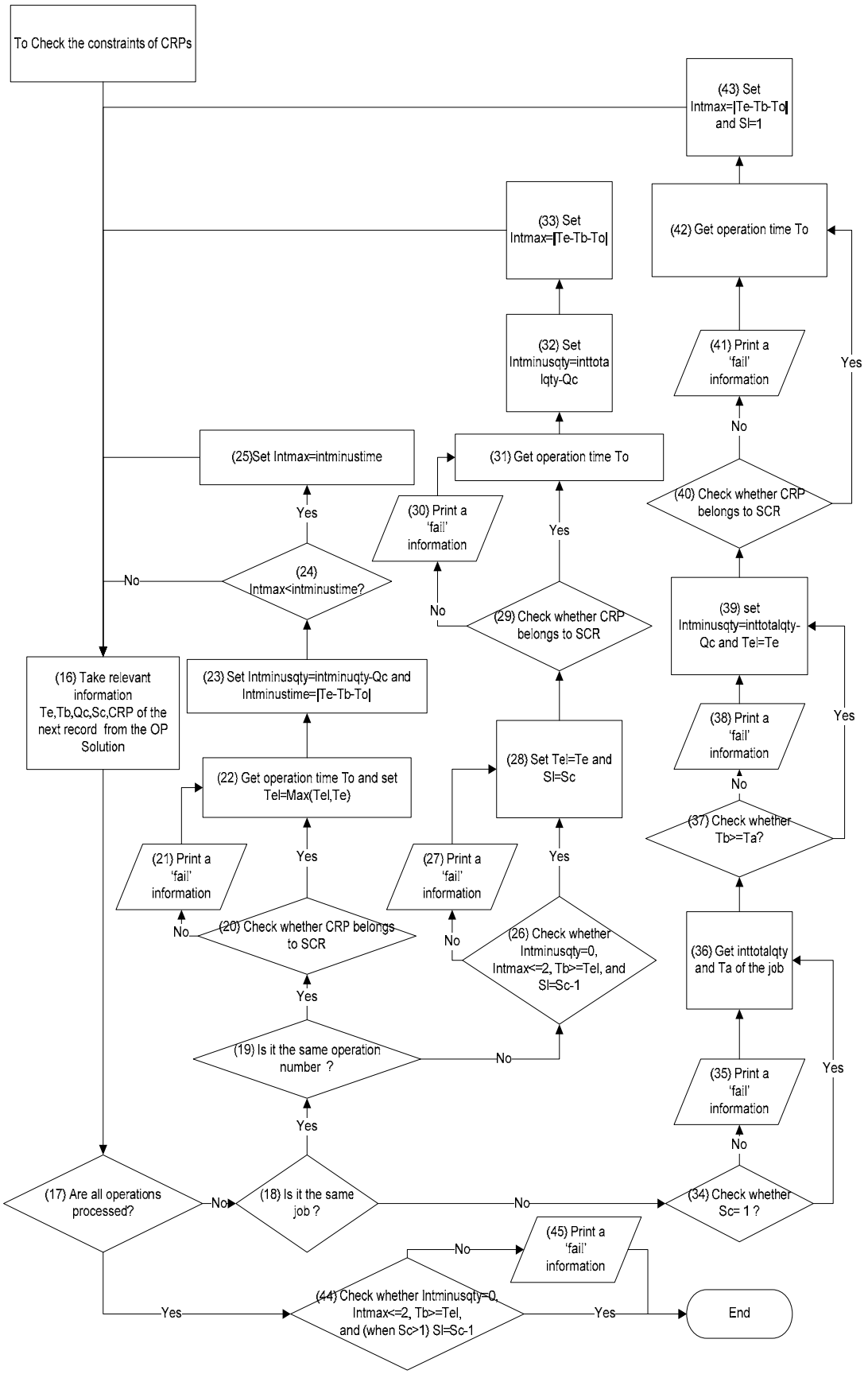


Figure 5.7 Workflow of Verification related to the Constraints of Operation Nodes

41. To print a ‘fail’ information of SCR, the job number, and the operation number
42. To get the operation time To of the current sub-job which occupies the CRP according to Qc by Equation 4.2
43. To set Intmax=|Te-Tb-To| and Sl=1, then go back to step 16
44. To check four verification equations. The first one is Intminusqty=0 and the second one is Intmax<=2. The third one is Tb>=Tel and the fourth one is Sl=Sc-1(when Sc>1). If any one or all of the equations fail, go to step 45. If Yes, go to End.
45. To print a ‘fail’ information of the job number and the operation number

Failure in any of the verification equations indicates that the solution is not feasible.

5.5 Conclusions

This chapter mainly focuses on a simulation model of a cell simulator and a verification program. Two events control mechanisms for *order coming* and *resource released* events are proposed and the cooperation mechanism of the two events is shown in Figure 5.1. These control mechanisms are successfully developed for a cell simulator. The relevant processes of building a simulation model are discussed based on the generic SME manufacturing processes. Inputs, outputs, experimental factors, model scope, active cycle and model simplification for this case study are outlined respectively. Finally, the mathematical constraints, namely conjunctive constraints and disjunctive constraints, and the generic logics of JSSP are outlined in designing a verification program to check the feasibility of the solution and the workflow of the program is discussed.

Chapter 6 Case Study

6.1 Introduction

Two types of simulation blocks are outlined in Chapter 3, namely a stock-driven block and a schedule-driven block. This chapter validates the designs of the adaptive simulation-based decision-making framework by developing two types of DMSs for two case studies. A simulation-based DMS which consists of a stock-driven simulation block was developed for a Make To Stock (MTS) gift SC of SMEs (Musalem and Dekker, 2005) and a key role, namely planner (a replenishment algorithm), is embedded in the cell simulator. The proposed replenishment algorithm is able to place orders according to the stock level volume and the performance of this algorithm is discussed under various scenarios. A simulation-based DMS including a schedule-driven simulation block is proposed for a UK-based Make To Order (MTO) case study and a complete optimisation block composed of CKPO and DOA is embedded into this block. The simple ERP system adopted by the case study is considered as a planner which takes the responsibility of making Master Production Schedule (MPS). A proposed cell simulator conducts scheduling from ERP and runs according to the scheduling. The optimisation block is able to provide 'good enough' solutions based on the initial solutions from the cell simulator and meets the requirements of decision makers. The performance of the optimisation block is evaluated by data obtained over a six month period from the UK SME. Several scenarios are designed to illustrate the capability of the simulation-based DMS of this case study.

6.2 Stock-driven Case Study

6.2.1 Stock-driven Case Study overview

This section introduces a simulation-based DMS which consists of a stock-driven

simulation block for a gift MTS SC. This SC has a Chinese assembly plant and a Netherlands' Distributor Center (DC) delivering goods from China to Holland. The company manufactures 23 products consisting of eight different families of song chips and other components. The chips are considered as the main items in the DMS. However, the numbers of chips needed to be taken into account in calculating the cost for delivery, shipment, and holding costs of the gifts. The *lead time* is 41 days including 14 days in assembling the products, 22 days in shipping from China to the port of Rotterdam and 5 days in delivering to the DC, as shown in Figure 6.1.

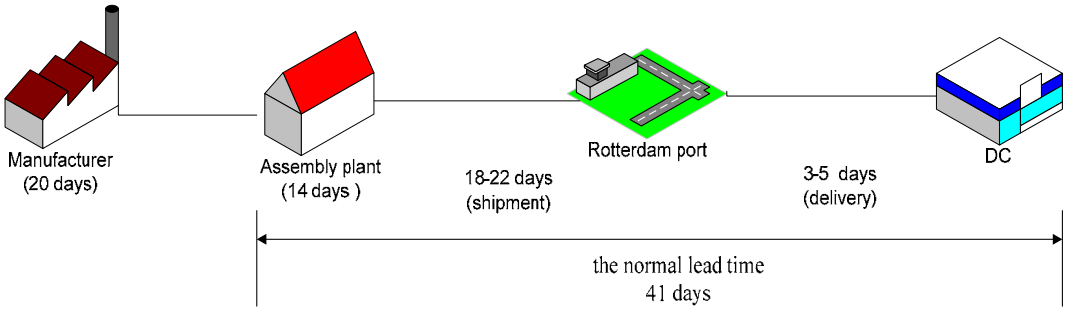


Figure 6.1 Diagrammatic Layout of the SC of the Gift Company

In the assembly plant, there are two prices for chips. If the number of chips of an order is no less than 50,000, the price is $0.375 per item. Otherwise, the price is $0.4 per item. The minimum order request is not less than 10,000 items and this information is outlined in Tables 6.1 to 6.2 in relation to costs (Musalem and Dekker, 2005). There are two types of shipment when delivering goods from China to the port of Rotterdam. They are either full or partial containers, and Table 6.1 shows the two types of costs associated with the containers types. A Free On Board (FOB) method is adopted for the shipment cost and product cost.

Table 6.1 Shipping Costs

Shipment type	Quantity	Cost FOB ($/item)
Full Container (FC)	45,000-52,000	1.25
Partial Container (PC)	10,000-44,999	1.27

The setup cost includes the handling cost at Rotterdam port and the delivery cost from the port to the DC. The setup cost of a full container is almost $700 and Table 6.2 represents the cost of a partial container.

Table 6.2 Setup Costs

Function	X (the total number of items in thousands)	Y
(210+Y)+5.45X	10=<X<15	80
	15=<X<20	100
	20=<X<25	140
	25=<X<30	170
	30=<X<35	200
	35=<X<40	230
	40=<X<45	240

6.2.2 Replenishment Algorithm

This section proposes an integrated optimisation algorithm including the cost function, namely Economic Order Quantity (EOQ) to deal with the issue associated with the multiple products sharing Common Resources (CRs) and this integrated optimisation algorithm is considered as a local optimisation embedded in the Rule layer of a cell simulator. The EOQ algorithm has been adopted to calculate the economic quantities and determine the Reorder Points (ROP). As the stocks descend to the ROP, this causes the placement of orders using several mechanisms to optimise the total costs by adjusting the trade-off between carrying costs and ordering costs for sharing CRs, which is not included in the original EOQ algorithm.

EOQ is the basic algorithm of the inventory replenishment model and it takes into account the carrying and ordering costs. In general, carrying (holding) costs which relate to stocks consist of taxes, interest, insurance, etc. Ordering (setup) costs

include the costs of machine setup, shipment, preparing invoices, checking goods, etc. EOQ's aim is to pursue the optimum order size and order point for minimising the total costs. However, one of the main limits of the EOQ algorithm is that it can only be used in one products situation and consequently this algorithm needs to be modified for the case study (Stevenson, 1998; Chase et al., 1998). EOQ algorithm (Evans, 1992; Chase and Aquilano, 2004; Heizer and Render, 2006)is:

TC = Carrying (holding) cost +Ordering (setup) cost +Purchasing cost (6.1)

According to Equation 6.1, the ordering cost is a variable depending on the order quantity and the selection of the shipment. It is assumed that the goods price and the product demand are relative constant during the year, and carrying cost is generally unchangeable.

$$TC = (\frac{Q}{2})P_{order}H + P_{setup}(\frac{D}{Q}) + P_{order}D \quad (6.2)$$

where TC is the total cost, Q is the economic quantity order, P_{order} is the order price (see Table 6.1 for Cost FOB), H is the holding cost per unit, D is the annual demand, P_{setup} is the setup cost for every order.

As most of the stocks are in the DC, H_{DC} (25% the holding cost in the DC) can almost replace H (total annual holding cost percentage) and H_{plant} (20% the holding cost in the assemble plant for chips) can be ignored for simplification purposes. For each family of chips

$$TC_i = (\frac{Q_i}{2})P_{order}H_{DC} + P_{setup}(\frac{D_i}{Q_i}) + P_{order}D_i \quad (6.3)$$

where TC_i is the total cost of family i, Q_i is the economic quantity of the family i's order, H_{DC} =25% (the holding cost percentage in the distributor center), D_i is the demand of family i (annual).

$$Q_i = \sqrt{\frac{2D_i P_{setup}}{P_{order} H_{DC}}} \tag{6.4}$$

$$\sigma_{Li} = \sqrt{L_i}\,\sigma_{di} \qquad \text{(when Lead Time } L_i \text{ is constant)} \tag{6.5}$$

$$E(z)_i = \frac{Q_i(1-P_i)}{\sigma_{Li}} \tag{6.6}$$

$$\overline{d_i} = \frac{D_i}{365} \tag{6.7}$$

$$R_i = \overline{d_i} L_i + Z_i \sigma_{Li} = \overline{d_i} L_i + Z_i \sqrt{L_i}\,\sigma_{di} \tag{6.8}$$

$$TC = \sum_{i=1}^{8} TC_i \tag{6.9}$$

where σ_{di} is the standard deviation of daily usage of family i, σ_{Li} is the standard deviation of usage of family i during *lead time*, $E(Z)_i$ is the expected number of family i of units short for each order cycle from a normalized table where σ=1, P_i is the service level desired of family i (satisfying a certain percentage of demand from items in stock), $\overline{d_i}$ is the average daily demand of family i, L_i is the lead time in days of family i, Z_i is the number of standard deviations for a specified service level of family i, R_i is the reorder point of family i in units.

According to Equation 6.4, it is assumed the full container is adopted (P_{setup}=\$700), D_8 = 70 units per day*365 day = 25550 units per year, and P_{order} = \$1.25 per unit, and H_{DC} =25%, then Q_8 =10679 units. If taking the partial container for a order with 30000 units , P_{order} = \$1.27, and P_{setup}=\$573.5 , then Q_8=9607 units. Therefore, Q_8 is set to 10000 units (the minimum order size). The same calculating process is used for family 7, $Q_7 \approx 10000$. The Q of other families is less than the minimum order size of 10000, so their Q is set to 10000. The ROP will be calculated according to the different standard deviation for the families respectively. It is assumed that the average daily demand of every family is constant and the sale distribution is a normal distribution with a zero standard deviation. According to Equation 6.8, the reorder points of each family are

shown in Table 6.3 (z=1.28 is to achieve a 90 percent probability of not stocking out, see Appendix C).

Table 6.3 ROP Numbers

Family (Chip) i	1,3	2,5	4,6	7,8
$\overline{d_i}$	48	57	28	70
$ROP_i(\sigma_{di}=0)$	1968	2337	1148	2870
$ROP_i(\sigma_{di}=5,z=1.28)$	2009	2378	1189	2911
$ROP_i(\sigma_{di}=20,z=1.28)$	2132	2501	1312	3034

If more full containers can be used, the cheaper the shipping and setup costs would be . Equations 6.10 to 6.13 have been developed to enable sharing the CRs of multiple products.

Adding holding cost: $C_h = N \times P_{order} \times H_{DC} \times T_a$ (6.10)

Saving setup cost and purchasing cost: $C_d = (P_{pc} - P_{fc})Q + CS_{setup}$ (6.11)

If $C_d > C_h$, the action in advanced is right and it means

$$N < \frac{(P_{pc} - P_{fc})Q + CS_{setup}}{P_{order} H_{DC} T_a} \quad (6.12)$$

where CS_{setup} is the saving setup cost, P_{pc} is the shipment cost per unit of partial container, P_{fc} is the shipment cost per unit of full container, T_a is the advance days of booking a order, N is the increasing quantity of the family above the normal order.

From Equation 6.12, there are two important factors (T_a and N) affecting the decision of advancing booking orders. In the simulation model, T_a and N will be adjusted for the best performance of the DMS. In the system, there exists a mechanism

for ensuring effective assign the replenishing *N* units depending on the priority of the families at the ROP points.

$$Ratio_i = \frac{S_{DCi}}{\overline{d_i}} \tag{6.13}$$

where $Ratio_i$ is the days of the current stocks in DC meeting the daily demand of family i, S_{dc} is the rest stocks in DC, $\overline{d_i}$ is the daily demand of family i.

There is a variable N_{max} (maximum N) to place the orders for the families of chips, which have not yet reach their ROPs in simulation with only a partial container. When O_{order} (the total number of items of the orders when the number of the families (chips) arrived down the ROP) + N_{max} >=45000 (full container), the order of N_{max} chips can be put into the order of related family of chips. This is represented by Equation 6.13 and N items (the family of chips) are allocated based on the days of current stock level in meeting the daily customer demand for the items. However, it is important to meet the condition of time ($Ratio_i$<=T_a) before replenishing N to the order of family (chip) i.

6.2.3 Proposed System and Cell Simulator

A simulation-based DMS is developed for a gift SC which adopts a replenishment algorithm previously outlined. A MTS case study with relatively simple data inputs and fixed relationships of different participants in the SC is evaluated. Consequently, only local datasets in the cell simulator are sufficient to support this DMS because of its simplicity. The proposed replenishment algorithm is not complicated and is able to be coded into the *Rule* layer of the cell simulator. This simulation-based DMS of this case study using a cell simulator of a stock-driven simulation block is able to run on a normal-performance PC with WITNESS software. Consequently, most SMEs can afford this type of system. The structure of the cell simulator of this case study is illustrated in Figure 6.2 and the diagram illustrates the normal processes of delivery and shipment, and shows the important process of

placing the order (planner). It also indicates that the operation time of each process i.e. assembling goods 14 days, shipment 22 days, and delivery 5 days respectively in the entities layer.

In the cell simulator, the buffer entities represent the processes of assembling, delivering, or shipping and the machine entities represent the actions of the places, such as warehouse, DC, and port. The machine can execute the operation of transferring items from the current buffer to the next related one, and set the attributes which are used to count the quantity of the items under different conditions. The important activities are *Ordergenerator* and *Rcalculateorder* which place orders depending on an attribute array *AintstockofDC* (the stock volume of DC) and calculate all costs like holding, shipping, and setup costs. In order to simulate the initial stock level, the number of items needs to be conveyed to the DC depending on the variable array *Ainitialsituation* and sales data refers to an attribute array *Aintnumberofsale*. The *Ordergenerator* sets an attribute array *Aintnumberorderofchip* from relative datasets at a decision dataset layer to drive the entities.

In Figure 6.2, the notations of partial variables and attributes are defined as follows:

VintnumberoflackFC (N_{max}): the maximum increasing quantity of the chip above the normal

Vreallimitofratio (Ta): the advance days of booking a order

Vintchipdemand(i) (O_i): the number of family i's orders during the period

VintROP(i): the reorder point of family i (Table 6.3)

VintEOQ(i): the economic order quantity of family i

Vrealpricebychip: two prices for chips

Vintqtyogorder: the number of items for each family items of an order

Vrealholdingcost(i): the holding costs of assembly plant, shipment, delivery, DC

Vrealshippingcost: the shipping cost (Table 6.1)

Vrealsetupcost: the setup cost

Vrealtotalcost: the total cost

Aintinitialsituation(i) : the initial stock of family i in DC

Aintnumberofsale(i) : the daily demand of family i

Aintnumberonorderofchip(i): the quantity of the order of family i

Aintcheckorderitem(i): the status of order of family i

AintstockofDC(i): the stock of family i in DC

The routes of products are relative fixed and simple, which belong to No Flexibility Configurations outlined in Section 3.3.6. Consequently, these routes are able to be conducted in the *Entities* layer of the cell simulator. The mechansim of c*oming order* event is adopted to deal with two main events. Firstly to place orders according to the stock volume each day. Secondly to take items from DC according to the sales data each day. Figure 6.3 shows the cell simulator of this case study coded by WITNESS.

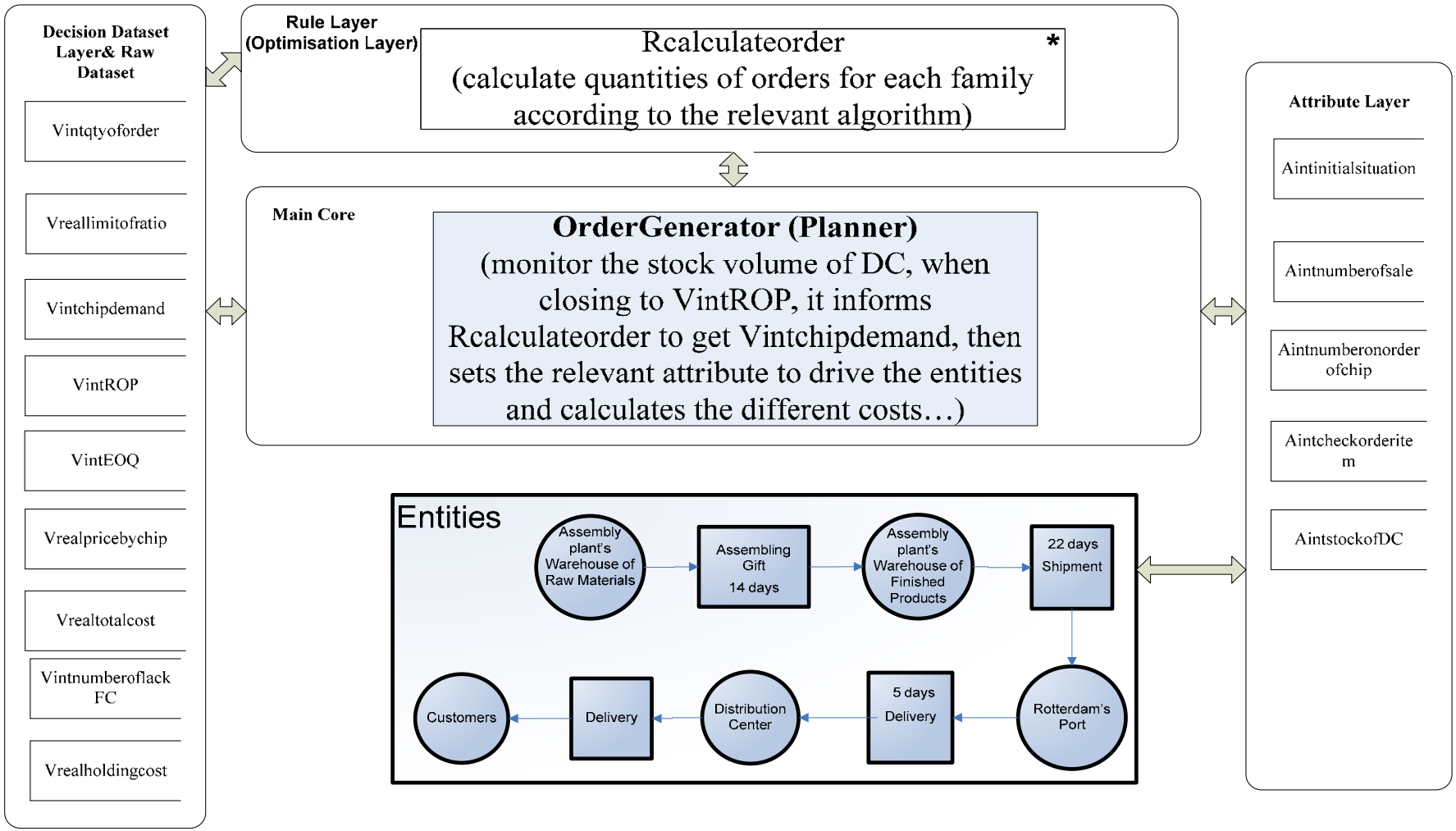


Figure 6.2 Concept Layout for a Stock-driven Case Study using WITNESS

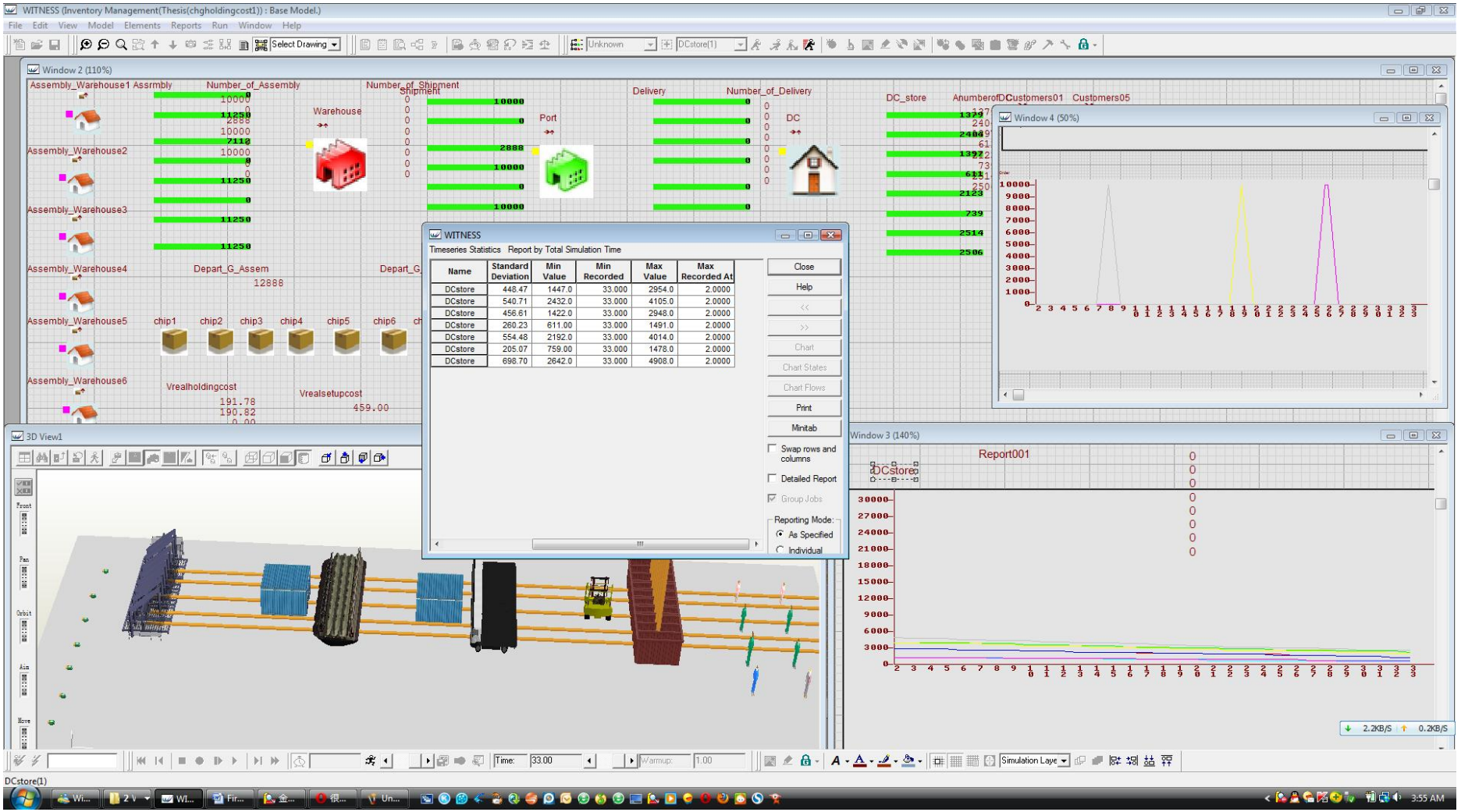


Figure 6.3 Cell Simulator for a Stock-driven Case Study using WITNESS

6.2.4 Experment Results

In order to analyse the performance of the replenishment algorithm in the different situation, the cell simulator has been run for several scenarios. In order to compare the holding costs of different areas, such as DC, assembly plant, shipment, and delivery. $H_{shipment}$ =25%, $H_{delivery}$= 25% are assumed. H_{dc}= 25% and $H_{assembly}$=20% (for chips) are outlined in Section 6.2.2. Consequently, total cost includes the holding costs of DC, assembly plant, shipment, and delivery. This includes multiple inputs and these results are compared and discussed as follow:

A. Different type of ship container

The assumptions of initial stock in DC of both scenarios are given as follows:

S1=10000, S2=10000, S3=10000, S4=10000, S5=10000, S6=10000, S7=10000, S8=10000. (Si means the stock number of family (chip) i in DC) simulation duration is 365 days.

Scenario 1: setting Nmax=0, Ta=0,

getting Total Cost= $171,693.89

Scenario 2: setting Nmax=30000, Ta=100,

getting Total Cost= $184,041.2

In Scenario 1, the variable Nmax is 0, which means there are no other families' replenishments except the necessary orders of the families that lowered to their ROP points. In Scenario 2, 30000 is set to Nmax for placing the other families' orders when one family arrives at its ROP points and the maximum number of other orders is 30000.

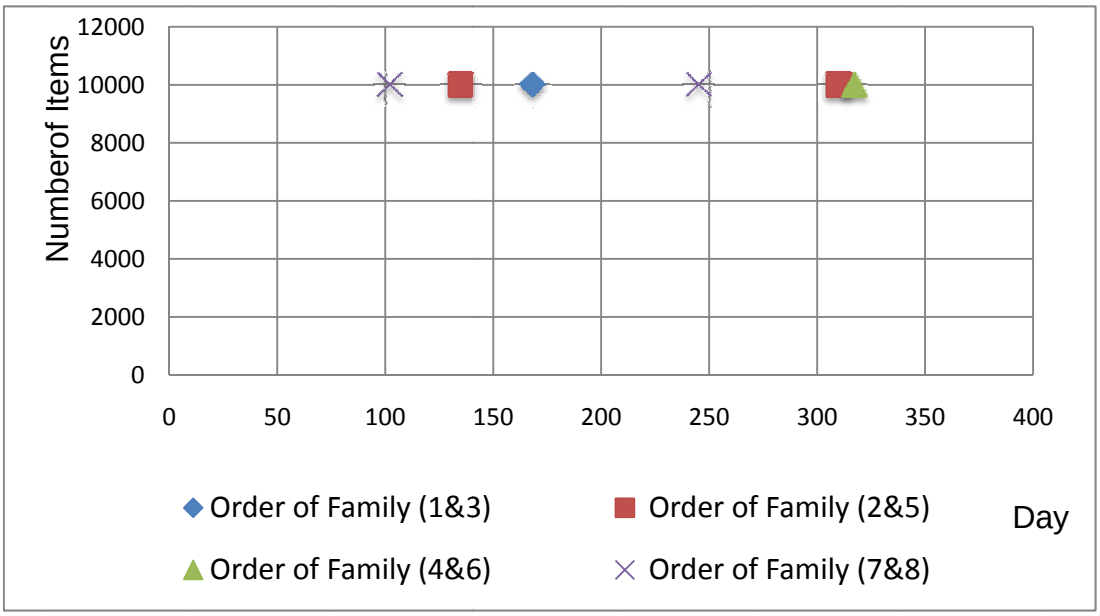


Figure 6.4 Orders States for Scenario 1

In Figures 6.4 and 6.5, only four markers represent the eight families of chips because the data for family 8 is similar to family 7, family 4 to family 6, family 2 to family 5, and family 1 to family 3 respectively. Figures 6.4 and 6.5 clearly illustrate that the orders are made six times in Scenario 1 and four times in Scenario 2 (first order for families of 2, 5, 7, and 8 at the day 102 and third order for families of 2, 5, 7, and 8 at the day 281). The parameter N_{max}=30000 reduces the number of the orders but increases the stocks in the DC.

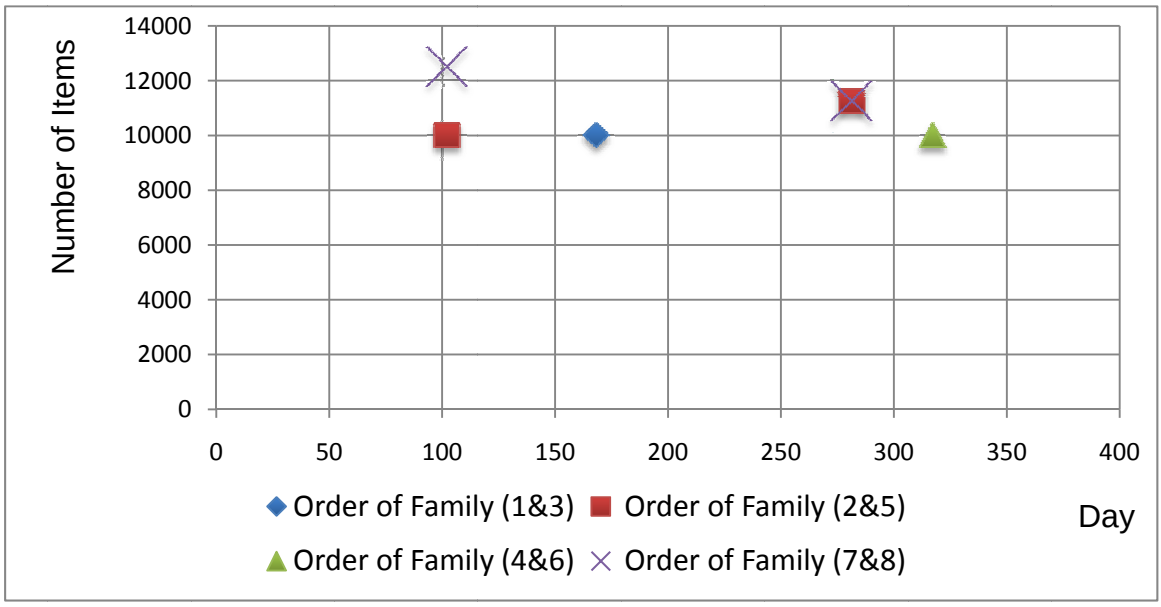


Figure 6.5 Order States for Scenario 2

Figure 6.6 shows the holding cost in DC is lower than in Figure 6.7 as there are more stocks of DC for Scenario 2.

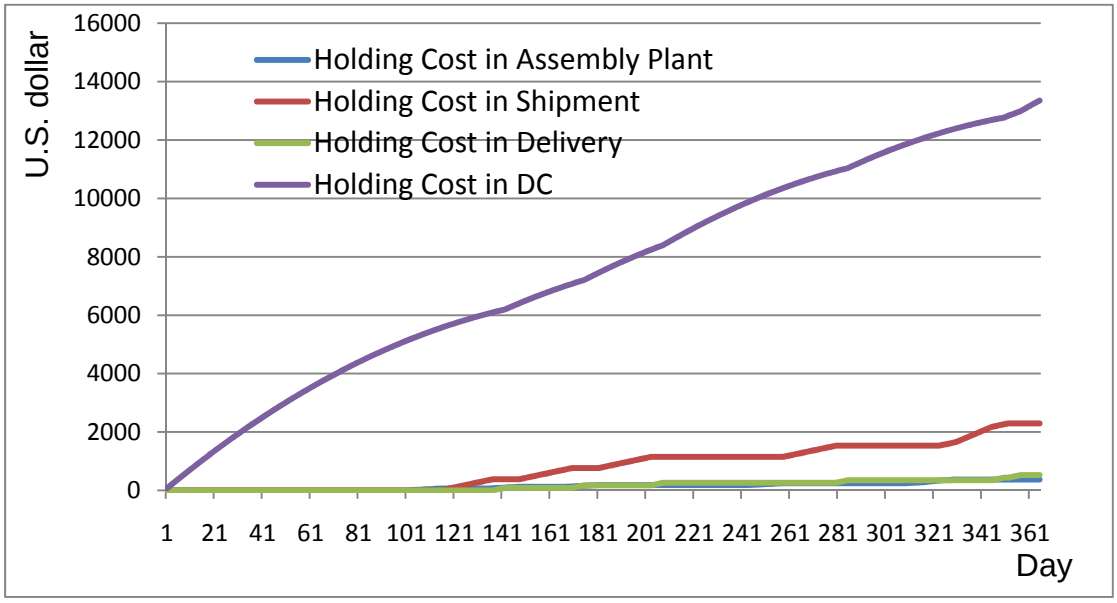


Figure 6.6 Holding Cost in DC for Scenario 1

Figures 6.6 and 6.7 indicate the holding cost in DC is larger than the holding costs during assembling, shipment, and delivery for Scenarios 1 and 2.

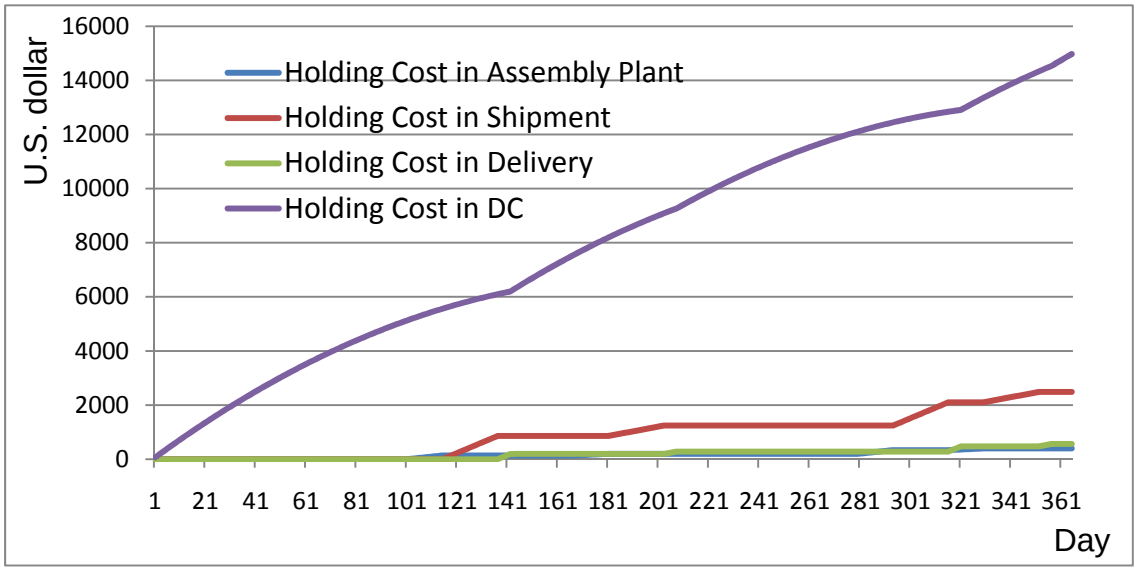


Figure 6.7 Holding Cost in DC for Scenario 2

Shipping cost in Scenario 2 is higher than in Scenario 1 as shown in Figure 6.8 because in this case study, the shipping (FOB) cost includes the purchasing cost and in Scenario 2 more number of items are ordered than in Scenario 1. The higher shipment cost is one of the main reasons which causes a large total cost in Scenario 2 compared to Scenario 1. In the meantime, the parameter Nmax=30000 reduces the setup cost.

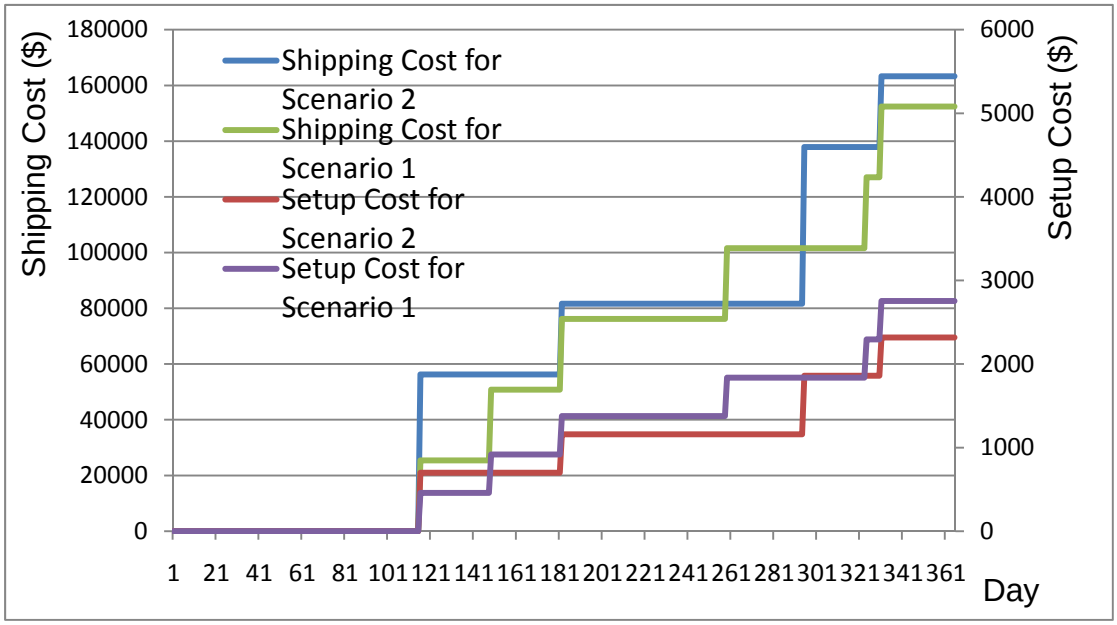


Figure 6.8 Shipping and Setup Costs for Scenario 1 and Scenario 2

B. Adding the shipment price of the partial container

It is assumed that the price of the partial container is $2.0 per item and the initial stock is similar to the data used in Scenario 1 and the simulation duration is 365 day.

Scenario 3: setting Nmax=0,Ta=0,

getting Total Cost= $262,646.30

Scenario 4: setting Nmax=30000,Ta=100,

getting Total Cost= $ 235,530.23

From the assumption outlined in Scenarios 3 and 4, the performance for 30000 Nmax in Scenario 3 is better than in Scenario 4. In Scenarios 3 and 4, there is only

change in the shipment cost and the other information is similar to the analysis of Scenarios 1 and 2. Shipment cost in Scenario 3 is $80,000 larger than in Scenario 4 and the large Nmax can benefit from adding the price of partial container, as shown in Figure 6.9.

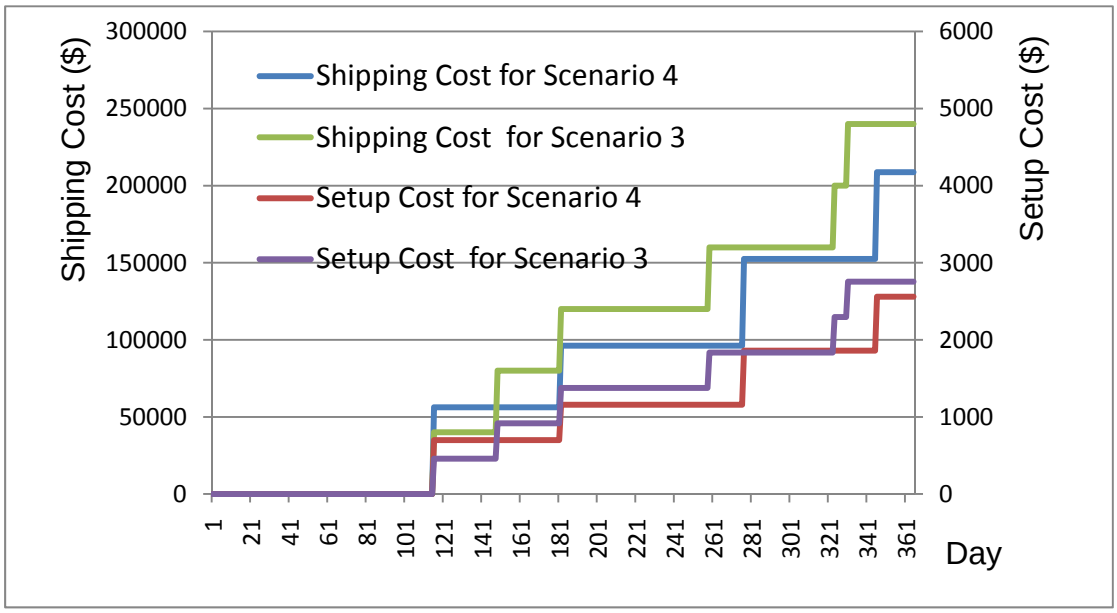


Figure 6.9 Shipment Costs for Scenario 3 and Scenario 4

C. Different initial stock in DC

The assumptions of initial stock in DC are given as follows:

S1=3000, S2=4150, S3=3000, S4=1500, S5=4100, S6=1500, S7=4930, S8=5000. (Si means the stock number of family i in DC)

Scenario 5: setting N_{max}=0,T_a=0,

getting Total Cost= $224,455.48

Scenario 6: setting N_{max}=35000,T_a=100,

getting Total Cost= $ 221,645.30

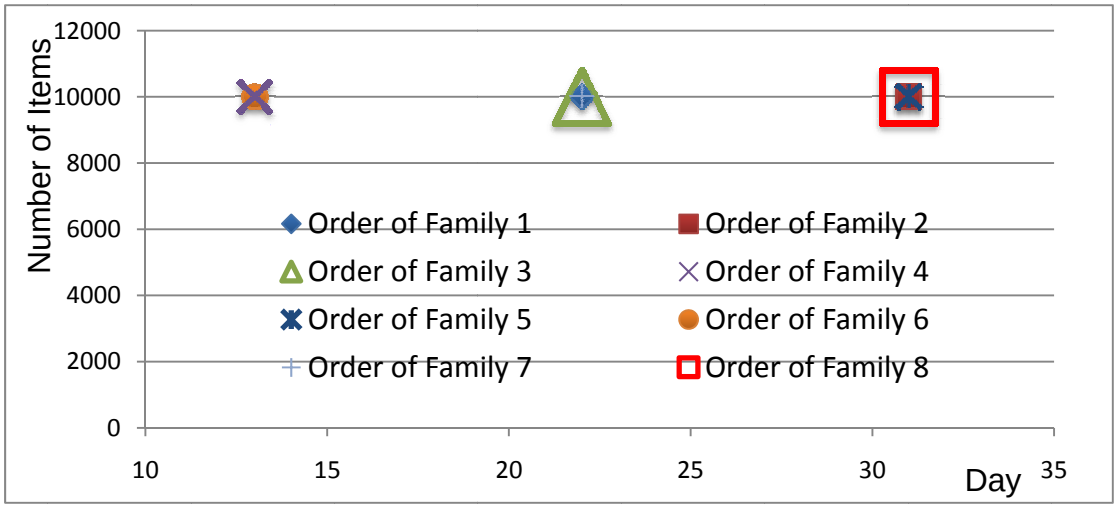


Figure 6.10 Order state from 10 day to 35 day for Scenario 5

In Figures 6.10 and 6.11, the number of booking orders is reduced from five to two times and the maximum quantity of order of families 2,5,7,8 increases from 10000 to 11250 at day 30. At day 13, the order consists of 10625 items for family 1, 625 for family 2, 10625 for family 3, 10625 for family 4, 625 for family 5, 10625 for family 6, 625 for family 7, and 625 for family 8 respectively are placed in Scenario 6.

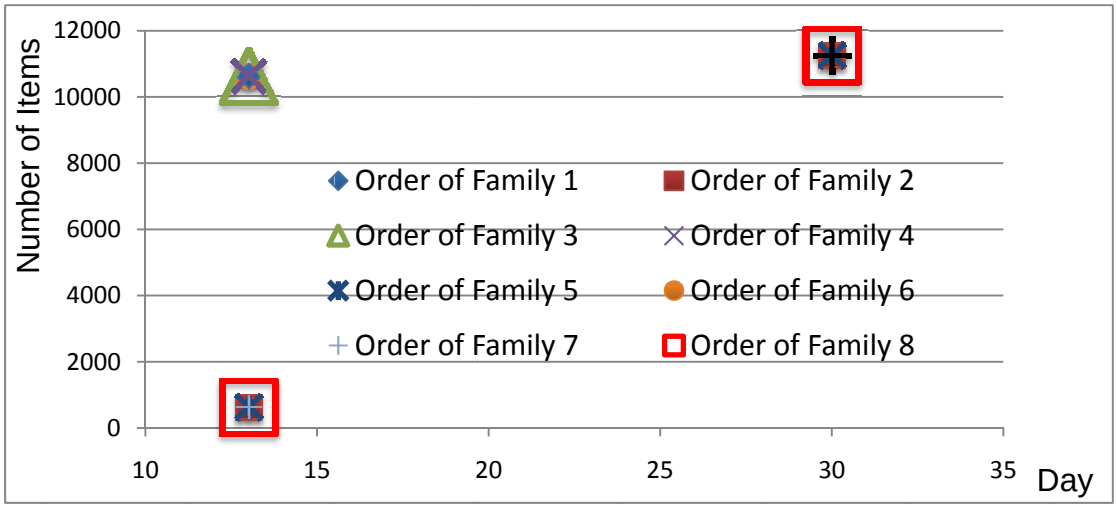


Figure 6.11 Order state from 10 day to 35 day for Scenario 6

The comparison between Figures 6.10 and 6.11 demonstrates the cost of reducing the order times is to increase the order quantity in the integrated algorithm.

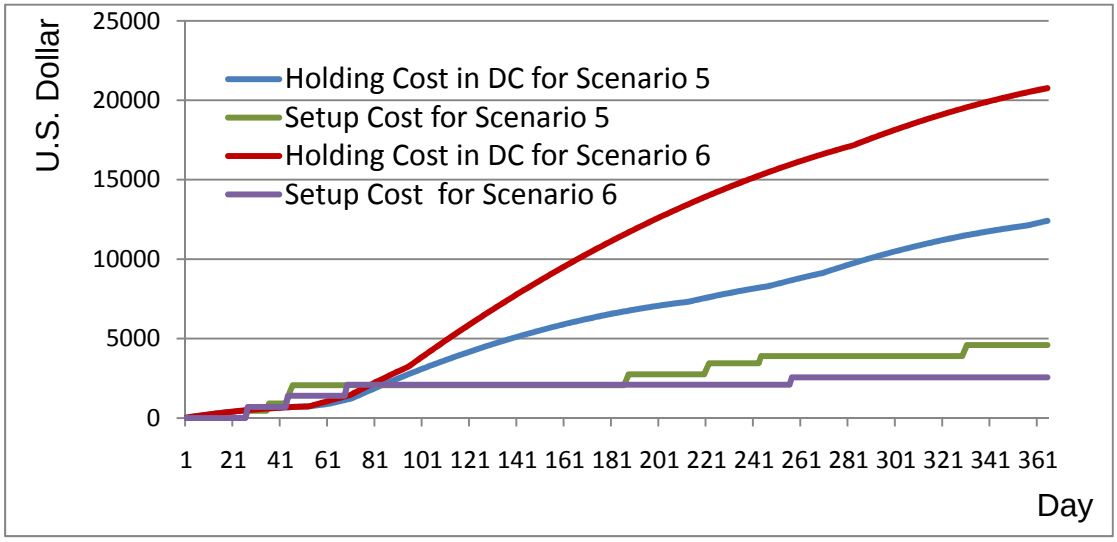


Figure 6.12 Holding Cost and Setup Cost for Scenario 5 and Scenario 6

Two costs are clearly showed in Figures 6.12 and 6.13 and the setup cost is reduced with holding cost increasing when setting N_{max} as 35000. However, it is difficult to determine the performance of the shipping cost for Scenarios 5 and 6 as shown from Figure 6.13 because using a *full container* method reduces the shipping cost and increasing the quantity of the order increases it in Scenario 6. In this case, setting Nmax as 35000 shows an improved performance of shipment cost and total cost.

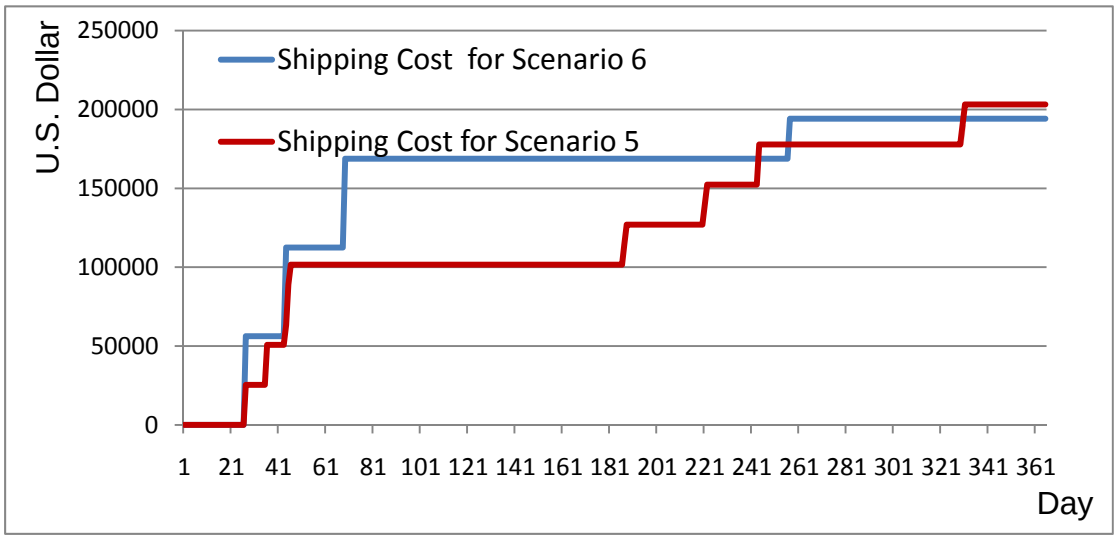


Figure 6.13 Shipping Cost for Scenario 5 and Scenario 6

D. Different standard deviation of sale distribution

The assumptions of initial stock using the data in Scenario 5 and setting ROP refers as Table 6.3.

Scenario 7: setting N_{max}=0,T_a=0, σ_L =5

getting Total Cost= $ 225,348.06

Scenario 8: setting N_{max}=40000,T_a=50, σ_L =5

getting Total Cost= $ 225,332.71

Scenario 9: setting N_{max}=0,T_a=0, σ_L =20

getting Total Cost= $ 238,542.46 (3 days out of stock)

Scenario 10:setting N_{max}=40000,T_a=50, σ_L =20

getting Total Cost= $ 243,036.47 (2 days out of stock)

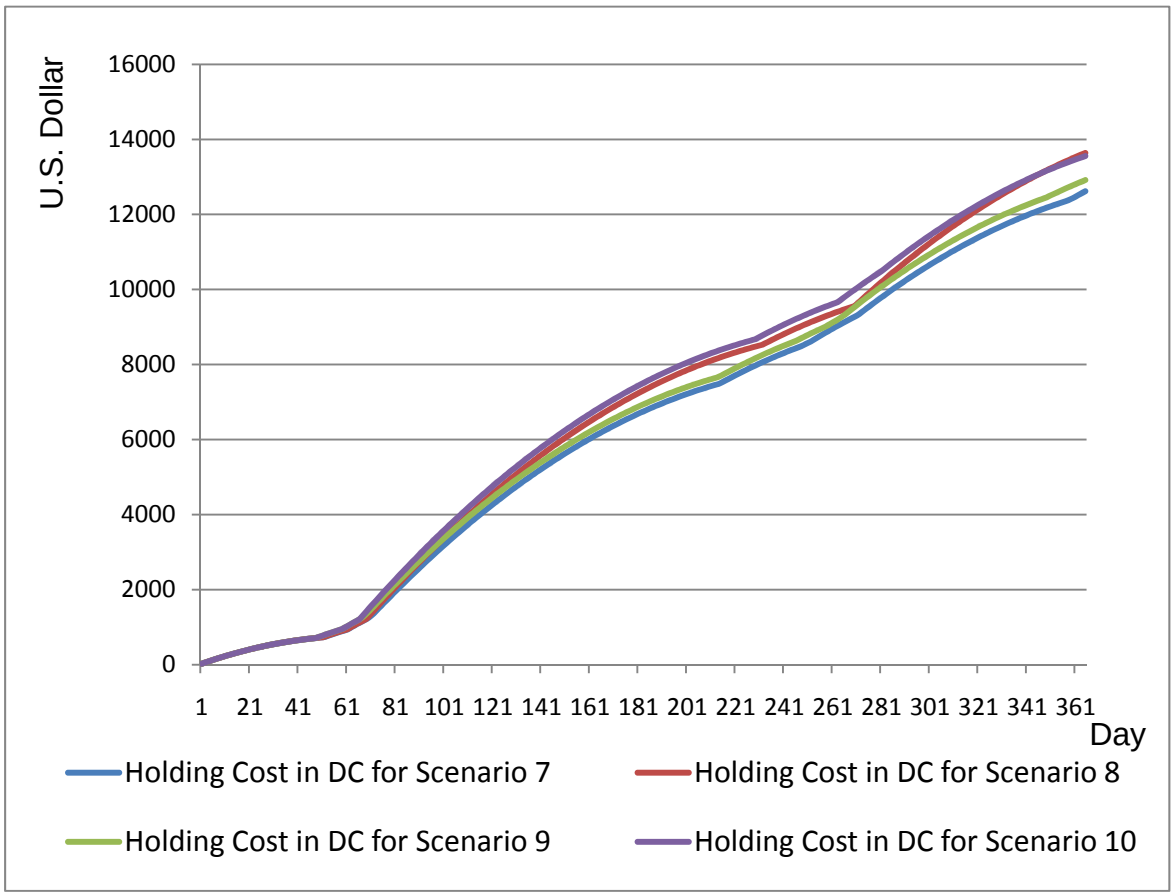


Figure 6.14 Holding Costs in DC for Different Scenarios

The holding cost is increased with the standard deviation whether setting N_{max} or not and the holding cost with N_{max} is higher than one without N_{max} at the same standard deviation, as shown in Figure 6.14.

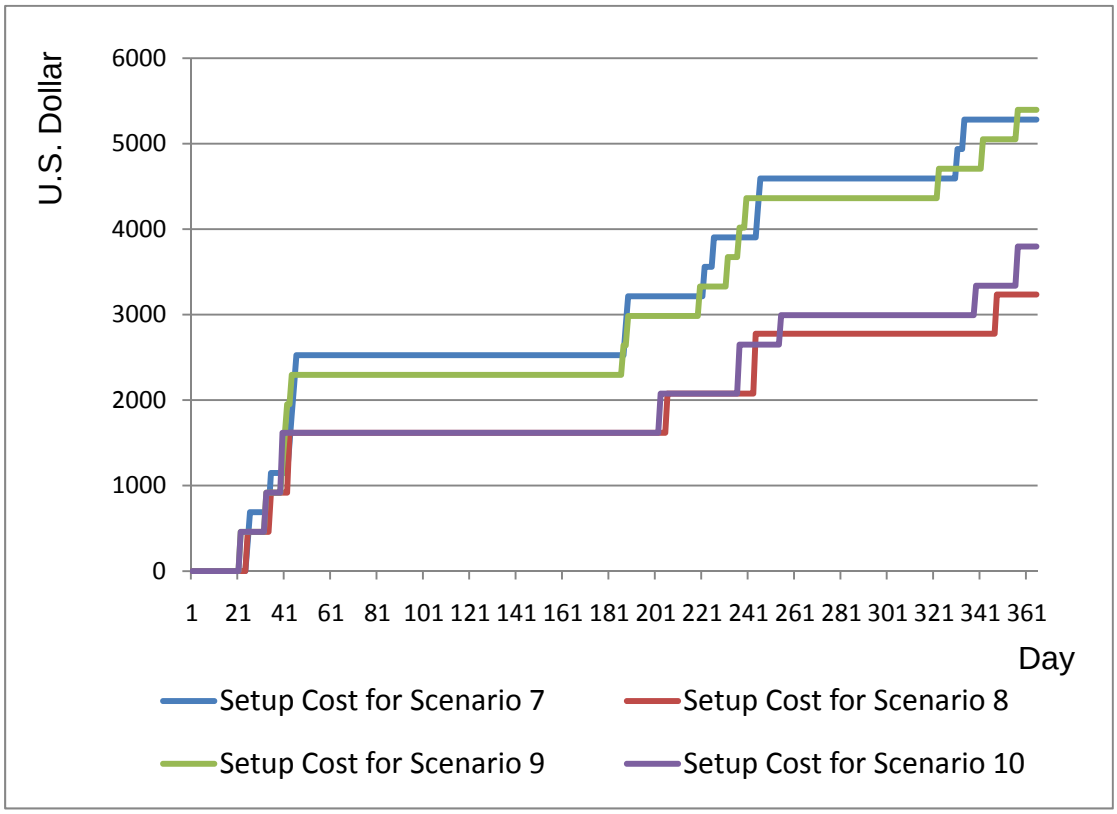


Figure 6.15 Setup Costs for Different Scenarios

Using the same standard deviation, setting N_{max} as 40000 can reduce the setup cost and the setup cost raises following the standard deviation, as shown in Figure 6.15.

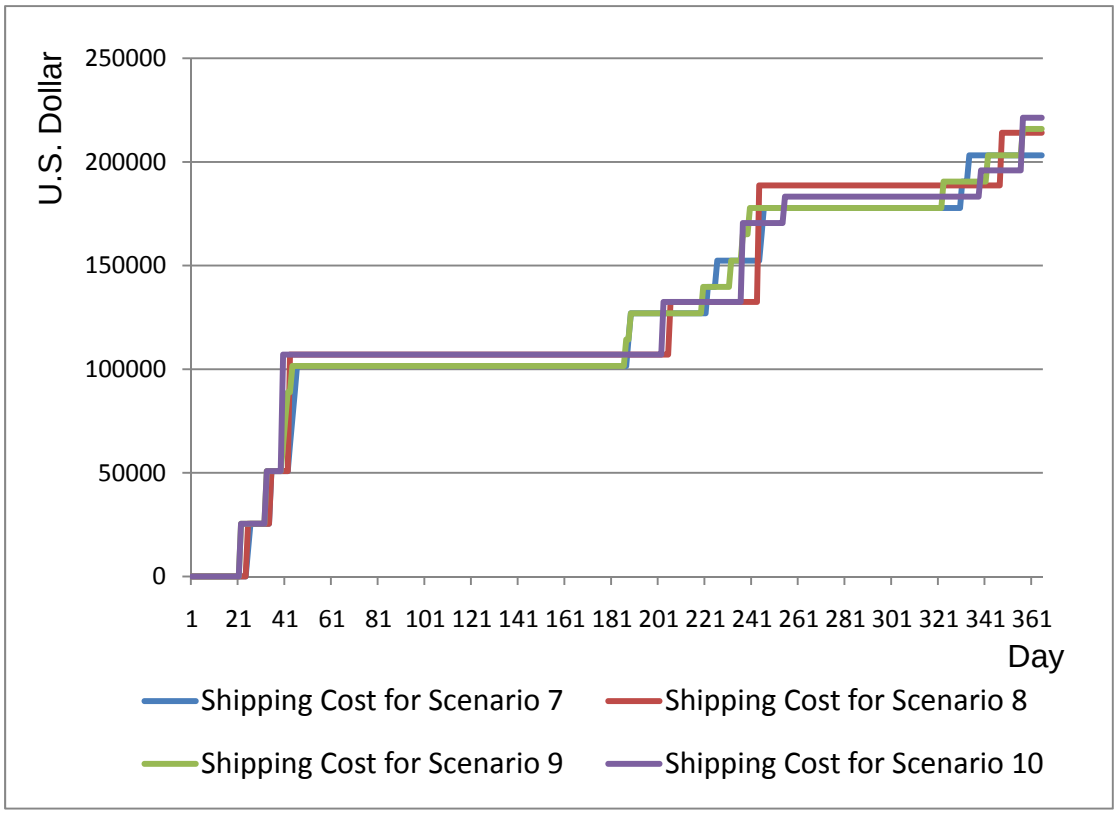


Figure 6.16 Shipment Costs for Different Scenarios

In Figure 6.16, there is no clear relationship between the shipping cost and the standard deviation. It is considered that the performance of the system using the replenishment algorithm is more than before when the demand fluctuates. In Scenarios 10, the replenishment algorithm is able to decrease the possibility of out of stock by increasing stock in DC.

The DMS of a gift supply chain, Case Study 1, is used to validate the capability of the proposed framework supporting the stock-drive control design for enterprises with a MTS product strategy. An improved replenishment algorithm is developed from the EOQ replenishment algorithm and is coded on the Rule Layer of the cell simulator, shown in Figure 6.2. The modularisation feature of this cell simulator gives an important advantage to the DMS, which is provided/supported by replenishment algorithms in the system as they can easily be adjusted or modified on the Rule Layer without revising the codes for the other layers. This case study is also used to verify the

design method of the hierarchical structure of the proposed framework on the supply chain layer.

Experiment results show the improved replenishment algorithm is able to reduce shipment and setup costs by merging orders under ten different scenarios. However, this algorithm increases holding costs. The improved algorithm can give a relative advantage when increasing shipment price of the partial container (Scenarios 3 and 4 outlined in Section 6.2.4). This algorithm can also decrease the possibility of out of stock compared to the original EOQ replenishment algorithm in a dynamic marketing environment (Scenarios 9 and 10 outlined in Section 6.2.4). The main benefits of the framework, such as optimisation and designing scenarios, are demonstrated in the UK-based SME, Case Study 2.

6.3 Schedule-driven Case Study

6.3.1 UK-based Company Overview

A UK-based engineering company has been selected to conduct the MTO case study of a simulation-based DMS. The company is a manufacturing enterprise with more than 20 years professional experience in the high-precision sheet metal machining field. It has accumulated more than 50 customers and created hundreds of different types of products to satisfy its customers' specific and individual requirements. This requires an agile DMS to help staff to manage an efficient production schedule.

The staff at the company has been divided into four groups according to employee responsibility, namely sales, technical, work and financial group respectively. Three directors are in charge of all the business processes of the company and a simple workflow is shown in Figure 6.17.

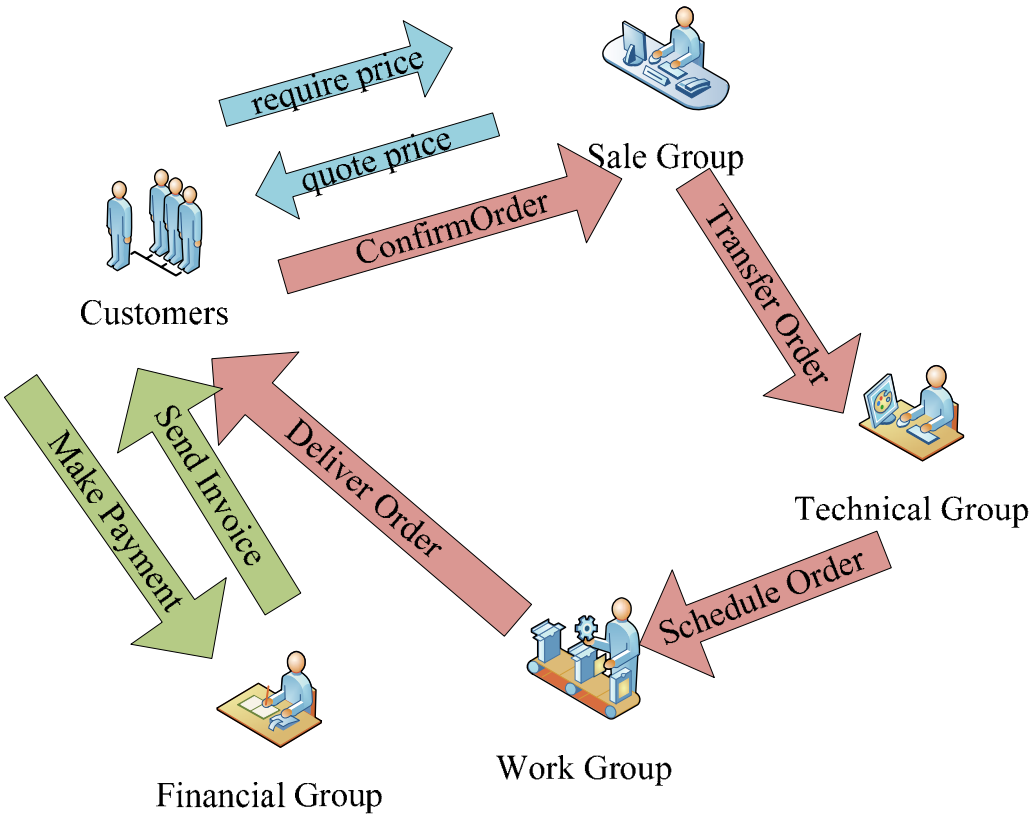


Figure 6. 17 Workflow of the UK-based Case Study

6.3.2 Challenge Issues

Although an ERP system – FabriTrak – has been successfully applied in improving the management level of the company, this system only analyses current data and provides various reports on the situation of the company to the managers. The system is not able to analyse data or simulate results based on different scenarios, and this means that the system lacks the ability to answer 'what-if' questions. The maximum function of the system is as an indicator to identify the current situation of the company, which is mainly focused on the shop-floor layer. When managers want to make decisions on adding equipment or employees, accepting urgent orders, or altering polices, they simply get real-time data from the ERP system more quickly, accurately, and completely than previously. The simulation results and expected performance data is not displayed, and they still have to visualise these results based on their previous experiences. This leads to a reduction in the real value of the data, thus limiting

improvement at a management level. Making incorrect decisions would result in a loss of customers, and might even put the company itself at risk.

Therefore, a new system is necessary for the company, which is capable of providing the necessary results and performance depending on different scenarios. The FabriTrak system enables employees working in the company to have an information flow concept, trace records and track product data. The data collected from FabriTrak would gradually become more accurate, and the new proposed system would benefit from the issues outlined.

6.3.3 Proposed Solution and System Design

There are two conventional methods of building a model to meet the demands of a company: mathematical modelling and simulation modelling. This SME has hundreds of routes and processes for making various products for different customers. Its complicated relationships cannot be completely expressed by mathematical equations, and consequently mathematical modelling is not suitable for the SME's case study. The only solution is to adopt a simulation modelling method.

There are several popular simulation software packages including WITNESS, Arena, AutoMod, etc. WITNESS has been adopted as a simulation program platform to develop a cell simulator because of its many advantages: popularity, customer relationship, and licences discounts (via the University). The proposed DMS adopts an individual normal-performance PC as a platform because an up-to-date workstation has a fast enough CPU, and a distributing cooperation method is not necessary to develop a calculating capacity.

A database is necessary to store the data and complicated relationships, and also to keep information independent and separate from the cell simulator in order that it can be modified easily. Later Section 6.3.8 will give further details on building the database

and discuss how to allocate tasks or functions into the cell simulator and the databases. On analysing the data records from ERP system of this SME, usually about a million records are stored in their database. Consequently, a small database is sufficient to support the SME's data. Compared to several small popular databases, such as MySQL, MS Access, Visual FoxPro, etc., MySQL is an open source database and is mostly used in the web design. However, MySQL lacks a friendly interface and good technical support. Visual FoxPro is a small database integrated with a professional programming tool and has a highly-efficient programming environment and a friendly user interface. However, it would require a development license. MS Access is a popular database integrated in the Microsoft Office suite which is widely used by many companies including the SME used as a case study.

This simulation system adopts MS Access 2007 database, which is quick, relative new, and the most popular version in the Access family. Its feature of supporting SQL standards enables it to be transferred to another database. For example, in order to meet the requirements of a large company, a professional database like SQL Server, DB2, Oracle, or Sybase would be required. Transferring the database developed from MS Access into a professional database like SQL Server is not an issue. A professional database support SQL standards, and SQL sentences created in MS Access are used to build the tables and their indexes, transfer data, and build the relationships between the tables in these databases. Alternatively, you could use an easy-to-use transfer tool for transferring into a professional database to achieve the above aims.

Figure 6.18 shows the architecture of the DMS consisting of a cell simulator, MS Access 2007 database, and an optimisation block operating on an individual PC. A normal-performance PC using WITNESS software package and MS Access 2007 is the requirements of this simulation-based DMS for the UK-based case study. Most SMEs are able to afford similar simulation-based DMS. The cell simulator, which is coded by WITNESS, is able to exchange data with MS Access 2007 through its embedded data

link. Users can input rules or parameters of scenarios into the database and get an initial solution from the simulations results. An optimisation block is then able to provide a group of optimum solutions to help users to make decisions. The designs of the optimisation block and the cell simulator were described in detail in Chapter 4 and Chapter 5 respectively.

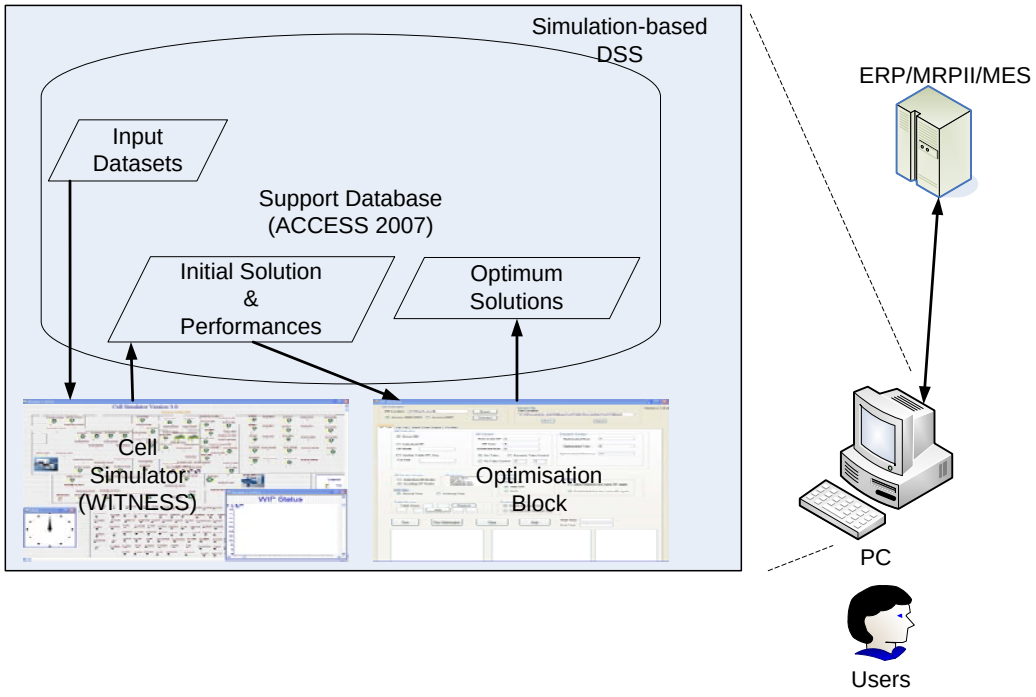


Figure 6.18 Diagram of System Design

6.3.4 Data Categories

To be able to collect data, the availability of the data is obviously an important factor. This data is required for the cell simulator such as time, quantity, labourers, etc., consequently this data must be either collected or estimated. Otherwise, the cell simulator would fail. As a result, the availability of the data used to support the simulator should be considered during the design of the cell simulator.

Stewart classed data into three categories, namely A, B and C respectively (Robinson, 2004). In practice, the data already available in the related forms or stored in

the system, such as machine setup time, cycle time, belong to category A. The data that is not available, but can be collected, like schedules and shop-floor dispatch rules, is classed as category B. Generally, the methods of collecting this data are interviewing and observing. The important point is that this data is collected within a time-scale. Category C represents the data which is either not available or not collected for reasons such as time constraints or limited resources, or not connected with the related process. For this data, there are two common methods of dealing with it. One is estimation and the other is to replace them with experimental data.

6.3.5 Required Data

In order to support the simulator, three key areas of data need to be collected. Firstly, data related to routes is important. The process flow on the shop floor must be following an identified route. Normally, the route taken should be at machine level and that means that the route defines which machine or type of machine is able to take a certain process. This gives a complete record to trace the machine entities. In order to build a flexible route in the system, the database builds the work centre level (machine group), the machine level, and their relationships. This is able to support a simulation based on either the work centre level or the machine level. Secondly, the data relating to the productivity of employees and machines need to be collated. Usually the same process with different employees show different performance on the same machine. An employee on different machines may also produce a different performance. Time and yield are the common factors used to measure the performance. According to the current situation of the company, time factors of machines are divided into setup time and process time. Thirdly, other basic information such as order including prices, arrival times, quantities, etc., are available in the ERP system.

6.3.7 Normalization Theory

Microsoft Office Access 2007 has been adopted as a Relational Database Management System (RDBMS) in the simulation system and the reasons for using this database are described in Section 6.3.3. In the database design, normalization is an important technique to reduce data redundancy and to keep information derived from one data source.

Normalization minimizes the storage space of the necessary data and avoids redundancy by setting relationships between primary keys and foreign keys of the relative tables. It can test and determine the degrees of the relationship between attributes. Usually, a database design is required to satisfy respectively three normal forms: First Normal Form (1NF), Second Normal Form (2NF) and Third Normal Form (3NF). 1NF indicates each row and column intersection only has one value and an attribute of the entity's own a value and not repeated values. 2NF is based on 1NF and requires that the other keys are functionally dependent on the primary keys. 3NF is based on 1NF and 2NF, and it does not require repeated primary-keys to exist in the database (Ponniah, 2003) .

6.3.8 Database Design

There are about 20 tables in the database shown in Figure 6.19. The database provides the information of routes, work orders, customers, labourers, etc. and is referred to a relationship database. The information concerning the tables shows the primary keys and the other tables can be referenced using the primary keys as the foreign keys. This keeps information normalized and reduces redundancy. Figure 6.19 shows the database structure of the DMS for the UK-base case study.

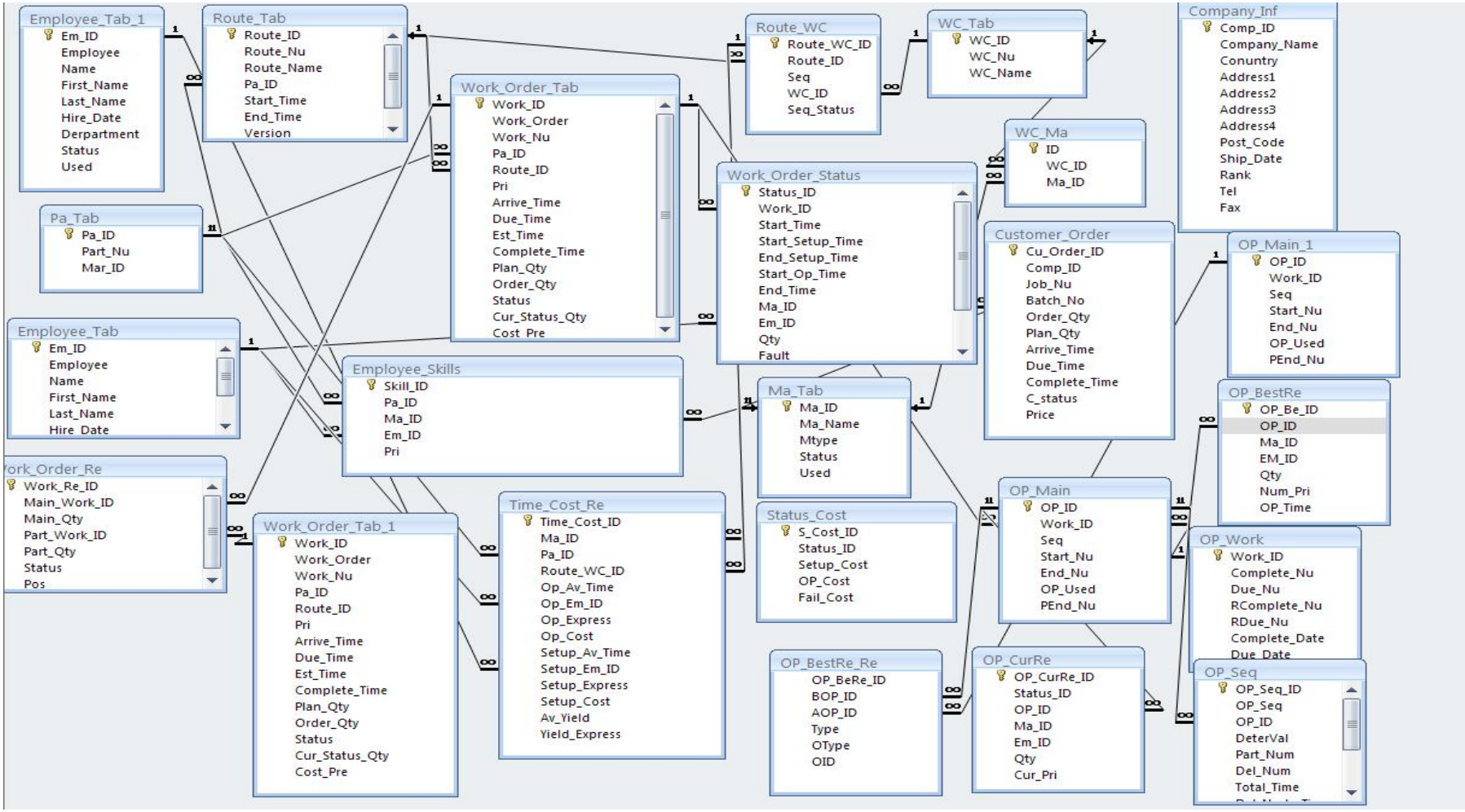


Figure 6.19 Database Structure

Firstly, the table details and their stored information are described. Table Company_Inf stores the relative basic information of the company, such as company name, country, address, etc. Table Customer_Order keeps the materials of orders, like price, due time, and quantity and so on. Table Employee_Tab shows the basic data of employees, including name, hired date, home address, etc. Table Ma_Tab provides the relative information for the machines and Table WC_Tab gives the necessary data for the work centres.

Secondly, the relations between the related tables are discussed. The database design meets the requirements of 1NF, 2NF and 3NF. The primary key Em_ID of Table Employee_Tab, Ma_ID of Table Ma_Tab and Route_ID of Table Route_Tab are three main items to represent the information of their tables respectively and to link to the most of tables. Table WC_Ma represents the 'many to many' relationship between work centres and machines and ids used to store their primary keys. This relationship means one work centre is the collection of machines and one machine can do the tasks of different work centres. Table Route_WC shows similar many to many relationships to the table WC_Ma. A route consists of several work centres and a work centre belongs to several different routes. The tables Work_Order_Tab, Work_Order_status and Status_Cost are typical of three layer tree construction. Table Work_Order_Tab represents the basic status and information of the work order. Table Work_Order_Status shows the information of every completed process of the work order and link to Table Work_Order_Tab by its foreign key Work_ID. Table Status_Cost expresses the information related to the cost of each process of the work_order, and status_ID is the foreign key to link Table Work_Order_Status.

There are six tables with 'OP_' in their table names to define an optimum solution. Table OP_Main stores the status of every operation node and Table OP_CurRe maintains the information of the OCR of every operation node and the remaining optimum tables are described as follows:

- Table OP_BestRe shows the information of the ECR of every operation node.
- Table OP_BestRe_Re describes the relationships of the relevant operation nodes.
- Table OP_Seq expresses the status of every iteration and tabu information.
- Table OP_Work shows the relevant information of the tardiness of every work order.

6.3.8 Verification and Validation

The white-box testing method is used to determine whether the results can be deduced from the design logic, and it has been adopted and completed at the debug stage of the programming design of this DMS consisting of a cell simulator using WITNESS and an optimisation block coded by C#. According to a report from the Department of Commerce U.S. concerning simulation modelling requirements (Umeda and Jain, 2004), an adaptive simulator of SC is required to provide performance measurements: resources utilisation, waiting queue length (waiting time), throughput rate, elapsed time for a system performance analysis tool. It is usually easy to identify the performance of basic problems such as capacity planning problem, resource planning, lead-time planning problems, and production planning problems through the analysis tool. The function testing can be checked to estimate whether it provides all of these measurements which are outlined in following sections.

6.3.9 Simulation Results and Analysis

A. Resources Utilisation

Figure 6.20 shows the machine utilisation of the case study and the utilisation of inspection machines has occupied 85.78% which is much higher than the utilisations of the other machines. Consequently, the necessary task is to lower the utilisation of the inspection machines. This leads to ask a ‘what-if’ question to add a new inspection machine.

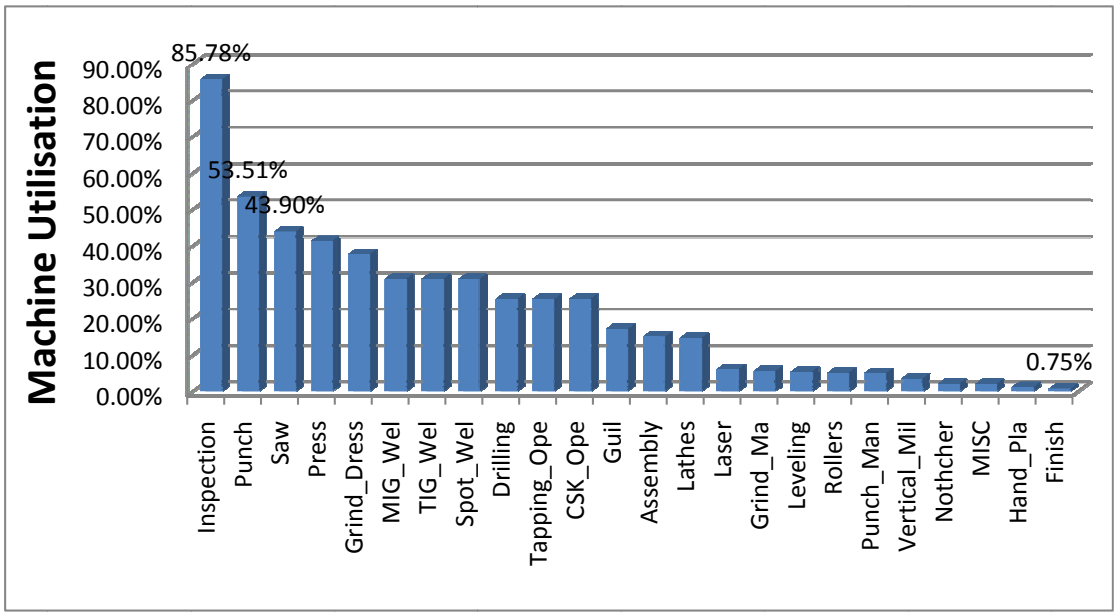


Figure 6.20 Machine Utilisation

The simulation result for adding a new inspection machine is shown in Figure 6.21. This result shows the utilisation of the inspection machine will be lowered 77.54% (decrease about 8%) and the utilisation of the punching machines will be raised from 53.51% to 74.02% (increase about 20%). Therefore, adding a new inspection machine decreases the utilisations from 85.78% to 77.54%. Utilisations of the other machines, such as punching, guillotine will be risen. The simulation results are helpful to support decisions making by answering 'what-if' questions.

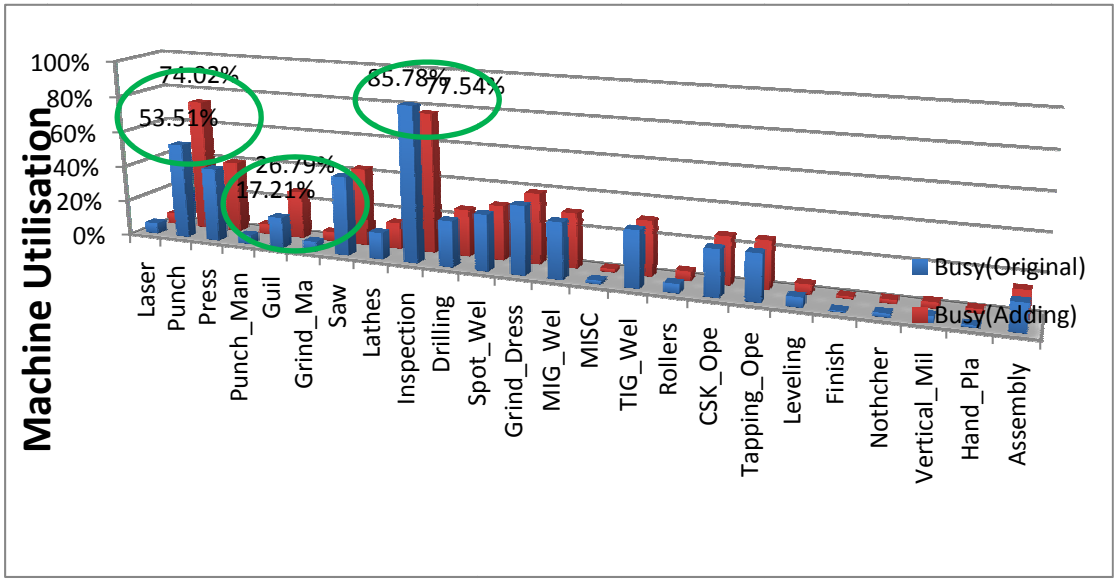


Figure 6.21 Machine Utilisation when Adding an Inspection Machine

B. Waiting Queue Length (Waiting Time)

In the cell simulator, a WIP monitor has been developed for real-time observing the situation of WIP of machines chosen as shown in Figure 6.22. The upper-levels (upper-limits) of WIP of different machines can be manually set to alarm users when real-time WIP is higher than its level (limit).

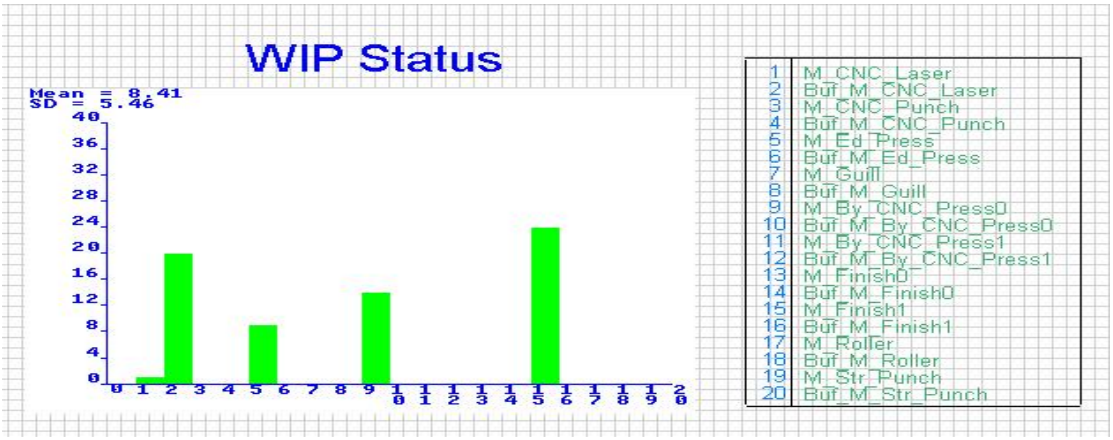


Figure 6.22 Real-Time WIP

After running a simulation, every machine can get its waiting queue length at a particular time from the Table Work_Order_status. Based on the information of Table Work_Order_status, a statistic parameter -the rate between the waiting time and total time can been calculated by Work Order (Work ID). Figure 6.23 shows the value between waiting time and total time of several work orders.

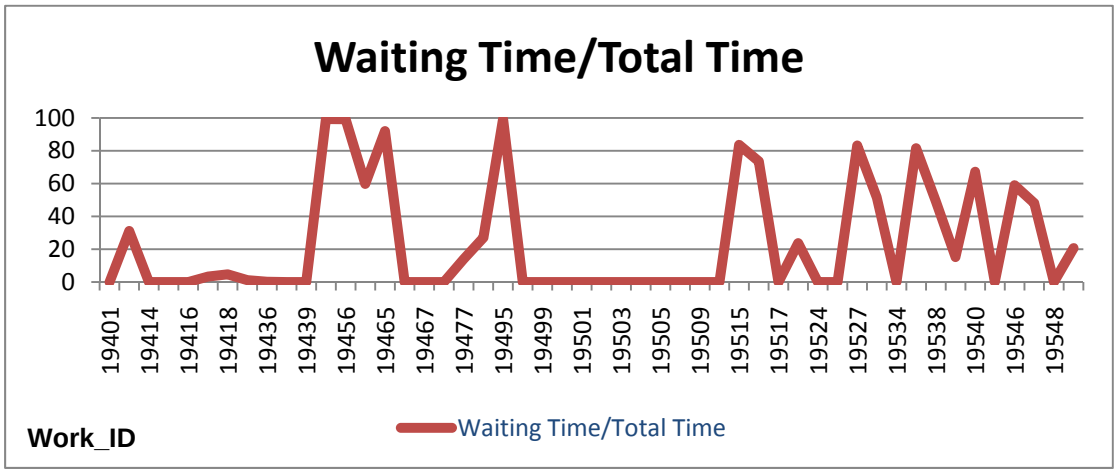


Figure 6.23 Waiting Time/ Total Time

C. Throughput Rate

Table Work_Order_status records the all details of every work order including processed machines, start time, end time, etc. Thus, every machine and labour can get their through rate via Table Work_Order_Status. Table Work_Order_Tab records the all statuses of every work order (job) in shop floor, such as arrival time and complete time and Table 6.4 shows the status of partial completed work orders of this shop floor.

Table 6.4 Status of Partial Completed Work Orders

Work_ID	Work_Nu	Arrive_Time	Complete_Time	Due_Time
23876	00004280	1/23/2008	1/25/2008 08:09	1/28/2008
23881	00004668	1/11/2008	1/14/2008 08:47	1/15/2008
23722	00004896	2/7/2008	2/11/2008 08:13	2/12/2008
20923	00004976	2/5/2008	3/3/2008 09:36	3/5/2008
24709	00005061	2/11/2008	3/10/2008 12:21	4/9/2008
20832	00005144	2/8/2008	2/29/2008 11:08	2/29/2008
21521	00005176	2/11/2008	2/14/2008 12:44	2/15/2008
22486	00005181	2/14/2008	2/18/2008 14:30	2/19/2008
21480	00005239	2/14/2008	3/12/2008 11:19	3/13/2008
21219	00005277	2/13/2008	3/10/2008 16:43	3/12/2008
21734	00005322	2/14/2008	3/12/2008 11:39	3/13/2008
21382	00005327	2/20/2008	3/3/2008 09:14	3/7/2008
24438	00005363	3/13/2008	4/1/2008 11:53	4/23/2008
21746	00005365	2/14/2008	3/12/2008 11:48	3/19/2008
22199	00005370	2/27/2008	3/24/2008 14:58	4/9/2008
21545	00005417	2/20/2008	3/13/2008 09:13	4/10/2008
22777	00005449	3/19/2008	3/24/2008 16:43	3/26/2008
21914	00005470	2/28/2008	3/3/2008 09:32	3/13/2008
21834	00005509	2/20/2008	3/13/2008 09:31	3/26/2008
22390	00005534	3/10/2008	3/31/2008 13:53	4/3/2008
22550	00005544	3/12/2008	4/1/2008 09:45	4/7/2008
21946	00005568	2/27/2008	3/24/2008 12:42	3/26/2008
21745	00005591	2/25/2008	3/14/2008 09:23	3/19/2008
22845	00005793	3/13/2008	4/1/2008 11:38	4/2/2008
24205	00005879	3/27/2008	4/2/2008 12:37	4/16/2008
27433	00005953	3/19/2008	4/2/2008 11:26	4/3/2008
24452	00006018	3/27/2008	4/3/2008 08:19	4/10/2008

D. Elapsed Time

In this section, an order with a quantity of 50 is assumed to be taken and its elapsed times of every process are shown in Table 6.5 through running a simulation.

Table 6.5 Status of Work_Nu 00004525

Work_ID	Seq	Start_Time	End_Time	Ma_ID	Em_ID	Qty
19560	1	1/10/2008 08:01	1/10//2008 12:22	31	64	50
19560	2	1/10/2008 12:22	1/10//2008 12:45	37	35	18
19560	2	1/10/2008 12:22	1/10//2008 12:45	38	69	17
19560	2	1/10/2008 12:22	1/10//2008 12:45	36	62	15
19560	3	1/10/2008 12:45	1/10//2008 12:47	45	33	50
19560	4	1/10/2008 12:47	1/10//2008 12:55	20	56	50

6.4 Performance of Optimisation Block

6.4.1 Overview

The data from a UK-based SME case study is adopted to evaluate the performance of the proposed optimisation block consisting of CKPO and DOA. The six months' data of order information and the other relevant data of machines, employees, processing, etc. are transferred from a FabriTrak ERP system or collected from the interviews of the staffs working in this company. The initial solution is provided after running the simulation model in the cell simulator illustrated in Section 6.3 designed for the company. The performance of the optimisation block is analysed through the solutions generated by the optimisation block based on the initial solution.

6.4.2 Differenterences of CKPO Settings

This section discusses the performance of different CKPO settings chosen based on an initial solution from the cell simulator which adopts a FIFO policy and uses data for a March period. A comparable analysis of performance, such as Best (the best

total tardiness of the optimum schedules), CPU time, Worst (the worst total tardiness of the optimum schedules), etc. are discussed to evaluate the effectiveness of CKPO under different CKPO parameters setting, as shown in Tables 6.6 to 6.8. The relevant notations are defined as follows:

About the settings of the initial schedule:

FIFO: the initial schedule got by a cell simulator adopted a FIFO rule

EDD: the initial schedule got by a cell simulator adopted an EDD rule

DV: maximum number of splitting a work order (job) and then assigning them into several CRPs

About the settings of CKPO:

OP: the number of the neighbours in every iteration

TB: the length of the tabu list

OT: the number of iterations

ACR: available common resource of optimisation nodes in a tabu list

ACRP: available common resource pair of optimisation nodes in a tabu list

AO: allowing reusing optimisation nodes in the tabu list

FO: forbidding reusing optimisation nodes in the tabu list

NU: the number on unimproved iterations (in this section, NU is assumed as ∞)

VO: a threshold value (in this section, VO is assumed as 0)

Original Tardiness: the total tardiness of the initial schedule

Cmax: the makespan of the initial schedule

Best: the best total tardiness of the optimum schedules

Cmax in Best: the makespan of the schedule with the best total tardiness

Worst: the worst total tardiness of the optimum schedules

AVG: the average value of the total tardiness of the optimum schedules from

an optimisation process under a particular setting

STD: the value of a standard deviation of total tardiness of the optimum schedules from an optimisation process under a particular setting

Partial Nodes: the quantity of operations after the earliest directly effective operation in the time horizon.

Dealed Nodes: the quantity of operations processed by a CKPO rescheduling method.

Total OT: the total number of optimum solutions of an optimisation process

First Best (the number of iterations): the first schedule obtains the best total tardiness after the number of iterations

Total Time: the time which the PC spends on the optimisation process

Time of Nodes: the time which the PC spends in processing the operation nodes of the optimisation process

After an optimisation node has been *swapped*, a benchmark schedule should be rescheduled for obtaining a new optimum schedule. A rescheduling method is used to indicate how to recalculate the relevant operations for generating a new schedule. The first general rescheduling method is to reschedule all operations of the schedule from the begin to the end, called **completed rescheduling**. The second general rescheduling method is only to recalculate the operations after the earliest directly effective operation in the time horizon and remain the operations unchanging before the earliest operation, called **partial rescheduling**. The third rescheduling method is the proposed CKPO rescheduling method only handles the effected operations. To compare two parameters 'Partial Nodes' and 'Dealed Nodes' is mostly to used for comparing the performance of a partial rescheduling method and a CKPO rescheduling method.

In this case study, there were 1194 CRPs and 481 work orders (jobs) from a

March period. The length of a tabu list (TB) is chosen using three constant values 3,5,10 and two dynamic values for comparing the performance of different TBs under OT20.

The first dynamic value is calculated using the method of Zhang et al. (2007) outlined in Section 2.10.3.2.

$$L=10+(n/m)=10+(481/1194)\approx 10$$

when n<=2m TB=Random(L~1.4L)=Random(10~14)

The second dynamic TB value is assumed as Random(6~9).

CKPO parameters setting consists of tabu setting and OP issues. OP issues represent the parameters in CKPO relating to stopping criteria and iterations including OP, OT, NU,VO. In this section, in order to explore the performance of CKPO under different tabu settings, NU and VO are assumed as ∞ and 0 respectively. The DOA's parameters are also mainly assumed as a (DV3, OI5, MV10) (means DV=3,OI=5,and MV=10) setting, where DV is the maximum number of job splitting, OI is the number of iteration of DOA, MV is the minimum value between p_{max} and p_{min}. No stock and left orders are assumed at the beginning of every month outlined in Sections 6.4 and 6.5. Tabu setting expresses the parameters relating to tabu status and consists of TB and TB design combination outlined in Section 4.4.3.

Table 6.6 Performance of Optimum Solutions under different Tabu Setting (FIFO,DV3,OP3,OT20) for March

CKPO		Original Tardiness(S)	Cmax	Best (S)	Cmax in Best	Improvement (%)	Worst	AVG	STD	First Best	Total OT	Dealed Nodes	Partial Nodes	Total Time(S)	Time of Nodes (S)
OP Issue	TB design														
OP3,TB0,OT20	~	3363481	579923	775204	588607	76.95	2681729	951275.23	312,955.69	6	60	18275	37688	3497	2449
OP3,TB3,OT20	ACR,AO	3363481	579923	775204	588607	76.95	2681729	951275.23	312,955.69	6	60	18275	37688	4470	3409
OP3,TB5,OT20	ACR,AO	3363481	579923	775204	588607	76.95	2681729	951275.23	312,955.69	6	60	18275	37688	3678	2636
OP3,TB6-9,OT20	ACR,AO	3363481	579923	775204	588607	76.95	2681729	951275.23	312,955.69	6	60	18275	37688	3688	2645
OP3,TB10,OT20	ACR,AO	3363481	579923	775204	588607	76.95	2681729	951275.23	312,955.69	6	60	18275	37688	3487	2441
OP3,TB10-14,OT20	ACR,AO	3363481	579923	775204	588607	76.95	2681729	951275.23	312,955.69	6	60	18275	37688	3688	2647
OP3,TB3,OT20	ACR,FO	3363481	579923	771594	588607	77.06	2681729	946138.52	297,914.25	12	66	19716	44819	3717	2544
OP3,TB5,OT20	ACR,FO	3363481	579923	775204	588607	76.95	2996604	1073894.17	374,639.63	6	76	26301	47467	5545	3969
OP3,TB6-9,OT20	ACR,FO	3363481	579923	775204	588607	76.95	2681729	966156.98	268,623.48	6	83	26662	55569	4751	3285
OP3,TB10,OT20	ACR,FO	3363481	579923	757974	588638	77.46	3147833	1017183.62	346,358.97	14	90	30890	61384	5334	3645
OP3,TB10-14,OT20	ACR,FO	3363481	579923	775204	588607	76.95	2681729	997821.38	259,320.90	6	93	28150	62734	5188	3645
OP3,TB3,OT20	ACRP,AO	3363481	579923	775204	588607	76.95	2681729	1037377.16	360,840.49	6	64	24487	41423	5755	4367
OP3,TB5,OT20	ACRP,AO	3363481	579923	775204	588607	76.95	2681729	990701.94	300,087.93	6	68	24371	41601	4727	3399
OP3,TB6-9,OT20	ACRP,AO	3363481	579923	775204	588607	76.95	2681729	1007460.51	304,894.16	6	68	24179	41667	4656	3315
OP3,TB10,OT20	ACRP,AO	3363481	579923	775204	588607	76.95	2681729	1006458.07	302,758.52	6	69	24524	42375	4397	3040
OP3,TB10-14,OT20	ACRP,AO	3363481	579923	775204	588607	76.95	2681729	1006458.07	302,758.52	6	69	24524	42375	5064	3689
OP3,TB3,OT20	ACRP,FO	3363481	579923	770958	588607	77.08	2681729	978211.37	305,159.07	15	68	21924	40546	3971	2736
OP3,TB5,OT20	ACRP,FO	3363481	579923	770691	588607	77.09	2681729	983974.95	281,025.28	19	78	24870	50614	5184	3682
OP3,TB6-9,OT20	ACRP,FO	3363481	579923	756870	588638	77.50	2681729	951153.23	275,538.34	20	77	25211	46866	5429	3922
OP3,TB10,OT20	ACRP,FO	3363481	579923	757974	588638	77.46	3147833	1017183.62	346,358.97	14	90	30890	61384	7318	5713
OP3,TB10-14,OT20	ACRP,FO	3363481	579923	775204	588607	76.95	2681729	1001570.57	273,324.21	6	88	28530	60029	7348	5519

Table 6.7 Performance of Optimum Solutions under different Tabu Setting (FIFO,DV3,OP1,OT20) for March

CKPO		Orignial Tardiness(S)	Cmax	Best (S)	Cmax in Best	Improvement (%)	Worst	AVG	STD	First Best	Total OT	Dealed Nodes	Partial Nodes	Total Time(S)	Time of Nodes (S)
OP Issue	TB design														
OP1,TB0,OT20	~	3363481	579923	844893	576809	74.88	1017254	910978.45	68888.87	13.00	20	3971	11317	799	591
OP1,TB3,OT20	ACR,AO	3363481	579923	844893	576809	74.88	1017254	910978.45	68888.87	13	20	3971	11317	795	588
OP1,TB5,OT20	ACR,AO	3363481	579923	844893	576809	74.88	1017254	910978.45	68888.87	13	20	3971	11317	828	627
OP1,TB6-9,OT20	ACR,AO	3363481	579923	844893	576809	74.88	1017254	910978.45	68888.87	13	20	3971	11317	801	612
OP1,TB10,OT20	ACR,AO	3363481	579923	844893	576809	74.88	1017254	910978.45	68888.87	13	20	3971	11317	815	617
OP1,TB10-14,OT20	ACR,AO	3363481	579923	844893	576809	74.88	1017254	910978.45	68888.87	13	20	3971	11317	805	596
OP1,TB3,OT20	ACR,FO	3363481	579923	844893	576809	74.88	1017254	910978.45	68888.87	13	20	3971	11317	806	614
OP1,TB5,OT20	ACR,FO	3363481	579923	772343	576809	77.04	1364548	931347.45	118215.55	13	29	5848	18142	1206	916
OP1,TB6-9,OT20	ACR,FO	3363481	579923	772409	588607	77.04	10932982	1695307.70	2536736.73	11	33	6498	20350	1385	1065
OP1,TB10,OT20	ACR,FO	3363481	579923	772409	588607	77.04	1407743	927257.39	108687.57	11	38	7868	23471	1685	1289
OP1,TB10-14,OT20	ACR,FO	3363481	579923	772409	588607	77.04	1017254	912064.54	69586.40	11	39	7716	24176	1627	1228
OP1,TB3,OT20	ACRP,AO	3363481	579923	772343	576809	77.04	1017254	890906.17	79791.60	14	23	5315	14337	1093	809
OP1,TB5,OT20	ACRP,AO	3363481	579923	772343	576809	77.04	1017254	889083.58	78546.87	14	24	5307	14677	1071	807
OP1,TB6-9,OT20	ACRP,AO	3363481	579923	772343	576809	77.04	1017254	889083.58	78546.87	14	24	5307	14677	1093	824
OP1,TB10,OT20	ACRP,AO	3363481	579923	772343	576809	77.04	1017254	889083.58	78546.87	14	24	5307	14677	1066	808
OP1,TB10-14,OT20	ACRP,AO	3363481	579923	772343	576809	77.04	1017254	889083.58	78546.87	14	24	5307	14677	1085	810
OP1,TB3,OT20	ACRP,FO	3363481	579923	772343	576809	77.04	1017254	888085.32	77016.00	14	25	5332	14959	1072	813
OP1,TB5,OT20	ACRP,FO	3363481	579923	772343	576809	77.04	1364548	925484.86	106033.46	13	35	7990	23201	1629	1238
OP1,TB6-9,OT20	ACRP,FO	3363481	579923	772409	588607	77.04	1407743	926300.97	109088.94	11	37	7676	22620	1636	1260
OP1,TB10,OT20	ACRP,FO	3363481	579923	772409	588607	77.04	1407743	927257.39	108687.57	11	38	7868	23471	1639	1249
OP1,TB10-14,OT20	ACRP,FO	3363481	579923	772409	588607	77.04	1017254	912064.54	69586.40	11	39	7716	24176	1628	1225

Table 6.8 Performance of Optimum Solutions under different Tabu Setting (FIFO,DV3,OP5,OT20) for March

CKPO		Orignial Tardiness(S)	Cmax	Best (S)	Cmax in Best	Improvement (%)	Worst	AVG	STD	First Best	Total OT	Dealed Nodes	Partial Nodes	Total Time(S)	Time of Nodes (S)
OP Issue	TB design														
OP5,TB0,OT20	~	3363481	579923	770475	588607	77.09	12406813	1356059.97	1411150.55	6	100	35647	73057	8492	6135
OP5,TB3,OT20	ACR,AO	3363481	579923	770475	588607	77.09	12406813	1356059.97	1411150.55	6	100	35647	73057	9697	7299
OP5,TB5,OT20	ACR,AO	3363481	579923	770475	588607	77.09	12406813	1356059.97	1411150.55	6	100	35647	73057	7343	4992
OP5,TB6-9,OT20	ACR,AO	3363481	579923	770475	2835007	77.09	12406813	1356059.97	1411150.55	6	100	35647	73057	6775	4694
OP5,TB10,OT20	ACR,AO	3363481	579923	770475	2835007	77.09	12406813	1356059.97	1411150.55	6	100	35647	73057	8486	6131
OP5,TB10-14,OT20	ACR,AO	3363481	579923	770475	2835007	77.09	12406813	1356059.97	1411150.55	6	100	35647	73057	8480	6107
OP5,TB3,OT20	ACR,FO	3363481	579923	770475	588607	77.09	12406813	1369719.94	1389589.32	6	103	38164	78683	10453	7905
OP5,TB5,OT20	ACR,FO	3363481	579923	770475	2835007	77.09	12406813	1349587.21	1376596.33	6	105	38847	80017	8392	5883
OP5,TB6-9,OT20	ACR,FO	3363481	579923	770475	2835007	77.09	12406813	1378928.23	1382969.64	6	109	41698	85075	9142	6402
OP5,TB10,OT20	ACR,FO	3363481	579923	770475	2835007	77.09	13672541	1779355.85	2420250.51	6	115	42841	85009	9651	6362
OP5,TB10-14,OT20	ACR,FO	3363481	579923	770475	2835007	77.09	13672541	1760669.08	2371301.94	6	120	45498	89934	13725	10651
OP5,TB3,OT20	ACRP,AO	3363481	579923	770475	588607	77.09	12406813	1342202.71	1385448.91	6	104	37211	76762	10142	7658
OP5,TB5,OT20	ACRP,AO	3363481	579923	770475	2835007	77.09	12406813	1342202.71	1385448.91	6	104	37211	76762	8100	5668
OP5,TB6-9,OT20	ACRP,AO	3363481	579923	770475	2835007	77.09	12406813	1341914.05	1366796.90	6	107	38470	79725	8481	5953
OP5,TB10,OT20	ACRP,AO	3363481	579923	770475	2835007	77.09	12406813	1340538.75	1360470.08	6	108	38962	80716	9236	6690
OP5,TB10-14,OT20	ACRP,AO	3363481	579923	770475	588607	77.09	12406813	1340538.75	1360470.08	6	108	38962	80716	9960	7406
OP5,TB3,OT20	ACRP,FO	3363481	579923	770475	588607	77.09	12406813	1355740.55	1365197.40	6	107	39728	82388	10356	7761
OP5,TB5,OT20	ACRP,FO	3363481	579923	770475	2835007	77.09	12406813	1336603.14	1352730.28	6	109	40411	83722	10404	7789
OP5,TB6-9,OT20	ACRP,FO	3363481	579923	770475	2835007	77.09	12406813	1339625.86	1341650.89	6	111	42304	86693	9875	7164
OP5,TB10,OT20	ACRP,FO	3363481	579923	770475	588607	77.09	13672541	1744306.49	2354795.77	6	122	45560	91677	11308	7854
OP5,TB10-14,OT20	ACRP,FO	3363481	579923	770475	588607	77.09	13672541	1719601.38	2301418.13	6	128	49741	97889	9952	6268

A. Total Tardiness

Figure 6.24 shows relevant performance of Best and Cmax in different tabu settings under a (OP3,OT20) setting for the March period. Cmax is calculated using work time (37 hours/week in this case study) and tardiness is calculated from the calendar days. The case study Cmax is still stable for approximately 160 hours under different tabu settings and Best is kept about 215 hours except three tabu settings, namely (TB10,ACR,FO), (TB6-9,ACRP,FO), and (TB10,ACRP,FO). When (TB6-9,ACRP,FO) is taken, the Best is achieved the minimum value about 210 hours and Improvement rises up to the maximum value 77.50%. (TB6-9,ACRP,FO) means TB=random (6,9) and TB combination chooses (ACRP,FO).When a tabu parameter is set as AO (OP=3), changing the number of TB has not affected Best. When a tabu parameter is set as FO and ACR (OP=3), Best is changed only under the situation of TB10. When a tabu parameter is set as FO and ACRP (OP=3), changing the number of TB relative obviously affects the Best. The relative minimum Best happens in (TB6-9,ACRP,FO) and (TB10,ACRP,FO), so the two tabu settings will be mainly adopted to evaluate the performance of CKPO of data over a six months' period.

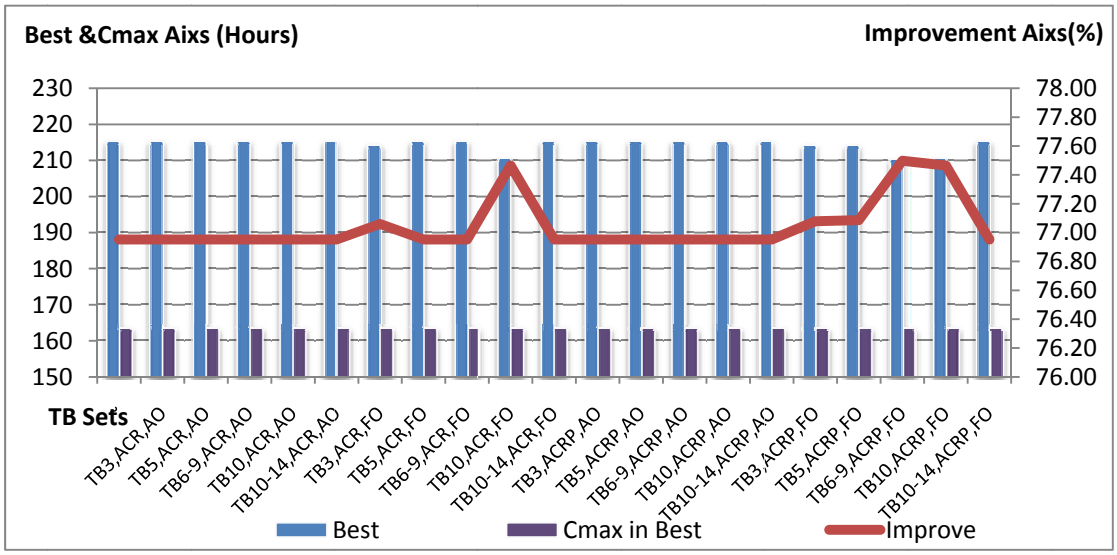


Figure 6.24 Best, Cmax and Improvement under Different Tabu Settings (OP3) for March

Figure 6.25 shows the total tardiness under different OPs. When OP is set to 1, different tabu settings make the maximum effects on Best. There are no changes on Best 214 hours approximately under different tabu settings (OP=5). The minimum Best is got in the set (OP3,TB6-9,ACRP,FO). In later Sections 6.4.3 and 6.5, OP3 setting will be adopted to evaluate the performance of the optimisation.

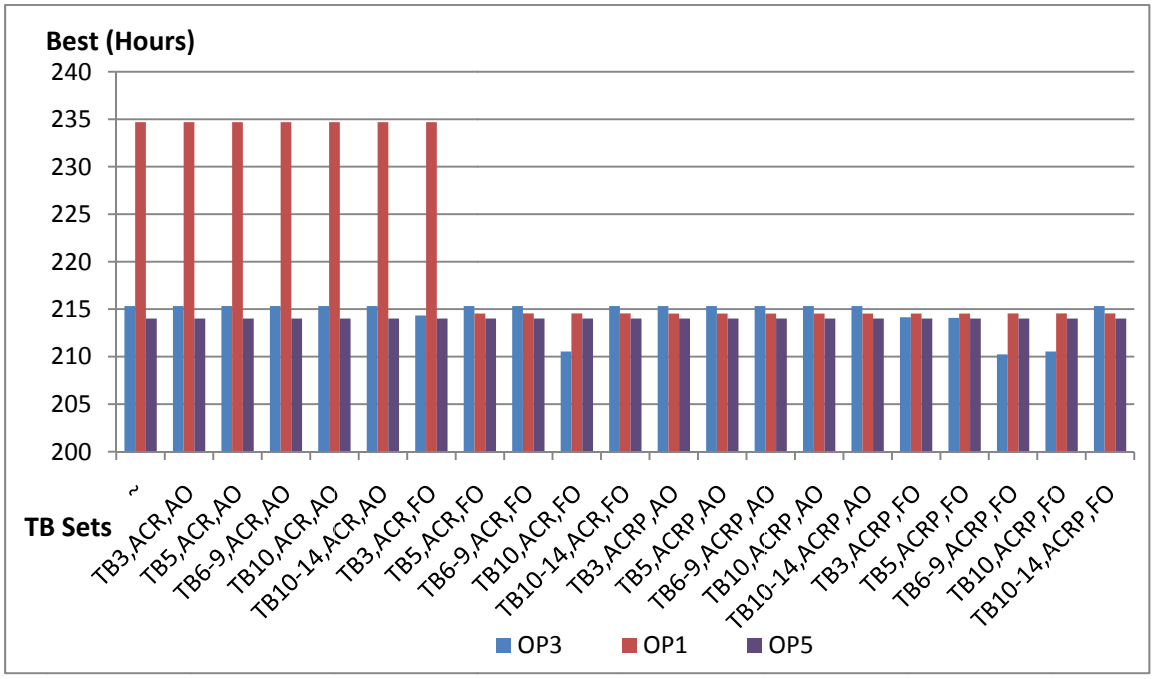


Figure 6.25 Best under different Tabu Settings for March

Figure 6.26 shows the changes of Worst, AVG and STD when setting different tabu settings (OP=3). There are obvious similar trends in the graph and the peak values happen at the same tabu settings.

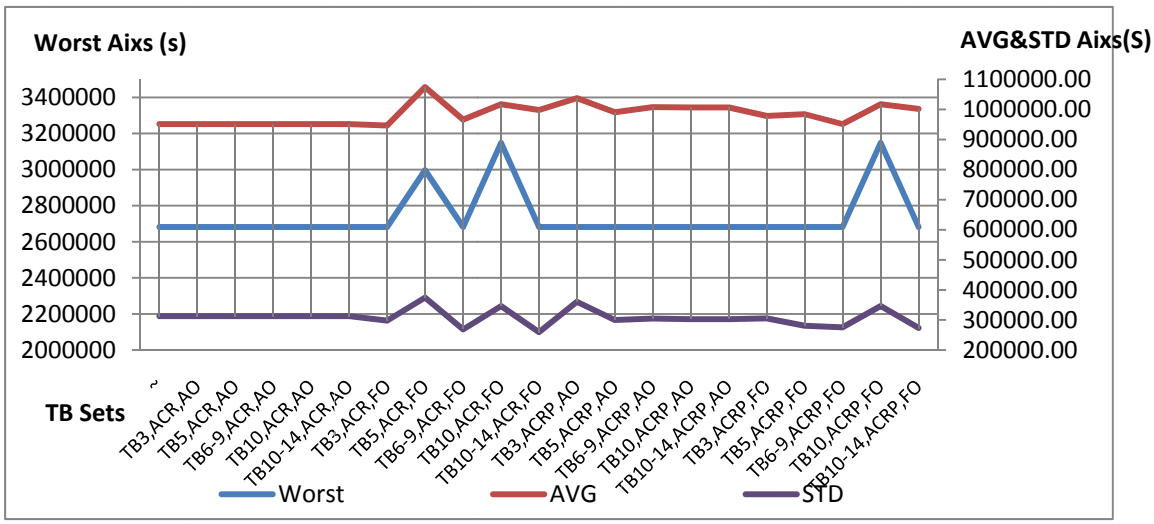


Figure 6.26 Worst, AVG and STD under different Tabu Settings (OP3) for March

B. CPU Time

20 similar computers at the University library are taken as the test platforms for implementing this comparison. Every PC embeds an Intel Core 2 Due E8500 3.2 Ghz CPU and 3.5G RAM and runs a Windows XP professional Service Pack 3. Figure 6.27 shows CPU Times under different tabu settings. Total OT rises higher in a FO setting than in an AO setting. This means a FO setting owns an obvious dynamic characteristic compared to an AO setting. In general, total OT is increased when TB rising. CPU Time basically rises following total OT.

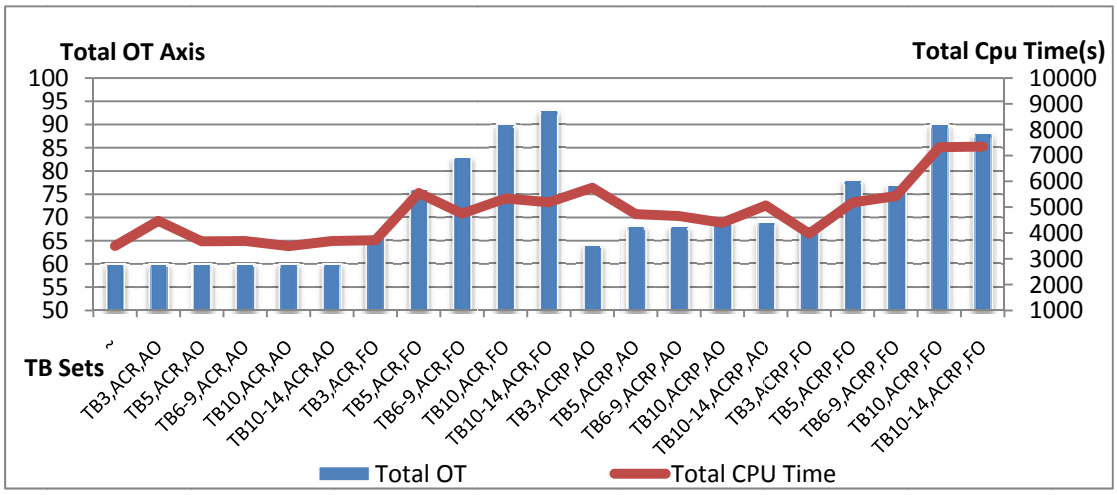


Figure 6.27 CPU Times and Total OTs under Different Tabu Settings (OP3) for March

C. Reschedule

Figure 6.28 illustrates the performance of two rescheduling methods of partial and CKPO (Improve=1-(Dealed Nodes/Partial Nodes)). CKPO reduces more than 40% the nodes which needs to be recalculated. The CKPO rescheduling process can only handle the effected operation nodes which are determined by relationships among operations (conjunctive and disjunctive constraints).

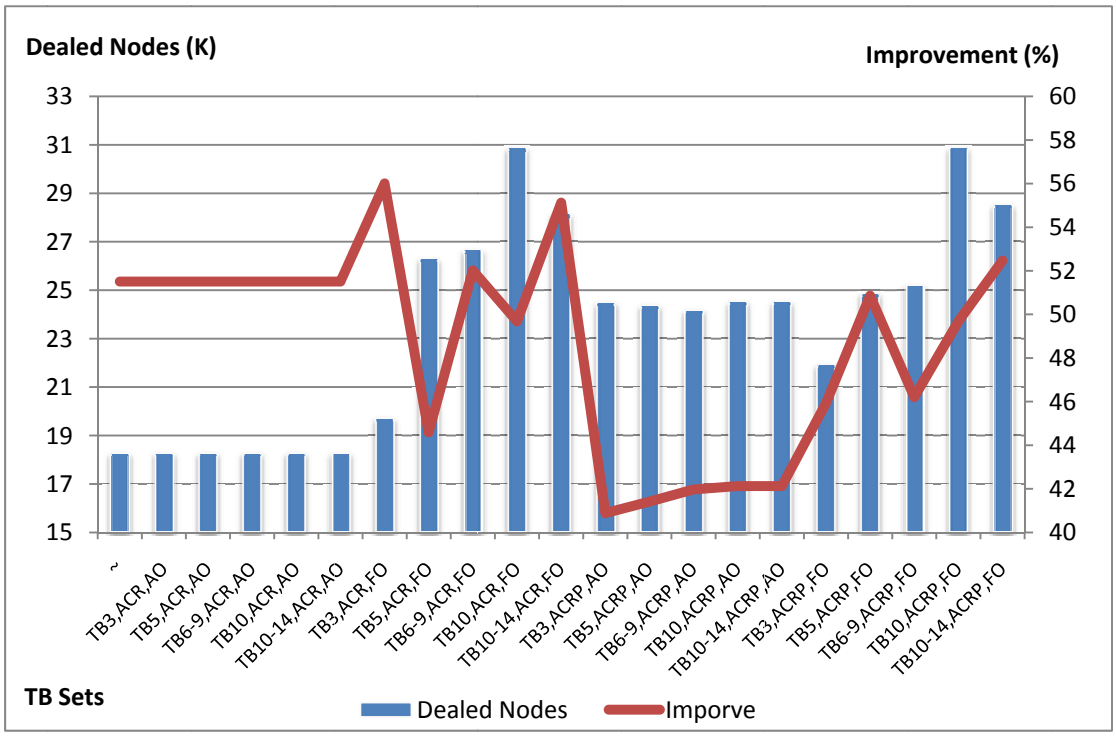


Figure 6.28 Dealed Nodes and Improve under different Tabu Settings (OP3) for March

6.4.3 CKPO of Different Months

The total tardiness is compared using six months' data and there are three types of initial solutions per month. The first initial schedule is obtained when the cell simulator using a (FIFO, DV3) setting. The cell simulator adopts a (EDD, DV3) setting and provides the second initial schedule. The third schedule is taken by adopting a (EDD, DV1) setting. Figure 6.29 illustrates the total tardiness over 6 months' period under different tabu settings and indicates 'good enough' schedules

can be achieved using this optimisation algorithm except for April. The graph mostly indicates there are improvements of more than 20% of the total tardiness (DV3) except for April. In Table 6.9, Total OT for April is less than 60, this means the optimisation process quits because all *swap* are all tabu. It is possible that the initial schedule is very close to the best solution. Therefore, this optimisation cannot improve it. Best using FIFO for April is worse than Original Tardiness because the OOP method of rescheduling may bring a negative effect to the optimisation and 'block' the search space where the 'best' schedules exist.

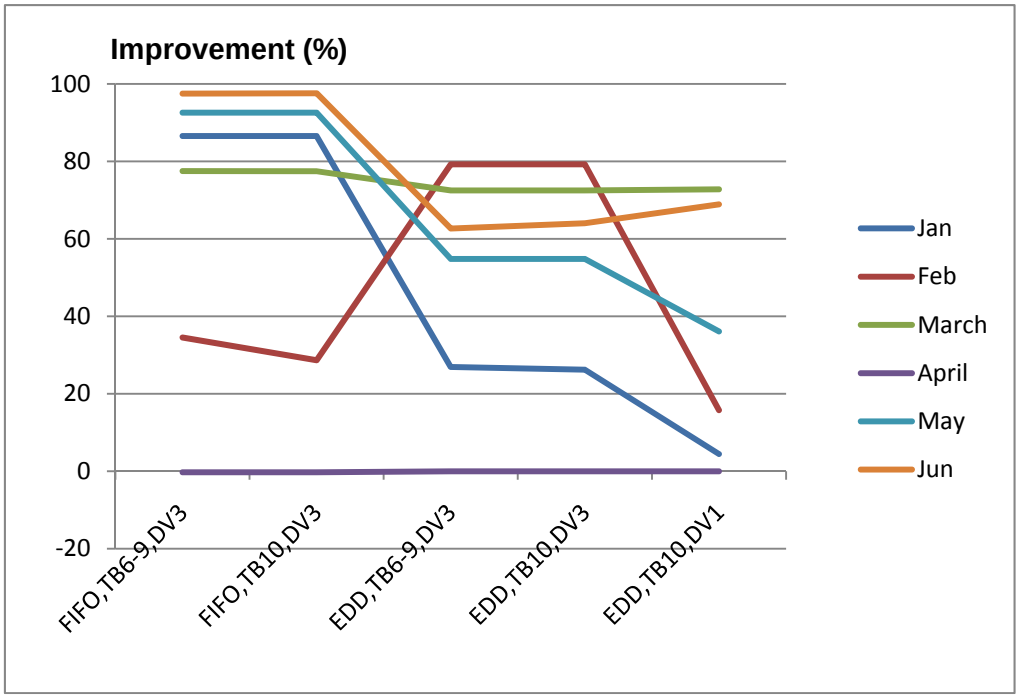


Figure 6.29 Improve of Optimisation under different Parameter Settings for Six Months

Best in (FIFO, DV3, TB10) is taken as a benchmark of each month and this benchmark was compared another settings as shown in Figure 6.30. The graph indicates no large fluctuations except for DV1 and February this shows the optimisation is able to provide the relative stable optimum value despite the initial solution by the cell simulator adopting FIFO or EDD. In general, the two initial

solutions produce a relative large gap of their total tardiness. Consequently, the proposed optimisation shows a relatively highly adaptable ability for the two initial solutions. The other factor illustrated shows the optimum schedules using DV3 have improved performance than using DV1, and this means the job splitting (LS) technique can contribute to the performance of this SME case. For February, these exists a huge difference between two initial schedules obtained from a cell simulator adopts EDD and FIFO rules respectively. This is a challenging issue to CKPO which only adopts a back forward method to choose optimisation nodes and reschedules effected operation nodes for saving the CPU time. In the future work, CKPO needs to be modified to effectively optimise this type of schedule.

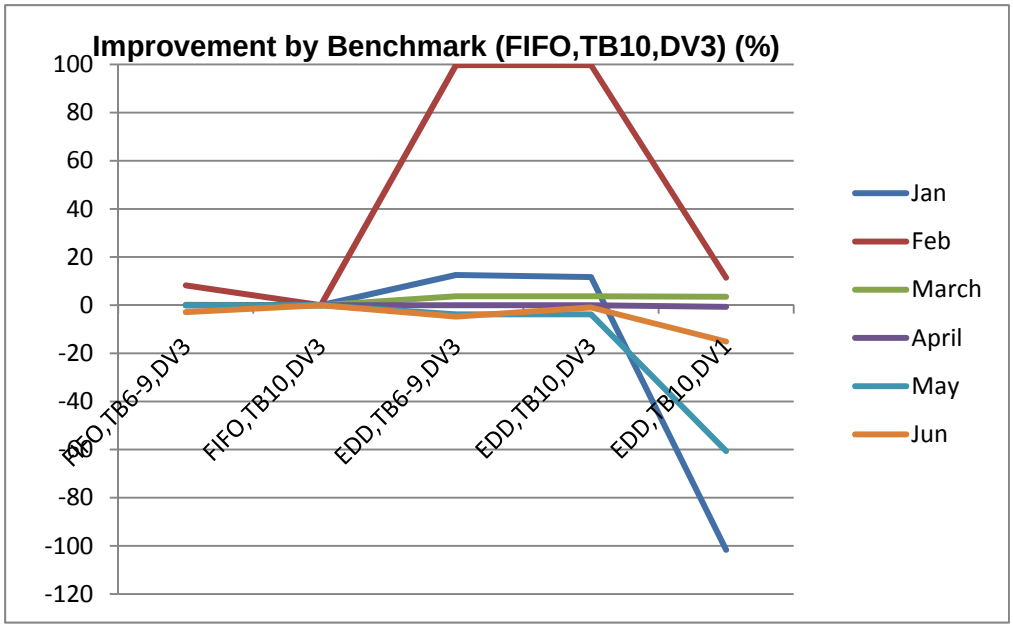


Figure 6.30 Comparisons of Optimisation under different Parameter Settings for Six Months

Table 6.9 Performance of Optimum Solutions under Different CKPO Setting (OP3,OT20) for Six Months I

Initial Schedule	CKPO		DOA	Original Tardiness (s)	Cmax	Best (S)	Cmax in Best	Improvement (%)	First Best	Total OT	Dealed Nodes	Partial Nodes
	OP Issue	TB design										
Jan(FIFO)	OP3,TB6-9,OT20	ACRP,FO	DV3	72537	542866	59508	542631	17.96	14	71	7464	39581
Jan(FIFO)	OP3,TB10,OT20	ACRP,FO	DV3	72537	542866	59508	542631	17.96	16	85	9763	65049
Jan(EDD)	OP3,TB6-9,OT20	ACRP,FO	DV3	71212	542866	52040	542982	26.92	20	71	10933	33110
Jan(EDD)	OP3,TB10,OT20	ACRP,FO	DV3	71212	542866	52546	542982	26.21	13	73	13331	35808
Jan(EDD)	OP3,TB6-9,OT20	ACRP,FO	DV1	125536	547204	119969	547204	4.43	11	99	11631	93858
Feb(FIFO)	OP3,TB6-9,OT20	ACRP,FO	DV3	407091744	996280	266570165	818823	34.52	17	132	109029	199361
Feb(FIFO)	OP3,TB10,OT20	ACRP,FO	DV3	407091744	996280	290409282	805199	28.66	16	152	130783	234022
Feb(EDD)	OP3,TB6-9,OT20	ACRP,FO	DV3	4242087	811838	881179	833502	79.23	4	113	74481	95914
Feb(EDD)	OP3,TB10,OT20	ACRP,FO	DV3	4242087	811838	881179	833502	79.23	4	115	75120	98040
Feb(EDD)	OP3,TB6-9,OT20	ACRP,FO	DV1	305389745	875363	257210800	808131	15.78	20	101	69018	157360
March(FIFO)	OP3,TB6-9,OT20	ACRP,FO	DV3	3363481	579923	756870	588638	77.50	20	77	25211	46866
March(FIFO)	OP3,TB10,OT20	ACRP,FO	DV3	3363481	579923	757974	588638	77.46	14	90	30890	61384
March(EDD)	OP3,TB6-9,OT20	ACRP,FO	DV3	2656143	588607	729846	588607	72.52	6	86	24546	56618
March(EDD)	OP3,TB10,OT20	ACRP,FO	DV3	2656143	588607	729846	588607	72.52	6	87	25834	57335
March(EDD)	OP3,TB6-9,OT20	ACRP,FO	DV1	2688819	576665	731502	576665	72.79	4	107	29545	77579

Table 6.9 Performance of Optimum Solutions under Different CKPO Setting (OP3,OT20) for Six Months II

Initial Schedule	CKPO		DOA	Original Tardiness (s)	Cmax	Best (S)	Cmax in Best	Improvement (%)	First Best	Total OT	Dealed Nodes	Partial Nodes
	OP Issue	TB design										
April(FIFO)	OP3,TB6-9,OT20(3)	ACRP,FO	DV3	577255	778535	578668	810086	-0.24	2	9	2153	4550
April(FIFO)	OP3,TB10,OT20	ACRP,FO	DV3	577255	778535	578668	810086	-0.24	2	9	2153	4550
April(EDD)	OP3,TB6-9,OT20(4)	ACRP,FO	DV3	578677	779319	578677	779319	0	2	11	2272	5368
April(EDD)	OP3,TB10,OT20(4)	ACRP,FO	DV3	578677	779319	578677	779319	0	2	11	2272	5368
April(EDD)	OP3,TB6-9,OT20	ACRP,FO	DV1	582698	781221	582698	781221	0.00	2	15	2860	7015
May(FIFO)	OP3,TB6-9,OT20	ACRP,FO	DV3	181590	633650	13453	635897	92.59	6	83	22180	50437
May(FIFO)	OP3,TB10,OT20	ACRP,FO	DV3	181590	633650	13453	635897	92.59	6	82	23418	48676
May(EDD)	OP3,TB6-9,OT20	ACRP,FO	DV3	30908	635897	13965	635897	54.82	13	47	8797	28468
May(EDD)	OP3,TB10,OT20	ACRP,FO	DV3	30908	635897	13965	635897	54.82	13	47	8797	28468
May(EDD)	OP3,TB6-9,OT20	ACRP,FO	DV1	33805	641700	21608	641700	36.08	14	68	9955	50974
Jun(FIFO)	OP3,TB6-9,OT20	ACRP,FO	DV3	963169	595991	112199	595991	88.35	11	83	39102	57376
Jun(FIFO)	OP3,TB10,OT20	ACRP,FO	DV3	963169	595991	109049	595991	88.68	19	75	34984	52064
Jun(EDD)	OP3,TB6-9,OT20	ACRP,FO	DV3	306058	595991	114218	595991	62.68	8	95	43081	65095
Jun(EDD)	OP3,TB10,OT20	ACRP,FO	DV3	306058	595991	110098	595991	64.03	18	106	48977	73350
Jun(EDD)	OP3,TB6-9,OT20	ACRP,FO	DV1	403839	602390	125565	602390	68.91	20	102	37102	78345

6.5 Improving Methods of Total Tardiness for the March Period

This section introduces methods, such as adding machines, training employees, decreasing the quantity of work orders to reduce the total tardiness for the March period. The first task is to analyse which work orders cause the major tardiness. The optimum solution with the minimum best tardiness is obtained with the setting (EDD, DV3, ACRP, FO, OP3, TB6-9,OT20) and is considered as a benchmark schedule for the analysis by decision makers. Table 6.10 shows the work orders which cause tardiness and the two work orders, 23170 and 24911 indicate more than two days tardiness. Therefore, reducing the tardiness from these two work orders will enable improvements in the performance of the case study for March.

Table 6.10 Tardiness of Work Orders of March

Work_ID	Tardiness (days)	Complete_Date	Due_date
23170	**3.03**	**3/14/2008 8:43**	**3/11/2008 8:00**
24911	**2.24**	**3/20/2008 13:46**	**3/18/2008 8:00**
22500	1.04	3/12/2008 8:55	3/11/2008 8:00
22163	2.53E-01	3/6/2008 14:04	3/6/2008 8:00
22084	2.51E-01	3/5/2008 14:01	3/5/2008 8:00
22074	2.51E-01	3/5/2008 14:01	3/5/2008 8:00
22073	2.49E-01	3/5/2008 13:59	3/5/2008 8:00
22376	2.49E-01	3/5/2008 13:58	3/5/2008 8:00
22061	2.48E-01	3/5/2008 13:56	3/5/2008 8:00
22719	1.93E-01	3/10/2008 12:38	3/10/2008 8:00
22131	1.44E-01	3/4/2008 11:26	3/4/2008 8:00
22580	1.21E-01	3/11/2008 10:53	3/11/2008 8:00
22291	5.78E-02	3/4/2008 9:23	3/4/2008 8:00
22433	5.07E-02	3/13/2008 9:13	3/13/2008 8:00
22201	2.79E-02	3/4/2008 8:40	3/4/2008 8:00
22508	2.00E-02	3/11/2008 8:28	3/11/2008 8:00
22424	8.66E-03	3/13/2008 8:12	3/13/2008 8:00
25526	6.56E-03	3/18/2008 8:09	3/18/2008 8:00
22371	4.20E-03	3/11/2008 8:06	3/11/2008 8:00
22403	2.94E-03	3/13/2008 8:04	3/13/2008 8:00

Table 6.11 shows the details of the operations of work orders 23170 and 24911 and there is no waiting time between two operations of the two work orders linked by a direct conjunctive arc. The only method is to reduce the operation time of the two work orders to achieve the aim of making their tardiness down. Their first operations occupy the most time of the total operation time, 116346/171399=67.88% in work order 23170 and 152160/159362=95.48% in work order 24911. Therefore, reducing the operation time of their first operations is a critical issue to be explored in the following Section 6.5.1. The relative ECRs have been occupied by the two first operations after checking the relative tables and their OCR=ECR expresses no Conflicted Disjunctive Arc (CDA) arrows to the two operations according to **Rule 4.4**. Referring to **Rule 4.6**, the relevant critical paths only take the conjunctive arcs arrows to the two operations. Consequently, the optimisation algorithm is an effective way to provide the best CRPs to the two operations and their operation times cannot be reduced by altering the schedules (optimisations). The two methods, namely adding CRPs and reducing the quantities of orders, are introduced to achieve this aim.

Table 6.11 Operation Nodes of Work Order 23170 and 24911

Work_ID	Seq	Start_Time	End_Time	Operation Time (S)	Ma_ID	Em_ID	Qty
23170	**1**	**1217234**	**1333580**	**116346**	**2**	**47**	**55**
23170	2	1333580	1344250	10670	9	10	47
23170	2	1333580	1344250	10670	8	19	8
23170	3	1344250	1374463	30213	52	24	55
23170	4	1374463	1382858	8395	24	67	11
23170	4	1374463	1382858	8395	28	24	44
23170	5	1382858	1388215	5357	37	80	8
23170	5	1382858	1388215	5357	38	62	7
23170	5	1382858	1388215	5357	36	30	40
23170	6	1388215	1388522	307	20	33	55
23170	7	1388522	1388633	111	45	33	55
24911	**1**	**1355400**	**1507560**	**152160**	**33**	**25**	**100**
24911	2	1507560	1514762	7202	20	9	100

6.5.1 Investment

The first scenario is to invest in adding a CNC_Punch machine (Ma_ID=2), training Em_ID 76 for completing the first operation of the work order 23170 and training Em_ID 25 for work order 24911. Table 6.12 shows the two initial solutions and their optimum solutions. The minimum best is 357384 seconds (4 days) and this method gives a 51% improvement compared to the previous 729846 seconds tardiness (8.5 days) without the investment.

Table 6.12 Initial and Optimum Solutions under an Investment Scenario

CKPO&DOA	Original Tardiness(S)	Cmax(S)	Best (S)	Cmax in Best(S)
FIFO,TB6-9,DV3	3363460	588616	567928	576797
FIFO,TB10,DV3	3363460	588616	545744	576797
EDD,TB6-9,DV3	2655010	588607	**357384**	588607
EDD,TB10,DV3	2655010	588607	359205	588616

Table 6.13 shows the situation of tardiness of work orders after taking an investment under a setting (EDD,DV3,OP3,TB6-9,OT20). Table 6.13 also indicates the tardiness of work order 23170 is successful reduced from 3 days to 1 day and no tardiness existing in work order 24911. The work order 22500 is almost unchanged as no improving method was adopted for this order.

Table 6.13 Tardiness of Work Orders under an Investment Scenario

Work_ID	Tardiness(days)	Complete_Date	Due_date
23170	**1.05**	**3/12/2008 9:14**	**3/11/2008 8:00**
22500	**1.02**	**3/12/2008 8:26**	**3/11/2008 8:00**
22084	2.51E-01	3/5/2008 14:01	3/5/2008 8:00
22074	2.51E-01	3/5/2008 14:01	3/5/2008 8:00
22073	2.49E-01	3/5/2008 13:59	3/5/2008 8:00
22376	2.49E-01	3/5/2008 13:58	3/5/2008 8:00
22061	2.48E-01	3/5/2008 13:56	3/5/2008 8:00
22719	1.89E-01	3/10/2008 12:31	3/10/2008 8:00
22163	1.86E-01	3/6/2008 12:27	3/6/2008 8:00
22131	1.44E-01	3/4/2008 11:26	3/4/2008 8:00
22580	1.26E-01	3/11/2008 11:01	3/11/2008 8:00
22291	5.78E-02	3/4/2008 9:23	3/4/2008 8:00
22433	5.05E-02	3/13/2008 9:12	3/13/2008 8:00
22201	2.79E-02	3/4/2008 8:40	3/4/2008 8:00
22508	1.77E-02	3/11/2008 8:25	3/11/2008 8:00
22424	8.09E-03	3/13/2008 8:11	3/13/2008 8:00
25526	6.56E-03	3/18/2008 8:09	3/18/2008 8:00
22403	2.94E-03	3/13/2008 8:04	3/13/2008 8:00
22371	2.48E-03	3/11/2008 8:03	3/11/2008 8:00

Table 6.14 shows the performance of optimum solutions based on an investment scenario of different CKPO setting and the best solution is occurred in a (OP5,TB10,OT20) setting.

Table 6.14 Performance of Optimum Solutions Based on a Investment Scenario under different CKPO setting (FIFO,DV3,OT20) for March

CKPO		Original Tardiness(S)	Cmax	Best (S)	Cmax in Best	Improvement (%)	Worst	AVG	STD	First Best	Total OP Turn	Dealed Nodes	Partial Nodes
OP Issue	TB design												
OP3,TB5,OT20	ACR,FO	3363460	588616	573758	588616	82.94	2681737	808856.49	355355.78	18	69	21189	49312
OP3,TB6-9,OT20	ACR,FO	3363460	588616	564839	588616	83.21	2681737	813570.97	349885.67	19	74	23745	53559
OP3,TB10,OT20	ACR,FO	3363460	588616	562987	588616	83.26	2681737	809621.19	343637.49	20	78	25342	56735
OP3,TB5,OT20	ACRP,FO	3363460	588616	573758	588616	82.94	2681737	808856.49	355355.78	18	69	21189	49312
OP3,TB6-9,OT20	ACRP,FO	3363460	588616	562987	588616	83.26	2681737	811599.14	346589.12	20	76	24876	55189
OP3,TB10,OT20	ACRP,FO	3363460	588616	562987	588616	83.26	2681737	809621.19	343637.49	20	78	25342	56735
OP5,TB3,OT20	ACRP,FO	3363460	588616	567928	576797	83.11	2698472	862790.88	403951.76	18	109	38674	87532
OP5,TB5,OT20	ACR,FO	3363460	588616	567928	576797	83.11	2698472	870803.62	401882.86	17	117	41164	95591
OP5,TB6-9,OT20	ACR,FO	3363460	588616	567928	576797	83.11	2698472	912107.52	418620.04	17	131	49745	109083
OP5,TB10,OT20	ACR,FO	3363460	588616	545744	576797	83.77	5294416	903456.55	541656.95	20	132	46626	101972
OP5,TB5,OT20	ACRP,FO	3363460	588616	567928	576797	83.11	2698472	870307.03	399203.09	17	119	41618	97358
OP5,TB6-9,OT20	ACRP,FO	3363460	588616	567928	576797	83.11	2698472	909002.62	417669.67	17	131	49284	108603
OP5,TB10,OT20	ACRP,FO	3363460	588616	545744	576797	83.77	5294416	901602.59	540024.75	20	133	46701	103011

6.5.2 Order Control

The second scenario is to reduce the quantity of the work order 231670 from 55 to 20 and the quantity of the work order 24911 from 100 to 70 based on observation. Table 6.15 shows the two initial solutions and their optimum solutions. The minimum best is 210716 seconds (2.5 days) and this method gets 71% improvement compared to the previous 729846 second tardiness (8.5 days) without reducing the orders.

Table 6.15 Initial and Optimum Solutions under an Order Control Scenario

CKPO&DOA	Original Tardiness(S)	Cmax (S)	Best (S)	Cmax in Best (S)	Improvement (%)	Worst	AVG	STD	First Best	Total OT
FIFO,TB6-9, DV3	2043570	588607	**210716**	588493	89.69	2845629	457013.99	405746.74	19	80
FIFO,TB10, DV3	2043570	588607	**210716**	588493	89.69	2845629	456352.98	400877.02	19	82
EDD,TB6-9, DV3	2029340	588607	211771	588607	89.56	2642964	421108.85	392848.74	19	78
EDD,TB10, DV3	2029340	588607	211771	588607	89.56	2642964	427508.58	390915.57	19	80

Table 6.16 shows the situation of tardiness of work orders after reducing the quantities of the orders under a setting (FIFO,DV3,OP3,TB6-9,OT20). It was found there was no tardiness for work order 23170 and work order 24911 respectively. Consequently, if 35 products for work order 23170 and 30 products for work order 24911 are in stock, the requirements of the customers can be met.

Table 6.16 Tardiness of Work Orders under an Order Control Scenario

Work_ID	Tardiness (days)	Complete_Date	Due_date
22061	1.10	3/6/2008 10:25	3/5/2008 8:00
22500	3.62E-01	3/11/2008 16:41	3/11/2008 8:00
22580	2.31E-01	3/11/2008 13:32	3/11/2008 8:00
22163	2.06E-01	3/6/2008 12:56	3/6/2008 8:00
22719	1.74E-01	3/10/2008 12:09	3/10/2008 8:00
22131	1.43E-01	3/4/2008 11:25	3/4/2008 8:00
22291	5.69E-02	3/4/2008 9:21	3/4/2008 8:00
22433	5.20E-02	3/13/2008 9:14	3/13/2008 8:00
22201	2.76E-02	3/4/2008 8:39	3/4/2008 8:00
22073	2.17E-02	3/5/2008 8:31	3/5/2008 8:00
22084	2.14E-02	3/5/2008 8:30	3/5/2008 8:00
22508	1.86E-02	3/11/2008 8:26	3/11/2008 8:00
22424	8.02E-03	3/13/2008 8:11	3/13/2008 8:00
25526	6.56E-03	3/18/2008 8:09	3/18/2008 8:00
22371	4.34E-03	3/11/2008 8:06	3/11/2008 8:00
22403	2.94E-03	3/13/2008 8:04	3/13/2008 8:00
22074	2.07E-03	3/5/2008 8:02	3/5/2008 8:00

6.5.3 Improvement

Table 6.17 shows the improvements of Best under two scenarios compared to the original situation and the improvement=(1-Best(what scenario)/Best(original)). This clearly indicates the performance of taking the *Order Control* method is better than the one of taking the *Investment* method.

Table 6.17 Improvements of Best under Two Scenarios

Scenario	Original		Investment			Order Control		
CKPO&DOA	Tardiness(s)	Best(s)	Tardiness(s)	Best(s)	Improvement (%)	Tardiness(s)	Best(s)	Improvement (%)
FIFO,TB6-9,DV3	3363481	756870	3363460	567928	24.96	2043570	210716	72.16
FIFO,TB10,DV3	3363481	757974	3363460	545744	28.00	2043570	210716	72.20
EDD,TB6-9,DV3	2656143	729846	2655010	357384	51.03	2029340	211771	70.98
EDD,TB10,DV3	2656143	729846	2655010	359205	50.78	2029340	211771	70.98

6.6 Compared to Other Optimisation Algorithms

The two-phase method integrated CKPO and DOA algorithms is an optimisation algorithm to solve the Jm|STsi,b with LS. To the best of our knowledge from the literature, there is not an existing optimisation algorithm implemented for this type of JSSP. The proposed CKPO and DOA algorithm is able to obtain a considerable stable 'good enough' schedule for the UK-based case study under different parameter settings. Table 6.18 shows the comparison and analysis among the proposed CKPO and DOA algorithm and several popular optimisation algorithms. It is clearly shown the proposed algorithm can be implemented for a more flexible application (Jm| STsi,b| Σ Tj with LS) than the other algorithms.

6.7 Compared to Other Existing Frameworks

Based on Table 2.1 and as demonstrated in Section 2.6, Table 6.19 is developed to compare the features of the proposed framework against existing frameworks on factors related to the adaptive framework.

Two case studies indicate the possible applications of this proposed framework in MTS and MTO strategies. Most frameworks normally only focus on one particular SC, e.g., MTS. Although Umeda's framework has developed solutions for different SCs with four product strategies, he has not supported them with real DMSs based on case studies.

Routing flexibility configuration method consisting of semi-flexibility and total flexibility configurations is developed in the proposed framework and described in Section 3.3.6. The proposed cooperation method integrated route information stored in the database and dispatching rules is used for the total routing flexible configuration. In the existing frameworks, only Wadhwa et al.(2008) briefly explored the issue of flexibility configuration of SCs and outlined a simple solution for a basic SC with

manufactures, distributors, warehouses, and retailers.

The multi-layer design, which covers hierarchical network, hierarchical structure, simulation block, and cell simulator, is embedded in the proposed framework. This design is a requirement of modularisation and also provides a foundation of building a cooperation mechanism of DMSs in a SC. The proposed hierarchical network has extended the range of the Pundoor's hierarchical structure from an internal SC to an enterprise SC.

Another novel aspect of the proposed framework is the proposal of the concept of optimisation block to provide the optimum solution. In the gift SC case study, a simple optimisation algorithm, which does not require a special designed optimisation block, is included in the Rule Layer of a cell simulator. The optimisation block includes a two-phase CKPO& DOA optimisation algorithm to deal with complex optimisation problems in the UK-based SME case study. The two case studies demonstrate an optimisation capability of the proposed framework. In the existing frameworks, only Mönch et al. (2008) implemented several optimisation algorithms, such as GA and shifting bottleneck heuristics, however the adaptive designing structure of optimisation methods was not explored.

To our knowledge, the proposed framework is novel and is specially designed for cooperating different simulation-based DMSs implemented in different SMEs of a SC. In the thesis, the cooperation mechanism of simulation blocks internal and external enterprise is explored and developed. Different SMEs are able to design their unique DMSs according to the architecture of the proposed framework and the structure of a simulation block, a cell simulator, and an optimisation block. This enables SMEs to provide a secure requesting service or adopt an account management directly giving authorised rights to their users of the DMSs. This design will achieve the aim of exchanging internal and external data for cooperation from different DMSs in a

complicated SC.

The remaining three features, application, data warehouse technology and performance measures, are shown in Table 6.19. The two case studies demonstrated the proposed framework can be used for order, inventory, production scheduling, and demand planning management. Normalisation technology was implemented in designing a database for the UK-based case study. In the performance measurement, different costs based on the EOQ algorithm, such as holding costs, setup costs, and shipping costs, were discussed in the gift SC case study. Further measurements including lead time, waiting time, resource utilisation, and total tardiness, etc. were explored in the UK-based SME case study. These features and their advantages are compared against existing frameworks and outlined in Table 6.19.

The proposed framework is able to provide a simulation and optimisation service based on the completed consideration of constraints of a SC. The framework also integrates the two characteristics, namely routing flexible configuration and hierarchical structure, to achieve a highly adaptive, modular, and reusability. Compared to other existing frameworks, it provides significant advantages and is able to provide 'good enough' solutions for SCs adopting MTO and MTS production strategies.

6.8 Conclusions

This chapter validates two types of simulation-based DMSs using two case studies. A simulation-based DMS is developed based on a gift MTS SC obtained from the literature. The proposed replenishment algorithm (including a cost function, namely EOQ to deal with the issue associated with multiple products sharing common resources) embedded in the cell simulator is evaluated under various scenarios. A simulation-based DMS consists of a schedule-driven cell simulator, which is designed for the UK-based case study, a support database, and an optimisation block which is programmed to reduce the total tardiness. Data from a UK-based case study collected

over six months' period was successfully implemented to validate the agility of the cell simulator and evaluate the features of integrated algorithms of CKPO and DOA according to various settings. The two-phase method integrated CKPO and DOA algorithms has proved to provide a 'good enough' schedule for the UK-based case study compared to the original schedule adopting FIFO or EDD rules. The two methods, namely investment and order control, were studied to reduce the total tardiness based on the data for the March period. Finally, a relative complete comparison between CKPO and DOA and several popular optimisation algorithms for JSSP are discussed.

Table 6.18 Popular Optimisation Algorithms for JSSP

Optimisation Algorithm	Methodologies	Novel Issues	Limitations	Applications
Lin and Jeng (2004)	Dynamic programming	A three-phases consisting of sequencing, dispatching and batching heuristic algorithms	Not considering unrelated parallel machine	Pm| STsi,b|Lmax and Pm| STsi,b|Tui with LS
KIM et al.(2004)	Dispatching rules	A two-phases methods consisting of SPT and EDD for dispatching and three rules for splitting jobs	Not considering unrelated parallel machine	Pm with LS
Chan (2008)	GA, Dispatching rules	A two-step method of hybrid GA is developed for solve AJSP	Not considering unrelated parallel machine	Jm with LS
Low(2005)	MIP, simulated annealing	MIP only handling FSMP less than four jobs and heuristics based on SA developed for more jobs	Not considering LS	Fm| STsi,b
Huang (2010)	ACO	Four different LS methods and multi-objectives function	Not considering unrelated parallel machine	Pm| STsi,b with LS
El-Bouri et al.(2007)	TS,simulated annealing	Four heuristic variants combined of several popular optimisation algorithm make it more efficient	Not considering LS	Jm| STsi,b
Li et al. (2010)	TS	An effective HTSA to optimise three objectives	Not considering LS	Jm| STsi,b
CKPO and DOA	TS	A two-phases method of CKPO for sequencing and DOA for dispatching for JSSP with LS (real case study with about 500 jobs and more than 1000 CRPs)	Not considering sequence-dependent batch or family setup time and removal time	Jm| STsi,b|Σ Tj with LS

Table 6.19 Features of Current Existing Simulation-based Decision Making Frameworks

Simulation Framework	Product Strategy	Multi-Layers	Routing Flexibility	Optimisation Capability	Cooperation Method	Application and case studies	Datawarehouse Technology	Performance Measurement
Proposed	MTS,MTO	A network consisting of three-layers hierarchical structure	Routing Flexibility Configuration	Tabu, CKPO, DOA	Simulation messages among simulation blocks	Manufacturing SC (gift SC and job shop)	Normalization Theory	Cmax, Total Tardiness, Costs
Umeda (1998,2004,2007)	MTS,MTO,ATO,ETO	Hierarchical structure with four data models	Null	Null	Null	Manufacturing SC	Null	Inventory level, lead time
Wadhwa(2008)	MTS	One layer	Routing Flexibility Configuration	Null	Null	Manufacturing SC	Null	Costs, lead time
Mönch (2003,2008)	Null	Control model and data model	Null	GA, shifting bottleneck heuristic	Message bus	Semiconductor factory	Null	The weighted tardiness
Wu (2002)	MTS	Three models	Null	Null	VBA linked to Excel	Inventory management	Null	Quantity
Lee (2002)	MTS	One layer	Null	Null	Null	Order management	Null	Inventory level
Pundoor (2006)	Null	Hierarchical structure with three layers	Null	Null	VBA linked to Excel	Product Scheduling	Null	Costs

Chapter 7 Conclusions and Future Works

7.1 Introduction

This chapter summarises the conclusions drawn from this research study. The main research contributions are highlighted and the current situation of the simulation-based DMS discussed. Future research works are outlined and discussed in this chapter.

7.2 Contributions

There are numbers of contributions in proposing and developing the adaptive simulation-based decision-making framework for SMEs in SCs and these contributions are outlined as follows:

- **Proposing a new adaptive simulation decision-making framework for SMEs in SCs.** Flexibility configurations and hierarchical structure are necessary characteristics of the proposed framework. In the thesis, a hierarchical network structure with *data filter and convert* layers is proposed to flexibly define the details for a simulation framework. A hierarchical network structure is developed from the original concept of a hierarchical structure in order to extend the coverage of this framework from an internal SC to a complete SC. This structure shows the details of SC, factory levels, work centres, and machines. A *data filter and convert* layer communicates with different layers and converts the relevant data into a required format. The aim of this flexibility configuration can be achieved by a dispatching method for semi-flexibility configurations and providing a route method for total-flexibility configurations.
- **Designing a simulation block consisting of optimisation blocks, cell**

simulators, support databases, etc. The proposed simulation block is a basic unit of the simulation framework for SMEs in SCs. In designing a simulation block, SMEs are able to take their characteristics, such as limit budgets and professional employees, into account to develop a suitable simulation block. A simulation block is a relative independent unit which links the outside of the simulation system using *data filter and convert* layers to limit outside influence.

- **Developing a optimisation block consisting of the proposed CKPO and DOA algorithms for Jm|STsi,b|ΣT with LS of JSSP.** Six months' data from a UK-based case study was used to evaluate the performance of the optimisation block under different parameter settings. This indicated which this optimisation block is able to provide a stable 'good enough' schedule for the decision makers under most parameters settings.
- **Conducting two types of simulation blocks (DMSs) to validate the proposed simulation framework.** The two DMSs are successfully developed based on two case studies which adopted two common production strategies: MTO and MTS respectively. A relational database is designed for UK-based SME case study with hundreds of products and the functions of the simulation blocks are verified under different scenarios.
- **Implementing a novel simulation worldview based on the *three phase* approach**. This worldview is based on the identified events in a SC, such as *order coming* events and *common resources released* events. The *cycle of event* which is a different control mechanism of *three phase* is developed. The details of the control mechanisms of *order coming* and *common resources released* events are outlined.
- **Applying 4P's closed-loop control and Multi-layer control concepts in designing the simulation block.** Multi-layer control concept has been used to classify the requirements for different people and to determine the

practical layers and methods of different functions of the simulation system.

7.3 Conclusions

The globalization of the world's economy is pushing companies to establish efficient and effective SCs to bring competitive advantages, such as Wal-Mart Stores, Toyota Motor and Dell (Choy et al., 2008). An inappropriately designed SC caused a U.S. food industry to lose about $30 bn. annually (Huemer, 2006). Dell inc. gained a 160% return on its invested capital and reduced its lead-time to 5 days by establishing a successfully SCM (Gunasekaran and Ngai, 2009). Toyota Motor raised 17.2% net income in 2006 by the implementation of a SCM. Consequently, many SMEs which are vital to the UK economy identified in Chapter 1 are trying to improve their SCM using IT. The main aim of the research work is to develop a flexible and adaptive simulation-based DMS for SMEs in SCs.

Chapter 2 reviews relevant methodologies of developing a simulation-based decision-making framework for SMEs. The proposed simulation system belongs to discrete-event model system and the relevant knowledge of a discrete-event system is the focused topic. A new simulation framework with flexibility configurations and hierarchical network structures is required for designing an adaptive simulation-based decision-making framework for SMEs from reviewing current simulation frameworks. The *three phase* approach of the simulation worldview as proved to be useful but still needs to be modified for the proposed simulation. WITNESS simulation software has proved to be one of the most suitable software development package after comparing 37 popular simulation software packages. Finally, the optimisation algorithms of JSSP are discussed in order to develop a DMS for UK-based case study. However, from the literature and to the best of our knowledge there are no existing optimisation algorithms suitable for Jm|STsi,b with LS (UK-based case study). Consequently, with regard to

the UK-based cast study, there is a need for a new optimisation algorithm.

Chapter 3 proposed an adaptive simulation-based decision-making framework for SMEs in SCs. The simulation framework with a flexible configuration and hierarchical network structure can be successfully developed based on a new concept of a simulation block. The hierarchical network is developed from the concept of a hierarchical structure and extends the proposed framework from an internal SC to a complete SC. The flexible configuration is divided into two types of configurations, namely semi-flexibility and total-flexibility configuration. A semi-flexibility configuration can be achieved by a dispatching rule, such as DOA. The flexible routes stored in the database and dispatching rules are used to develop a total-flexibility configuration. A simulation block may consist of a cell simulator, support database and an optimisation block. A cell simulator is able to provide initial solutions for future optimisations by an optimisation block. The adaptive structure of a cell simulator includes decision dataset layer, rule layer, main core, attribute layer, entities layer, etc. It is proposed to guide the developing simulation models for different applications. Also other new concept such as 4 Ps' closed-loop control, multi-layer control, a proposed simulation worldview, etc. are utilised in this framework for effectively meeting the requirements of SMEs.

Chapter 4 proposes and implements an optimisation block including proposed CKPO and DOA algorithms which is developed to reduce Cmax or total tardiness of Jm|STsi,b with LS. An optimisation block consists of optimisation module, performance analysis module, and related data module and is able to provide optimum solutions for a particular problem. For the situation of Jm|STsi,b with LS, a two-phase optimisation method including CKPO and DOA is developed. A CKPO algorithm is able to optimise the schedule on the top layer for Jm|STsi,b, which assigns suitable jobs to suitable resources at suitable time points. Then a DOA algorithm is adopted to deal with a LS issue which means appropriate splitting the assigned jobs and to place

them into the assigned resources on the bottom layer. An improved disjunctive graph representation is used to express schedules of Jm|STsi,b with LS and a OOP method is proposed to generate a protection cover to avoid deadlock in rescheduling process of the CKPO.

Chapter 5 develops the processes of designing a cell simulator. Two event controlling mechanisms for two identified events, namely *order coming* and *common resource released* events, are developed. A simulation worldview with a cooperation function for two events mechanisms is investigated for building a controlling mechanism of a cell simulator. The conceptual simulation model is built and inputs, outputs are identified for UK-based case study. Experimental factors, model scope, active cycle and model simplification for this case study are discussed respectively. Finally, a verification program is programmed to check the feasibility of the optimum solutions generated by the optimisation block outlined in Chapter 4 according to two mathematic constraints of JSSP, namely conjunctive constraint and disjunctive constraint, and the generic logics.

Chapter 6 uses two case studies to validate two types of simulation-based DMSs of SMEs and demonstrates the framework is suitable for the two popular product strategies MTO and MTS. A stock-driven simulation-based DMS with the proposed replenishment algorithm is successfully developed for a gift SC with a MTS strategy. Under different scenarios, performance of the proposed replenishment algorithm is studied by analysing results from the cell simulator in this DMS. A schedule-driven simulation-based DMS is successfully developed for UK-based case study with a MTO strategy. Six months' data obtained from the company are linked from the simple ERP system and used to verify the functions of this DMS and evaluate the performance of the optimisation block. The proposed system's agility is validated after running the simulations based on various scenarios for different applications. It was demonstrated which the integrated method between CKPO and DOA is able to provide a stable 'good

enough' optimum solution after testing its behaviour under different parameter settings. Two improvement methods to reduce the total tardiness for a March data period are investigated and compared. Finally, several popular optimisation algorithms of JSSP and the proposed CKPO and DOA are compared and analysed to highlight the advantages of the proposed algorithms.

7.4 Future Research

There are several research aspects to my future research works for establishing a more flexible and efficient simulation-based DMS for SMEs in SCs and extending its application from the manufacturing sector to other sectors such as hospital, banking services. These aspects are outlined as follows:

- **A knowledge database:** A knowledge database of optimisation algorithms in the simulation block needs to be developed. In order to achieve different aims, the components such as cost functions, support data structure, and optimisation algorithm are able to be chosen and constructed for an optimisation block to provide optimum solutions. Different reasoning methods about case-base reasoning and rule-based reasoning need to be reviewed and compared. The classification methods, characteristics identifications of the simulation models and problems will be developed.
- **Interface**: The interfaces between popular information systems, such as ERP, MRPII, CRM, etc. and simulation-based DMS will be explored and the standard of these interfaces will be defined. The synchronization and cooperation mechanisms between these systems will be discussed and developed.
- **Applications**: More practical case studies will be used to extend the proposed framework from a manufacturing field to other services such as hospitals, banks, government, etc. Most of these service cases are able to

be imitated by discrete-event simulation models. Consequently, their structure of cell simulators are similar to that in the manufacturing field. The other components of a simulation block may be modified and redeveloped.

- **Internet of Things**: The Internet of Things will be applied into the framework. Real-time collection devices, such as RFID, GPS, 3G devices, etc., smart control devices, and mobile devices, such as smart mobile, ipad, tablet, etc., could be integrated into the proposed framework. These devices are able to support real-time planning, optimisation and decision making.
- **Distribution Configurations**: The proposed framework needs to be evaluated to support distribution configurations to be implemented in a complete SC. Different data formats and supports of *data filter and convert* layers need to be researched and the standard of data structures of the layer need to be developed. The authority and security mechanisms of data filter/convert layers between two different simulation blocks need to be developed based on different relationships among different companies in SC.
- **Cloud Computing**: The cloud computing needs to be researched to evaluate an appropriate framework. SMEs maybe able to use cloud computing to reduce the computation time of simulation and optimisation processes to implement a real-time decision making approach.

References

Aiex, R. M., Binato, S. & Resende, M. G. C. 2003. Parallel GRASP with path-relinking for job shop scheduling. *Parallel Computing,* 29, pp. 393-430.

Aldakhilallah, K. A. & Ramesh, R. 2001. Cyclic scheduling heuristics for a re-entrant job shop manufacturing environment. *International Journal of Production Research,* 39, pp. 2635-2657.

Allahverdi, A., Ng, C. T., Cheng, T. C. E. & Kovalyov, M. Y. 2008. A survey of scheduling problems with setup times or costs. *European Journal of Operational Research,* 187, pp. 985-1032.

Aloini, D., Dulmin, R. & Mininno, V. 2007. Risk management in ERP project introduction: Review of the literature. *Information & Management,* 44, pp. 547-567.

Anderson, E. G., Fine, C. H. & Gilboy, G. J. 2000. Upstream Volatility in the Supply Chain: The Machine Tool Industry as a Case Study. *Production and Operations Management.* MIT Sloan School.

Andradottir, S. 1996a. A global search method for discrete stochastic optimization. *SIAM Journal on Optimization*, pp.513-530.

Arsham, H. 1996. stochastic optimization of discrete event systems simulation. *Microelectronics and Reliability*, pp.1357-1368.

Artigues, C. & Roubellat, F. 2001. A Petri net model and a general method for on and off-line multi-resource shop floor scheduling with setup times. *International Journal of Production Economics,* 74, pp. 63-75.

Babbie, E. R. 1990. *Survey Research Methods*, Wadsworth Publishing.

Balas, E. & Vazacopoulos, A. 1998. Guided Local Search with Shifting Bottleneck for Job Shop Scheduling. *Management Science,* 44, pp. 262-275.

Bechhofer, R. E. 1954. A single-sample multiple decision procedure for ranking means of normal populations with known variances. *Ann. Math. Stat*, pp.16-39.

BIS. 2010. Small and Medium-sized Enterprise (SME) Statistics for the UK and Regions 2009. Department for Business Innovation &Skills.

Blazewicz, J., Pesch, E. & Sterna, M. 2000. The disjunctive graph machine representation of the job shop scheduling problem. *European Journal of Operational Research,* 127, pp. 317-331.

Bofinger, E. & Lewis, G. J. 1992. Two-stage procedures for multiple comparisons with a control. *Amer. J. Math. Manage. Sci.,* 12, pp. 253-275.

Botsali, A. R. 1999. Lot Streaming Techniques. Technical Note.

Bruque, S. & Moyano, J. 2007. Organisational determinants of information technology adoption and implementation in SMEs: The case of family and cooperative firms. *Technovation,* 27, pp. 241-253.

Bryman, A. & Bell, E. 2007. *Business Research Methods*, OUP Oxford.

Burns, J. F. & Sivazlian, B. D. 1978. Dynamic analysis of multi-echelon supply systems. *Computer .&Industrial Engineering,* 7, pp. 181-193.

Carson, J. S. 2004. Introduction to modeling and simulation. *Proceedings of the 36th conference on Winter simulation.* Washington, D.C.: Winter Simulation Conference.

Cassandras, C. G. 1993. *Rapid Learning Techniques for Discrete Event Systems: Modeling and Performance Analysis,* Irwin, Homewood, Illinois.

Chan, F. T. S., Wong, T. C. & Chan, L. Y. 2008. Lot streaming for product assembly in job shop environment. *Robotics and Computer-Integrated Manufacturing,* 24, pp. 321-331.

Chan, F. T. S., Wong, T. C. & Chan, L. Y. 2009. The application of genetic algorithms to lot streaming in a job-shop scheduling problem. *International Journal of Production Research,* 47, pp. 3387-3412(26).

Chang, J. H. & Chiu, H. N. 2005. A comprehensive review of lot streaming. *International Journal of Production Research,* 43, pp. 1515-1536.

Chase, R. B. & Aquilano, N. J. 2004. *Operations Management for Competitive Advantage*, Irwin Professional Pub.

Chase, R. B., Aquilano, N. J. & Jacobs, F. R. 1998. *Production and Operations Management: Manufacturing and Services*, Richard D Irwin.

Chen, C.-L. & Chen, C.-L. 2009a. A bottleneck-based heuristic for minimizing

makespan in a flexible flow line with unrelated parallel machines. *Computers & Operations Research,* 36, pp. 3073-3081.

Chen, C.-L. & Chen, C.-L. 2009b. Bottleneck-based heuristics to minimize total tardiness for the flexible flow line with unrelated parallel machines. *Computers & Industrial Engineering,* 56, pp. 1393-1401.

Chen, C. H. 1996. A lower bound for the correct subset-selection probability and its application to discrete-event systems simulations. *IEEE Transactions on Automatic Control*, pp.1227-1231.

Choy, K. L., Chow, H. K. H., Tan, K. H., Chan, C.-K., Mok, E. C. M. & Wang, Q. 2008. Leveraging the supply chain flexibility of third party logistics - Hybrid knowledge-based system approach. *Expert Systems with Applications,* 35, pp. 1998-2016.

Collis, J. & Hussey,R. 2003.*Business research: A practical guide for undergraduate and postgraduate students*. Palgrave Macmillan.

Dauzere-Peres, S. & Lasserre, J.-B. 1993. An iterative procedure for lot streaming in job-shop scheduling. *Computers & Industrial Engineering,* 25, pp. 231-234.

Dauzere-Peres, S. & Lasserre, J.-B. 1997. Lot Streaming in Job-Shop Scheduling. *OPERATIONS RESEARCH,* 45, pp. 584-595.

Dell'Amico, M. & Trubian, M. 1993. Applying tabu search to the job-shop scheduling problem. *Annals of Operations Research,* 41, pp. 231-252.

Ding, H., Ren, C., Wang, W. & Dong, J. 2006. Applying simulation in a supply chain transformation case. *Proceedings of the 38th conference on Winter simulation.* Monterey, California: Winter Simulation Conference.

Dong, J., Ding, H., Ren, C. & Wang, W. 2006. IBM SmartSCOR - a SCOR based supply chain transformation platform through simulation and optimization techniques. *Proceedings of the 38th conference on Winter simulation.* Monterey, California: Winter Simulation Conference.

Dunnett, C. W. 1955. A multiple comparisons procedure for comparing several treatments with a control. *J. Ameri. Stat. Assoc*, pp. 965-971.

El-Bouri, A., Azizi, N. & Zolfaghari, S. 2007. A comparative study of a new heuristic

based on adaptive memory programming and simulated annealing: The case of job shop scheduling. *European Journal of Operational Research,* 177, pp. 1894-1910.

Ervolina, T. R., Ettl, M., Lee, Y. M. & Peters, D. J. 2006. Simulating order fulfillment with product substitutions in an assemble-to-order supply chain. *Proceedings of the 38th conference on Winter simulation.* Monterey, California: Winter Simulation Conference.

Essafi, I., Mati, Y. & Dauzère-Pérès, S. 2008. A genetic local search algorithm for minimizing total weighted tardiness in the job-shop scheduling problem. *Computers & Operations Research,* 35, pp. 2599-2616.

European Commission. 2003, The new SME definition user guide and model declaration European Commission.

European Commission. 2010. European SMEs under Pressure-Annual Report on EU Small and Medium-Sized Enterprises 2009. European Commission.

Evans, J. R. 1992. *Applied Production and Operations Management*, West Publishing Company.

Fayez, M., Rabelo, L. & Mollaghasemi, M. 2005. Ontologies for supply chain simulation modeling. *Proceedings of the 37th conference on Winter simulation.* Orlando, Florida: Winter Simulation Conference.

Forrester, J. W. 1958. *Industrial Dynamics:A Major Breakthrough for Decision Makers*, Harvard Business Review.

Frank, J. M. & Barry, L. N. 1993. Simultaneous ranking, selection and multiple comparisons for simulation. *Proceedings of the 25th conference on Winter simulation.* Los Angeles, California, United States: ACM.

Fu, M. C. 1990. Convergence of a stochastic approximation algorithm for the GI/G/1 queue using infinitesimal perturbation. *J. Optim. Theory Appl.,* 65, pp. 149-160.

Fu, M. C. 1994. A tutorial review of techniques for simulation optimization. *Proceedings of the 26th conference on Winter simulation.* Orlando, Florida, United States: Society for Computer Simulation International.

Fu, M. C. 2001. Simulation optimization. *Proceedings of the 33nd conference on Winter simulation.* Arlington, Virginia: IEEE Computer Society.

Fu, M. C., Glover, F. W. & April, J. 2005. Simulation optimization: a review, new developments, and applications. *Proceedings of the 37th conference on Winter simulation.* Orlando, Florida: Winter Simulation Conference.

Fu, M. C. & Ho, Y.-C. 1988. Using perturbation analysis for gradient estimation, averaging and updating in a stochastic approximation algorithm. *Proceedings of the 20th conference on Winter simulation.* San Diego, California, United States: ACM.

Fulya, A., Berna, D. & Akif, A. B. 2002. Simulation of manufacturing operations: optimization of buffer sizes in assembly systems using intelligent techniques. *Proceedings of the 34th conference on Winter simulation: exploring new frontiers.* San Diego, California: Winter Simulation Conference.

Gaither, N., & Frazier, G. 2002. *Operations management* (9th ed.). Mason, OH.

Gao, J., Gen, M., Sun, L. & Zhao, X. 2007. A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling problems. *Computers & Industrial Engineering,* 53, pp. 149-162.

Gen, M., Lin, L. & Zhang, H. 2009. Evolutionary techniques for optimization problems in integrated manufacturing system: State-of-the-art-survey. *Computers & Industrial Engineering,* 56, pp. 779-808.

Glover, F. 1989. Tabu Seach-Part I. *ORSA Journal on Computing,* 1, pp. 190-206.

Glover, F. 1990. Tabu Search-Part II. *ORSA Journal on Computing,* 2, pp. 4-32.

Glover, F. 2001. Tabu search. *In:* Harris, S. G. a. C. (ed.) *In Encyclopedia of Operations Research and Management Science.* 2n ed. Boston: Kluwer Academic Publishers.

Glover, F. & Kochenberger, G. A. 2003. *Handbook of Metaheuristics*, Springer.

Gordon, V., Proth, J.-M. & Chu, C. 2002. A survey of the state-of-the-art of common due date assignment and scheduling research. *European Journal of Operational Research,* 139, pp. 1-25.

Grabowski, J. & Wodecki, M. 2004. A very fast tabu search algorithm for the permutation flow shop problem with makespan criterion. *Computers & Operations Research,* 31, pp. 1891-1909.

Gray, D. & Goldsman, D. 1988. Indifference-zone selection procedures for choosing the best airspace configuration. *Proceedings of the 20th conference on Winter*

simulation. San Diego, California, United States: ACM.

Gunasekaran, A. & Ngai, E. W. T. 2009. Modeling and analysis of build-to-order supply chains. *European Journal of Operational Research,* 195, pp. 319-334.

Gupta, A., Whitman, L. & Agarwal, R. K. 2001. Supply chain agent decision aid system (SCADAS). *Proceedings of the 33nd conference on Winter simulation.* Arlington, Virginia: IEEE Computer Society.

Gupta, S. S. 1956. On a decision rule for a problem in ranking means. *Mimeograph Series No. 150.* N.C.: Institute of Statistics. University of North Carolina, Chapel Hill.

Haddock, J. & Mittenthal, J. 1992. Simulation optimization using simulated annealing. *Comput. Ind. Eng.,* 22, pp. 387-395.

Heizer, J. & Render, B. 2006. *Operations Management*, Prentice Hall.

Ho, Y. C. 1989. Introduction to special issue on dynamics of discrete event systems. *In:* Proceedings of the IEEE.

Ho, Y. C. & Deng, M. 1994. The problem of large search space in stochastic optimization. *In:* Proceedings of the IEEE Conference on Decision and Control, pp. 1975-1977.

Ho, Y. C., Sreenivas, R. & Vakili, P. 1992. Ordinal optimization of discrete event dynamic systems. *Discrete Event Dynamical Systems*, pp.61-88.

Hsu, J. C. 1984. Constrained simultaneous confidence intervals for multiple comparisons with the best. *Ann. Stat.*, pp. 1136-1144.

Hsu, J. C. & Nelson, B. L. 1988. Optimization over a finite number of system designs with one-stage sampling and multiple comparisons with the best. *Proceedings of the 20th conference on Winter simulation.* San Diego, California, United States: ACM.

Huang, R.-H. 2010. Multi-objective job-shop scheduling with lot-splitting production. *International Journal of Production Economics,* 124, pp. 206-213.

Huemer, L. 2006. Supply Management: Value Creation, Coordination and Positioning in Supply Relationships. *Long Range Planning,* 39, pp. 133-153.

Jeong, H.-I., Park, J. & Leachman, R. C. 1999. A batch splitting method for a job shop scheduling problem in an MRP environment *International Journal of Production*

Research, 37, pp. 3583 - 3598.

Jin, B., Luh, P. B. & Thakur, L. S. 1999. An effective optimization-based algorithm for job shop scheduling with fixed-size transfer lots. *Journal of Manufacturing Systems,* 18, pp. 284-300.

Kapuscinski, R. & Tayur, S. R. 1999. Optimal policies and simulation based optimization for capacitated production inventory systems. *In:* S. R. Tayur, R. G., M. J. Magazine (ed.) *Optimal policies and simulation based optimization for capacitated production inventory systems.* Kluwer Academic.

Kemper, P. 2000. Logistic Processes go Petri nets. *In:* Workshop Algorithmen und Werkzeuge für Petri Netze, July 2000 Koblenz: Universität Koblenz-Landau. pp. 69-74.

KIM, Y.-D., SHIM, S.-O., KIM, S.-B., CHOI, Y.-C. & YOON, H. M. 2004. Parallel machine scheduling considering a job-splitting property. *International journal of production research* 42, pp. 4531–4546.

Koo, L. Y., Chen, Y., Adhitya, A., Srinivasan, R. & Karimi, I. A. 2006. Evaluating refinery supply chain policies and investment decisions through simulation-optimization. *Proceedings of the 38th conference on Winter simulation.* Monterey, California: Winter Simulation Conference.

Laarhoven, P. J. M. v., Aarts, E. H. L. & Lenstra, J. K. 1992. Job Shop Scheduling by Simulated Annealing. *Operations Research,* 40, pp. 113-125.

Laudon, K. C. & Laudon, J. P. 2006. *Management Information Systems: Managing the Digital Firm*, Prentice Hall.

Lauff, V. & Werner, F. 2004. Scheduling with common due date, earliness and tardiness penalties for multimachine problems: A survey. *Mathematical and Computer Modelling,* 40, pp. 637-655.

Law, A. M. & Kelton, W. D. 2000. *Simulation Modeling and Analysis*, McGraw Hill.

Lee, L. H., T. W. E. Lau., and Y. C. Ho. 1999. Explanation of goal softening in ordinal optimization. *IEEE Transactions on Automatic Control*, pp.94-98.

Lee, Y. H., Choa, M. K., Kimb, S. J. & Kimc, Y. B. 2002. Supply chain simulation with discrete-continuous combined modeling. *Computers & Industrial Engineering,* 43, pp. 375-392

Lee, Y. M. 2006. Simulating impact of available-to-promise generation on supply chain performance. *Proceedings of the 38th conference on Winter simulation.* Monterey, California: Winter Simulation Conference.

Li, J.-q., Pan, Q.-k. & Liang, Y.-C. 2010. An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems. *Computers & Industrial Engineering,* 59, pp. 647-662.

Li, S., Ragu-Nathan, B., Ragu-Nathan, T. S. & Subba Rao, S. 2006. The impact of supply chain management practices on competitive advantage and organizational performance. *Omega,* 34, pp. 107-124.

Liang, H. & Xue, Y. 2004. Coping with ERP-related contextual issues in SMEs: a vendor's perspective. *The Journal of Strategic Information Systems,* 13, pp. 399-415.

Lin, B. M. T. & Jeng, A. A. K. 2004. Parallel-machine batch scheduling to minimize the maximum lateness and the number of tardy jobs. *International Journal of Production Economics,* 91, pp. 121-134.

Lipták, B. G. 2003. *Process measurement and analysis*, CRC Press.

Liu, R., Kumar, A. & Stenger, A. J. 2006. Simulation results for supply chain configurations based on information sharing. *Proceedings of the 38th conference on Winter simulation.* Monterey, California: Winter Simulation Conference.

Low, C. 2005. Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines. *Comput. Oper. Res.,* 32, pp. 2013-2025.

Low, C., Hsu, C.-M. & Huang, K.-I. 2004. Benefits of lot splitting in job-shop scheduling. *The International Journal of Advanced Manufacturing Technology,* 24, pp. 773-780.

Ma, Y., Chu, C. & Zuo, C. 2010. A survey of scheduling with deterministic machine availability constraints. *Computers & Industrial Engineering,* 58, pp. 199-211.

Martin, P. Y. & Turner, B. 1986. Grounded Theory and Organizational Research. *The Journal of Applied Behavioral Science,* 22, pp. 141-157.

Martinez, S., Dauzère-Pérès, S., Guéret, C., Mati, Y. & Sauer, N. 2006. Complexity of flowshop scheduling problems with a new blocking constraint. *European Journal of Operational Research,* 169, pp. 855-864.

Mascis, A. & Pacciarelli, D. 2002. Job-shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research,* 143, pp. 498-517.

Mehta, A. & Rawles, I. 1999. Business solutions using WITNESS. *Proceedings of the 31st conference on Winter simulation: Simulation---a bridge to the future - Volume 1.* Phoenix, Arizona, United States: ACM.

Michalewicz, Z. & Schoenauer, M. 2001. *Evolutionary algorithms. In Encyclopedia of Operations Research and Management Science,* Boston, Kluwer Academic Publishers.

Mönch, L. & Habenicht, I. 2003. SIMULATION-BASED ASSESSMENT OF BATCHING HEURISTICS IN SEMICONDUCTOR MANUFACTURING. *In: Proceedings of the 35th conference on Winter simulation*, New Orleans, Louisiana. ACM, pp. 1338-1345.

Mönch, L., Rose, O. & Sturm, R. 2003. A Simulation Framework for the Performance Assessment of Shop-Floor Control Systems. *Simulation,* 79.

Morabito, V., Pace, S. & Previtali, P. 2005. ERP Marketing and Italian SMEs. *European Management Journal,* 23, pp. 590-598.

Moyaux, T., Chaib-draa, B. & D'Amours, S. 2003. Multi-Agent Coordination Based on Tokens: Reduction of the Bullwhip Effect in a Forest Supply Chain. *In:* Proceedings of the second international joint conference on Autonomous agents and multiagent systems, Melbourne, Australia pp. 670 - 677.

Mullarkey, P., Gavirneni, S. & Morrice, D. J. 2000. Strategies for successful simulation of manufacturing systems: dynamic output analysis for simulations of manufacturing environments. *Proceedings of the 32nd conference on Winter simulation.* Orlando, Florida: Society for Computer Simulation International.

Musalem, E. P. & Dekker, R. 2005. Controlling inventories in a supply chain: A case study. *International Journal of Production Economics,* 93-94, pp. 179-188.

Nelson, B. L. & Banerjee, S. 2001. Selecting a good system: procedures and inference. *IIE Trans,* 33, pp. 149-166.

Nelson, B. L. & Matejcik, F. J. 1995. Using common random numbers for indifference-zone selection and multiple comparisons in simulation. *Manage. Sci.,* 41, pp. 1935-1945.

Normal Distribution. 2004. *Area under the Normal Distribution* [Online]. Available:

http://business.statistics.sweb.cz/normal01.jpg [Accessed 15 April 2011].

Nowicki, E. & Smutnicki, C. 1996. A fast taboo search algorithm for the job shop problem. *Management Science,,* 42, pp. 797-813.

Oey, K. & Mason, S. J. 2001. Scheduling and dispatching: scheduling batch processing machines in complex job shops. *Proceedings of the 33nd conference on Winter simulation.* Arlington, Virginia: IEEE Computer Society.

Ólafsson, S. & Jumi, K. 2002. Simulation optimization: simulation optimization. *Proceedings of the 34th conference on Winter simulation: exploring new frontiers.* San Diego, California: Winter Simulation Conference.

Orlikowski, W.J. & Baroudi, J.J.1991. *Studying Information Technology in Organizations: Research Approaches and Assumptions*, Information Systems Research (2), pp. 1-28.

Parunak, V. & Vanderbok, R. 1998. The DASCh Experience:How to Model a Supply Chain? *In Proceedings of the Second International Conference on Complex Systems.* Nashua, NH.

Pidd, M. 1998. *Computer Simulation in Management Science*, John Wiley & Sons.

Pidd, M. 2004. Simulation worldviews: so what? *Proceedings of the 36th conference on Winter simulation.* Washington, D.C.: Winter Simulation Conference.

Pinedo, M. L. 2000. *Planning and Scheduling in Manufacturing and Services,* New York, Springer.

Pinedo, M. L. 2008. *Scheduling Theory, Algorithms, and Systems,* New York, Springer.

Ponniah, P. 2003. *Database Design and Development: An Essential Guide for IT Professionals* John Wiley & Sons.

Potts, C. N. & Kovalyov, M. Y. 2000. Scheduling with batching: A review. *European Journal of Operational Research,* 120, pp. 228-249.

Pundoor, G. & Herrmann, J. W. 2006. A hierarchical approach to supply chain simulation modelling using the Supply Chain Operations Reference model. *International Journal of Simulation and Process Modelling* 2, pp. 124 - 132

Rabe, M., Jaekel, F.-W. & Weinaug, H. 2006. Reference models for supply chain design

and configuration. *Proceedings of the 38th conference on Winter simulation.* Monterey, California: Winter Simulation Conference.

Rapoport, R. N. 1970. Three Dilemmas in Action Research. *Human Relations,* 23, pp. 499-512.

Remenyi, D., Williams, B., Money, A. & Swartz, E. 1998. *Doing research in business and management: An introduction to process and method*. Sage Publications Ltd.

Ribas, I., Leisten, R. & Framiñan, J. M. 2010. Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. *Computers & Operations Research,* 37, pp. 1439-1454.

Robinson, S. 2004. *Simulation:The Practice of Model Development and Use*, John Wiley & Sons Ltd.

Rockwell Automation. 2009. *Arena Comparison Statement* [online]. Rockwell Automation .Available: http://www.systemsnavigator.com/sn_website/Documents/Arena_comparison_statement_2008.pdf [Accessed 14th Sep 2011].

Ross, D. F. 2003. *Introduction to e-Supply Chain Management: Engaging Technology to Build Market-Winning Business Partnerships* St. Lucie Press.

Rossetti, M. D., Aylor, B., Jacoby, R., Prorock, A. & White, A. 2000. Simulation modeling and the web: Simfone': an object-oriented simulation framework. *Proceedings of the 32nd conference on Winter simulation.* Orlando, Florida: Society for Computer Simulation International.

Rossetti, M. D. & Chan, H.-T. 2003. Supply chain management simulation: a prototype object-oriented supply chain simulation framework. *Proceedings of the 35th conference on Winter simulation: driving innovation.* New Orleans, Louisiana: Winter Simulation Conference.

Rossetti, M. D., Miman, M., Varghese, V. & Xiang, Y. 2006. An object-oriented framework for simulating multi-echelon inventory systems. *Proceedings of the 38th conference on Winter simulation.* Monterey, California: Winter Simulation Conference.

Ruiz, R. & Vázquez-Rodríguez, J. A. 2010. The hybrid flow shop scheduling problem. *European Journal of Operational Research,* 205, pp. 1-18.

Saker. 2007. *Witness* [Online]. Saker Solutions. Available: http://www.sakersolutions.com/simcentre/witness.html [Accessed 29 March 2011].

Saunders, M. N. K., Thornhill, A. & Lewis, P. 2009. *Research Methods for Business Students*, Financial Times/ Prentice Hal.

SBA. 2011. Small Business Profile. The US Small Business Administration.

Schmidt, K. 2001. *Using tabu search to solve the job-shop scheduling problem with sequence dependent setup times.* Master's thesis, Brown University.

Shi, L. & Ólafsson, S. 1997. An integrated framework for deterministic and stochastic optimization. *Proceedings of the 29th conference on Winter simulation.* Atlanta, Georgia, United States: IEEE Computer Society.

Sotskov, Y. N., Tautenhahn, T. & Werner, F. 1999. On the application of insertion techniques for job shop problems with setup times. *RAIRO Recherche Operationnelle,* 33, pp. 209 - 245.

Sprenger, R. & Mönch, L. 2008. A SIMULATION FRAMEWORK FOR ASSESSING THE REPFORMANCE OF COOPERATIVE TRANSPORTATION PLANNING ALGORITHMS. *In:* Proceedings of the 40th Conference on Winter Simulation. pp. 2769-2776.

Sterman, J. D. 1989. Modeling Managerial Behavior: Misperceptions of Feedback in a Dynamic Decision Making Experiment. *Management Science,* 5, pp. 321-339.

Stevenson, W. J. 1998. *Production/Operations Management*, Mcgraw-Hill.

Sullivan, D. W. & Wilson, J. R. 1989. Restricted subset selection procedures for simulation. *Oper. Res.,* 37, pp. 52-71.

Swisher, J. R., Hyden, P. D., Jacobson, S. H. & Schruben, L. W. 2000. Simulation optimization: a survey of simulation optimization techniques and procedures. *Proceedings of the 32nd conference on Winter simulation.* Orlando, Florida: Society for Computer Simulation International.

Swisher, J. R. & Jacobson, S. H. 2002. Evaluating the design of a family practice healthcare clinic using discrete-event simulation. *Health Care Manage. Sci,* 5, pp. 75-88.

Swisher, J. R., Jacobson, S. H. & Yücesan, E. 2003. Discrete-event simulation

optimization using ranking, selection, and multiple comparison procedures: A survey. *ACM Trans. Model. Comput. Simul.,* 13, pp. 134-154.

Taillard, É. D. 1994. Parallel Taboo Search Techniques for the Job Shop Scheduling Problem. *ORSA JOURNAL ON COMPUTING,* 6, pp. 108-117.

Tashakkori, A. & Teddlie, C. 1998. *Mixed Methodology Combining Qualitative and Quantitative Approaches.* Sage Publications Ltd.

Towill, D. R. 1982. Dynamics analysis of an inventory and order based production control system. *International Journal of Production Research,* 20, pp. 671-687.

Towill, D. R. 1991. Supply Chain Dynamics. *International Journal of Computer Intergrated Manufacturing,* 4, pp. 197-208.

Towill, D. R. 1996. Time compression and supply chain management- a guided tour. *Ligistics Information Management* 9, pp. 41-53.

Truong, T. H. & Azadivar, F. 2003. Simulation optimization in manufacturing analysis: simulation based optimization for supply chain configuration design. *Proceedings of the 35th conference on Winter simulation: driving innovation.* New Orleans, Louisiana: Winter Simulation Conference.

Trochim,W. & Donnelly, J.2006. *The Research Methods Knowledge Base. 3rd.* Atomic Dog Publishing Inc

Tukey, J. W. 1953. The problem of multiple comparisons. Unpublished manuscript.

Umeda, S. 2007. Supply-chain Simulation Integrated Discrete-event Modeling with System-Dynamics Modeling. Conference on Advances in Production Management. Linköping, Sweden. pp.329-336

Umeda, S. & Jain, S. 2004. ISSS:Integrated Supply Chain Simulation System--Modeling Requirement and Design Issues,NISTIR 7180. Gaithersburg: Mary Land National Institute of Standards and Technology.

Umeda, S. & Jones, A. 1998. AN INTEGRATION TEST-BED SYSTEM FOR SUPPLY CHAIN MANAGEMENT. *In:* Proceedings of the 30th conference on Winter simulation, Washington, D.C., United States. pp. 1377-1385.

Umeda, S. & Lee, Y. T. 2004. DESIGN SPECIFICATIONS OF A GENERIC SUPPLY CHAIN SIMULATOR. *In:* Proceedings of the 36th conference on Winter simulation Washington, D.C., pp. 1158-1166.

van Laarhoven, P. J. M., Aarts, E. H. L. & Lenstra, J. K. 1992. Job shop scheduling with simulated annealing. *Operations Research,* 40, pp. 113 - 125.

Vorst, J. G. A. J. v. d., Beulens, A. J. M. & Beek, P. v. 2000. Modelling and Simulating Multi-Echelon Food Systems. *European Journal of Operational Research,* 122, pp. 354-366.

Wadhwa, S., Saxena, A. & Chan, F. T. S. 2008. Framework for flexibility in dynamic supply chain management. *International Journal of Production Research,* 46, pp. 1373-1404.

Wang, X. & Takakuwa, S. 2006. Module-based modeling of production-distribution systems considering shipment consolidation. *Proceedings of the 38th conference on Winter simulation.* Monterey, California: Winter Simulation Conference.

Wang, Y. & Zhou, C. 2005. Fluid based simulation model for high volume DC conveyor systems. *Proceedings of the 37th conference on Winter simulation.* Orlando, Florida: Winter Simulation Conference.

Wu, Y., Frizelle, G., Ayral, L., Marsein, J., Merwe, E. V. d. & Zhou, D. 2002. Tackling Industrial Complexity: the ideas that make a difference. *In:* Conference of Manufacturing Complexity Network ,Cambridge, UK.

Xie, X. 1997. Dynamics and convergence rate of ordinal comparison of stochastic discrete-event systems. *IEEE Transactions on Automatic Control*, pp.586-590.

Yalaoui, F. & Chu, C. 2003. An efficient heuristic approach for parallel machine scheduling with job splitting and sequence-dependent setup times. *IIE Transactions,* 35, pp. 183-190

Yang, W. N. & Nelson, B. L. 1991. Using common random numbers and control variates in multiple-comparison procedures. *Oper. Res.,* 39, pp. 583-591.

Yin, R. K. 2003. *Case Study Research: Design and Methods*, Sage Publications, Inc.

Zeng, J. & Wu, J. 1993. DEDS (discrete event dynamic systems) simulation-optimization algorithm using simulated-annealing combined with perturbation analysis. *Zidonghua Xuebao Acta Automatica Sinica*, pp. 728-731.

Zhang, C., Li, P., Guan, Z. & Rao, Y. 2007. A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem. *Computers & Operations Research,* 34, pp. 3229-3242.

Zhang, C. Y., Li, P., Rao, Y. & Guan, Z. 2008. A very fast TS/SA algorithm for the job shop scheduling problem. *Computers & Operations Research,* 35, pp. 282-294.

Zikmund,W.G.2003. *Business research methods.* Thomson Learning.

Appendix A Notations of a Three-field α|β|γ

The following describes the notations of a three-field α|β|γ (Allahverdi et al., 2008, Pinedo, 2008).

Shop type (α field)

1: single machine

Fm: flow shop

FFc: flexible (hybrid) flow shop

AF: assembly flow shop

Pm, Qm, Rm: parallel machines (Pm: identical machines; Qm: uniform machines; Rm: unrelated machines)

Jm: job shop

FJc: flexible job shop

Om: open shop

Shop characteristics (β field)

prec: precedence constraints

r_j: non-zero release date

pmtn: pre-emption

nwt: no-wait

block: blocking

brkdwn: breakdowns

M_j: machine eligibility restrictions

Setup information (β field)

ST_{si}: sequence-independent setup time

SC_{si}: sequence-independent setup cost

ST_{sd}: sequence-dependent setup time

SC_{sd}: sequence-dependent setup cost

$ST_{si,b}$: sequence-independent batch or family setup time

$SC_{si,b}$: sequence-independent batch or family setup cost

$ST_{sd,b}$: sequence-dependent batch or family setup time

$SC_{sd,b}$: sequence-dependent batch or family setup cost

R_{si} :sequence-independent removal time

R_{sd} :sequence-dependent removal time

$R_{si,b}$:sequence-independent batch or family removal time

$R_{sd,b}$:sequence-dependent batch or family removal time

Performance (γ field)

C_{max}: makespan

L_{max}: maximum lateness

T_{max}: maximum tardiness

D_{max}: maximum delivery time

TSC: total setup/changeover cost

TST: total setup/changeover time

$\sum f_j$: total flowtime

$\sum C_j$: total completion time

$\sum E_j$: total earliness

$\sum T_j$: total tardiness

$\sum U_j$: number of tardy (late) jobs

$\sum w_j C_j$: total weighted completion time

$\sum w_j U_j$: weighted number of tardy jobs

$\sum w_j E_j$: total weighted earliness

$\sum w_j T_j$: total weighted tardiness

$\sum w_j f_j$: total weighted flowtime

More details of the notations refer to (Pinedo, 2008).

Appendix B Rule Collection in Chapter 4

Rule 4.0 (To determine the critical job or an ordered list of critical jobs):

If a general objective function (cost function) is chosen as $Min(\sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij}F_{ij})$, subject to $\sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij} = 1$, the critical job i is

$$\text{Max}(\sum_{j=1}^{m} W_{ij}F_{ij})$$

or the ordered list of critical jobs is

the list of the job i in a non-increasing order of their $(\sum_{j=1}^{m} W_{ij}F_{ij})$

If a general objective function (cost function) is chosen as $Max(\sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij}F_{ij})$, subject to $\sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij} = 1$, the critical job i is

$$\text{Min}(\sum_{j=1}^{m} W_{ij}F_{ij})$$

or the ordered list of critical jobs is

the list of the job i in a non-decreasing order of their $(\sum_{j=1}^{m} W_{ij}F_{ij})$

Definition 4.1: $a_o \in$ DAS is a disjunctive arc from an operation node mi (N_{mi}) to an operation node mj (N_{mj}). a_p represents the conjunctive arc from an operation node nj (N_{nj}) to an operation node mj (N_{mj})

Rule 4.1: $r_{mj}=\max(r_{nj}+p_{nj}, r_{mi}+p_{mi})$

If $r_{mi}+p_{mi} > r_{nj}+p_{nj}$

$a_o \in$ CDAS

Else

$a_o \in$ RDAS

Where r_{nj} is the release time of operation nj (N_{nj}) determined by the longest path in the graph G from the *source* U to $N_{nj;}$ r_{mj} is the release time of operation mj (N_{mj}) determined by the longest path in graph G from the *source* U to $N_{mj;}$ p_{nj} is the operation

time of operation nj; r_{mi} is the release time of operation mi (N_{mi}) determined by the longest path in graph G from the *source* U to N_{mi} ; p_{mi} is the operation time of operation mi; operation mi is a predecessor operation of operation mj on the resource (machine) task; operation nj is a predecessor operation of operation mj on the job task.

Rule 4.2 (To determine longest (critical) paths for Jm||C_{max})

When a schedule is provided, the last operation (node) with the completion time is taken as the **begin node**. The longest path only consists of nodes, CDAs, and conjunctive arcs. The method of determining the longest path is as follows`:

Current node = Begin node

If there are a CDA and a conjunctive arc which come to the current node (end node)

The longest path includes the CDA

Current node = New node (from the previous node along the CDA)

Else

The longest path includes the conjunctive arc

Current node = New node (from the previous node along the conjunctive arc)

This procedure above will continue in the backward manner until reaching the source node. The path obtained using **Rule 4.2** is the longest path. If there exists more than one last operation with the same completion time, these critical paths can be determined by **Rule 4.2** one by one.

Rule 4.3 (To construct an ECR_{ij})

To an operation ij of a job (batch) with Q (number of items in the batch), the CRP_x with the lowest operation time p_{ij}= (T_{p*} Q/DV +T_s) (T_p is process time, T_s is set up time) in SCR_{ij} is chosen as the first CRP of ECR_{ij}. CRP_x with the first lowest operation time is removed from SCR_{ij}. This concept can be used iteratively to select the best CRPs for ECR_{ij} with the lowest operation time.

Rule 4.4 (To determine CDA,RDA, and DA)

There are two operation nodes N_{ij}, N_{kt}, and N_{ft} is the predecessor operation of operation N_{kt} on job t. If between r_{kt} and Te_{ft}, N_{ij} occupies a CR or several CRs of ECR_{kt} and OCR_{kt}, there exists a CDA directed from N_{ij} to N_{kt}. If N_{ij} is a predecessor operation of operation N_{kt} on a CR task of OCR_{kt} before r_{kt}, there exists a RDA directed from N_{ij} to N_{kt}.

$DA= CDA \cup RDA$

Rule 4.5 (To determine ef)

To describe the relationship of two operations (nodes) N_{ij} and N_{kt} linked by a CDA, a conflicted effective parameter ef is introduced to evaluate the relationship level of a CDA ($N_{ij} \rightarrow N_{kt}$).

$CM_n= M_{ij} \cap M_{kt}$ and $CL_n= L_{ij} \cap L_{kt}$

If ($M_{kt}= CM_n$ and $L_{kt}= CL_n$) % means ($M_{kt} \in M_{ij}$ and $L_{kt} \in L_{ij}$) %

ef=1

p_{ij} = operation time of N_{ij} on DA

Else

ef=0

p_{ij} =0 on DA

where M_{ij} is the machine set of OCR_{ij}, M_{kt} is the machine set of OCR_{kt}, L_{ij} is the labour set of OCR_{ij}, L_{kt} is the labour set of OCR_{kt}.

Rule 4.6 (To determine the critical path for $Jm|ST_{id,b}$ with LS)

The begin node is the last operation of the critical job i using **Rule 4.0**. The critical path only consists of nodes, CDAs with ef=1, and conjunctive arcs. The method of finding the longest path is as follows:

Current node = Begin node

If there is a CDA with ef=1 which comes to the current node (end node)

The critical path includes the CDA with ef=1

Current node= New node (from the previous node along the CDA)

Else

The critical path includes the conjunctive arc

Current node = New node (from the previous node along the conjunctive arc)

This will continue in the backward manner until reaching the source node. The path obtained using **Rule 4.6** is the critical path. If there exists more than one last operation with the same completion time, these critical paths can be determined by **Rule 4.6** one by one.

Rule 4.7 (To determine the tabu move)

The TL is considered as the list of the optimisation nodes and only TB recently optimisation nodes own a tabu status. There are several proposed methods of combinations of four parameters to define a tabu move. The notations of the four parameters are described as follows:

ACR: If a *swap* moves all CRs in the ECR of an optimisation node with a tabu status to the other node, this *swap* should be forbidden except when meeting aspiration criterion.

ACRP: If a *swap* moves all CRPs in the ECR of an optimisation node with a tabu status to the other node, this *swap* should be forbidden except when meeting aspiration criterion.

AO: allowing reusing optimisation nodes in the tabu list

FO: forbidding reusing optimisation nodes in the tabu list

Every tabu design combination needs to choose two parameters, one from ACR and ACRP and the other from AO and FO. The combination (ACRP, FO) expresses if a *swap* is determined a tabu move by the ACRP method or by the FO method, this *swap* is a tabu move. A combination (ACRP, FO) means if a *swap* moves all CRPs in the ECR of an optimisation node with a tabu status to the other node or this current optimisation

node owns a tabu status, this *swap* should be forbidden except when meeting aspiration criterion.

Rule 4.8 (priority rule of assigning CRs to operations)

Firstly the status of the node, if the node is an optimisation node in the tabu list, it is processed before the other nodes.

Secondly the priority of a job, the high priority job gets the CRs prior to the next higher one.

Thirdly the due date, the early node gets the CRs before the later node.

Fourthly the arrival date, the early node are assigned CRs before the later node.

Rule 4.9 (To determine optimisation nodes on a critical path)

A critical path of the job is generated referring to **Rule 4.6**. In order to reduce the calculated time of the reschedule, a back forward method is adopted which means considering the later node first, and then go backward to consider the earlier node on the time horizon. The node directly linked by CDAs is taken as a suitable optimisation node. If the node is only linked by RDAs or conjunctive arcs, the node is ignored because it has already occupied its ECR and *swapping* (optimising) it is unnecessary.

Zhang's method (definition in Section 4.3.2)

The length of tabu list is determined by Zhang's method (2007) for n jobs and m machines.

L=10+n/m

If n<=2m

  tn=int(random(L,1.4L))

else

  tn=int(random(L,1.5L))

Appendix C Normal Curve Areas

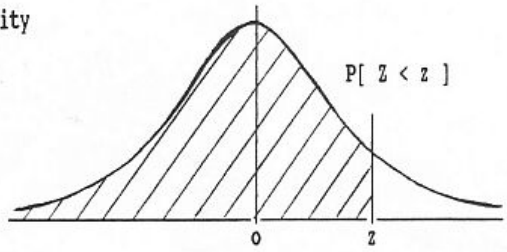
The following table shows relationship between the z value and an area under the normal curve (Normal Distribution, 2004).

STANDARD STATISTICAL TABLES

1. Areas under the Normal Distribution

The table gives the cumulative probability up to the standardised normal value z

i.e. $P[\ Z < z\ ] = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}Z^2)\ dZ$



z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.5000	0.5040	0.5080	0.5120	0.5159	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7854
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8804	0.8830
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
1.6	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.7	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.8	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
1.9	0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	0.9750	0.9756	0.9761	0.9767
2.0	0.9773	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817
2.1	0.9821	0.9826	0.9830	0.9834	0.9838	0.9842	0.9846	0.9850	0.9854	0.9857
2.2	0.9861	0.9865	0.9868	0.9871	0.9874	0.9878	0.9881	0.9884	0.9887	0.9890
2.3	0.9893	0.9896	0.9898	0.9901	0.9904	0.9906	0.9909	0.9911	0.9913	0.9916
2.4	0.9918	0.9920	0.9922	0.9924	0.9927	0.9929	0.9931	0.9932	0.9934	0.9936
2.5	0.9938	0.9940	0.9941	0.9943	0.9945	0.9946	0.9948	0.9949	0.9951	0.9952
2.6	0.9953	0.9955	0.9956	0.9957	0.9959	0.9960	0.9961	0.9962	0.9963	0.9964
2.7	0.9965	0.9966	0.9967	0.9968	0.9969	0.9970	0.9971	0.9972	0.9973	0.9974
2.8	0.9974	0.9975	0.9976	0.9977	0.9977	0.9978	0.9979	0.9980	0.9980	0.9981
2.9	0.9981	0.9982	0.9982	0.9983	0.9984	0.9984	0.9985	0.9985	0.9986	0.9986
z	3.00	3.10	3.20	3.30	3.40	3.50	3.60	3.70	3.80	3.90
P	0.9986	0.9990	0.9993	0.9995	0.9997	0.9998	0.9998	0.9999	0.9999	1.0000

Appendix D Schedule-driven Case study

The following pictures show the interfaces of an optimisation block and a verification program, a cell simulator coded by WITNESS for the UK-based case study.

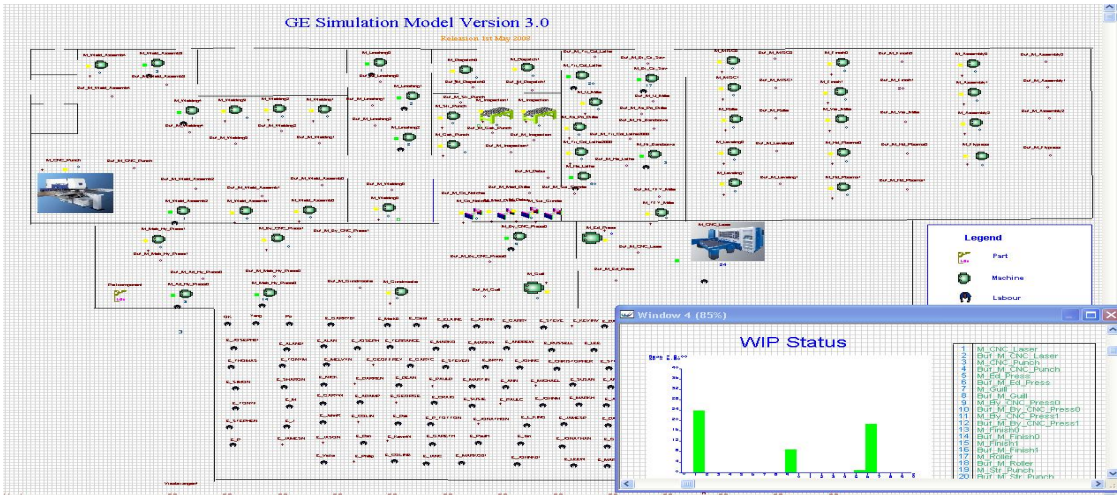


Figure D1 Cell simulator for a schedule-driven case study

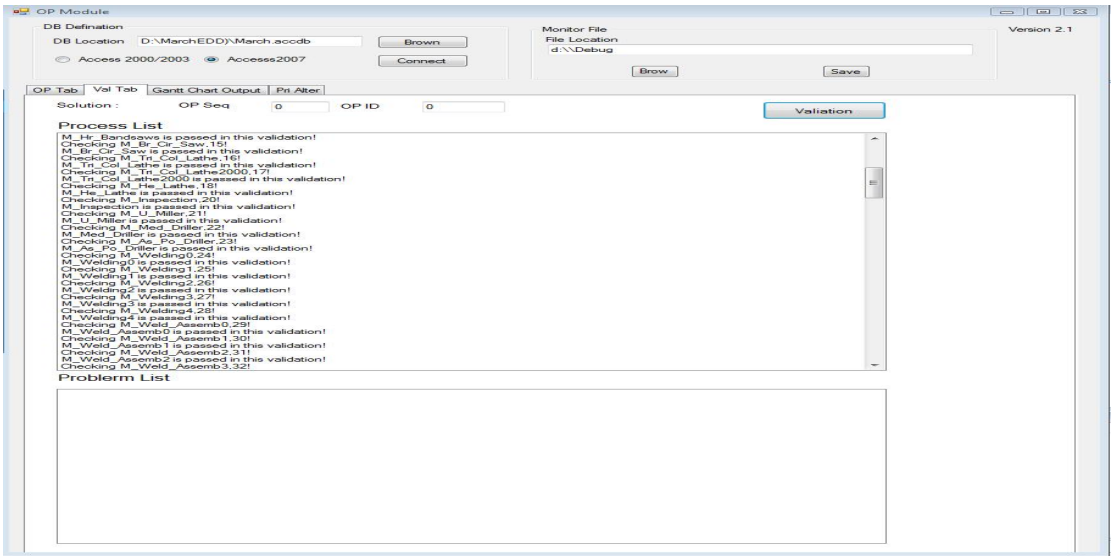


Figure D2 Interface of a Verification Program

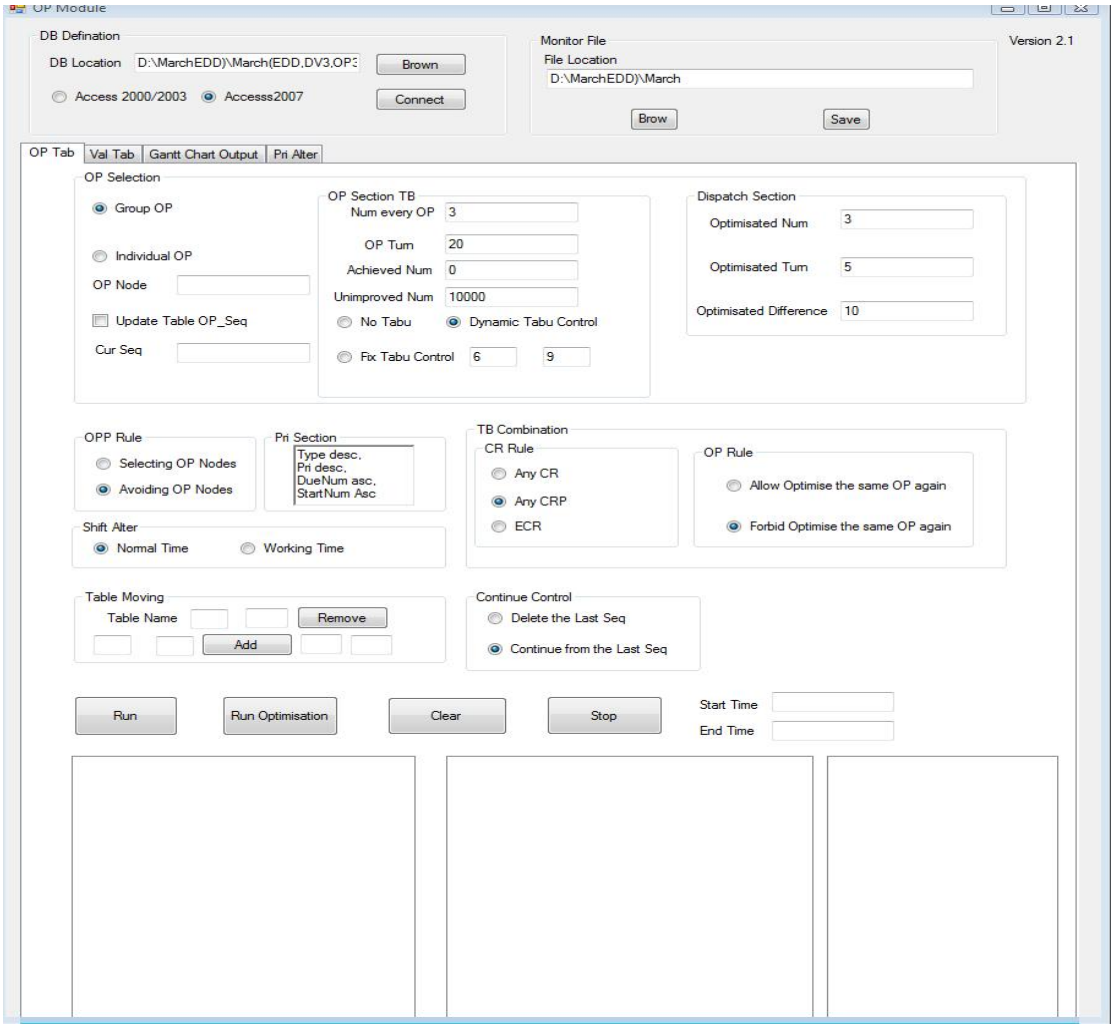


Figure D3 Interface of an optimisation block