DOCTOR OF PHILOSOPHY

Modelling and Optimisation of a Micro Brewery Production Process

Shen, Zhonghua

Award date:
2015

Awarding institution:
Coventry University

Modelling and Optimisation of a Micro Brewery Production Process.

Shen, Z.

Modelling and Optimisation of a Micro Brewery Production Process

By

Zhonghua Shen

December 2015

Coventry University

Modelling and Optimisation of a Micro Brewery Production Process

By

Zhonghua Shen

December 2015

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

Control Theory and Applications Centre

Coventry University

Acknowledgements

Undertaking this research has been a truly life-changing experience for me, and it would not have been possible to progress without the support and guidance that I received from these people.

First, I would like to express my most sincere gratitude to my first principal supervisor, Professor Keith Burnham, for his invaluable advice, support, guidance, patience and kindness throughout the course of this study, which has made the time as both a very wonderful learning and researching experience in my life.

I would also like to express my most sincere gratitude to my second principal supervisor, Dr. Leon Smalov, for his insightful comments, suggestions, encouragement and constant support throughout my research.

Lastly I am also grateful to my supervisor, Dr. Saad Amin and my colleagues of Control Theory and Applications Centre, for their assistance, encouragements and friendships. Special thanks are given to Walid Allafi and Ionut Gheorghe, for all the good times we have shared together.

Abstract

Production scheduling is a decision-making process that is applied in the manufacturing and service industries to achieve efficiency, minimise production costs and maximise the profit. Production process planning and scheduling are critical functions for the sustainable development of manufacturing processes that not only minimise the time or cost, but also improve adaptability, responsiveness and robustness. Therefore, effective production process planning and scheduling is imperative in order to achieve sustainable manufacturing. This study presents a production scheduling problem and its optimal solution, for a typical real-life micro-brewery production process, based in Coventry, UK. In the brewery, various orders of product types arrive dynamically to form a queue for production in a variety of vessels with limited capacity. The operation of brewery production is determined by the processing time, the setting up time, the changeover time of each product type and the cleaning time of each vessel. The due date for delivery the product to customers is another important factor. For the brewery production system, a multi-objective optimisation problem of minimising the overall production time in a job shop is considered in this research. A novel optimisation approach for the sustainable process and scheduling is presented.

The objective of the study is to formulate a mathematical model of a scheduling problem and to develop a Simulink model to simulate the scenario of the brewery production system. Subsequently, the primary focus of this thesis is the design and application of meta-heuristics methods, namely, genetic algorithm (GA), simulated annealing (SA) and ant colony optimisation (ACO), to optimise the brewery production system. In addition, it proposes a hybrid method to solve the production problem, which is comprised of an improved GA with the improved SA to minimise the total production time. The advantage of the hybrid method is not only to achieve the combination of the global search capability of GA and the local search capability of SA, but also an effective avoidance of the premature convergence and strengthen the global optimal solution at a higher temperature; at a lower temperature, the hill climbing of the SA can speed up the convergence. The proposed hybrid method is effectively applied to the brewery production system. The result demonstrates that the proposed method provides better performance and effectiveness when it is compared with other heuristic algorithms that include traditional GA, SA, ACO, the improved GA and improved SA.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACO: Ant Colony Optimisation

BB: Branch and Bound Method

EDD: Earliest Due Date

FSP: Flow-shop Scheduling Problem

GA: Genetic Algorithm

IA: Immune Algorithm

JSP: Job-shop Scheduling Problem

MILP: Mixed Integer linear Programming

MINLP: Mixed Integer Nonlinear Programming

NP: Non-deterministic Polynomial-time

ANN: Artificial Neural Network

PSO: Particle Swarm Optimisation

QAP: Quadratic Assignment Problem

SA: Simulated Annealing

SPT: Shortest Processing Time

TS: Tabu Search

TSP: Traveling Salesman Problem

Chapter 1: Introduction

1.1 Research background and motivation

With rapid economic developments and increasingly fierce competition, many companies are pursuing diverse ways to reduce costs and improve their performance and efficiency. The successful competition of manufacturing enterprises largely depends on the product’s supply cycle, quality and after service level. Advanced production management is an important factor towards achieving these objectives. Production scheduling is widely considered an essential component of production management that its task is determined by the equipment processing sequence and processing time, under limited resource constraints, in order to ensure the selected production goal is optimal. Recently, the production scheduling problem has been a research hotspot in modern science and customised production mode is becoming the mainstream of manufacturing. In a sense, whether the enterprise can survive market competition is determined by its management ability to meet the customer demand and satisfaction, especially in beer production scheduling (Huang 2002), which due to the lean operation method is popularised continuously. Effective production scheduling plays a very important role in improving the performance, market response and comprehensive competitiveness of the beer company. Consequently, there is significant interest in beer manufacturing process control and optimisation, and there is currently a great practical justification for the current investigative study.

The beer industry plays a most important role in economic development and also significantly impacts on people's daily life. The British Beer and Pub Association (2015) reported in 2014 that there are over 14000 breweries in the UK, which contributed 22 billion pounds GDP and generated 13 billion pounds in tax revenue, and also the beer industry and pub sector supports almost 900,000 jobs. The forecast of the beer industry is still increasing in the consecutive years due to the government announcement to cut beer duty by 1 penny per pint in 2013, which will boost the development of the beer market in the future. In addition, most breweries are small-medium scale and have general issues that need be improved, such as poor automatic control ability and technology, lack of management skills and techniques in the production process and human resources, high energy consumption, etc. These problems are severely hindering the sustainable development of the productive forces in the beer industry. Subsequently, most companies are still using the traditional manpower scheduling method in production plans that are of low efficiency or are even inefficient in a complicated and dynamic environment. Consequently, it is easy to encounter the phenomena of overstock or even out of stock of products. Therefore, there are many issues that the beer industry faces in regards to improving efficiency for the brewery production system. Zheng, at el. (2011) and Zheng (2008) identified that beer production is not only a very complex batch production system, but there are also many constraints and limitations that need to be considered within the process dynamics, e.g. delays, feedback, uncertainty non-linearity and varying parameters. Beer production processes have the same characteristics of semi-continuous operations as other batch production processes that are based on the sequence of operation and process conditions. Due to the frequent change of products, and the sharing of resources (including time, equipment, raw materials and human resource), this makes all the activities of production and subsequent economic benefits greatly dependent on the production planning and scheduling; thus making the beer production scheduling becomes a very complicated problem. At present, research on the mathematical model of beer production scheduling is still scarce, beer companies still rely on artificial scheduling production, and its effect is not ideal. Moreover, beer production scheduling problems are classified in the non-deterministic polynomial time known as NP-hard problems due to the batch production environment, dynamic change of customer's demand and production scheduling problems (Huang 2002). It has also been identified that most of these problems can be solved by using intelligent control and optimisation methods.

Mathematical optimisation is a process of search and selection of the best fit values which is used to solve various engineering problems. As an important branch of science, it has been popularised rapidly and applied widely in various engineering and manufacturing fields, such as system control, artificial intelligence, pattern recognition, production scheduling, computer engineering, etc. Research on theories and methods of production scheduling is an area with many complicated factors involved as well as it is crucial for improving productivity and efficiency for a company or even an industry. Based on a large amount of literature review, intelligent scheduling methods are the most effective way to solve practical complex scheduling problems (Huang 2002).

Their usage has been increasing in the beer industry, including the use of genetic algorithms (GA), simulated annealing (SA), ant colony optimisation (ACO), tabu search (TS), etc. Starkweather, *et al.* (1992) have applied a GA to solve the multi-objective JSP in a brewery, these objectives contain the minimum average time of inventory, minimum waiting time of customer's orders, etc. GAs have also been applied for optimisation based on the dynamic model of the beer fermentation process (Carrillo, Roberts & Becerra 2001). In the latter, the GA is used to obtain a series of curves at different temperatures during the fixed fermentation time, in order to discover an optimal temperature curve. This ensures where the final amount of alcohol in the fermentation process reaches the maximum and at the same time ensuring that the product concentration is lowest, which protects the beer quality. Subsequently, Xiao and Zhou 2004 implemented ACO to optimise the same dynamic model of beer fermentation process and better optimised results are obtained. Zheng, *et al.* (2011) have applied ACO to optimise beer production scheduling in order to solve the limitations of the traditional scheduling methods. It formulated a mathematical model for production scheduling and results show that the optimised scheme has robustness and practicability. In comparison with intelligent control, intelligent optimisation technology is still relatively scarce in the beer industry, and also there are many practical issues which need to be optimised and resolved.

This research aims to enhance comprehensive production effectiveness for the beer production system in small breweries. The research scenario is based on the actual demand of the beer production process and provides ideal solutions of the existing problems as described in Section 4.2.3. It considers the approach when the randomly placed orders are to be accumulated to form batches for production, and looking for the application of intelligent control under realistic terms. The Matlab/Simulink software is applied to observe and control the operation process of a micro-brewery production system. As a consequence, it is necessary to develop a new model of the production system which takes into account some constraints. The problem brought into focus on the beer production system is where to minimise the time cost that is formulated by a mathematical model. The optimisation problem is solved by using intelligent algorithms that include GA, SA and ACO, whilst providing some effective improvement strategies based on their shortfalls of application.

1.2 Research questions

There are many uncertain and unpredictable factors in the real-life brewery production. The beer production process is time-consuming in terms of brewing fermentation. In a typical micro brewery, there are numerous different types of product to be made in several fermentation vessels. However, the capacities of vessels are limited. Appropriate time management is necessary in order to ensure demand satisfaction. Dynamic orders will be arriving to form a queue for allocation by decision making. Subsequently, orders will be accumulated to meet the required capacities of vessels and to be assigned for the several parallel fermenters to be processed over a given time period. In the meantime, new jobs cannot be added to the batch to process until the current job is finished when the vessel is in operation. Then vessels need to be cleaned after each production. The cleaning time is determined on the basis of the different capacities of vessels. In addition, the changeover time occurs when the new batch changed the different vessels to be produced after the previous batch is finished. Also, the due date is required to hand over the orders based on the customer demands and satisfaction. Therefore, the main problem is how to schedule the sequence of orders to be optimal in order to minimise the production time, while satisfying all conditions and constraints.

1.3 Research aim and objectives

The aim of this research is to achieve a significant improvement in efficiency and performance in the brewery production system to meet varying market demand. On the basis of the development of a modelling approach, algorithms analysis, and optimisation techniques which minimise the production time and maximise the profits are proposed. The research objectives of this study can be divided into specific parts to achieve the aim as follows:

- To investigate the existing literature available on optimisation methods for the scheduling problems of the manufacturing production system in order to give a better understanding of the current problems in this domain and to seek out some possible methods to solve the problems.
- To analyse the rationale and development of the GA, SA and ACO in the various domains.
- To formulate the problems using mathematical models and find out the constraints and conditions.
- To simulate the scenario of the brewery production system for improving performance and efficiency based on the simulation results, obtained by the Simulink model. The following tasks should be achieved.
    - To obtain the result of the sequence of orders

- To obtain the result of the accumulated batch based on the decision making
- To obtain the result of the different routes of operations
- To obtain the result of the total production

- To develop and implement heuristic algorithms to optimise a real-life brewery production system in order to minimise the total production time. The following tasks are considered in this part.
  - To apply the GA, SA and ACO to optimise the brewery production system
  - To modify and improve the GA and SA
  - To integrate the improved GA with the improved SA as a hybrid algorithm
  - To validate the hybrid algorithm as contrasted with other algorithms for optimising a micro-brewery production system

1.4 Contributions of the research

Four main contributions of this research are summarised as follows:

- First of all, an optimisation based on a simulation model is formulated mathematically in Chapter 4 in order to maximise the profits and minimise the costs of the process operation for the job shop scheduling problems (JSP) of a brewery production system. A sequencing of orders for requests for production from the brewery forms the basis of a varying demand which is applied to the business process. A sequence of orders, whilst satisfying constraints on meeting customer demand, is subsequently adjusted to form a basis for developing a model-based control-theoretical approach. This generic model is part of a new approach that it is used to tackle the problems.

- The second contribution of the research is a Simulink model as explained in Chapter 5 which makes use of MATLAB/Simulink to model the scenario of a brewery production system in order to observe the performance and to improve capacity planning which allocates resources optimally and identifies the bottlenecks that include dynamics, delays, feedback, uncertainty and non-linearity due to constraints. The queue of a sequence of batch orders will be changed based on the objective function as formulated in Chapter 4. It will follow a set of conditions and constraints to schedule the sequence of orders to be optimal. This simulation model is considered the most flexible to identify the different situations in dynamic production processes. Initially, it employed the 'first come, first-served rule' accumulating arriving orders to meet the maximum capacity of vessels for the production plan. Moreover, the model of decision making can be changeable in order to alter the resource profile in different ways and observe the changes in the simulation results. In addition, the simulation system can show the numbers of accumulated batches of each product type produced in the same or different vessels.
- The emphasis of the third contribution in Chapter 6 is to achieve an optimal design of the controller parameters with a novel intelligent optimisation algorithm which is proposed to combine an improved genetic algorithm (GA) with an improved simulated annealing (SA) for optimising the job-shop scheduling problem (JSP) in a micro-brewery, in order to minimise the total

production time. It adopts the acceptance probability of SA to improve the convergence of the advanced GA, which improves the computational efficiency and accuracy by real-number encoding and also improves the diversity of the population of the adaptive adjustment of crossover probability and mutation probability. Consequently, the improved GA and SA not only achieve the combination of the global search capability of GA and the local search capability of SA, but also it can help SA to take full advantage of the global information from GA. The convergence of crossover rate and mutation rate is optimised to 0.92 and 0.08, respectively, from initial values of 0.8 and 0.2 respectively.

- The fourth contribution of the research is to validate the novel proposed a hybrid algorithm with the different heuristic algorithms and to analyse the results in Chapter 7. We have applied GA, SA, Ant colony optimisation (ACO), improved GA and improved SA to optimise a typical brewery production system compared with the proposed hybrid algorithm. Comparison results of the brewery production system have demonstrated that the proposed algorithm gives better performance and effective ability to search optimisation solutions. In a typical scenario, it saved approximately 22%, 44%, 24%, 20% and 37%, respectively, in comparison with GA, SA, ACO, improved GA and improved SA in terms of total production time (This is one of three cases).

1.5 Organisation of the thesis

The structure of this thesis is organised into eight chapters. Each of the following chapters is introduced briefly as follows:

Chapter 2: Literature Review on Scheduling Problems for Manufacturing Production Processes

This chapter introduces the theory, concepts and developments of scheduling problems for manufacturing production systems. The scheduling problems are classified and summarised related to this research background of brewery manufacturing production processes. Also, different optimisation methods are compared and critically reviewed, which will potentially be used to solve these problems.

Chapter 3: Relevant Heuristic Algorithm

This chapter introduces both the traditional (mathematical) and intelligent (heuristic). Three heuristic algorithms are discussed and compared that will be applied in this research. It provides the reader with a full and thorough understanding and analysis of rationales and research development. More specifically, the GA, SA and ACO are discussed as there are essential to the following chapters.

Chapter 4: Brewery Industry Investigation and Mathematical Model Formulation

It aims to investigate an overview of the brewery production system. The problems are formulated by a mathematical model. The brewery manufacturing production system will be analysed based on the model-based control-theoretical approach. The approach is based on a dynamical mathematical model of the type commonly used in control systems engineering.

Chapter 5: Simulink Model Building and Simulation for a Brewery Production System

This chapter is dedicated to designing a Simulink model which simulates the scenarios presented in a real-life brewery production system to determine resource requirements and identify bottlenecks that include dynamics, delays, feedbacks, uncertainty and non-linearity due to constraints. The results of the model can clearly demonstrate the production performance that includes sequences of orders, accumulation, decision making and total production time.

Chapter 6: Hybrid Algorithm

This chapter proposes a hybrid algorithm which combines the improved GA and the improved SA to optimise the brewery production system. The GA is improved by the encoding representation and adaptive adjustment of crossover rate and mutation rate. The SA is improved by the improvement of the generator and the improvement of the acceptance probability of temperature drop function.

Chapter 7: Applications of Presented Methods and Algorithms

It makes use of different optimisation algorithms to compare in optimising a typical brewery production system to validate the proposed hybrid algorithm as described in Chapter 6. The result of the hybrid algorithm gives better performance than other algorithms in terms of the total production time.

Chapter 8: Conclusion and Future Work

Finally, the main contributions of this research are summarised. The obtained results in the preceding chapters are compared and discussed. The directions of future research in this area are suggested.

1.6 Publications

Shen, Z., Burnham, K. J., and Smalov, L. 2012. Towards formulating a business process simulation model for a brewery production system: preliminary steps. In *Proceeding of 22nd International Conference on Systems Engineering,* 11-13 September, Coventry, UK, 2012.

Shen, Z., Burnham, K. J., and Smalov, L. 2013. Modelling a complex production scheduling problem. In Proceeding of 13th Polish-British Workshop, 6-9 June, Wroclaw, Poland, 2013 [Abstract]

Shen.Z, Burnham.J.K, Samlov.L, and Amin.S. 2014. Formulating scheduling problem for a manufacturing production system. In *Proceeding of 2014 International Conference on Computer, Network Security and Communication Engineering.* DEStech Pubilications, Inc. (pp. 683-687).

Shen, Z., Burnham, K. J., and Smalov, L. 2015. Optimised Job-Shop Scheduling via Genetic Algorithm for a Manufacturing Production System. In *Progress in Systems Engineering* (pp. 89-92). Springer International Publishing.

Shen, Z., Burnham, K. J., and Smalov, L. 2015. An improved genetic algorithm for optimising a manufacturing production process. In *Proceeding of 13th Annual Industrial Simulation Conference,* 1-3 June, Valencia, Spain, 2015.

Shen, Z., Burnham, K. J., and Smalov, L. 2015. Comparative performance of genetic algorithms, simulated annealing and ant colony optimisation in solving the job-shop scheduling problem, *23rd* for *International Conference on Systems Engineering, ICSE,* 8-10 September, Coventry, UK, *2015*[Abstract]

Chapter 2: Literature Review on Scheduling Problems for Manufacturing Production Processes

2.1 Introduction

Scheduling is one of the most important activities of operation control in manufacturing, as well as service firms (Pinedo 2008). It allocates resources and tasks optimally to be executed within a certain time period in the production of goods and service, whilst also meeting the demand of satisfying the customers. It is playing a most important role in the management level of companies to improve their performance and efficiency. However, the manufacturing production scheduling process is a most important hotspot in the research and is one of the hardest problems in theoretical research. In the manufacturing system, there are a variety of products, process and production levels.

Production schedules can enable better coordination to increase productivity and minimise operating costs. In particular, JSP and FSP are classically the most important problems in the manufacturing (Rodammer & White 1988).

This chapter aims to provide a concise survey of scheduling theories, concepts, and developments in the manufacturing that leads to a general understanding of the production scheduling. Moreover, it analyses the characteristic and general framework of the JSP in relation to the brewery production process by dealing with the classification of related optimisation methods, which will potentially be applied to optimise the beer production. Finally, optimisation in the beer production system is reviewed.

2.2 Production Scheduling

Scheduling plays an important role in most manufacturing production systems and engineering as well as service industries (Pinedo 2012, Suwa and Sandoh 2012, and Cerdá *c.*2006). It is a decision making process that allocates resources optimally to tasks over given time period that maximise the efficiency and to minimise the costs of operations of the companies in terms of some specific performance criterion (Baker1974, Lopez and Roubellat 2008). Production scheduling is also the operational plan for the production process which is the core of the development of the entire advanced production manufacturing to achieve management technology, operations research, optimisation, automation and computer technology, etc. The theory of production scheduling is generally concentrated on the modelling and optimisation of the production process. It can be divided into two aspects: Modelling and scheduling algorithms (Garey, Johnson and Sethi 1976). A production schedule can be identified that resource conflicts, control the tasks of jobs during production, and ensures the raw materials are ordered in time. Wight (1984) identified the two crucial problems in production scheduling which are priority and capacity, such as: what should be processed first? And who is making the decision? Scheduling is also defined by Wight, it is establishing the timing for performing a task and observing in manufacturing company. Scheduling problems have been classified to be NP-hard which has no known algorithms for finding optimal solutions in polynomial time. Some crucial types of scheduling problems have been classified by (Cerdá *c.*2006) as follows:

2.2.1 Static and dynamic

In static scheduling, all production orders and arrival times are scheduled and the job machines are continuously available. On the other hand, it is often triggered by unexpected events when performing the dynamic scheduling in the practical production. Tang (2000) identified four sources of unexpected events: uncertainty in external demand; uncertainty in supply conditions; effect of the rolling planning horizon; and a system effect which is caused by the above three uncertainty sources. Furthermore, the typical events of problems which may occur have included machine breakdown, job priority, cancellation, shortage of materials, operator mistakes and tardiness of individual workers, etc (Li, Shyu & Adiga 1993). The traits of dynamic scheduling are that: firstly, it can generate real-time scheduling online; secondly, it can realise online identification of random disturbance; and thirdly, it can quickly carry out automatic weight scheduling. Dynamic scheduling can also be classified as follows:

- Feedback scheduling (Szelke & Kerr 1994): it is a new concept in recent years, there is still no widely accepted definition, which is often associated with dynamic scheduling. Feedback scheduling is a dynamic and stochastic environment; it emphasises the response capability of the environment change, so it can be considered as a type of processing mode or feedback mechanism of dynamic scheduling.
- Adaptive scheduling (Nof & Hank Grant 1991): it is proposed based on the following facts: if the original scheduling has better scheduling performance and robustness, when the disturbance occurs, too frequent re-scheduling is not only unnecessary, but also easily causes system instability and therefore should reduce the rescheduling times to attempt to restore the original scheduling. Adaptive scheduling can be considered as a kind of realisation method of dynamic scheduling.
- Real-time scheduling: for the batch scheduling, the real-time scheduling emphasises the feedback that can operate effectively and efficiently when the conditions change. It is a typical event-driven method (Li, Shyu & Adiga 1993).
- On-line scheduling: for the offline scheduling, the online scheduling requires a production process that is continuous, to make timely decisions on environmental changes. On-line scheduling is a continuous scheduling method (Li, Shyu & Adiga 1993).

There are a variety of dynamic events in the scheduling, (Suresh & Chaudhuri 1993) which have briefly been classified into four parts as follows:

- Related to the production job: it includes the random arrival of jobs; the processing time of jobs is uncertain; the change of delivery time; the dynamic priority and the change of orders;

- Related to machine: it includes the machine damage; limited capacity of machines, machine blocking / deadlock and the conflicts of the production capability;
- Related to the process: it includes the processing delay; the quality and the unstable outputs;
- Other events: the personal problems of the operator, the delay of raw material, defective raw material, and the dynamic processing route, etc.

2.2.2 Flow shop and job shop

Flow shop scheduling is defined by (Seda 2008) such that all jobs pass through all the machines in the same order. There is more than one machine and each job has the same processing operation order which must be processed on each of the machines. A flow shop is illustrated in Figure 2.1. It shows that jobs start to process on machine 1, then machine 2, machine 3, …, to the final machine n. A flow shop means an operation where jobs must be processed on each machine in exactly the same order.

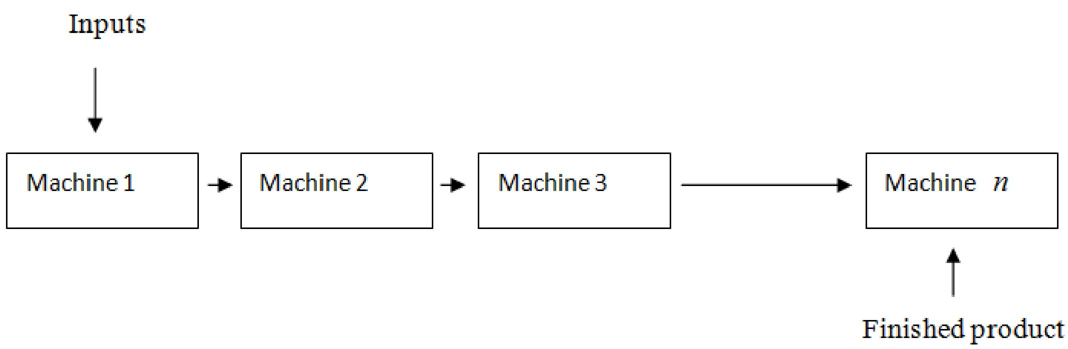

Figure 2.1: A simple flow-shop

Furthermore, a typical job shop can usually be described as: n jobs are processed on m machines, the processing time of jobs in the machine is known, the sequence of production is given for each job to be processed in each machine (i.e. technical constraints), then the operation is required to satisfy with the technical constraints of all jobs to be processed on each machine according to the processing sequencing, the optimal processing performance index can be achieved. An example of a job-shop environment is shown in Figure 2.2.

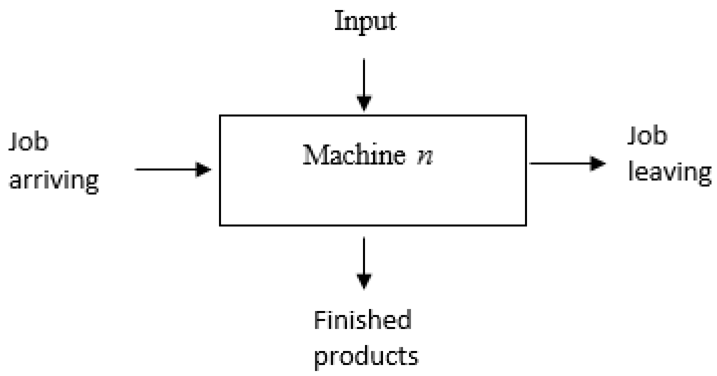

Figure 2.2: One machine in Job shop

The JSP is one of most difficult discrete or combinatorial optimisation problem which belongs to the class of NP-hard problems. Each job can be processed in different machines (Grary and Johnson 1979). In addition, if all jobs have the same technical constraints (Sadeh & Fox 1996; Van Laarhoven, Aarts & Lenstra 1992), then a JSP can be transferred to the simple FSP; if the sequence of jobs of machines is also same, the problem can be further converted to the FJP. So FJP is a simplified form for JSP. JSP have many different descriptive forms, there is usually a linear programming model and disjunctive graph model. The research methods can be divided into two categories: exact algorithms and approximate algorithms. The exact algorithms are mainly applied to the disjunctive graph model and the enumeration method or other methods based on the event scheduling generated mixed integer programming model (Floudas, Aggarwal

& Ciric1989; Abdelmaguid 2009). The approximate algorithms mainly apply to the priority rule of the scheduling algorithm, the heuristic algorithm and the local improvement algorithm based on the local search algorithm, etc. (Panwalkar & Iskander 1977).

2.2.3 Make-to-stock and make-to-order production facilities

In the make-to-stock manufacturing facility, the products are produced for inventory to supply ex-stock before orders arrive based on demand forecasts. The advantages and disadvantages of make-to-stock have been identified by (Chen and Ma 1999) that included a high class of production standardisation, high production efficiency, short time of orders and high inventory levels. It can be referred to as open shop. However, the companies need to be able to forecast demand accurately to determine how much product to be made and stocked. Otherwise, it could lead to excessive inventory and stockouts. On the other hand, make-to-order jobs are produced on the basis of the specific due date, size and quantities by the customers (Cerdá *c.*2006). On the contrary, make-to-order is summarised as a low degree of standardisation, low production efficiency as well as low inventory (Chen and Ma 1999). It also can be referred to as closed shop.

Moreover, the research of production scheduling is a cross-research field that involves many subjects such as operations research, mathematics, computer engineering, control engineering, industrial engineering and so on. The production scheduling problem is very complex, usually expressed as multi-constraints, multi-objectives optimisation problem. It has been proved that it belongs to the NP-complete problem.

2.3 Scheduling in the discrete and continuous system

The discrete system is a kind of classical complex system; Production scheduling for continuous process needs to meet the requirements of devices, equipment and process conditions (or capacity limit) in advance and to schedule and plan for a variety of feasible products in the time and space which is determined by the product structure, resource allocation, and process route of the production process in order to achieve the goals.

Figure 2.3: Comparison of discrete systems and continuous systems (Cristina 2005)

As indicated above (Figure 2.3) a discrete system is one in which the state variable changes at a discrete point in time and use computational procedures to solve mathematical models. On the contrary, a continuous system is one in which the state variables change continuously over time and use deductive mathematical reasoning to define and solve the system.

There is a very large difference between the continuous process and the discrete manufacturing process, the production object not only has physical changes, and also chemical reactions, such as nonlinear, stochastic, uncertainty, etc. which are difficult to express in the traditional mathematical model (Mockus & Reklaitis 1999). In addition, the continuity and stability of the continuous process require a higher standard, so the continuous process scheduling is more concentrated on the scheduling system strategy, expert system and dynamic scheduling. The study of the mathematical model is usually transformed from the mechanical manufacturing scheduling model, or to the establishment of a model containing equipment. However, these models are either too simple or so complicated that it cannot explicitly describe the production process and it is too difficult to find out the solutions (Pinto & Grossmann 1994). The resulting solution needs to be used as an implementation of the program to be executed after a complex decision-making process and correction, and sometimes a feasible solution cannot be found because of the complexity of the production environment and conditions. Hence, it is not realistic to rely on the classical mathematical model completely for the complex process, but it is an effective way to combine experience and to improve benefits.

2.4 Scheduling in the batch process

Batch production refers to a similarity of product production process that can be classified as either multi-product or multi-purpose (Rippin 1983).

2.4.1 Multi-product

In the multi-product production process, all products are in accordance with the same sequence of operations through the same production approaches. The whole production process consists of several production stages, each stage contains varied equipment, each product requires an order passing through all stages of production, and hence it is analogous to the flow-shop scheduling problems (FLP).

Scheduling of multi-product batch production process has two main aspects: 1) the completion time of the product determines the number and size of products. 2) product completion time is determined not only by the final demand for the product and the impact of the delivery date of the product, but also by the processing capacity of the equipment, storage conditions and operation switching time, etc.

2.4.2 Multi-purpose

In the multi-purpose production process, each product can be produced in one or more production lines for a number of operations. A batch cannot be transferred to a machine to produce the next operation unless that machine is currently idle. It is analogous to the JSP.

In addition to the two aspects of the multi-product batch production process scheduling problem, the sequencing problem of the same products can have the same properties on multiple paths for processing.

2.5 Scheduling methods

In order to commence production, decisions need to be made regarding customer demands. Estimation and forecasting methods are mainly used in Elasyed and Bouncher (1994), which primarily allows companies to predict the future values in accordance with past data. In this regard, it is divided into qualitative and quantitative techniques. The former are utilised for predictions when none or very little historical data are available. This approach is normally used to predict based on historical trends, market research, customer surveys, panel consensus, etc. The latter, i.e. quantitative techniques are used for data forecasting, which has divided the techniques of time-series analysis and structural models. Time-series analysis involves sequences of data collected over time, which are classified into short-range forecasting (from hours to a year), medium-range (from a year to 5 years) and long-range (over 5 years). The structural models are widely used for understanding economic behaviour. The approach can also be used in the research to forecast the demand and to estimate the impact of future product development.

Scheduling problems have been studied for several decades and many approaches have been proposed to solve the scheduling problems. Problems in manufacturing are all highly uncertain and dynamic. The problem of uncertainty mainly refers to the randomness of characteristics and constraints that the range of them can only be determined as most time, but the specific values in a period cannot be determined. The dynamics refer to the property that the characteristics and constraints of problems are changing with time. The values can be determined only in a period, but they will change gradually. In most manufacturing systems, researchers and engineers always simplify the problems of the uncertainties and dynamics to certain values, which will make the design and application of algorithms more convenient. However, the simplification will normally bring inaccuracy and instability. Therefore, there are several methods to deal with the uncertainty and dynamic problems in order to improve the stability and solving efficiency in manufacturing as follows (Tao, *et al.* 2015):

(1) Replicated simulation: This method is mainly for the modelling of uncertainty. It takes repeated measurements to obtain the mean value and variance of uncertain parameters. Then it conducts a number of decisions in a small range around the value to obtain a set of good solutions. It is quite time-consuming if it implementing all algorithms, the inaccurate solutions obtained are often due to only limited tests which cannot cover all situations.

(2) Description with fitting function: This method can be used for the solving of either uncertainty or dynamics. From the mathematical point of view, it obtains the fitting functions of uncertainty or dynamic by capturing the relationship between the actual environment and the variation rules of uncertain or dynamic parameters.

(3) Cyclical forecasting: It is used primarily for the modelling of dynamics. It predicts the variation characteristics of the problems at regular intervals. Predicting rules are also conducted according to some tests or fuzzy relation among problem features and the environment.

(4) Feedback control: This method can be applied to deal with both uncertainty and dynamics. It does not need to analyse the characteristics of a problem and its environment in advance. It refers to the design of an adaptive feedback control strategy in an optimisation algorithm to automatically adjust the decision making parameters

with variant characteristics during the optimisation process. It can be seen that this scheme is generally carried out with multi-period problem simulation.

Effective research and application of scheduling methods and optimisation techniques are most important for achieving advanced manufacturing and improving production efficiency. Improving the production scheduling scheme can greatly improve the production efficiency and resource utilisation, and then enhance the competitive ability of enterprises. There is much published literature which focuses on the new approaches in the last decade for formulating the short-term scheduling problem in order to reduce the computational complexity of the resulting mathematical model, and most of them can be classified on the basis of time representation. However, there is still no exact method and theory for scheduling problems. The approaches proposed can be divided into the classical scheduling methods and artificial intelligence methods.

2.5.1 Classical scheduling methods

The classical scheduling method is mainly applied to the scheduling scheme, and the core problem is one or multiple optima of the objective function. There are three main types of methods for scheduling problems in classical scheduling theory that include, 1) analytical optimisation methods, 2) numerical optimisation methods and 3) heuristic algorithms.

Analytical optimisation is a class of methods which can achieve the optimal scheduling in polynomial time based on the specific scheduling objectives (Zheng 2008). Although the method is very effective, the scope of application is limited for specific problems. However, it is very difficult to find an analytical optimisation algorithm for large-scale scheduling problems in addition to a single and simple scheduling, such as a small number of scheduling problems.

Numerical optimisation methods offer an alternative scheduling method, which is widely used in some feasible scheduling sets. From the point of view of data planning, production scheduling problem can be attributed to the equality constraints or inequality constraints. It can be represented as the mixed-integer linear programming (MILP) or mixed-integer nonlinear programming (MINLP) optimisation model for the optimisation of one or more objective functions.

The traditional method of solving MILP is a branch and bound method (BB), which is one of the few effective methods for solving combinatorial optimisation problems (Patterson 1984). In order to improve the solving efficiency, (Shah, Pantelides & Sargent 1993) various improved strategies have been applied for BB or simplified calculation techniques. (Harjunkoski & Grossmann 2001) proposed the application of the decomposition strategy of the mathematical programming method to solve the problem of the large-scale scheduling. (Ierapetritou & Floudas 1998) proposed a new algorithm based on continuous time to represent the MILP mathematical model, which can significantly reduce the total number of variables, so as to effectively reduce the computation time. Although the mathematical programming method is relatively mature, it can only solve the problem of small scale optimisation effectively. For a large-scale complex production scheduling problem, with the increasing the number of the devices and tasks, the scale of the numerical model is dramatically increasing, then to find the optimal solutions often belong to the NP-hard problem, which it is difficult to solve; heuristic methods have great superiority in this aspect for optimisation.

Heuristic methods are based on heuristic reasoning to ensure the local optimal in the scheduling is in accordance with the decision of the equipment, the status of the task. Kudva *et al.* (1994) applied the heuristic method to generate a scheduling scheme for multi-product batch and semi-continuous enterprise with limited intermediate storage in the case of considering the priority of the order and reducing the switching cost. (Al-Khayyal, Griffin & Smith 2001) proposed a tree-based heuristic method based on a decomposition technique, which is applied to the production scheduling of flat glass. The heuristic method can ensure the local optimum is based on the current point of view, but it is difficult to guarantee the global optimum.

2.5.2 Intelligent optimisation methods

Artificial intelligence methods are a class of approximation methods which are designed to handle hardest combinatorial optimisation problems where the classical methods are not that very effective. Although the classical scheduling problem theory has achieved great development, there is a big difference between the actual scheduling problem and the theory. Due to its need to simplify the scheduling problems in the research, many classical scheduling theories still cannot solve the practical scheduling problem which is difficult to express in the mathematical model. Therefore, how to narrow the gap between the theoretical research and a practical solution becomes a common concern. Since 1980, many scholars have attempted to solve the scheduling problem in the actual application, steering by the theory of scheduling research. Hence, the abundant research results provide a better way for the field of artificial intelligence. Intelligent scheduling method mainly includes expert system method, artificial neural network method, fuzzy optimisation method and biological evolution algorithm, dispatching rules, etc.

- Expert system

Expert system methods form a database through the collection of operational experience, and then to search the optimal online (McBride & O'Leary 1993). Advantages of this method are simple and easy to apply. The disadvantages of this method are hard to collect and cover all of the aspects, as well as hard to quantify, so the method is generally used as a supplementary method with other mathematical programming methods and artificial intelligence techniques.

- Artificial Neural network (ANN)

ANN method does not need to be accurate to the process model, which is the use of process input and output data in accordance with the connection weights of the network, the network can accurately reflect the process characteristics of the time for optimisation calculation (Cochocki & Unbehauen 1993). Generally, the major advantages of using ANN are as follows: 1) it is suitable to be used in a larger amount of data sets; 2) it has the ability to implicitly detect complex nonlinear relationships among concerned variables. 3) it can be used to extract patterns and identify trends that are too complex to be noticed by either humans or other techniques. However, the main drawbacks of ANN are summarised as follows: 1) it is difficult to specify mathematically; 2) it cannot extrapolate the results; 3) it cannot handle uncertainties and cannot interpret the relationship between input and output.

Based on above review, there is not enough data for training ANN in this research and the research problem can be described using a mathematical model. The ANN is better suited for use with a larger amounts of data. As such the ANN is not suitable to be used in this research.

- Fuzzy optimisation

Fuzzy optimisation is an area of soft computing that enables a computer system to reason with uncertainty (Castillo & Melin 2001). The probability of achievement of global optimal solutions is larger when compared with the model free optimisation methods for nonlinear optimisation problems. Fuzzy set theory is focused on the use of language and concept as a representative of the macro function of the brain to solve the ambiguity of the language information in a vague way. The main advantage of fuzzy is contrary to ANN that it is good to handle uncertainties and can interpret the relationship between input and output by producing rules. On the other hand, the disadvantage of fuzzy optimisation that it is tedious; fuzzy rules and membership functions and fuzzy outputs can be interpreted in a number of ways making analysis difficult (Zheng 2008). However, there are too many situations in the real world that it is difficult to decide in an unambiguous manner. So it is not suitable to adapt to changing situations as in this research.

- Dispatching rules

Dispatching rules have been applied consistently to scheduling problems. They are procedures designed to provide good solutions to complex problems in real-time. The terms, dispatching rule, scheduling rule, sequencing rule, or heuristic are often used synonymously (Panwalker and Islander 1977; Blackstone, Phillips and Hogg 1982; Baker 1974). Dispatching rules have been classified mainly according to the performance criteria for which they have been developed. A basic dispatching rule is a rule that prioritises all the jobs that are waiting for processing on a machine. The prioritisation scheme may take into account jobs' attributes and machines' attributes as well as the current time; a dispatching rule inspects the waiting jobs and selects the highest priority job next to process whenever a machine is idle. Dispatching rules can be classified into static and dynamical rules (Wu 1987). A static rule is not time-dependent but just a function of the job data, the machine data or both (EDD-earliest due date first, SPT-shortest processing time first). Dynamical rules are time-dependent since they also take into account, in addition to the job and machine data, the current time (Example: minimum slack time-first). Dispatching rules can also be categorised into two classes: local and global rules; a local rule uses only information related to either the queue or the machine and work centre to which the rule is applied. A global rule may use information related to other machines, such as either the processing times of the jobs or the current queue length on the next machine. In addition, dispatching rules has a number of advantages as follows: 1) it is easy to implement; 2) it can find a reasonably good solution in a relatively short time; 3) it obtains optimal for special cases. The disadvantages of dispatching rule also classified that included limited use in practice and it can find unpredictably bad solution. According to advantages of dispatching rules, some of important methods can be employed in the model for production scheduling that includes priority, EDD, SPT.

- Evolutionary algorithms

Evolutionary algorithms is widely applied in many fields and have many developments in the basic theory and applied research (Kim, Jung & Lee 1996). GA and evolutionary algorithms are optimisation methods based on principles of inspired by nature and can be viewed as searching algorithms since they explore a space using heuristics approaches. They can also be used to optimise a general objective function. However, there are still many problems which need further study to solve them, such as proof of convergence, to avoid the premature convergence problem, to deal with complex constraints, environmental parameters selection method, etc. In order to solve these problems, the intelligent optimisation methods are widely used in many domains. The application of the process of intelligent optimisation algorithms in manufacturing engineering has been presented by (Tao, *et al.* 2015). It mainly consists of five parts as shown in Figure 2.4, including problem modelling, variable encoding, operator design, simulation and algorithm implementation. Also, it is emphasised that problem modelling and variable encoding are the most critical parts of algorithm application. Thus, the design of the operator in the algorithm depends largely on the specific environment and ways of coding.

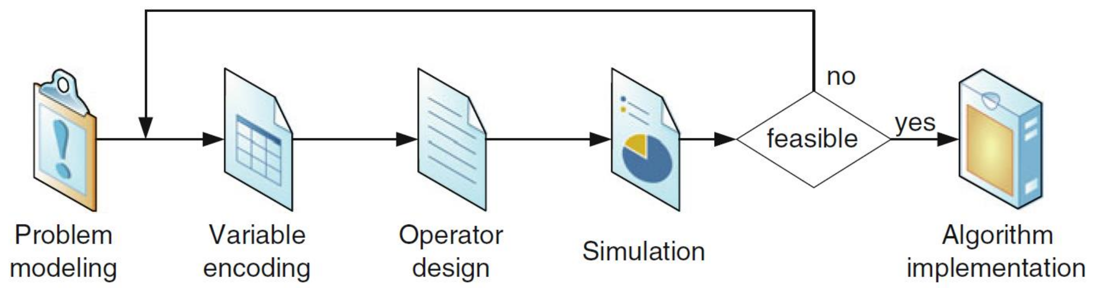

Figure 2.4: The application process of intelligent optimisation algorithm

- Process modelling: the core of the modelling uses variables and formulas to concisely and comprehensively express the problems in accordance with the environment and requirement that include variables, objectives and constraints. Based on literature review, environmental parameters and the relationship between variables should be given in concise mathematical expression.

- Variable encoding: encoding scheme is the relation between problem and intelligent optimisation algorithm. It is the solution space of problem based on operators in the algorithm. There are different searching capabilities of the algorithms based on the levels of randommness by different encoding schemes.
- Operator design: operators need to be selected and designed with population-based iteration in accordance with the above encoding scheme, such as crossover, mutation and so on. It decides the evolutionary direction of the population and the whole searching method of the algorithm. Different kinds of operator have different abilities of exploration and exploitation to be suitable for the different sorts of problems.
- Simulation: it is the most effective way to verify the performance of algorithms with the theoretical analysis. Moreover, parameters need to be modified based on several experiments. The algorithm can be adopted and applied if the expected performance is reached; otherwise, the encoding scheme or the operators need to be reanalysed and adjusted for the specific problem.
- Algorithm implementation: the algorithm can be developed in practical systems for application after the process of the design and simulation.

Although the mathematical basis of the intelligent optimisation method still needs to be improved, the intelligent optimisation method has been widely used to solve complex industrial optimisation problems in different domains, such as the beer industry. The beer production process is extremely complex, with many constraints and limitations. Although the traditional control and theoretical optimisation procedures are relatively mature, the application of the brewery production system is a non-trivial task. Not surprisingly, there are many intelligent optimisation methods which have been applied in the beer production processes in the literature.

2.6 Optimisation in the beer production system

With the development of artificial intelligence, many intelligent optimisation algorithms have been proposed, such as GA, ACO, SA, TS, neural network (NN), particle swarm optimisation (PSO), Immune Algorithms (IA), etc. These algorithms have been developed to provide new ideas and methods for solving complex problems via simulation or explanation of some natural phenomena or processes. Intelligent optimisation has been identified by (Hopfield 1982) which can effectively solve the problem of combinatorial optimisation, such as TSP (Travelling Salesman Problem), QAP (Quadratic Assignment Problem), JSP (Job-shop Scheduling problem), etc. (Shah, Pantelides & Sargent 1993; McBride & O'Leary 1993; Ankenbrandt 1994; Dasgupta & Forrest 1999; Tsai & Tsai 2002).

The optimisation of beer fermentation control has been reported by (Xiao and Zhou 2004) which applied the ACO to optimise the process. However, the authors only focus on a series of different temperature profiles for the mixture during a fixed period of fermentation to model and simulate the system. The optimal results are reported to be readily obtained. Similarly, a mathematical model of the temperature controlling system of a fermentation process has been created by (Wang 2005). The advantages reported in this paper are focused on a brewery company to develop the scenario of a beer production system; the mathematical model is constructed to allow the fermentation temperature to be detected. (Chen and Hu 1992) analysed the beer production as a kind of multi-segment and multi-species batch production process and combined the expert knowledge to develop a hybrid optimisation scheduling strategy for production. In addition, Zheng, et al. (2011) and (Zheng 2008) have applied the ACO to optimise the beer production process. The former is just concentrated on beer production scheduling. The latter, not only applies the intelligent algorithm to optimise the scheduling of beer production, it also provides studies on the automation technology and the applications of the beer production system. Furthermore, an integrated control comprising a fuzzy control system and a PID control system has been developed. It successfully detects the filtering process and discusses an auto-control system for the process. It also compared different intelligent algorithms to optimise the beer production process that included ACO, GA, SA etc. The cost and potential uncertainty issues of beer production are additionally considered for future work. The intelligent algorithms need further research and analysis before they can be implemented.

Manufacturing processes of the future will be more concerned with profits as businesses become tightly squeezed. In this regard, (Shi 2006) identified that the beer saccharification processing auto-control system can be analysed in either the hardware or software aspects to optimise the techniques. The PID algorithm has been applied which is a primary control method in the paper to modify and adjust parameters for the achievement of system performance in terms of profit. In addition, Yan, et al. (2009) illustrated that a small brewery production process can be modelled using the Flash 8.0 platform and ActionScritpt 2.0 software to model and simulate the entire process. Furthermore, a number of decomposition approaches have been proposed by (Wu & Ierapetritou 2003), which are used for a solution for the short-term scheduling problems.

Moreover, an integrated architecture of integrated information system (ARIS) and unified modelling language (UML) method has been proposed by (Ren 2010) to model the life-cycle for the brewery. It used a genetic algorithm and Matlab to optimise the three kinds of energy consumption for water, electricity and steam.

From the review of the literature, it is found that various researchers have applied numerous optimisation techniques in the manufacturing production system for partitioning optimisation problem with mixed results. Simplified versions of the problem can be solved exactly by the early approach. However, it is very difficult to find any exact solution for real problems, which are too large and complicated. Heuristic method was then devised to find good solutions, or find simply feasible solutions for the really difficult problems. Therefore, most research now consists of designing better heuristic solutions for specific instances of scheduling problems. In this present work three popular optimisation methods of GA, SA and ACO have been applied to the beer production scheduling problems. These three methods are widely used to apply in optimisation problems and will be discussed in Chapter 3.

2.7 Conclusion

In this chapter, the history and concepts of production scheduling have been described. It is leading to understanding the existing production scheduling systems and problems, to find out the ways to improve them. The chapter covers not only techniques used to support decision making in real-life production scheduling, but also the intelligent optimisation methods in the production scheduling problems. Based on a comprehensive survey on production scheduling problems, JSP is representative of the production scheduling domain which is more and more close to the actual production in that the character consists of randomness, dynamic, uncertainty, constraint, multi-objectives, etc. This research concentrates on business process modelling and optimisation for a micro brewery production process.

Chapter 3: Relevant Heuristic Algorithms

3.1 Introduction

There are many different optimisation methods that are used for rescheduling problems in the variety of scheduling environments. It is well known that for NP-hard problems, e.g. job shop scheduling problems (JSP), is one of most difficult discrete or combinatorial optimisation problems in the planning and managing of manufacturing processes, which belongs to the class of (non-deterministic polynomial time) known as NP-hard problems (Garey, Johnson and Sethi 1976), which does not generate form explicit solutions for JSP. Heuristic methods have been identified by (Seda 2008), which could provide an optimal solution for complex systems by using genetic algorithms (GA), simulated annealing, Tabu search, etc. Algorithms will be used to find values of discrete or continuous variables that optimise system performance or improve system reliability. Sun, Cheng and Liang (2010) also complemented this approach and have identified two main classes of meta-heuristics. One is the construction and improvement heuristic (Tabu search, simulated annealing, etc.), and another is the population-based heuristic (GA, particle swarm optimisation (PSO), artificial immune system and their hybrids, etc.). However, seeking a suitable intelligent algorithm for large-scale parallelism becomes a major research goal in relevant disciplines based on the view of practical engineering problems, such as complexity, constraints, nonlinear, multiple minima, difficulties in modelling, etc.

Due to its good versatility and independence, intelligent optimisation algorithms has largely shortened the time of decision-making in large-scale optimisation problems of manufacture. However, lower searching time often conflicts with the searching accuracy in most cases. To improve the problem solving capability, research in intelligent optimisation algorithm based on different domain characteristics never stopped. From the view of manufacturing production scheduling, this chapter classified and comprehensively analysed the basic concept, basic principle, rationale, convergence, character application features and research development of the optimisation methods that included the GA, SA and ACO.

3.2 Traditional optimisation methods

3.2.1 Linear programming

Optimisation methods are used to find the best values of decision variables for certain types of models which can be either linear or non-linear. Linear programming (LP) is an extremely powerful tool in modelling many applications when used to solve various optimisation problems represented as a mathematical model. Many scheduling problems can be formulated in traditional linear or integer programming form. These problems may be defined as the problem of maximising and minimising a linear function subject to linear equality and linear inequality constraints. Each optimisation problem consists of three elements: *decision variables*, *objective function*, and *constraints*. Decision variables are the variables in the model that represent production levels, transportation levels, etc. which are under the control of the decision makers. The objective function can be the result of an attempt to express a business goal in mathematical terms that needs to be either minimised (e.g., cost) or maximised (e.g., profit, income, customer satisfaction). Constraints are restrictive limitations which need to be satisfied by the decision variables.

LP problems that can be expressed as follows:

Minimise

$$\sum_{j=1}^{n} c_j x_j \tag{3.1}$$

subject to:

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, i = 1, 2, ..., m$$
$$x_j \geq 0, j = 1, 2, ..., n$$

or it can be presented in a canonical form as follows:

Minimise

$$c^T x \tag{3.2}$$

subject to:

$$Ax \leq b$$
$$x \geq 0$$

where x is the vector of variables, c and b are vectors of coefficients of the objective function, A is a matrix of coefficients of the constraints.

In general, a method is based on the characteristics of a specific constraint formulation, such as single model task only; and the objective function, such as strictly integer values (Davis 1985).

3.2.2 Constrained and unconstrained optimisation

3.2.2.1 Constrained optimisation

There is an important method to solve the constrained optimisation problems, which is to obtain the penalty function method for solving a series of unconstrained optimisation problems (Box 1965). An unconstrained optimisation method is used to solve the constrained optimisation problem, and the feasibility of the iteration point is also required to decrease the value of the objective function. Penalty function method of the unconstrained optimisation methods will execute punishment to the infeasible iterative point and to increase the penalty amount with the iteration progresses, forcing the iteration point can be gradually closer to the feasible region; once the iteration point becomes a feasible point, which it is the optimal solution for the original problem. In addition, there are also some other methods which can solve the constrained optimisation problem that include sequential quadratic programming, the Augmented Lagrangian method, the Zoutendijk feasible direction method and the gradient projection method, etc (Rao & Rao 2009). Constrained optimisation involves the optimisation of a process subject to constraints that have two basic types: equality constraints and inequality constraints. Equality constraints define that some factors have to equal constraints; inequality constraints defines that some factors have to be less than or greater than the constraints, normally called upper and lower bounds (Simon 2013).

3.2.2.2 Unconstrained optimisation

The unconstrained optimisation method is very important. This is not only because of many problems in scientific engineering practice, but also that most optimisation problems are transformed into the unconstrained problem for solving (Di Fonzo & Marini 2011; Simon 2013). Such as Newton method, conjugate gradient method, variable metric method, etc.

3.3 Heuristic algorithms

Since the early of the 1980s, some novel optimisation algorithms have been developed by simulation to reveal some natural phenomenon or process development and its principle and content relates to mathematics, physics, biological evolution, artificial intelligence, neural science and statistical mechanics. These provides new ideas and methods for solving complex problems, as NP-hard problems, such as neural network, chaos, SA, evolutionary programming, GA, ACO, tabu search and hybrid optimisation strategies etc. The unique advantages and mechanisms of these algorithms have attracted worldwide attention and set off a wave of research in this field, and have been successfully applied in many areas (Tao, *et al.* 2015). In the optimisation field, due to the algorithm for constructing the intuitive and natural mechanism, which is usually called intelligent optimisation algorithm or modern heuristic algorithms.

There are various characteristics of the GA, SA and ACO have compared as shown in Table 3.1, these three algorithms are to be applied in this research for optimising a brewery production system.

Table 3.1: Comparison of characteristics of GA, SA and ACO

	GA	**SA**	**ACO**
Innovator and Emergence time	John Holland 1975	A Kirkpatrick 1985	Marco Dorigo 1992
Source of inspiration	Evolution principle	The foraging behavior of ant colonies	Physical annealing
Originally purpose	For solving combinatorial problems	For solving combinatorial problems	For solving combinatorial problems
Using memory	Memory less	Memory less	Using memory store amount of pheromones
Population or single solution orientation	Population-based algorithm	Single solution	Population and single based
Parameters	Population size	Annealing rate	Pheromone

	Crossover rate Mutation rate	Initial temperature Cooling factor	Evaporation rate
Convergence	Rapid	Rapid	Slow
Generating initial solution	Random	Random	Random and local search
Finding local optimum	Mutation operator	Deceasing temperature and limiting search space	Accumulation on better solution
Escaping from local optimum	Random search of search space, using crossover operator	Evaporation mechanism	Probabilistic acceptance of non-improving solutions, based on acceptance function and temperature parameters

3.3.1 Genetic algorithms (GA)

Genetic algorithms were proposed by John Holland (1975). It is inspired by the biological evolution of random search algorithm based on natural selection and natural genetic mechanisms for solving both constrained and unconstrained optimisation problems. The procedure of GA is to repeatedly modify a population of individual solutions. At each step, the current population will be selected randomly to be parents which produce the children for the next generation. Over successive generations, the population evolves toward an optimal solution. Likewise, (Mitchell 1998; Shaw *et al*

2000) has defined that GA is a biological simulation in the natural environment of the survival of the fittest genetic and evolutionary process to form a kind of adaptive ability and global search probability. The possible solution of each problem will be considered as an individual (chromosome) of the population that form clusters of each chromosome as encoding, to carry on the appraisal according to the predetermined objective function for each individual, and also to give a fitness value. The algorithm will be based on the fitness value of its search process. Three main types of rules at each step are used to create the next generation from the current population by selection, crossover and mutation of three genetic operators.

3.3.1.1 Encoding

Encoding is the primary problem that needs to be solved by GA. The Holland coding method is binary code, but this simple coding method is difficult to directly describe the nature of the problem in many GAs applications, especially in industrial engineering. Over the past decade, there are some main encoding methods which have been proposed for the special issues as follows (Holland 1975; Gerst 1971; Zhou and Sun 1999):

- Binary encoding
- Gray code
- Real number encoding
- Symbolic coding

3.3.1.2 Initial population and the evaluation of fitness

First of all, it needs to determine the number of individuals in the population, namely the population size ($popsize$), and then generate an initial population randomly, using the fitness function to evaluate the performance of each individual of the initial species as the initial solution which is to calculate the fitness of each initial solution. If the fitness is higher, the individual performance is better, and then it is closer to the optimal objectives, so the definition of the fitness function plays an important role in the GA. In addition, as many GA solutions require a significant amount of computation time to solve some practical problems, it generally has a large population size and needs to apply more substantial genetic and evolutionary operations for many individuals, especially in the calculation and evaluation of the individual fitness of large numbers. It may lead to the low efficient of the evolutionary computation process, and may fail to meet requirements of the computation speed. It is recognised that there is the possibility of parallel processing of the GA. Hence, a number of parallel GAs have been proposed in past decades (Tomassini 1995; Konfršt 2004). These algorithms have obtained even better optimisation quality than the classical GA.

3.3.1.3 Selection

Selection is to select the superior individual for producing the next generation based on the size of fitness, so it guarantees the population of the evolution. Selection operation is the operation to select the superior and eliminate the inferior, the survival of the fittest individuals of the population. The Higher fitness of individuals has higher probabilities to select to a large group in the next generation; for the lower fitness individuals, it has lower probabilities to select for the next generation. The task of selecting operation is to select some individuals from the parent population in accordance with some methods as follows:

- Roulette selection

Roulette is one of the most commonly used methods among the various methods. (Goldberg 1989) have explained that if all the individual in the population is placed on the roulette wheel according to their fitness value, then the higher the fitness of the individual, the more probability of selection to produce more offspring, whereas the lower fitness individuals have less probability of selection. The specific procedure can be stated as follows:

1) To compute each fitness of chromosome in the population, the fitness can be denoted f_i, where $i = 1, 2, ..., \mathrm{M}$, M is the size of the population

2) To calculate the sum of all fitness of chromosome, it can be obtained as follows:

$$Sum = \sum_{j=1}^{N} f_j \tag{3.3}$$

where, N is the number of individuals in the population

3) To calculate the probability of each individual to be selected for the next generation, the selection probability can be denoted p_i, then it can be obtained:

$$p_i = \frac{f_i}{Sum} \tag{3.4}$$

4) To calculate the accumulation probability of each chromosome, it can be obtained:

$$q_i = \sum_{j=1}^{i} p_j \tag{3.5}$$

where, is accumulation probability of chromosome i, ($i = 1, 2, ..., \mathrm{n}$). For example:

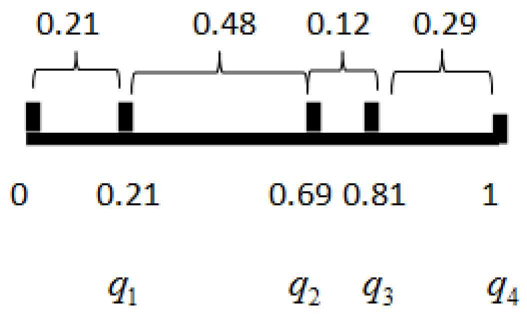

Figure 3.1: Example of the accumulation probability

Figure 3.1 is a basic example which is how to calculate the accumulation probability. In this case, 0.21, 0.48, 0.12 and 0.29 are selection probability of each individual; q_1, q_2, q_3 and q_4 are accumulation probability for 0.21, 0.69, 0.81 and 1, separately.

5) To generate a pseudo random number, denoted r of the uniform distribution between 0 and 1, denoted [0, 1].

6) If $r \le q_1$, select the individual 1; otherwise, to select the individual k, then it requires to meet $q_{k-1} \le r \le q_k$

7) Repeat 4) and 5) M times

- Rank selection

Rank selection sorts the population first according to their fitness value and ranks them. The probability of each individual is selected based on its rank (Baker 1985). The operation process of rank selection can be described as follows:

1) All individuals in the population sort descending order according to their fitness value.

2) To design a probability distribution table according to the specific problem, and to assign each probability value to each individual according to the above arrangement;

3) To apply the selection method of the roulette wheel to generate the next generation population based on the selection probability value of the individuals.

- Tournament selection

Tournament selection is also a more commonly used method by which to select a number of individuals randomly from the population to carry on the tournament, then to choose the winner (best fitness of the individual) to be the parent. The operation can be summarised as follows (Ji 2004):

1) To select the N (In general, N value is 2) individuals randomly in the population to compare their fitness value, and then to select the highest fitness of the individual to the population of the next generation.

2) Repeat the above M times, then the new population can be obtained.

- Elitism selection

Elitism selection selects the best one or more best individual from the current population to the new population. The individual with the highest fitness in the present population does not participate in the crossover and mutation operation and is used to replace the individual with the lowest fitness of the population after crossing and mutation. This method can guarantee that the optimal individual is not destroyed by the crossover and mutation operation, which is an important guarantee for the convergence of GA. On the other hand, it is also easy to lead to the local optimal and individual is not easy to be eliminated; so that the global search ability of the algorithm is not strong. Therefore, this method is generally used in conjunction with other selection operations, which can have a good effect (Ji 2004).

3.3.1.4 Recombination

Recombination is a new method to generate a new individual (chromosome), then to recombine after the selection, the most common method of the recombination that includes crossover and mutation.

- Crossover

Crossover is the operation where two individuals as the parent are chosen by the selection methods to generate, and replace the two new individuals. Crossover operation is the most important feature of GAs, which is different from other evolutionary algorithms. It plays a key role in the convergences of GAs. Chromosome crossover operation is executed in accordance with a certain probability, called P_c, so that it has $P_c \times popsize$ individual for crossover operation. More specifically, each individual will generate a random number r in between 0 and 1, if $r \le P_c$, the individual will be selected for crossover. Then it will randomly match pairs of the chromosome to generate a random number pos (where $pos = 1...m-1, m$. m is the number of genes in the chromosome), pos is a crossover point which is for crossover and replace of an individual gene. Crossover can be described as following ('|' is the crossover point) in Figure 3.2:

Before:

Chromosome 1: **11011 | 00100110110**

Chromosome 2: 11011 | 11000011110

After:

Offspring 1: **11011** | 11000011110

Offspring 2: 11011 | **00100110110**

Figure 3.2: Example of crossover operation at the single crossover points

There are also other methods to make the crossover, such as two crossover points, uniform, etc. specific crossover made for a specific problem can improve the performance of the GA. In the GA, it is necessary to pair the individuals in the population before the crossover operation, and the common matching strategy is random matching. Crossover operator is normally designed to include the contents of two aspects: how to determine the position of the cross point? How to carry out the exchange of genes? Here some kinds of crossover operator have been classified that are applicable to binary coding or real number coding as follows (Barros, de Carvalho & Freitas 2015):

1) Single point crossover; also known as the simple crossover, which is to select one crossover point randomly in the individual encoding cluster, and then to exchange a pair of individual parts of the gene at that point.

2) Two-point crossover; the specific procedure of implementation is to select two crossover points in the pairing between two individuals of the encoded string, and then to exchange the part of the genes at two crossover points.

3) Uniform crossover; this refers to every gene in two pairs of individuals having the same probability to exchange, so as to form two new individuals.

4) Arithmetic crossover; it refers to the linear combination of two individuals in order to generate new individuals.

- Mutation

The mutation operation is defined when some of the gene values of the individual encoding cluster are randomly rearranged from the crossover operations, so as to mutate, and then to obtain a new individual. Mutation is intended to break one or more individual and to jump out of a local optimum to discover a better minimum or maximum space. It maintains genetic diversity and avoids premature convergence on a local minimum or maximum. A mutation operation can also be described by binary encoding as follows:

Before:

Original offspring 1: 1101100100110110

Original offspring 2: 1101111000011110

After:

Mutated offspring 1: 1101110100110110

Mutated offspring 2: 1111111000011010

Figure 3.3: Example of a mutation operation

In the example in Figure 3.3, we have selected one and two random values corresponding to the bit length of the chromosome. In this case, 6 have been selected in the original offspring 1; 3 and 14 have been selected in the original offspring 2. Then simply take the bits from the chromosome and swap them. Chromosome mutation operation is determined by the specified P_m . The design of mutation operation includes two aspects: how to determine the mutation position? How to process the replacement of gene value? Some types of mutation operators have been classified that are applicable to binary coding or real number coding as follows (Barros, de Carvalho & Freitas 2015): 1) Flip bit; 2) Boundary; 3) Uniform mutation; 4) Non-uniform; 5) Gaussian.

All in all, crossover and mutation have both co-operation and competition (Eiben & Smith 2003). Crossover is explorative that to discover promising areas in the search space, such as gaining information on the problem. It makes a big jump to an area somewhere in between two (parent) areas; mutation is exploitative that to optimise within a promising area, such as using information. It creates random small diversions, thereby staying near the area of the parent.

3.3.1.5 Convergence

For the selection, crossover and mutation operation, in order to produce new species, the fitness of the new species is requested to be evaluated. The above steps are repeated until the algorithm reaches a pre-determined condition, or the fitness of the population will no longer increase.

3.3.2 Simulated annealing (SA)

The SA is an extension of the local search algorithm, which is different from the local search based on a certain probability that is given to the value of the neighbourhood state, which is subject to the metal annealing process inspired by the Metropolis criteria and the composition of the annealing process. The earliest SA was invented by (Metropolis, *et al*. 1953), and (Kirkpatrick, Gelatt & Vecchi 1983; Creny 1985) and was successfully applied in the combinatorial optimisation problem in 1983. It proposed a probabilistic as the SA for finding the global minimum of a cost function that may possess several local minima. The process consists of two steps as follows:

- Increase the temperature of the heat bath to a maximum value at which the solid melts;
- Decrease carefully the temperature of the heat bath until the particles arrange themselves in the ground state of the solid;

In the liquid phase, all particles arrange themselves randomly, whereas, in the ground state of the solid, the particles are arranged in a highly structured lattice, for which the corresponding energy is minimal. The ground state of the solid is obtained only if the maximum value of the temperature is sufficiently high and the cooling is performed sufficiently slowly. Otherwise, the solid will be frozen into a meta-stable state rather than into the true ground state (Kirkpatrick, Gelatt & Vecchi 1983).

- The SA is self-adaptive. The basic procedure of the SA can be summarised as follows:
    - Step 1: Initialisation: start with a random initial placement, to initialise a very high “temperature”.
    - Step 2: Move: perturb the placement through a defined move.

- Step 3: Calculate score: to calculate the change in the score due to the move made.
- Step 4: Choose: to choose whether to accept or reject the move depends on the change in score. The probability of acceptance is based on the current "temperature".
- Step 5: Update and repeat: update the temperature value by cooling the temperature. Go back to Step 2.
- The process is done until "Freezing Point" is reached.

SA is a mathematical analogy for the cooling system which can be used to sample highly nonlinear, multidimensional functions. There are many flavours around and the efficiency strongly depends on the particular function to sample. Therefore, it is extremely difficult to make general statements as to what parameters work best. A proof of convergence of SA with general acceptance probability functions has been identified by (Anily and Federgruen 1987), which is applied to a general discrete optimisation problem to prove convergence under the essential and sufficient conditions as follows:

- Reachability of the set of global optimal.
- Asymptotic independence of starting solution.
- Convergence in distribution
- Convergence to a global optimum

A comparison of the traditional iterative optimisation algorithm, the SA has the following characteristics:

- Not easy to fall into a local optimum. It is possible to jump out of the local optimum when temperature of the system is in the non-zero

- The total characteristics of the final state of the system can be seen at higher temperatures by statistical thermodynamics. At the low temperature, it restricts exploration.

The main advantages and disadvantages of SA are summarised as follows (Anily and Federgruen 1987):

- Advantages: high efficiency, flexible, the initial value is robust, suitable for parallel processing, and useful for solving the complex nonlinear issues.
- Disadvantages: SA needs higher initial temperature, the slower decreasing rate of temperature, lower end temperature and multiple samples, then the convergence is slow, processing time takes longer. In addition, it may not obtain the entire/global optimum solution/local convergence if the temperature is decreasing fast.

3.3.3 Ant colony optimisation (ACO)

Initially ant system was developed by Marco Dorigo in his thesis in 1992; ACO has been defined as a meta-heuristic optimisation and probabilistic technique, which will search for the optimal path in the graph based on the behaviour of ants seeking a path between their colony and food source (Dorigo, Maniezzo & Colorni 1996). The composed of a large number of ants group as the collective behaviour actually constitutes a positive feedback phenomenon of learning information: ants passed through a path, then other ants behind have more possibility to choose this path. So the individual ant seeks the shortest path to food based on this information. ACO is based on this characteristic, by imitating the behaviour of ants, so as to achieve the optimum.

Initially, the path is hardly optimal when the program is beginning to search the target, and may even contain a myriad of wrong choices and extremely lengthy. However, the program can search for food by ants in according to the principle of pheromone, and constantly amend the original route, so that the whole route is getting shorter and shorter, and ultimately find the best route. The original algorithm of ACO was specially designed for the travelling salesman problem ((Dorigo, Maniezzo & Colorni 1996; Dorigo & Gambardella 1997). At the beginning, an ant will move from node i city to node j city with probability as according to the equation (3.6).

$$P_{ij}^{k}(\mathrm{t}) = \frac{\tau_{ij}{}^{\alpha}(\mathrm{t})\eta_{ij}{}^{\beta}}{\sum_{l \in \Lambda} \tau_{il}{}^{\alpha}(\mathrm{t})\eta_{il}{}^{\beta}} \tag{3.6}$$

where

P_{ij}^{k} is the probability from the i to j at the t time. ; τ_{ij} is the amount of pheromone on edge i, j ; α is a parameter to control the influence of τ_{ij} ; η_{ij} is the desirability of edge i, j (typically 1/di,j) β is a parameter to control the influence of η_{ij} ; $l \in \Lambda$ is all of the other cities which have not been visited yet.

Moreover, In order to avoid the problem of information overload caused by pheromone, after the end of each ant cycle, it must update the pheromone that to imitate the characteristics of human memory for weakening the old information. At the same time, the latest information of the ant access path needs to be updated according to the equation (3.7)

$$\tau_{ij} = (1-\rho)\tau_{ij} + \Delta\tau_{ij} \tag{3.7}$$

where

τ_{ij} is the amount of pheromone on a given edge i, j ; ρ is the rate of pheromone evaporation; $\Delta\tau_{ij}$ is the amount of pheromone deposited, typically given by the equation (3.8)

$$\Delta\tau_{ij}^{k} = Q / L_k \tag{3.8}$$

where

Q is constant; L_k is the cost of the k^{th} ant's tour, typically length.

The essence of the ACO optimisation process can be classified as follows:

- Selection mechanism: the more pheromone path, the more probability of being chosen.
- Update mechanism: the pheromone of the path will grow with the ants, but also with the passage of time, gradually disappear.
- Coordination mechanism: the ants are actually communicating and cooperating with each other through secretion. Through the information exchange between individuals and their mutual cooperation and they will ultimately find the optimal solution, it having a strong ability to be an improved solution.
- Error mechanism: obviously if the ants are moving in a more pheromone area, it will lead to the problem of the local optimal solution. However, some of them will not go to the more pheromone places, so as to jump out of the local optimal solution, to find the global optimal solution.

Advantages of the ACO can be summarised as follows

- It is versatile; it can effectively solve a lot of problems, such as travelling salesman problem and similar problems
- It has the characteristics of positive and negative feedback in the same time; the characteristics of positive feedback are used to solve the local solution and to rapidly search the optimal solution, and the feature of negative feedback is the evaporation of the pheromone that can avoid the trap of local optimal;
- Inherent parallelism and it can be used in dynamic applications (adapts to changes such as new distances, etc.)

Disadvantages are also presented as follows:

- Theoretical analysis is difficult
- Sequence of random decisions, not independent
- Probability distribution changes by iteration
- Research is experimental rather than theoretical
- Time to convergence is uncertain. It normally takes longer for searching solution as contrasted with other algorithms.

3.4 Conclusion

This chapter has introduced the traditional and intelligent optimisation methods and has given an overview of GA, SA and ACO algorithms. The basic concept, principles rationales have been discussed. In the development of the application of intelligent optimisation algorithms, there are existing contradictions between the optimisation results and computational time due to the computational speed and time constraints. Hence, it is difficult to guarantee the computational results for the global optimum and the optimisation effect is not very ideal. In order to solve these problems, the GA and SA are improved and combined to optimise the complex brewery production system as described in Chapter 6. It is effective to avoid the local optimal solution, to speed up the convergence and to obtain the better global searchability, etc.

Chapter 4: Brewery Industry Investigation and Simulation Model Formulation

4.1 Introduction

A brewery production is a typical batch production process that mainly consists of saccharification, fermentation, filtration storage, and packaging. Each process contains many sub-processes, and they also interrelate with each other. Likewise, the beer production process is the same as other batch processes in that it is neither discrete nor a continuous production process, however, it follows the operational sequences and process conditions for batch production. It has the character of semi-continuity. The brewery manufacturing production system will be analysed based on the model-based control-theoretical approach. The approach is based on a dynamical mathematical model of the type commonly used in control systems engineering.

From the management level, managers need to analyse the market demand and various production conditions, to determine the variety and quantity of beer, as well as orders of each batch. From the analysis of the operational management, the operator needs to decide on the equipment and various parameters of production in each stage, and the scheduling operation can directly affect the overall production capacity of the production line, and may also affect the management decisions. Therefore, the whole beer production of the global optimisation scheduling is a very difficult optimisation problem. In this chapter, we have considered the production equipment as a whole system from the input of raw materials to the output of end product, in order to formulate the mathematical model of the beer production scheduling.

4.2 Overview of the brewery production process

4.2.1 Business process

A business process is defined by (Ruth 2004) as a set of activities in an enterprise which are designed to generate the desired result. Likewise, (Havey 2009) identified that a business process can be described as a transformation of an input into an output and is an organised group of interrelated activities that work together to create a result for customers. Business process modelling refers to the design, analysis and execution of the business process, and a simple transformation model of a business process has been presented by (Laguna and Marklund 2013) as shown in Figure 4.1:

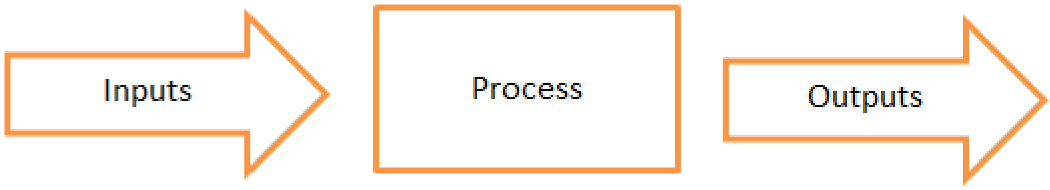

Figure 4.1: The transformation model of a business process

According to Figure 4.1, it shows the importance of a business process in achieving the basic business goals of any company which is basically operated from inputs to carry out a series of processes in order to come up with an output for customer's satisfactions. (Laguna and Marklund 2013) is also expanded the model of a business process that highlights the process network and the significance of resources as presented in Figure 4.2.

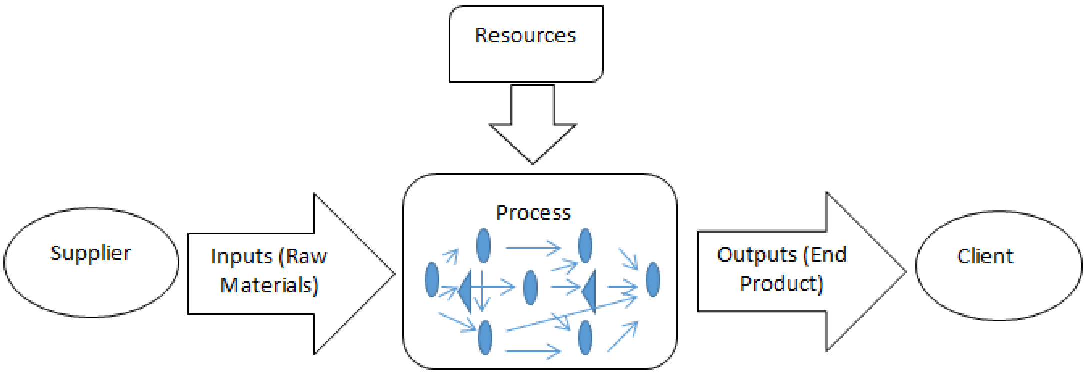

Figure 4.2: The process network of activities

The above diagram illustrates a process network of activities that transforms inputs (raw materials and resources) to process, and then outputs (end product) will be processed and delivered to clients.

4.2.2 Brewery process

The basic beer production process makes use of the constituent ingredients (Wunderlich & Back 2009): barley, malt, sugar and yeast (and possibly others) as presented in Figure 4.3:

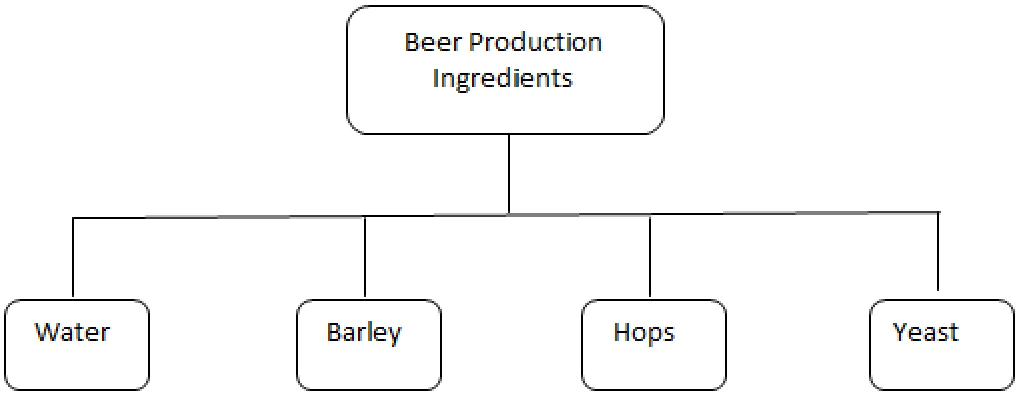

Figure 4.3: Overview of basic ingredients for beer production

The various combinations of these basic ingredients are processed to give various beer product types that include: light, heavy, ale, bitter, draught and stout, etc. Beer products will be sold to customers via various modes, including public bars, wholesale and retail, etc. Breaking down the manufacturing process further, Zheng, et al. (2011) stated that the entire beer production can be divided into saccharification, fermentation, filtering, storage and packaging. A brewery process has been identified by (Sabmiller 2012) as shown in Figure 4.4.

Figure 4.4: The brewery process (SABMiller c.2012)

Generally, all beers contain four main ingredients: barley, water, hops and yeast. Accordingly Figure 4.4 shows that there are many steps in the production of the beer as follows.

- **Milling-** The first step in the brewing process is called milling. The barley malt is passed through a mill that crushes it, preparing the crushed grain (known as grist) to be cooked.
- **Mashing-** In the mashing step, the crushed grist soaks into hot water to convert the starches in the malt to fermentable sugars.
- **Lautering-** Now the beer enters the lautering phase. Here, the sweet liquid, called wort, is separated from the grain solids in the lauter tun.
- **Boiling-** During the boiling phase, the wort is boiled and hops are added to provide the right amount of bitterness and aroma.

- **Whirlpooling**- The hopped wort is then spun in a whirlpool. This separates the spent hops and unwanted proteins from the brew.
- **Cooling-**The wort is then cooled in this process and then moved into a fermentation vessel.
- **Fermentation**- After cooling, yeast is added to ferment the wort that produces the alcohol, carbon dioxide and many of the other flavour compounds. The brewery can determine what character of the beer produces its product range.
- **Filtration**- This step gives the beer the sparkly appearance. For darker brews, filtration creates sharpness in the beer.
- **Packaging**- After passing final inspection, the batch of beer is placed into kegs or bottles and distributed to the final customers.

Beer production belongs to the typical batch production process, thus, it has the main characteristics of the intermittent production process as follows (Bonvin 1998):

(1) The beer production operation follows the sequence of formula rules. Production formula is the required information during the production which includes instrumentation process, methods, etc. The instrument society of America ISA has defined that formula model to include five parts: product name, process, formula, equipment requirements and security.

(2) The discontinuity of batch production. The beer production process is an operation from input to outputs, from the required raw materials to the end product. The whole production process is completed by a series of sequential tasks to be executed. The discontinuity of batch production process is not only reflected in the material, but also the equipment operation.

(3) Non-steady state of beer production status. The continuous production process runs at steady state or close to the steady state. Conversely, as a batch production process, the status of materials and equipment are changeable. Hence, the batch production process identification and system modelling cannot apply the linear approximation model which is commonly used in the continuous production process. Therefore, it can only be used for the nonlinear model based on the actual measured value, or the nonlinear recursive model based on artificial intelligence. The beer production process optimisation operation generally is not a constant steady-state value, but the change over time of the optimal trajectory, for example, the temperature curve.

(4) Shared resource processing. Many units and resources are commonly used in the process of beer production, which requires the control system to have a very good capacity of coordination and distribution. If a shared resource can only be used alone, then the control system would need to be implemented which would prevent the ability that needs to share the resource by two units at the same time, whilst the scheduling system can use the priority queuing method for scheduling. If shared resources are used by several devices simultaneously, it may be necessary to consider whether the resource capacity's needs are met the equipment in use.

4.2.3 Research setting

Based on Figure 4.4, the brewery process shows that there are four main ingredients as inputs via processing to be transformed into beers as outputs. Therefore, the modelling of a brewery production process can be envisaged as in Figure 4.5. The raw ingredients are shown as inputs, as are packaging, cleaning, workforce and energy. The outputs are related to the product.

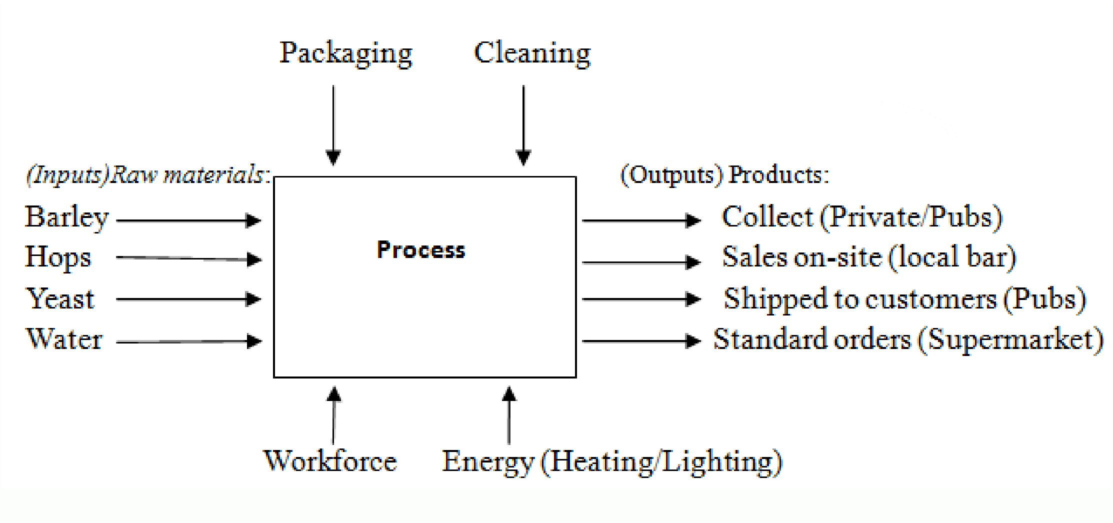

Figure 4.5: Modelling of a Brewery Production Process

It is clear that all of the inputs (raw materials, workforce, energy, packaging and cleaning, etc.) are the costs of expenditure in the system as well as fixed costs (tax, rent, etc.). Furthermore, as almost all companies seek the maximum profits, then the revenue must be greater than the expenditure costs. Denoting revenue as R, expenditure as E and fixed costs as F, it may be stated that for a successful company, R> (E+F).

In addition, outputs have been transformed from inputs. There are several factors and assumptions to be considered. Firstly, the customers may collect the products. Then it would save delivery costs and collection appointments need to be made. Secondly, products could be sold in a local bar that could be considered a sequence of orders. Thirdly, products may be delivered to customers when they are being ordered without collection. Lastly, a standard order may be served by supermarkets. All these types will be considered whilst developing the business process modelling for brewery operations.

Due to increasingly global competition and price erosion, the manufacturing industry is facing an unpredictable environment. Manufacturing is one of the most important implementation of simulation among the various industries (Benedettini 2008). Church End Brewery (CEB), the collaborating company, has provided an interesting challenge for modelling and simulation to optimise their business process operation. There are over 100 different variations of products and approximately 100 different customers (outlets). Some products are constantly being produced and others are produced to order. There is a guaranteed turn-around from order acceptance to delivery. There are numerous constraining factors and costs which are to be met, thus warranting a model based approach to optimisation and production analysis/design/improvement of the brewery process.

4.3 Problem description

After an extensive literature search regarding the beer production, the mathematical model is a priority requirement to be taken into consideration in real-life beer production industrial conditions, which are different to be usual scientific scenarios. In a commercial brewery, it is considered that the beer production process is the JSP. There are a certain number of orders of various beer product types which have been arriving continuously within a given time period and form the queue waiting for production within the limited capacity in terms of fermentation vessels; Each order accumulates towards a batch production, each batch production can only be processed once in each vessel, and also each vessel has to be cleaned after each operation. Therefore, the beer production is a time-based operation of brewing fermentation and other constraints and conditions. The model of a brewery production process has been shown in Figure 4.5, the real-life CEB production process is further described in Figure 4.6.

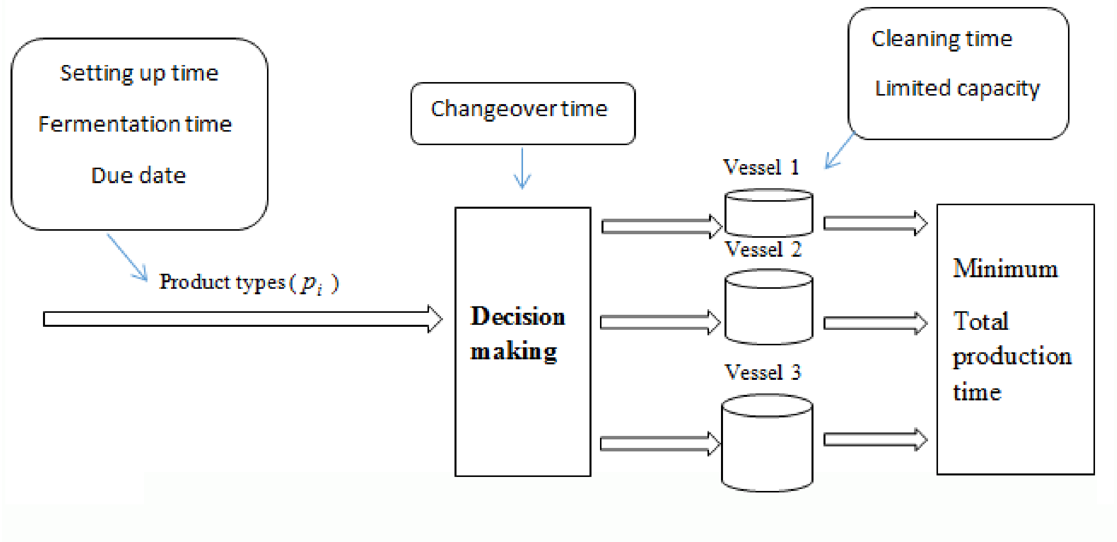

Figure 4.6: Model of CEB brewery production process

Subsequently, the operation of a brewery production is determined by the setting up time, fermentation time, cleaning time and changeover time. The setting up time and fermentation time is fixed for each product type. The changeover time may occur when the next batch production is to be changed to different vessels, and then it requires additional time for vessel cleaning. To formulate the problem it is assumed that the time required to produce different types of beer may be calculated from the duration of the fermentation process. The cleaning times for each vessel will depend on vessel capacity and it is increased by a nominal period when changing over from one type of beer to another, so that when no changeover takes place the cleaning time is a minimum. In addition, the due date required for each product is dependent on customer demand. Customer satisfaction is adversely affected if demand cannot meet the handover date. Also, some orders may need to take priority in production over other orders when the handover is due.

It is considered that a mathematical model based on these problems with the constraints and a control theoretical approach will be derived to provide an initial framework to formulate and solve the problem. Having refined/validated the model framework using data from CEB, the method will be tested.

4.4 Mathematical model formulation

4.4.1 Notation

The following notation is used:

Table 4.1: List of notation

Abbreviation	Definition
T_p	Fermentation time of each product
i	Number of products
j	Number of vessels
x_{ij}	Number of occurrences of operation
c_{ij}	Decision making: if coefficient is 1, working in the same vessel, if coefficient is 0, working in different vessel
T_{due}	Due date for each product
T_{setup}	Setting up time
T_{change}	Changeover time
T_{clean}	Cleaning time
T_1	Production time
T_2	Time of setting up and cleaning
T_3	Time of setting up, cleaning and changeover

v_j	Capacity of vessel j
N_i	Demand for each product
m_1	Number of times of product to be produced as same as previously finished batch
m_2	Number of times of product to be produced as different as previously finished batch
n_1	Number of processing times for product to be produced as same as previously finished batch
n_2	Number of processing times for product to be produced as different as previously finished batch

4.4.2 Objective function

The production period of products can be formulated as follows:

$$T_1 = T_p \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij} \tag{4.1}$$

In addition, even when there is no vessel change, the setting up time and cleaning time is still required before the next batch operation is no vessel change. It can be formulated as follows:

$$T_2 = \sum_{j=1}^{m_1} \left(\sum_{i=1}^{n_1} T_{setup} + \sum_{i=1}^{n_1} T_{clean} \right) \tag{4.2}$$

Furthermore, the changeover time may occur when the next batch production is to be changed to different vessels. Then, it can be denoted as follows:

$$T_3 = \sum_{j=1}^{m_2}\left(\sum_{i=1}^{n_2} T_{setup} + \sum_{i=1}^{n_2} T_{clean} + \sum_{i=1}^{n_2} T_{change}\right) \tag{4.3}$$

Therefore, the objective function can be derived from the above equation (4.1), (4.2) and (4.3) that total production time, denoted T, can be represented:

$$\begin{aligned} T &= T_1 + T_2 + T_3 \\ &= T_p \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij} x_{ij} + \sum_{j=1}^{m_1}\left(\sum_{i=1}^{n_1} T_{setup} + \sum_{i=1}^{n_1} T_{clean}\right) + \sum_{j=1}^{m_2}\left(\sum_{i=1}^{n_2} T_{setup} + \sum_{i=1}^{n_2} T_{clean} + \sum_{i=1}^{n_2} T_{change}\right) \end{aligned} \tag{4.4}$$

4.4.3 Constraints

However, to accurately simulate a commercial brewery production system, certain constraints and conditions need to be considered in the model, such as the due date of a product handover, a production delay, level of order priority, etc. In the actual brewery, each vessel can only produce one batch order for one product type during a production process. This fact can be denoted as follows:

$$\sum_{i=1}^{n} c_{ij} = 1 \tag{4.5}$$

$$\sum_{j=1}^{n} c_{ij} = 1 \tag{4.6}$$

where

$c_{ij} \in \{0, 1\}$, $\forall i \in \{1, 2, 3, \ldots, \mathrm{n}\}$, $\forall j \in \{1, 2, 3, \ldots, \mathrm{n}\}$

The production needs to be completed before or on the due date of delivery due to the lateness will have an effect on customer satisfaction:

$$T_p + \left(\sum_{i=1}^{n_1} T_{setup} + \sum_{i=1}^{n_1} T_{clean} \right) < T_{due} \quad (4.7)$$

$$T_p + \left(\sum_{i=1}^{n_2} T_{setup} + \sum_{i=1}^{n_2} T_{clean} + \sum_{i=1}^{n_2} T_{change} \right) < T_{due} \quad (4.8)$$

Furthermore, each product type produced in each vessel should satisfy the following conditions:

$$x_{ij} \geq 0 \quad (4.9)$$

$$x_{ij} v_j \geq \sum N_i \quad (4.10)$$

4.5 Conclusion

This chapter has fully explored the factors, issues and rationales of a business process that leads to a general understanding of the operation of the brewery production process. The main difference between what already exists in literature and the proposed work is used to be made of a control-theoretical approach. This will allow a better understanding of the underlying physical/conceptual behaviour of a business process due to higher fidelity of the mathematical models employed. Therefore, a new mathematical model is formulated in order to maximise the profits and minimise the costs of the process operation for the JSP of a brewery production system. A sequencing of orders for requests for production from the brewery forms the basis of a varying demand which is applied to the business process. A sequence of orders, whilst satisfying constraints on meeting customer demand, is subsequently adjusted to form a basis for developing a model-based control-theoretical approach. This generic model is part of a new approach which is used to tackle this problem.

Chapter 5: Simulink Model Building and Simulation for a Brewery Production Process

5.1 Introduction

The simulation, modelling and analysis of manufacturing systems are becoming increasingly important for the performance improvement of systems in the last decade (Sandanayake, et al. 2009). This chapter is dedicated to developing a simulation model to observe the performance and efficiency of a complex brewery production system as presented in Chapter 4, which makes use of a production time-based representation. (Bosilj-Vuksic, Ceric & Hlupic 2007) stated that there are different methods and tools that have been used by most companies, which are able to measure the performance of business processes in terms of dynamic systems. The approach makes use of the MATLAB/Simulink environment to simulate the scenarios presented in a brewery manufacturing production system as given to determine resource requirements and to identify bottlenecks that include dynamics, delays, feedback, uncertainty and non-linearity due to constraints.

It is assumed that three products are to be produced simultaneously in three vessels in parallel. There are some important factors to be considered; Firstly, it models a situation where three random sequential orders are received and these form three queues waiting for production. Secondly, it describes an accumulation concept in which products are allocated to vessels so they are working at maximum efficient capacity thus maximising profits; accumulating each daily order to meet the maximum capacity of vessels for each production. Thirdly, it represents how to make an optimal decision regarding which batches of orders are to be produced in each vessel based on the time constraints involved. Finally, the result in terms of total production time for each product type is obtained.

5.2 Business process modelling and simulation for the manufacturing production process system

5.2.1 Business process simulation

Simulation is widely used as a tool for analysing business processes. The term 'business process simulation' has been defined in (Banks, et al. 2000). Essentially a simulation is the imitation of the operation of a real-world processes or systems over time. It is also stated that business process modelling plays a significantly important role for organisations and process improvement that determine the value of the outcome of the production process.

5.2.2 Types of simulation model

There are types of models which have been classified by (Sidnev, et al. 2005) as follows in Figure 5.1.

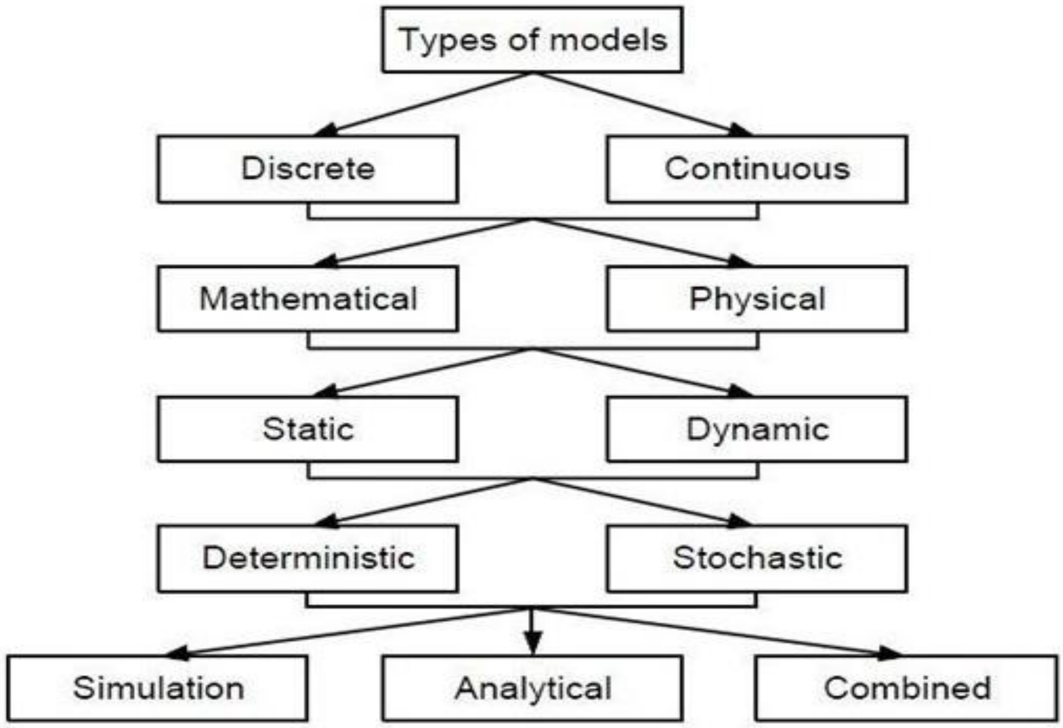

Figure 5.1: Types classification of model

- Static simulation, also called Monte Carlo simulation represents a system at a particular point in time. Dynamic simulation represents systems as they change over time.
- Deterministic simulation contains no random variables. A known set of inputs will result in a unique set of outputs. In contrast, stochastic simulation estimates the true characteristics of the model from random inputs to random outputs.

- Mathematical model uses symbolic notation and equations to represent the systems as opposed to the physical model in converse.
- A discrete system is one in which the state variable changes at a discrete point in time and uses computational procedures to solve mathematical models. Conversely, a continuous system is one in which the date variables change continuously over time and use deductive mathematical reasoning to define and solve the system.

5.2.3 Tools of business process simulation (BPS)

(ProModel 2011) has been reported that "anyone can perform a simple analysis manually". However, with complex analysis, there is an increasing need to apply computer-based tools. There are many simulation tools that are available in the market. (Merkuryev and Pecherska 2005) reported that many universities have adopted simulation software around the world as shown in the following diagram:

Figure 5.2: Occupancy rate of simulation software in universities (Merkuryev and Pecherska 2005)

It is clear that Arena based on Figure 5.2, 2005 data is the greatest implemented simulation tool. According to (Systems Navigator 2012) it is reported that the Arena simulation software is a most used simulation platform with more than 350,000 users in the world. Similarly, (Advantage 2008) stated that Arena is a leading simulation software that has been used successfully by organisations around the world. There are three main advantages that have been reported by (Rockwell 2011), which are important as follows:

- Easier and faster to learn than other simulation tools
- Easier to validate, verify and debug
- Easier to communicate the intricacies of complex processes to others

A number of BPS tools have been discussed and evaluated by (Jansen and Netjes 2010) based on output analysis for capabilities of modelling and simulation, and other possibilities. They have concluded that the FlOWer, FileNet, and Protos are considered unsuitable for real-life realistic BPS studies, and ARIS, Arena and CPN, all qualify for BPS studies. In particular, Arena has been strongly recommended as an appropriate system tool to be used for BPS. However, MATLAB/Simulink is used here due to its versatility and increasingly wide acceptance in industry.

In addition, (MathWorks 2012) reported that MATLAB/Simulink is the leading developer of mathematical computing software for engineers and scientists in industry, commerce, government and education. (Christian and Filippo 2004) identified that Simulink is an extension of MATLAB by Mathworks Inc. The primary interface consists of a graphical block diagramming tool and a set of tailored block libraries. Both MATLAB and Simulink are widely used in control and digital signal processing for multidomain simulation and model-based design. "Simulink is a software package for modelling, simulating, and analysing dynamical systems. It supports linear and nonlinear systems, modelled in continuous time, sampled time, or a hybrid of the two". Also “Simulink is a block diagram environment for multidomain simulation and model-based design. It supports simulation, automatic code generation, and continuous test and verification of embedded systems” (Mathworks 2011). It provides multiple blocks that can be dragged around a workspace and connected through ports with lines and establish an input-output relationship as well as dependencies between those two blocks. There are two main advantages that have been reported by (EETimes 2001)

- The graphical tools are comprehensive and very easy to use.

- Simulink has an extensive control library, which allows any control algorithm, such as linear control, fuzzy logic, neural networks, and others to be easily implemented.

Based on the reviews above, the MATLAB/Simulink package has a wide range of toolboxes that can be chosen. It can be used as a mathematic model for calculation and optimisation, etc. especially in quantitative research. In this research makes use of MATLAB/Simulink to model the scenario of a brewery production system as presented in Chapter 4, which is to allocate resource optimally and to identify the bottlenecks that include dynamics, delays, feedback, uncertainty and non-linearity due to constraints. The development details of all the blocks of a holistic model are discussed which includes modelling of random orders, modelling of the production process, modelling of decision making and modelling of total production time.

5.3 Simulink model of the scenario of a brewery production system

It is well known that a brewery production system is an extremely complex batch production system as described in Chapter 4. Therefore, this chapter will model a typical brewery production system to observe the operation performance and to minimise the production time in accordance with decision making by managers. The decision making will take into consideration three product types to be produced in three vessels in parallel. Three product types can be denoted p_1, p_2 and p_3 separately. The brewery production can be modelled in Figure 5.3.

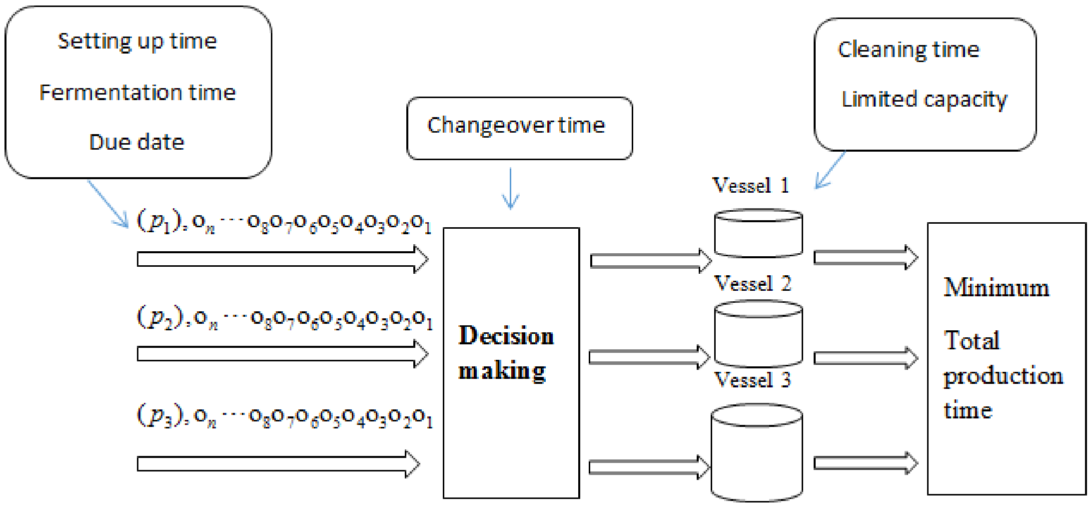

Figure 5.3: Model of three product types of a brewery production system

Orders will be receiving randomly to form a queue for production based on decision maker by management. The sequences of orders are constant and it can be denoted o_n, where $n = 1, 2, 3, ..., n$. Each order will be accumulated to the full capacity of each given vessel. It will be formed as a batch if accumulated back orders are equal to the full capacity of the vessel.

Also, the initial production parameters are assumed as in Table 5.1:

Table 5.1: Production parameters for Simulink model

		p_1	p_2	p_3
T_{p_i} (hours)		72	96	120
T^i_{due} (hours)		7*24	10*24	15*24
T^i_{setup} (hours)	v_1	1	2	3
	v_2	2	3	4
	v_3	3	4	5

Initial parameters are defined in order to help the model which can be revised easily. It is flexible allowing the change of any parameters as the model requires, due to brewery production being based on the fermentation time, setting up time, cleaning time and changeover time. In this case, it is assumed that fermentation time will take 72, 90, and 120 hours respectively; the setting up time is determined by different product types having different setting up time; the cleaning time of v_1, v_2, and v_3 will take 2, 3 and 5 hours respectively; the changeover time may occur when the next batch production is changed to the differing vessels for production, it will be 5 hours delay. In addition, the due date of products is 7, 10, and 15 days, respectively, based on customer demand.

The scenario of the production process is simulated by the Simulink model as following in Figure 5.4. It clearly shows a holistic micro-brewery production system in which four main parts are achieved in the model that includes the sequences of orders, accumulation, decision making and production time. The decision making method is most important to a company. The purpose of this model is to achieve the optimal total production time, whilst satisfying with constraints and conditions. Also, the uncertain changeover time needs to be analysed when it happens after a previous batch is being finished. The following three approached are considered to make decisions in the model:

a) If one product type is unfinished and two product types are finished in the current operation; it then needs to change the vessel to produce that one product type if the surplus of accumulated batch is large.

b) If two product types are unfinished and one product type is finished in the current operation; then it needs to analyse the surplus of the batch in order to make the decision which product needs to change vessel in advance when the production time is short. In addition, the last product type is to be produced according to approach a) until the previous batch finished.

c) If one product type is finished, and there some orders of this product type left that have not been produced because orders cannot meet the required production quantity, then the vessel would become idle. So this vessel will be changed to produce a different product type until orders are accumulated to meet the required conditions.

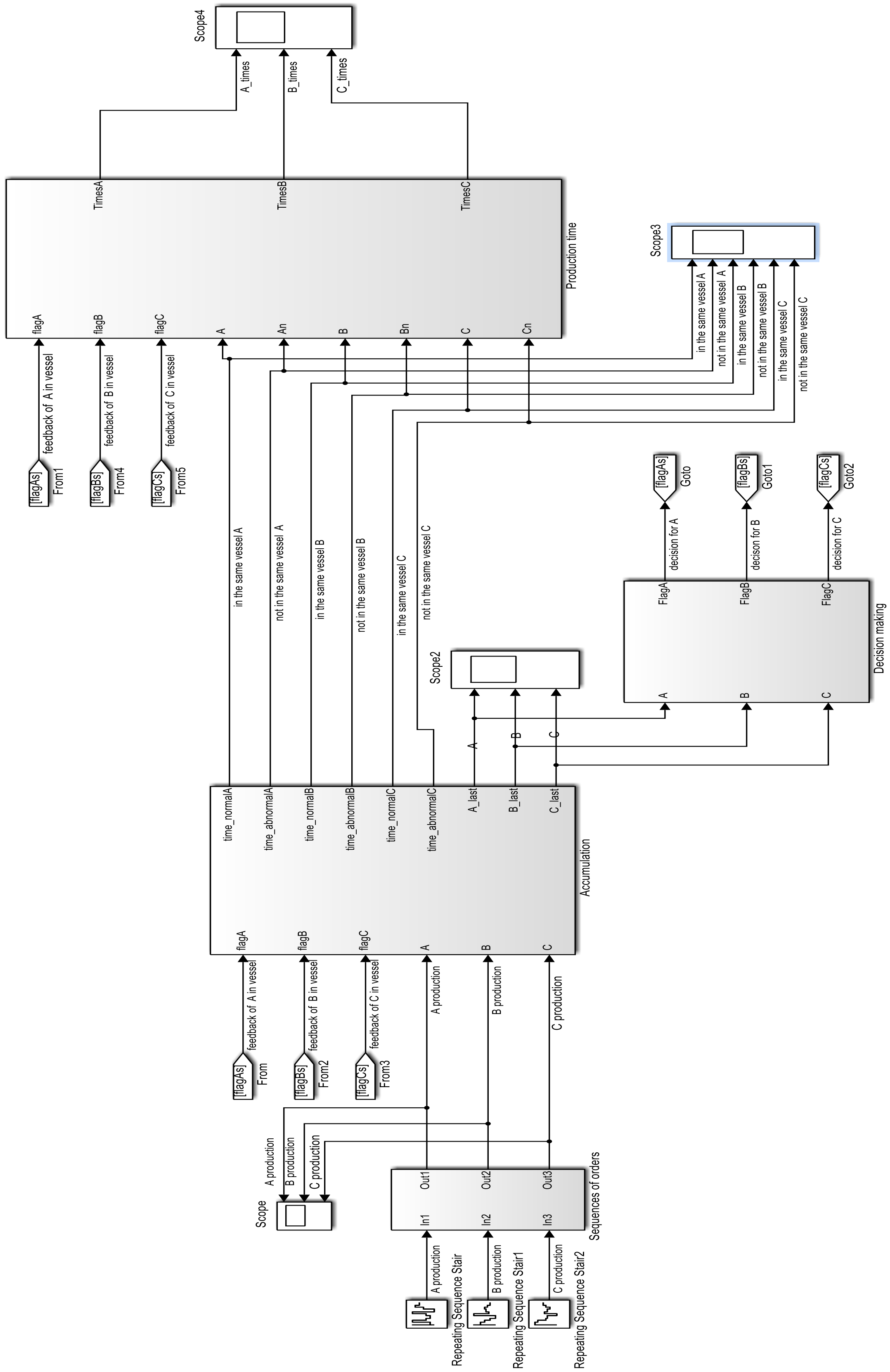

Figure 5.4: Model of a brewery production system

The Simulink model is shown in Figure 5.4. The internal structure of Simulink model can be pictured logically as following in Figure 5.5:

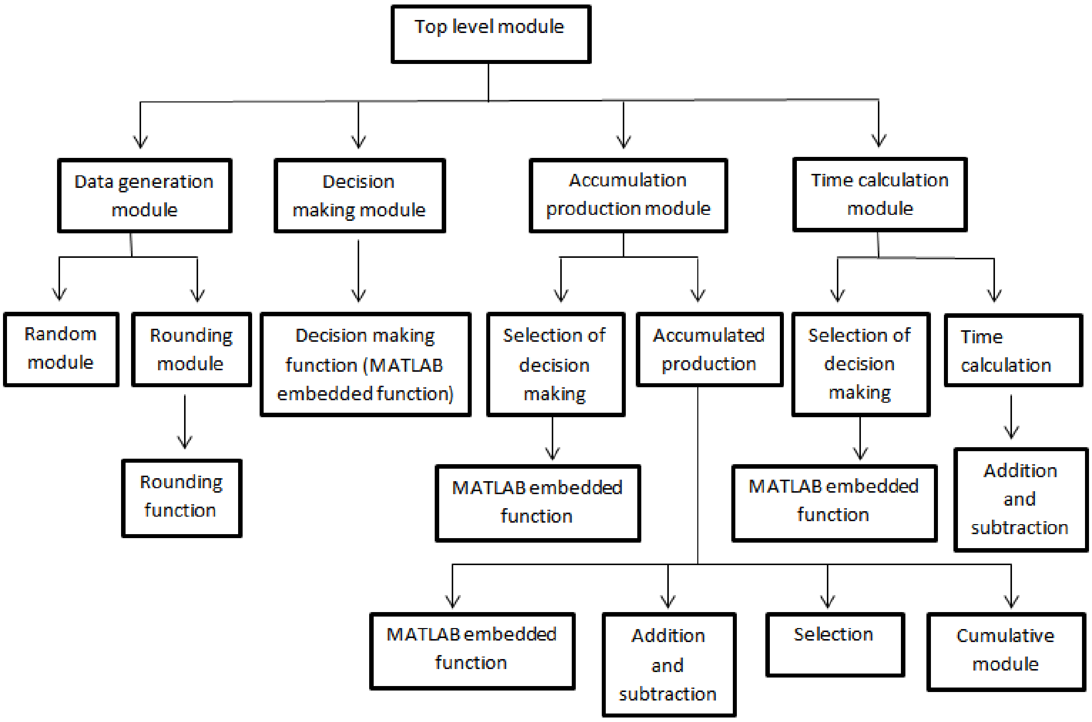

Figure 5.5: The structure of Simulink model

5.3.1 Modelling of sequences of orders

In a brewery, quantities of orders of each product will be received randomly from customers. Then, sequences of orders for three products form queues to be separately produced in the three parallel vessels. This process assumes the use the fixed sequences of orders of 10 days and the sequences of random orders to model a brewery production system as following in Figure 5.6 and Figure 5.7. The 10 days back orders of three products are assumed in Table 5.2. Three product types are denoted A, B, and C, respectively. Each number of product type represented each order which has been received in a day in a 10 days sequence. Such as, 3 barrels of product A are received on the 1st day.

Table 5.2: 10 days of back order

Days	1	2	3	4	5	6	7	8	9	10
A(barrels)	3	8	4	3	4	8	2	5	7	6
B(barrels)	14	12	8	5	19	9	11	12	10	8
C(barrels)	30	27	16	15	4	18	14	11	17	20

Moreover, the brewery will not accept orders that are only part of a barrel, half, or one-third etc. In the subsystem of three sequences of orders, a block of rounding function is applied in this model as shown in Figure 5.8, which can be obtained all of the numbers of orders are to be the nearest integer if the block of the random number generating the number of the decimal fraction.

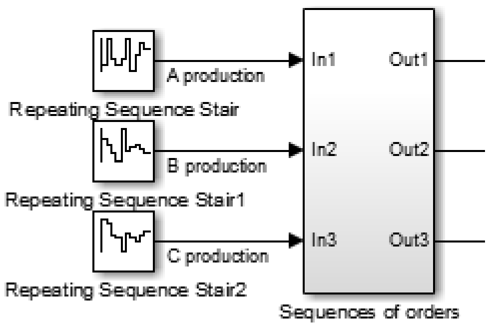

Figure 5.6: Subsystem of three sequences of fixed orders

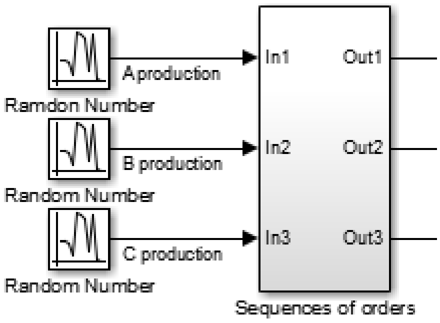

Figure 5.7: Subsystem of three sequences of random orders

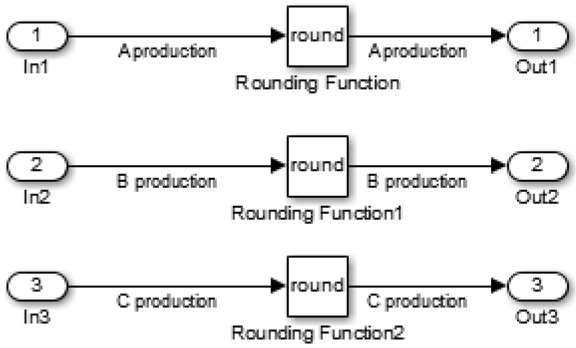

Figure 5.8: Sequences of orders

Subsequently, the result of the sequences of orders can be obtained as shown in Figure 5.9. It applies a block of the repeat sequencing stair which repeats a stair sequence of 10 days of back orders and assumes that it has been specified with the vector of output values parameter. It is easy to picture each quantity of order forming a queue in this stage of the operation. In addition, a block of random numbers is also applied to generate normally distributed random orders as shown in Figure 5.10.

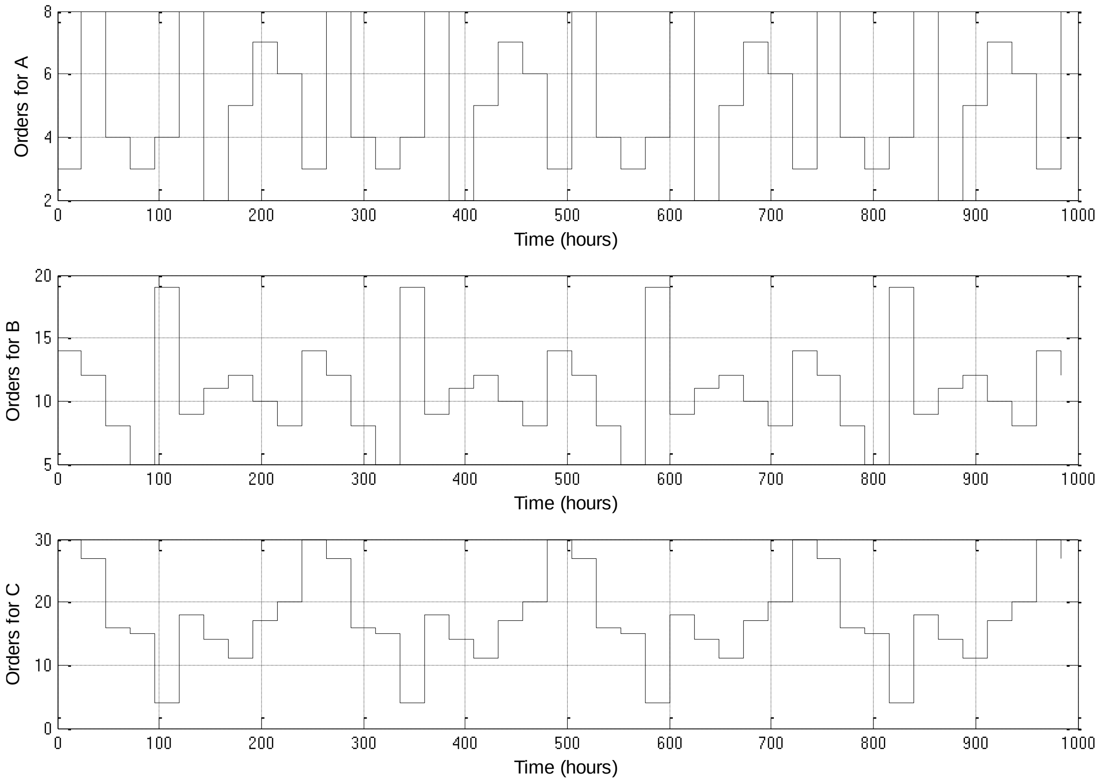

Figure 5.9: Sequences of fixed orders for product A, B and C

The result in Figure 5.9 clearly shows that three products have arrived to form three queues at this stage. The model uses a block of repeating sequencing mix which makes it easy to understand how the production operates for 10 days back orders. It has modelled 10 days back orders of each product repeatedly in 1000 hours.

Figure 5.10 shows how many sequences of random orders of each product have arrived in 1000 hours. The quantities of orders of three products are generated randomly based on the mean value of the Gaussian and random seed is 10, 15 and 20, respectively.

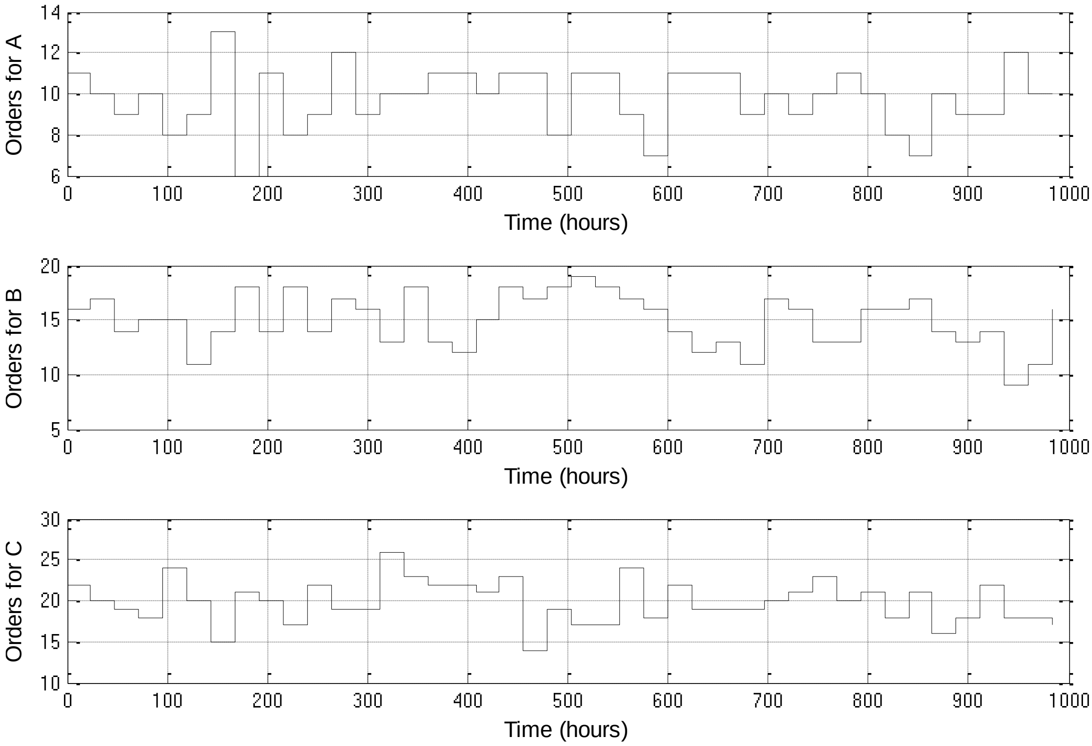

Figure 5.10: Sequences of random orders for product A, B and C

5.3.2 Modelling of accumulation

The production process system is formulated based on the method of accumulation of product orders, which consists of vessel selection and the production plan. These are shown in Figure 5.11, which includes a model of selection and, also, the production model as shown in Figure 5.12.

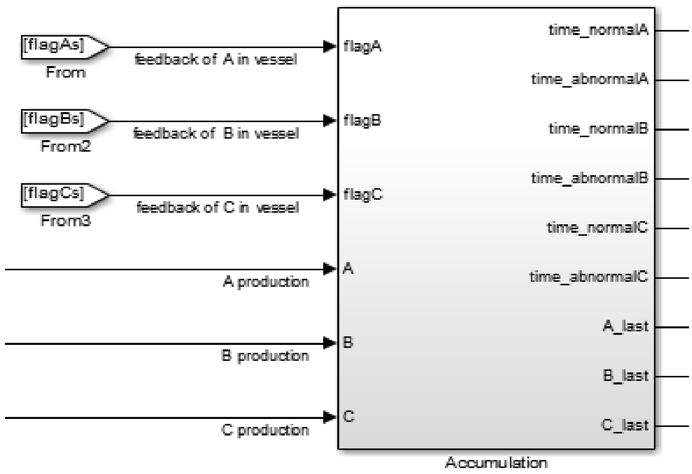

Figure 5.11：Subsystem of accumulation block

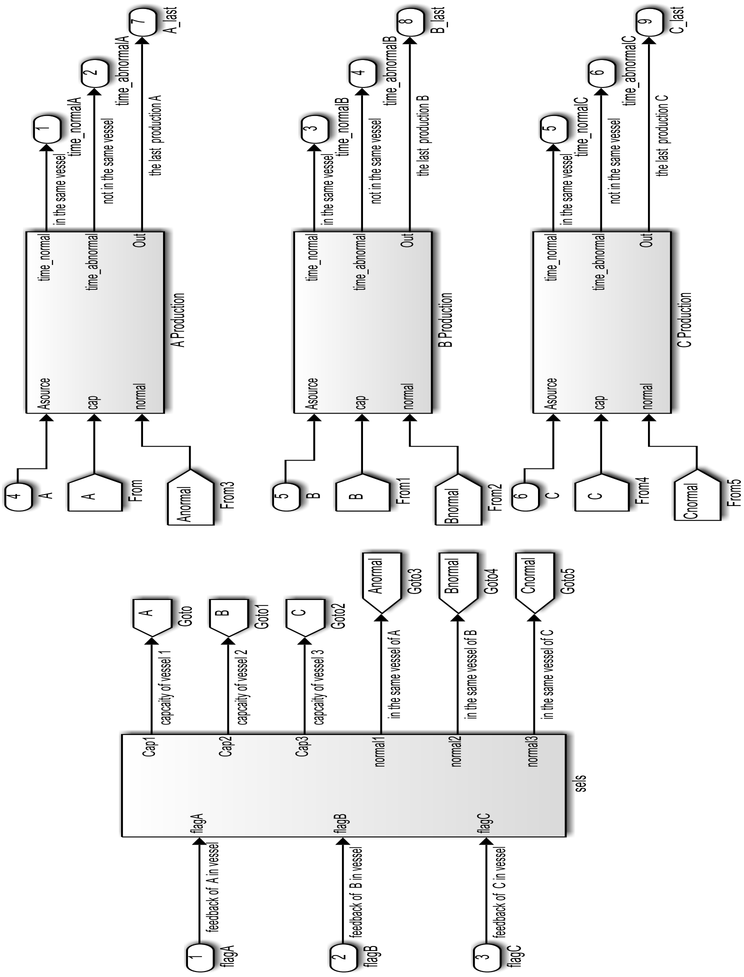

Figure 5.12：Subsystems of vessel selection and production plan

Subsequently, once orders have arrived their production is prioritised according to their due date; otherwise the company could lose customers. Therefore, it can be assumed that quantities of orders gradually add up during each day and are then produced if the sum of orders is equivalent to the maximum capacity of the vessels. In this case, the capacity of the three vessels, denoted v_1 , v_2 and v_3 which are 20, 30 and 50 barrels respectively. So that, if accumulated batch of product is allocated to production in v_1, and then it will subtract 20 of the capacity of the v_1. Similarity, it will be subtracted 30, 50 for v_2 and v_3, separately. Each vessel will be selected to produce a batch of orders based on the feedback of decisions then allocates resources according to requirements. Decision making model will be discussed in the next Section 5.3.3 in details. The selection possibilities for production can be shown in Figure 5.13. The math function of interpreted cap_sel is embedded to interpret which vessel will be allocated to process the next batch of orders; it has different possibilities of operations, such as, A v_1 -B v_2 - C v_3 ; A v_1 -B v_3 - C v_2 and so on.

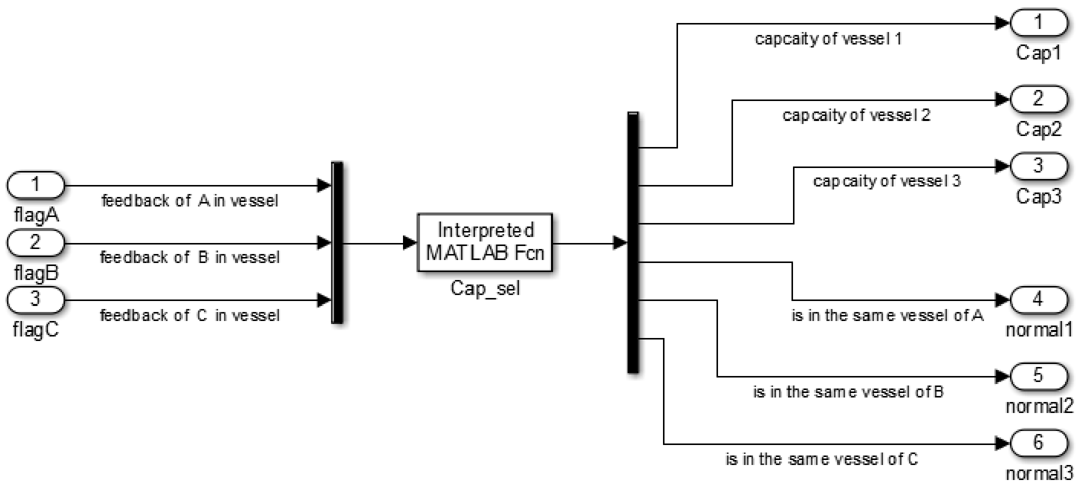

Figure 5.13: Vessel selection for production

According to the model of vessel selection, the batch of orders will determine which vessel is selected for production in the next operation. The production process of each product is formulated as shown in Figure 5.14, to demonstrate that the accumulation of orders is to be modelled. The model will count how many batches of each product are to be made in the same vessel or different vessel.

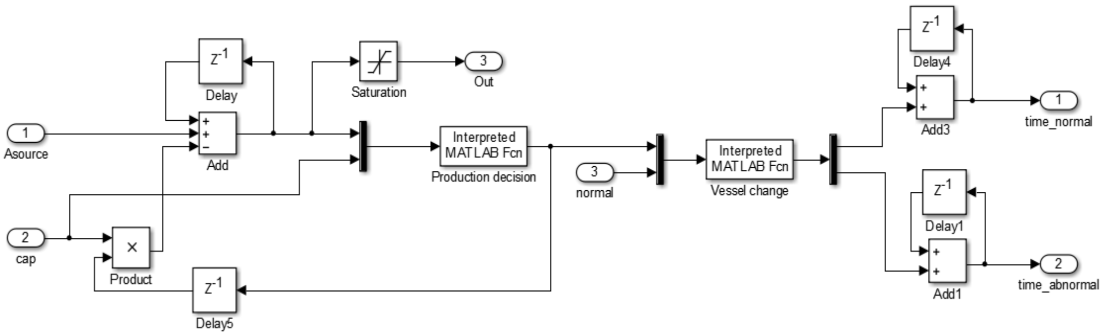

Figure 5.14: Model of product production

The function of p_1, p_2 and p_3 production is to decide whether the next product will be produced or not. It makes use of the Boolean algebra which is denoted 0 and 1. In this case, the production system is according to the feedback of decision making which judges whether has met the required batch of orders; if yes, the signal will be 1 which is processing right now. Otherwise, it will be 0 which is not processing currently. Likewise, the function of vessel change applies the same method accordingly in order to decide which vessel is to be utilised to produce the next batch of orders.

5.3.3 Modelling of decision making

Accordingly, it is most important how to decide the allocation of orders to be produced in the vessel optimally. Based on the fact that most companies are seeking maximum profits, then it will be optimal and efficient if vessels are working near maximum capacity in this case. Figure 5.15 represents the model of decision making where orders are to be made and decided, and then reported as feedback to the model of the accumulation and next model of production time.

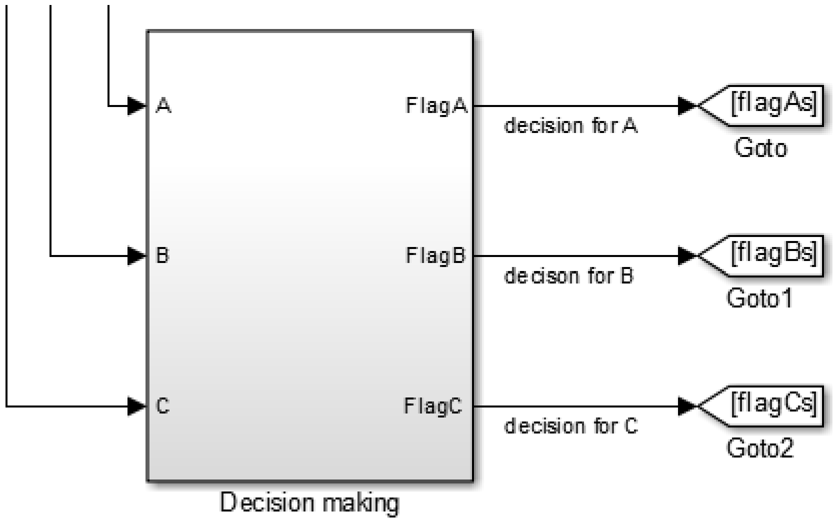

Figure 5.15: Subsystem of decision making

Hence, the idea of accumulation of orders considers to how many batches will be made in the three vessels. There are some constraints on production are concerned as follows: the accumulated amount of orders is greater and equal to 16 and also smaller than 20, then this batch will be produced in the v_1, it denoted $16 \le p_1 \le 20$. Analogically, it can be denoted $26 \le p_2 \le 30$ for the v_2, and $46 \le p_1 \le 50$ for the v_3. In addition, if the accumulated amount of orders is larger than the limited capacity of vessels, the surplus orders will be delayed to the next batch production. Such as, accumulated batch of orders for three days is 25 barrels, the maximum capacity of the v_1 is 20 barrels, then it will take out 20 barrels to produce in advance, the remaining 5 barrels will be delayed to the next batch production. This idea is being used in the model of decision making in Figure 5.16

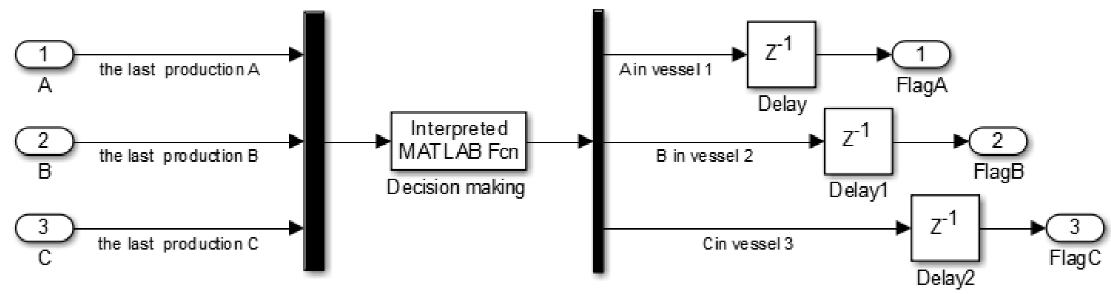

Figure 5.16: Model of decision making

Therefore, the result of sequences of batch orders can be shown in Figure 5.17 as follows:

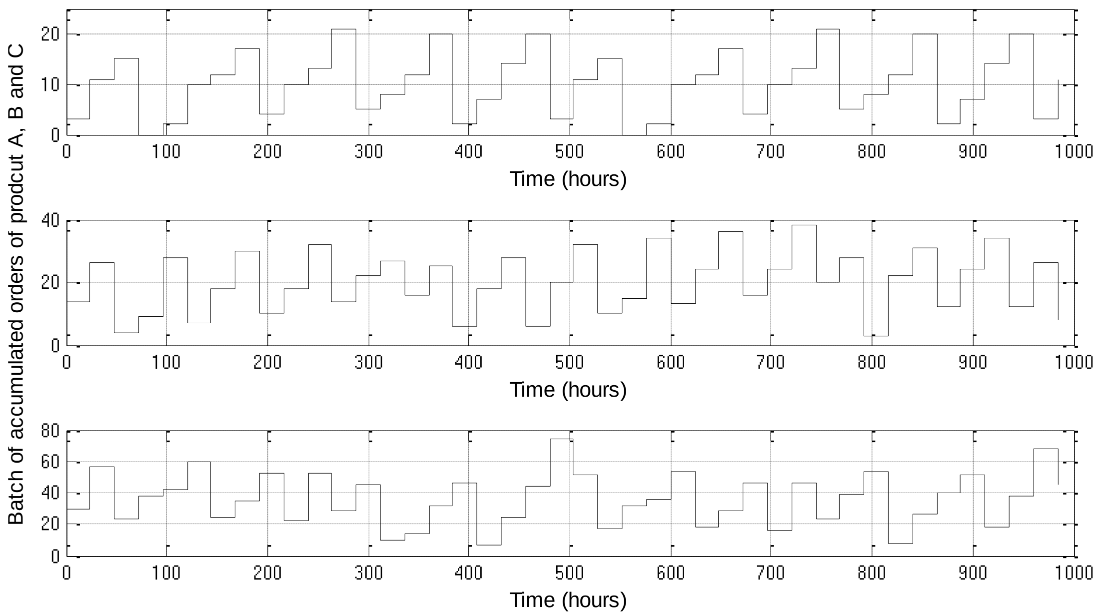

Figure 5.17: Accumulated orders for product A, B and C

Subsequently, the results of batch production and vessel change can be obtained as follows in Figure 5.18.

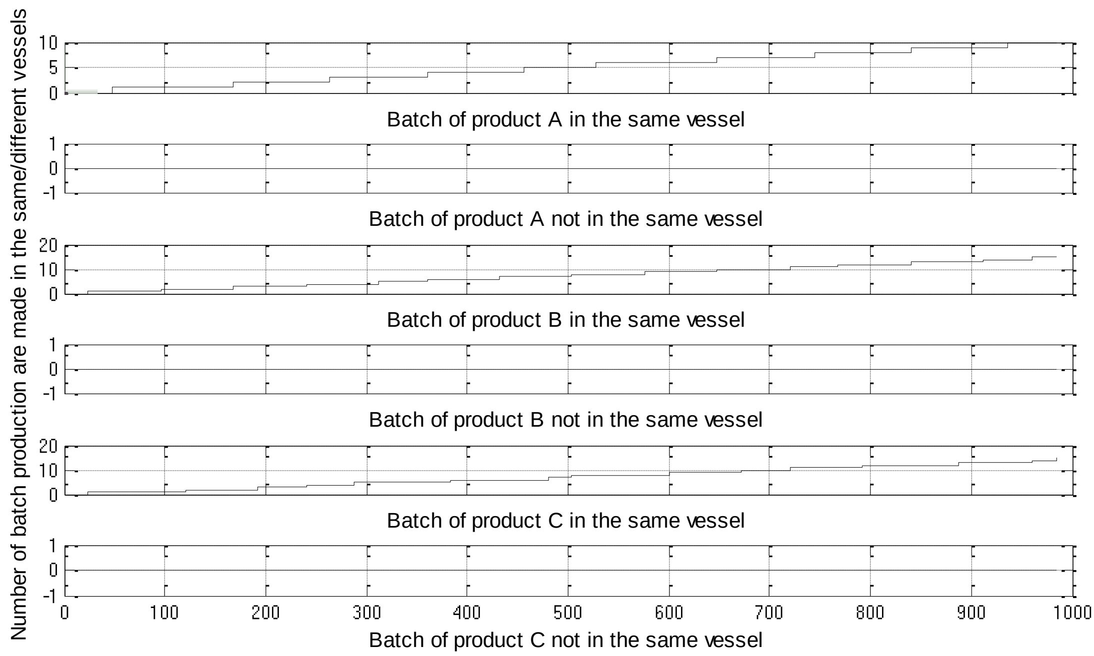

Figure 5.18: Number of batch production are made in the same and different vessels

Based on the model of production and decision making, there is no batch production to be changed to the different vessel. If it changes the number of values randomly, the different result is obtained in Figure 5.19.

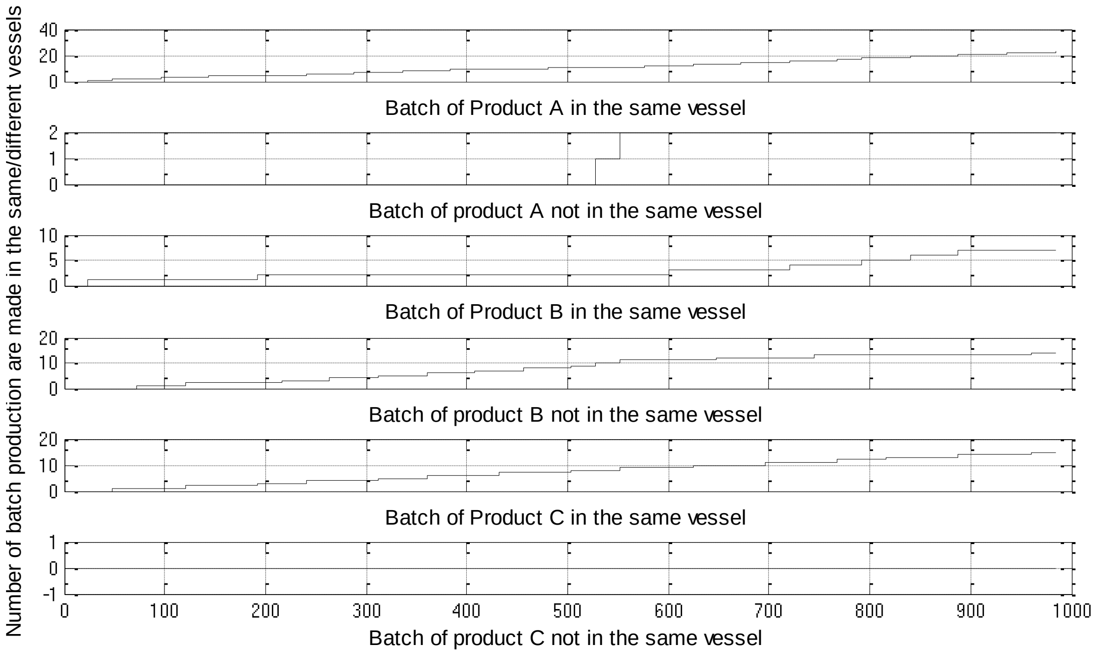

Figure 5.19: Number of batch production with random orders are made in the same and different vessels

According to the simulation result in Figure 5.19, it is obviously shown that batch production of product A has been changed twice; also the batch production of B has been changed 14 times; there is no change for product C which is working in the same vessel always.

5.3.4 Modelling of production time

Finally, the model of production time is modelled as shown in Figure 5.20. It will be calculated how much time will be required to produce batch orders in accordance to the feedbacks from both models of accumulation and model of decision making.

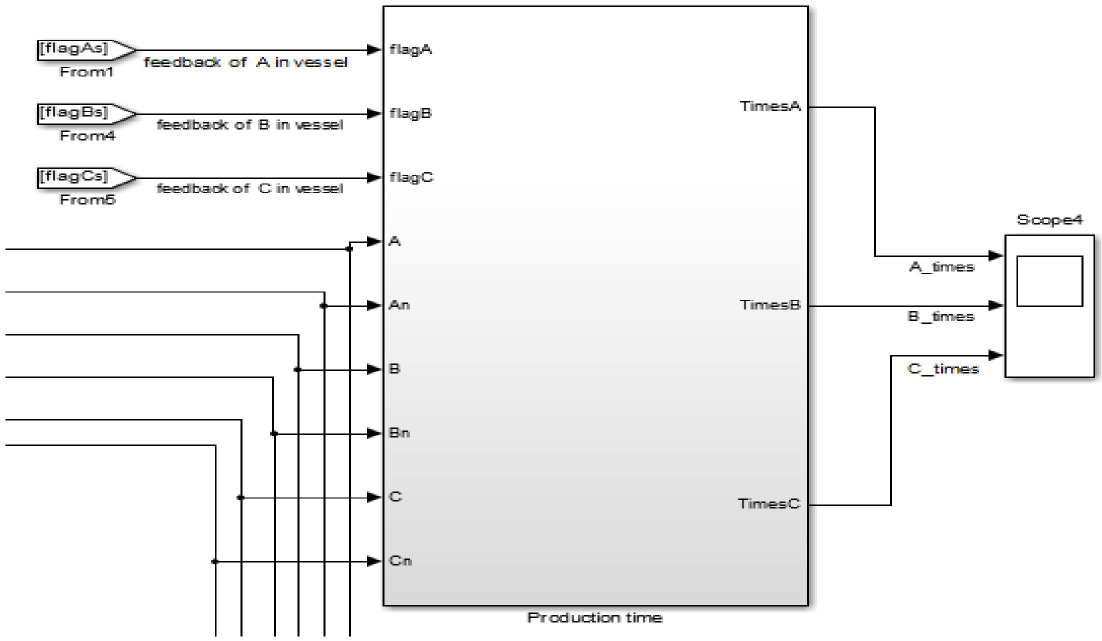

Figure 5.20: Subsystem of production time

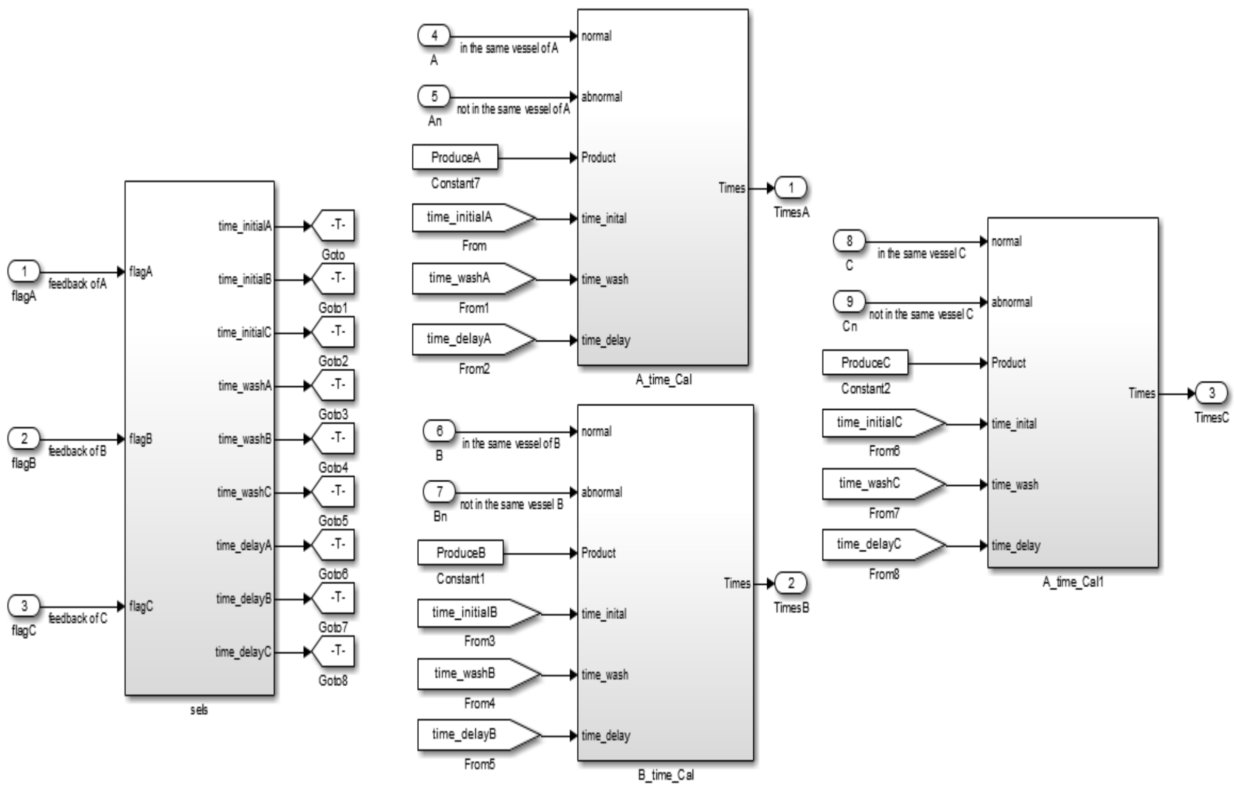

Figure 5.21: Subsystems of selection of production time and calculation of total production time

The model of selection of production time is illustrated in Figure 5.21 based on the feedback of decision making regarding how much time will be spent for each batch production including setting up time and cleaning time. In addition, changeover time may be incurred when product changes require different vessels for production.

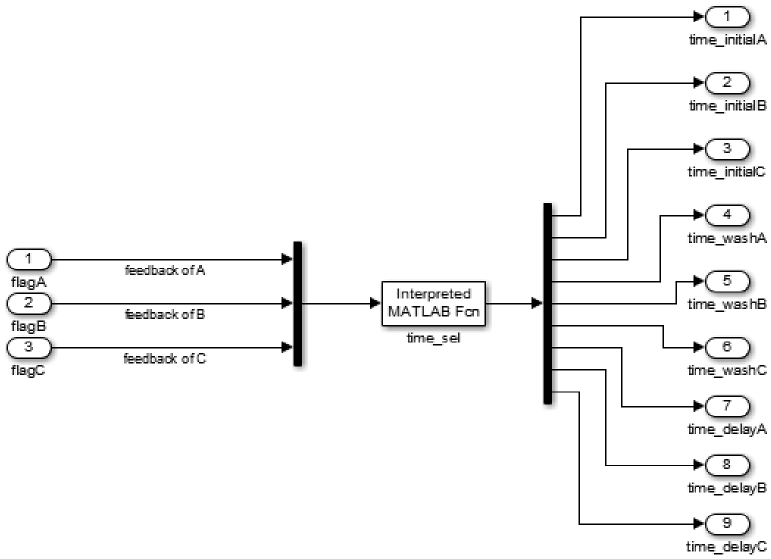

Figure 5.22: Model of selection of production time

The function of the selection of production time shows that generated time is based on each product to be made in each vessel as in Figure 5.22. The model will then decide what potential time could happen.

Afterwards, the total production time can be calculated as follows in Figure 5.23:

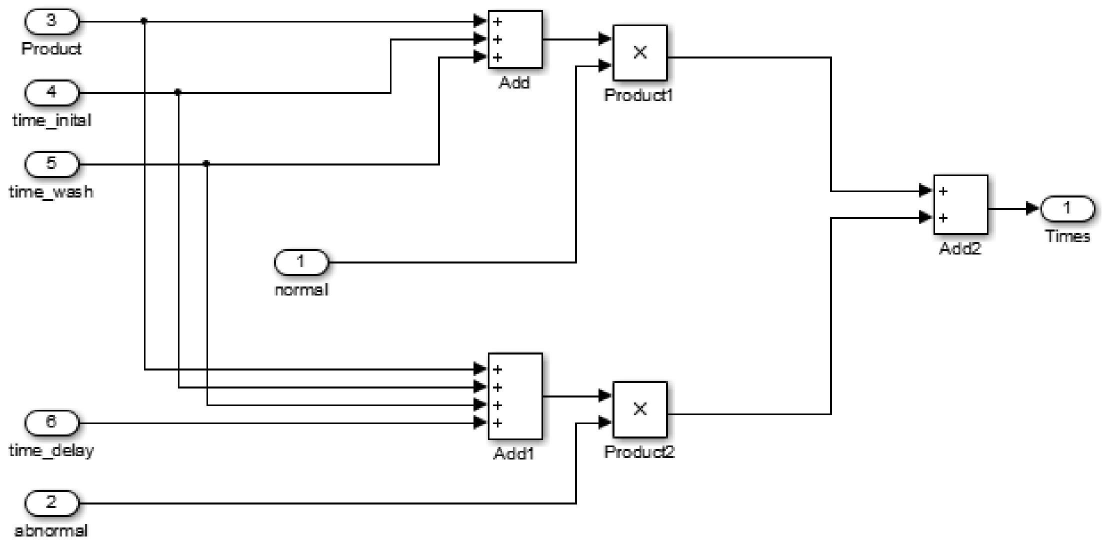

Figure 5.23: Model of calculation of production time

Lastly, the total production time of product A, B, and C can be obtained as in Figure 5.24:

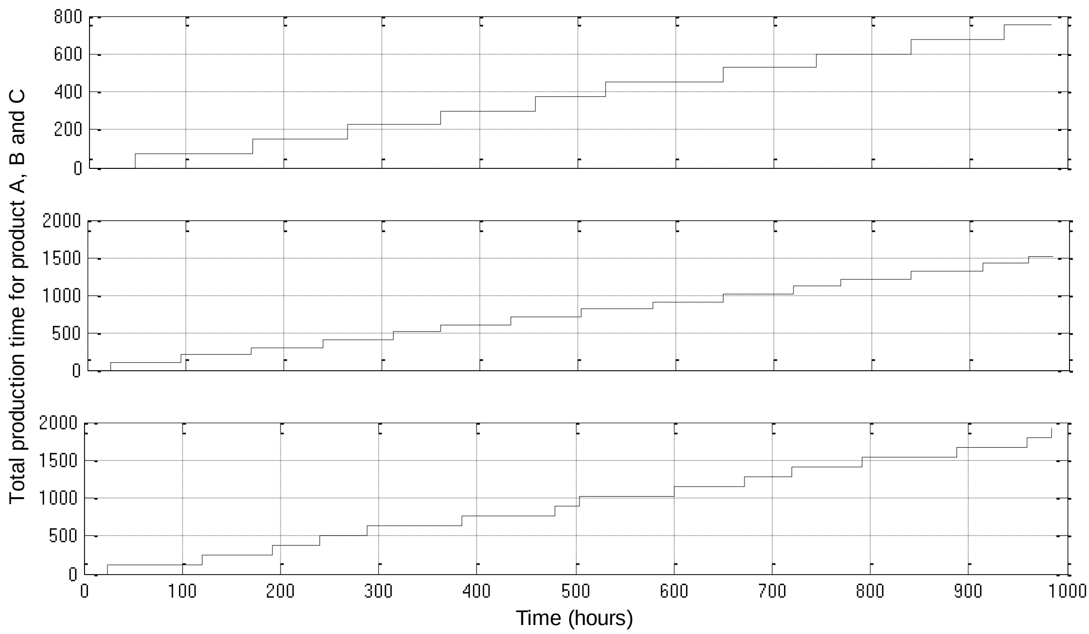

Figure 5.24: Total production time for product A, B and C

5.4 Conclusion

This chapter focused on business process modelling and simulation for a complex dynamic production system of a micro-brewery. The approach of the MATLAB/Simulink environment is implemented to simulate the scenario of a brewery production system. The dispatching rules are applied to allocate the resources optimally based on the decision making at the manager level. It is entirely feasible to change any parameters in the model if it needs to be. It identifies how many batch productions have been allocated for production in the same or different vessels on the basis of decision making, and the total production time is also obtained as a result. The results of the simulation are demonstrated in the production performance, that includes modelling of random orders, modelling of the production process, modelling of decision making and modelling of production time.

Chapter 6: A Hybrid Algorithm

6.1 Introduction

It is well-known that a GAs is a general and effective method for solving optimisation problems. However, the traditional GAs is not very effective in many cases, such as easy to produce premature convergence, poor local search optimisation ability and other issues (Zhou and Sun 1999). Therefore, many scholars have proposed various hybrid algorithms based on GAs for solving optimisation problems. For example, Ackley (2012) recommended genetic hill climbing method; Yu et al (2000) proposed the hybrid method of combined GA and SA; Miller *et al*. (1993) improved a GA for the NP-hard problem optimisation problem that added a local improvement in computation, and so on. The basic idea of the Hybrid GA is to apply local optimisation tools (such as hill climbing method, SA, etc), for each new offspring in a generation to move to the nearest local optimal point before it enters the next generation groups. In the hybrid GA, the heuristic method is used for local optimisation, and the GA is used to explore the global optimum. So the hybrid GA is usually superior to a standard single algorithm.

From the mathematical model as described in Section 4.4, it can be seen that the beer production planning and scheduling problem involves a large number of variables and constraints, including a large number of integer variables, including variable delays, which belong to the mixed integer nonlinear programming problem (MINLP). Currently, the branch and bound algorithm, the generalised benders decomposition method, and the outer approximation method have already been proposed (Zheng 2008), which are used to solve the problem of mixed integer programming. However, the calculation amount will be increased exponentially and the complexity of time and space cannot be acceptable when the scale of the problem is large (i.e. too many variables). Due to the defects of the traditional deterministic search method, the intelligent optimisation algorithm is gradually applied to the planning problem and has shown many advantages. The GA is widely applied in areas where it has good robustness and better global optimisation ability.

Therefore, this chapter implements the intelligent optimisation algorithm to optimise a brewery production system. A typical beer production process is considered in that different product types are to be produced in three vessels with different capacity in parallel. The operation of production is determined by the setting up time, fermentation time, cleaning time and changeover time. This chapter concentrates on optimisation of the JSP which is improved by heuristic algorithms based on its identified drawbacks. Firstly, the traditional GA is improved by a coding representation, the adaptive adjustment of mutation probability and crossover probability. Secondly, the improved SA is proposed by improvement of generator and temperature drop function. Finally, a novel optimisation algorithm is proposed in which an improved GA is integrated with the improved SA.

6.2 Related works

With the current development of manufacturing, the optimisation of the JSP is becoming a more and more important issue. Scheduling plays a crucial role in most manufacturing systems and engineering as well as service industries (Pinedo 2012, Suwa and Sandoh 2013). It is a decision making process which allocates resources optimally to tasks over a given period of time in order to maximise efficiency and to minimise the costs of operations of the companies.

There are various optimisation methods for solving JSP problems. Heuristic methods have been identified by (Šeda 2007), which could be used to obtain an optimal solution for complex tasks that includes GA, SA, tabu search, etc. Some main research achievements in optimisation for JSP have been clarified in the literature. According to (MacCarthy and Liu 1993), it has been proposed that a GA is based on its effect on the number of iterations and optimal solutions via initial population size, crossover probability, and mutation probability. The most important factor of a given proposed GA is whether the parameter selection is appropriate. (Tozkapan, Kirca and Chung 2003) have also proposed a GA which is based on the model of a schema theorem to select parameters. It is the evaluated effect of the algorithm performance by the probability of constituting schema and built relations between the GA parameters and performance. The main disadvantages of the traditional GA have been analysed by (Prügel-Bennett 2004) which included poor local searching capability, inefficient searching ability after evolution, and premature convergence, etc. In addition, (Kirkpatrick, Gelatt and Vecchi 1983; Dekkers and Aarts 1991) introduced the SA, which can approach the global minimum point of the objective function by a randomly searching method.

In recent years, many researchers have been engaged in flow-shop and job-shop scheduling optimisation problems to advance or combine the algorithms in order to successfully improve the performance in a diversity of domains. It is also well-known that heuristic methods are widely used to solve the JSP, such as GAs, SA, tabu search, ant colony optimisation, etc. (Huang, Zhao & Ma 2014) have proposed an improved GA which redesigned the chromosome encoding schema, crossover operator and mutation operator to minimise the makespan for JSP with process sequence flexibility. (Wang & Tang 2011) have also proposed an improved GA to minimise the makespan for JSP which redesigned the adaptive crossover probability and adaptive mutation probability based on a hormone modulation mechanism. The advantages are characterised by simplifying operations, high search precision, overcoming premature phenomenon and a slow evolution. Moreover, the proposed SA algorithm is implemented to solve the JSP by Damodaran & Vélez-Gallego (2012), which is considered to minimise makespan of parallel batch processing machines with unequal job ready times. (Tamilarasi 2010) has proposed a hybrid algorithm which is a combined GA and SA that adopts the real space as the search space and the chromosome represents the permutation of all operation of all jobs. The advantages of the algorithm are stated that it can narrow the field of search and speed up the rate of convergence continually during the optimising process, higher searching efficiency and escape from the local minima. (Peng, Lu and Cheng 2015) proposed a hybrid method of tabu search and path relinking algorithm for JSP, which incorporated a number of distinguishing features, such as a path solution construction procedure based on the distances of the solutions and a special mechanism to determine the reference solution. (Thamilselvan & Balasubramanie 2012) have also proposed a new algorithm based on three algorithms of GA, Tabu search and SA to solve the JSP. The important features of the proposed algorithms include chromosome representation, effective genetic operators and restricted neighbourhood strategies.

6.3 Improved GA

According to the shortfalls of the standard GA in the solving the problems outlined above it is easy to witness premature convergence, falling into local optimum and poor efficiency at the late evolutionary stage, etc. This chapter presents an improved GA to overcome these problems. A real number method is applied instead of traditional binary coding and a dynamic adaptive strategy is introduced to adjust the crossing probability and mutating probability. Numerical results illustrate that the algorithm is feasible and effective. According to the literature regarding the use of GAs (Song, *et al.* 1997; Stender 1993; Subbu, Sanderson & Bonissone 1998; Zhou and Sun Yun & Gen 2003; Liu, Xu, & Abraham 2005; Lei 2012) and combined with the selected input variables in this research, the proposed qualitative fuzzy control rules for crossover probability P_c and mutation probability P_m can be summarised for the minimisation problems as follows:

- In comparison with the parent, if the average fitness value of the offspring is lower, then the current evolutionary operation is carried out in the direction of the global or local optimal solution. Then it should increase the probability P_c and P_m; otherwise, decrease their probability. The choice of probabilities of crossover and mutation that are shown in Section 6.3.3.

- In comparison with the parents, if the standard deviation of the sub-generation population is higher, the offspring individuals are more and more decentralised, which is very beneficial for searching the global optimal solution. Then it should increase the P_c and reduce the P_m. Conversely, it should decrease the P_c and increase P_m.
- In comparison with the parents, if the average fitness function value and the standard deviation of the sub-generation population are similar; this shows that the strength of the P_c and P_m may not be enough; then the P_c and P_m should then be substantially increased to prevent the occurrence of the premature convergence phenomenon.

In the GA operating process, a set of parameters has a great influence on the performance of the GA. This set of parameters needs to be reasonably selected and controlled during the initial stage or population evolution that includes the length of the chromosome, the population size, the probability of crossover and mutation. A lot of experiments have been carried out successfully (Bvack 1993; Jong 1980; Goldberg & Holland 1988; Grefenstette 1986), and some suggestions are given as follows:

- Bit string length l: it depends on the accuracy of the pending problem. The longer the string the higher the required accuracy, but more computing time is needed. In order to improve the efficiency of the operation, the variable length string or re-encoding is a feasible method in the current small feasible region, and it can obtain good performance;
- Population size n: it contains a number of chromosomes that are in the population. Large scale populations contain more patterns that provide enough

sampling points for the GA, which can improve the quality of GA search and prevent premature convergence. But the large population increases the computation quantity of the individual adaptability evaluation, which makes the convergence rate decrease. On the other hand, if there are too few chromosomes in the population, then crossover possibilities of GA is very small and only a small part of search space is explored. It is recommended that the value range is 20~100;

- Crossover probability P_c : it controls the frequency of the crossover operator. In each new population, it is necessary to crossover the chromosomes of the selected individuals. P_c is bigger, the new structure in a species is introduced faster, the gene loss rate of the obtained fine gene structure is higher; Yet P_c is too small, it will lead to block search, resulting in premature convergence. It is recommended that the value range is 0.4~0.99;

- Mutation probability P_m : mutation operation is an effective means to maintain population diversity. In the mating pool, each individual allele will be changed according to random probability after the end of the crossover, so every generation will be occurred about n times mutation. If P_m is too small, it may prematurely lose some information that cannot be restored. If P_m is too large, the search will become a random search. Generally speaking, if GA is not using the crossover operator, then P_m takes a larger value range for 0.4~1; otherwise, P_m takes a smaller value range for 0.0001~0.5; when the mutation operator is used as the core search operator, the ideal is set adaptive mutation probability, in order to achieve the GA from the “overall search” gradually transiting to the “local search” (Wang, Ding and Li 2005).

In this research, the model is complex, and the multi-parameters are too difficult to achieve by traditional GAs. Furthermore, the constraints are complex, and the algorithm has difficulty in meeting all the constraints. The optimisation speed is also slow and cannot reach the optimal solution. Due to the population size of the traditional GA is limited; the higher fitness individuals have more chance to reproduce in the next generation after reproduction, selection, crossover, and mutation, which leads to local convergence by the GA. In this case, the non-global optimal solution will be of low efficiency for searching ability and fitness calculation, etc. Therefore, several main aspects of the GA are improved in order to solve these problems as follows:

6.3.1 Selection operator

In the GA, the main function of the selection operator is to obtain the best individuals from a genetic population. The selected individual from the population will have a better chance to reproduce in the next generation. Here we implemented a selection method of fitness value proportion, also as known as roulette wheel selection. The selection probability of each individual is obtained by the sum of each individual fitness and population fitness to generate a new population (Back 1996). The probability of an individual being selected can be denoted:

$$P_{mi} = \frac{f_i}{\sum_{j=1}^{m} f_j} \tag{6.1}$$

where P_{mi} is the selection probability of the i th individual; f_i is the fitness of the individual i in the population; $\sum_{j=1}^{m} f_j$ is the cumulative fitness of the population.

Elitism selection is also applied to select the best individuals for the new population. It can rapidly increase the performance of a GA and prevents losing the best found solution. According to the traditional selection methods (Goldberg & Rudnick 1990), the excellent schema may be damaged by the genetic operator, and then it will decrease the average fitness value. So we will select the highest fitness of individuals to the next population, and to use the current best individuals to replace the lowest fitness of the individuals for genetic operation.

6.3.2 Encoding representation

Traditional GAs employs binary strings for the gene encoding. There are many advantages of binary encoding, such as ease of implementation, the encoding and decoding are simple; crossover and mutation operation can be easily achieved, and so on. However, there is the lower computation efficiency when processing the calculation of encoding and decoding. It will generate the deviation of the encoding and decoding for solving the continuous parameter optimisation problem, and then it will lower the computational accuracy. The length of strings needs to be extended in order to improve the computational accuracy, which will affect the calculation efficiency. Due to the poor search ability of the continuous optimisation problem, it could lead to premature convergence for the multi-dimension and high precision optimisation problems. So it is difficult to meet the requirement of accuracy in numerical optimisation. An important issue of the binary representation implementation has been identified by (Beasley and Chu 1996); the resulting solutions are no longer guaranteed to be feasible. It has also been stated that real number encoding is best employed and gives better performance than binary or Gray encoding for the function optimisation problem. The real-number encoding of the GA is the natural description without the process of encoding and decoding for the continuous numerical optimisation problem, it can greatly improve the accuracy and convergence of the solution. In order to explore a larger search space, here we applied real number encoding for gene encoding in the GA, so it is beneficial to retrieve the special heuristic information and to improve the computational efficiency and accuracy.

6.3.3 Adaptive adjustment of crossover probability and mutation probability

In the standard GA, the probability of crossover and mutation are fixed parameter values, which may be sub-optimal. Typically it is leads to the phenomenon of premature convergence. Conversely, the adaptive adjustment of crossover and mutation probability in the algorithm have been studied by Srinvas and Patnaik (1994), and has shown advantages over the shortcomings of the standard GA. In the adaptive GA, the probability of crossover and mutation are varied depending on the fitness of each solution. The higher fitness solutions are retained and the lower fitnesses are totally discarded (Song and Xiao 2013). An adaptive GA is able to find a more general adaptive crossover and mutation probability to improve the efficiency of the GA. The parameters are determined by a maximum fitness value and an average fitness value of each chromosome difference value, defined as in equation (6.2):

$$\Delta f = f_{\max} - f_{ave} \tag{6.2}$$

where the different value, denoted Δf, is also known as the iteration error; $f_{\max}$ is the maximum fitness value; f_{ave} is the average fitness value. When Δf is large, it can improve the diversity of the population and the global search capability; when Δf is small, it is easy to lead to local convergence and it can become trapped in a local minimum (Vasconcelos, *at al*. 2001). In order to avoid local convergence in optimisation, P_c and P_m need to be adaptively adjusted. Individuals with higher fitness values will be retained, P_c and P_m should be decreased when Δf is large, while individuals with lower fitness values, P_c and P_m should be increased when Δf is small. These can be represented, respectively, as:

$$P_c = \frac{1}{1+e^{(-k_1 \Delta f)}} \tag{6.3}$$

$$P_m = 1 - \frac{1}{1+e^{(-k_2 \Delta f)}} \tag{6.4}$$

where P_c is the crossover rate; P_m is the mutation rate; k_1 and k_2 are constants and $k_1, k_2 \leq 1.0$; the two parameters should be adjusted according to a given problem; we assign k_1 and k_2 a value of 1, this ensures that all solutions with a fitness value are low or equal to f_{ave} compulsorily undergo crossover. Then the probability of crossover decreases as the fitness value tends to $f_{\max}$ and equal to $f_{\max}$ acoording to the improvement of diversity of individuals and the global searchability when Δf is large . In the iterative procedure, essentially, Δf is repeatedly calculated from an objective function (4.4) and subsequently fed back to update equations (6.3) and (6.4) to adaptively adjust based on the chromosome conditions. In the adaptive case, P_c and P_m will be changeable and will tend to optimal values for a given problem. This can be shown in Figure 7.10, initially, we set default values of P_c and P_m to be 0.8 and 0.2, respectively, and their values are adjusted adaptively to 0.87 and 0.13, respectively. The results clearly show that adaptive adjustment performs better in optimisation as presented in Figure 7.12. In this way, we can obtain optimal P_c and P_m during each iteration.

6.4 Improved SA

Based on the literature, the advantages of the SA can be simplified as follows: high efficiency, flexible, general purpose, the initial value is highly robust, suitable for parallel processing, and can be used to solve complex nonlinear optimisation problems. However, the convergence of the SA is slow and execution time is too long due to the higher initial temperature, the slower cooling rate, the lowering of the end temperature, and the temperature of the sample; so it is difficult to get the global optimal solution if the cooling process is too fast. Therefore, SA is improved by the calculation method of the jump distance and the acceptance probability of the temperature drop function in order to solve these shortcomings, in the following several aspects (Ning and Guo 2008):

6.4.1 Improvement of generator

The improvements of generator addresses the improvement of the calculation method of the jump distance, in which it is assumed that the maximum and the minimum of the individual species is U and L, respectively, then the information of population in upper and lower bounds can be obtained from the following formula:

$$L' = \frac{L + \beta^k \cdot LB}{1 + \beta^k} \tag{6.5}$$

$$U' = \frac{U + \beta^k \cdot UB}{1 + \beta^k} \tag{6.6}$$

The formula (6.5) and (6.6) is the information of population in upper and lower bounds,

LB and UB is the upper and lower bounds of the feasible region respectively, β is shrinkage coefficient, k is constant. Then jump distance can be calculated by the following formula:

$$D = \min\left(x - L', U' - x\right) \tag{6.7}$$

through the improved method, which can solve the problem of local minimum effectively when the population is decentralised , and solve the problem of zeros jump distance to prevent single population. (Kirkpatrick, Gelatt & Vecchi 1983; Dekkersn & Aarts 1991)

6.4.2 Improvement of acceptance probability of the temperature drop function

The acceptance probability of the temperature drop function can be expressed by the following formula:

$$p_{ij} = e^{\left(-\frac{f(x+dx)-f(x)}{\alpha^{k-1}\cdot T_{0k}}\right)} \tag{6.8}$$

where,

$dx = D \cdot r = \beta^{k-1} \cdot A \cdot r$, D is the jump distance, A is equal to $\min\left(x - LB, UB - x\right)$, r is a random number, β is shrinkage coefficient, α is the temperature drop function coefficient, T_{0k} is the initial temperature, where the k is bigger and bigger, dx is smaller and smaller, then $f\left(x+dx\right) - f\left(x\right)$ can equal to $f'\left(x\right)\cdot dx$ So the acceptance probability can convert to the following formula:

$$p_{ij} = e^{\left(-\frac{f'(x)\cdot\beta^{k-1}\cdot A\cdot r}{\alpha^{k-1}\cdot T_{0k}}\right)} = e^{\left[\ln\left(1-\frac{1}{nPopAnn}\right)\left(\frac{\beta}{\alpha}\right)^{k-1}\cdot A\cdot r\cdot\frac{f'(x)}{f_{\max}-f_{\min}}\right]} \tag{6.9}$$

In the formula (6.9), we can see that when $\beta/\alpha > 1$, the acceptance probability tends to zero after evolution, when $\beta/\alpha < 1$, the acceptance probability tends to one after evolution, when $\beta/\alpha = 1$, the acceptance probability after evolution is associated with the derivative of x, when $f'(x) \to 0$ the acceptance probability tends to one. The main advantage of this function can speed up the convergence (Dekkersn & Aarts 1991)

6.5 Hybrid method of improved GA and improved SA

The GA has a strong ability of global search and is easy to implement in optimisation. However, it will lead the error or missed optimal results in the crossover procedure due to slow simulation speed, local optimisation, premature phenomenon, etc. Therefore, it is proposed to integrate an improved GA with the improved SA as a hybrid method and to combine advantages of GA (globalisation and parallelism, etc.) and SA (local search ability and ergodicity, etc.). The procedure of the hybrid method is presented as following:

Step 1: Randomly generate the n th initial populations, denoted as p_1, p_2, $p_3 \ldots$, p_n respectively. Parameters need to be clarified, including population size, chromosome length, crossover probability, mutation probability, initial temperature, cooling temperature and cooling approach.

Step 2: Apply parameters into the objective function (4.4) to obtain initial fitness values. These are denoted T_{p_1}, T_{p_2}, T_{p_3} $\ldots$, T_{p_n} respectively, which need to satisfy the constraints and conditions in the optimisation process.

Step 3: Implement the improved GA. This employs a roulette wheel selection method to select some of the optimal individuals from the populations and reproduce to the next generation based on the procedure of selection, mutation, crossover, etc. The selected individuals form a new population, denoted p_1(t) ; the unselected individuals, are denoted p_2(t) and to crossover and mutate the unselected individuals to form a new population, are denoted $p'_2(t)$

Step 4: Implement the improved SA after crossover and mutation, Equation (6.9) is applied whether it is acceptable for a cooling factor. This generates a new population, denoted $p''_2(t)$.

Step 5: Calculate fitness of all individuals after improved SA.

Step 6: Combine p_1(t) and $p''_2(t)$ to form the new population as a new generation of species.

Step 7: Repeat above procedures until iteration ends.

The improved algorithm is presented in Figure 6.1:

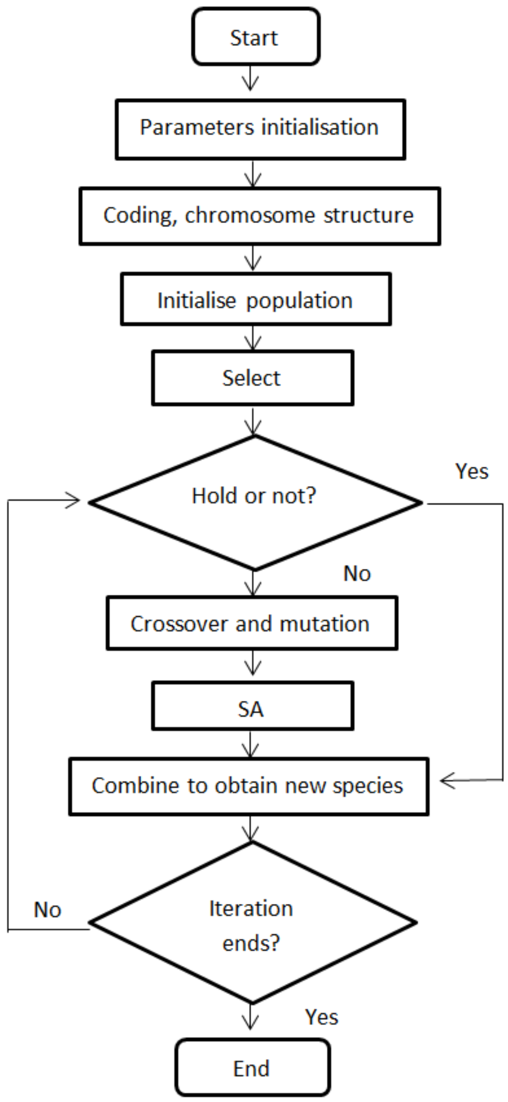

Figure 6.1: Hybrid method of improved GA and improved SA

6.6 Conclusion

In this chapter, we proposed a novel hybrid algorithm for the fusion of combined an improved GA and an improved SA for minimising the total production time in a micro-brewery, as given in Chapter 4. It adopts the acceptance probability of SA to improve the convergence of the improved GA which has improved the computational efficiency and accuracy by real-number encoding and also improved the diversity of the population of the adaptive adjustment of crossover probability and mutation probability. Consequently, the improved GA and SA not only achieves the combination of global search capability of GA and local search capability of SA, it can also help SA to take full advantage of the global information from GA. The convergence of crossover rate and mutation rate is optimised and also the proposed hybrid algorithm is validated as described in Chapter 7. The hybrid approach designed in this work differ from other hybrid approaches in that it has been developed specifically for the complex micro brewery production process.

Chapter 7: Application of Proposed Models and Algorithms

7.1 Introduction

In order to validate the proposed hybrid algorithm and to obtain the optimal sequences of orders of each product, the research is applied to use the traditional GA, SA, and ACO to optimise a brewery production system. In addition, the traditional GA will be improved by encoding representation, and adaptive adjustment of crossover rate and mutation rate. The SA is also improved by the advanced jump distance and acceptance probability of the temperature drop function. Then we use the traditional GA, SA, ACO, an improved GA and an improved SA to compare with the hybrid algorithm in order to optimise a typical brewery production system in terms of the total production time. Thus, altogether six algorithms are compared.

7.2 Research setting for a micro-brewery

In this research, a typical micro brewery system assumes that three beer products are to be produced simultaneously in three parallel fermentation vessels of differing capacity. The problem is how to schedule the orders to be produced in the vessels, such that the total production time is optimal.

Furthermore, three products can be denoted p_1, p_2 and p_3 respectively. The production period is denoted T_{p_i} of p_1, p_2 and p_3 are 15, 20 and 30 days, respectively, where $i = 1, 2, 3$. The sequence of the quantity of orders are constant; it can be denoted o_n, where $n = 1, 2, 3, \ldots n$; Three vessels can be denoted v_1, v_2 and v_3, respectively. The maximum capacity of v_1, v_2 and v_3 are expressed in terms of barrels and denoted 20, 30 and 50 barrels, respectively.

Subsequently, the operation of production is determined by the setting up time, fermentation time, cleaning time and changeover time as shown Table 7.1. The setting up time is denoted T_{setup} for p_1, p_2 and p_3, are 24, 48 and 72 hours, respectively. The cleaning time is denoted T_{clean} for v_1, v_2 and v_3 are 2, 3 and 5 hours, respectively. Moreover, the changeover time might occur when the next batch production is to be changed in different vessels, and then it requires an additional 12 hours for vessel cleaning. In addition, the due date is denoted T_{due} for p_1, p_2 and p_3 are 20, 30 and 40 days, respectively, based on customer demands.

Table 7.1: Production parameters

		p_1	p_2	p_3
T_{p_i} (hours)		15*24	20*24	30*24
T^i_{setup} (hours)		24	48	72
o_n (barrels)		100	100	100
T^i_{due} (hours)		20*24	30*24	40*24
T^i_{change} (hours)	v_1	0	12	12
	v_2	12	0	12
	v_3	12	12	0

The following are the assumptions made in solving this problem (the assumptions are based on the commercial brewery operation process):

- The 100 random raw orders for each product type which are given in Appendix A.
- Three different products working in three parallel vessels simultaneously with limited capacity.
- A set of fixed processing time and setting up time for each product as well as cleaning time for each vessel and changeover time. They are considered deterministic and known in advance.
- Each vessel must process one batch of production only, once a vessel starts to process a batch of orders, no interruption is allowed, and then it needs to be cleaned when finished.
- The arriving orders will be within accumulated batches in the same order to meet the required capacity of the vessels.
- The priority order is allowed based on the due date.

- The production time cannot exceed the due date

7.3 Optimisation results

Based on the complex brewery production scheduling problem, if three products are always working in the same vessel in parallel, then there are no vessel changes for production. Therefore, the total production time without optimisation can be obtained as follows:

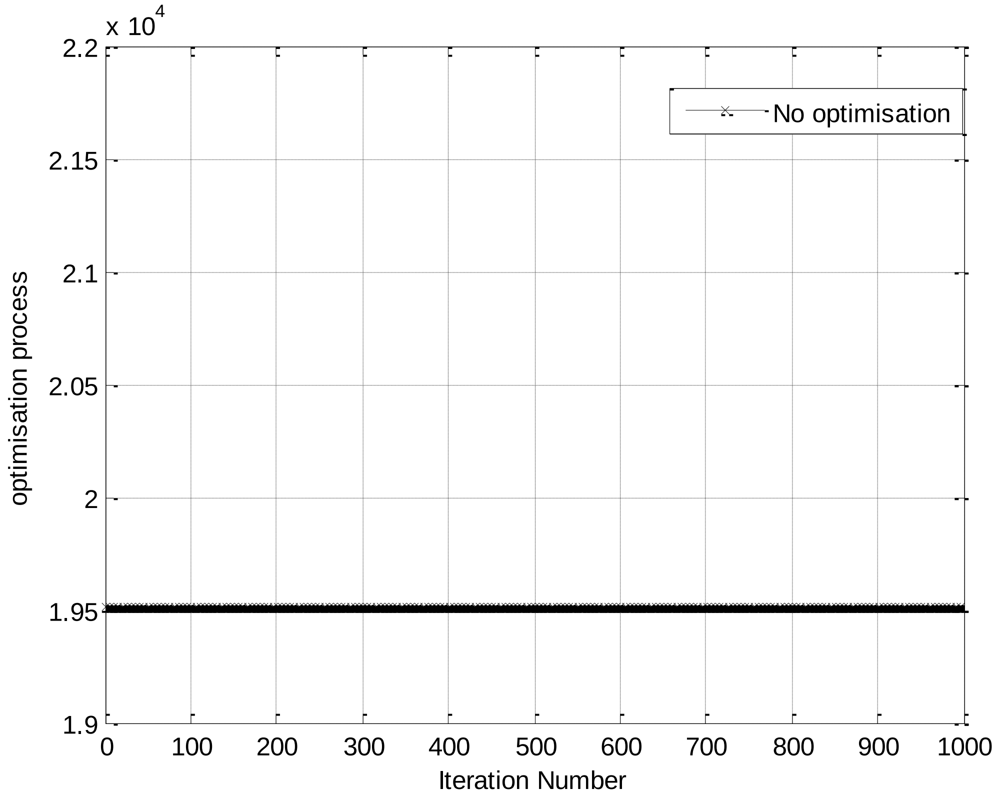

Figure 7.1: Result without optimisation

According to the result in Figure 7.1, it is shown that the result of production is 19512 hours without the use of optimisation methods. The same orders were passed to each system. The unoptimised case simply processes the order in the sequence they arrive. The other methods re-order the batches to make the vessel more effectively approach will lead changeover.

7.3.1 Genetic algorithm (GA)

In the application of the GA, the choice of crossover probability P_c and mutation probability P_m is known to critically affect the behaviour and performance of GAs. The P_c controls the capability of GAs in exploiting a located hill to reach the local optima. The higher P_c, the quicker exploitation proceeds. If the P_m is too large that would disrupt individuals faster than they could be exploited. The P_m controls the speed of GAs in exploring a new area. Small P_m values are commonly adopted in GAs. The values of P_c are normally recommended to be in the range 0.5~1.0, while the values of P_m are recommended to be in the range 0.001~0.5. The workflow of the GA can be described as follows in Figure 7.2:

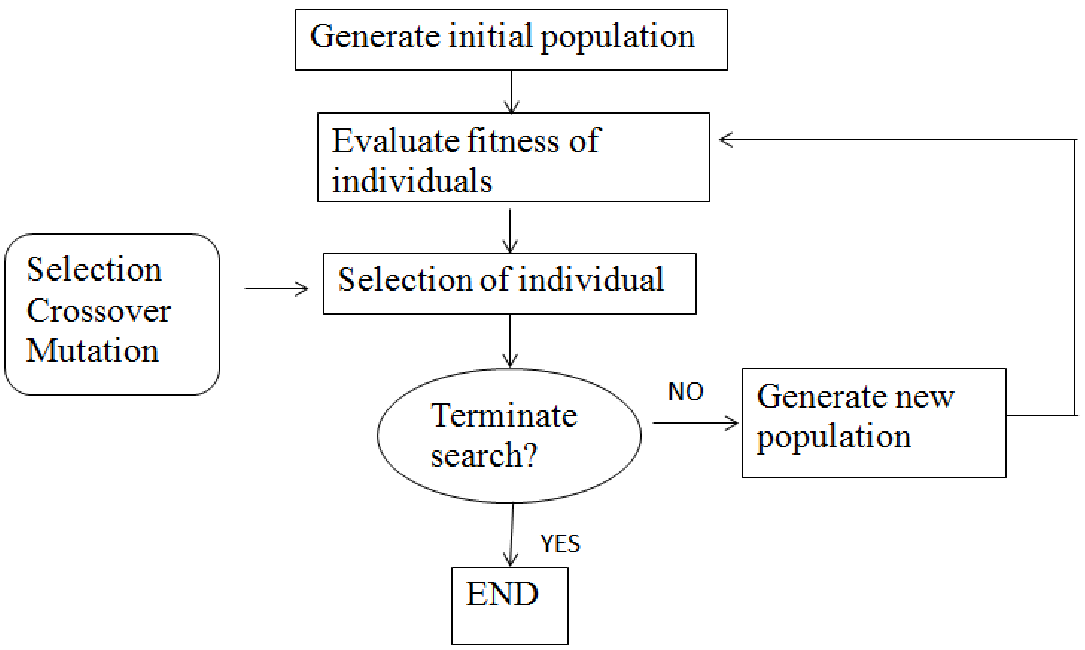

Figure 7.2: Workflow of the GA

The main parameters are used as follows:

- *Number of iterations: 1000*
- *crossover probability: 0.8*
- *mutation probability: 0.2*
- *generation gap: 0.9*
- *population size: 20*

The result of GA can be obtained as follows:

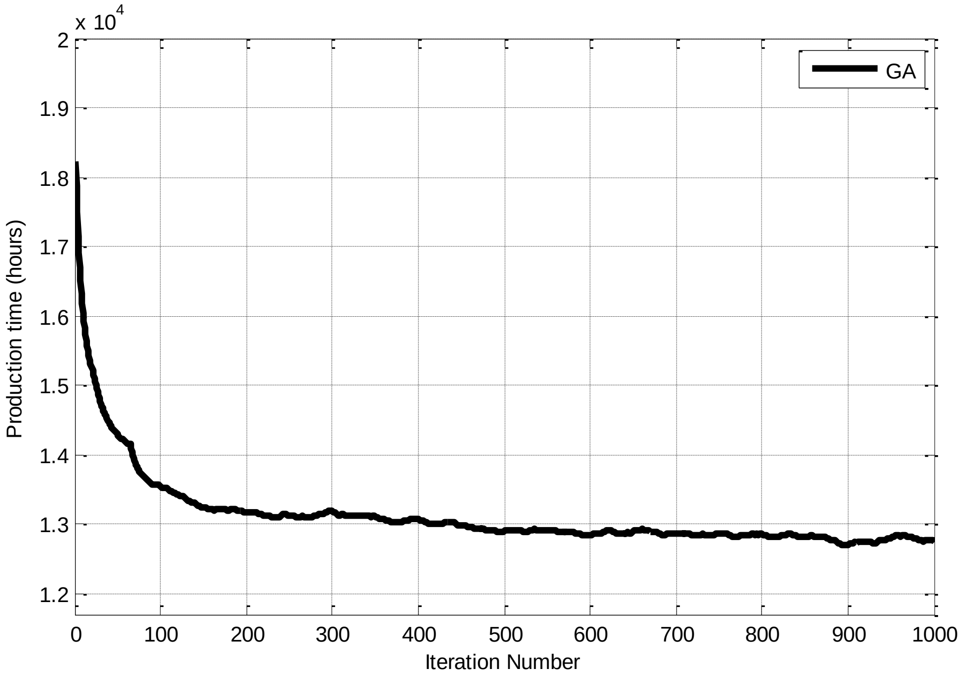

Figure 7.3: Result of GA

The above result is clearly an improvement in the unoptimised case. The production time of the GA is 12744 which has saved approximately 35% in contrast with the unoptimised case.

7.3.2 Simulated annealing (SA)

The algorithm SA is a mathematical analogy to a cooling system which can be used to sample highly nonlinear, multi-dimensional functions. There are many variations around and the efficiency strongly depends on the particular function to sample. It is extremely difficult to make general statements as to what parameters work best. The procedure of the SA can be described as follows:

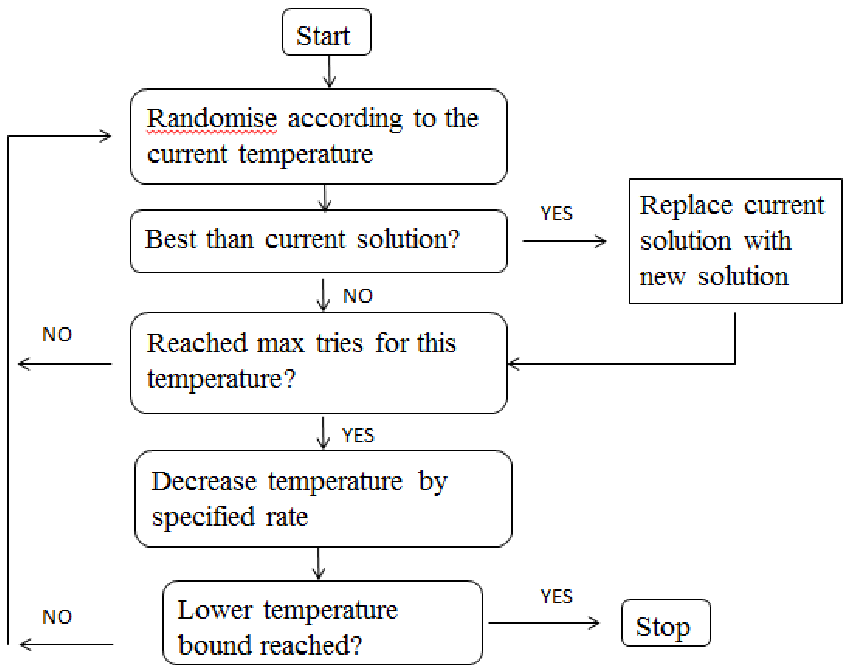

Figure 7.4: Workflow of the SA

The main parameters are used in this model as follows:

- *Number of iterations: 1000*
- *Population size: 20*
- *Initial temperature: 0*
- *End temperature: 1*

- *Current temperature: 100*
- *Cooling factor: 0.99*

The result of the SA is obtained as follows:

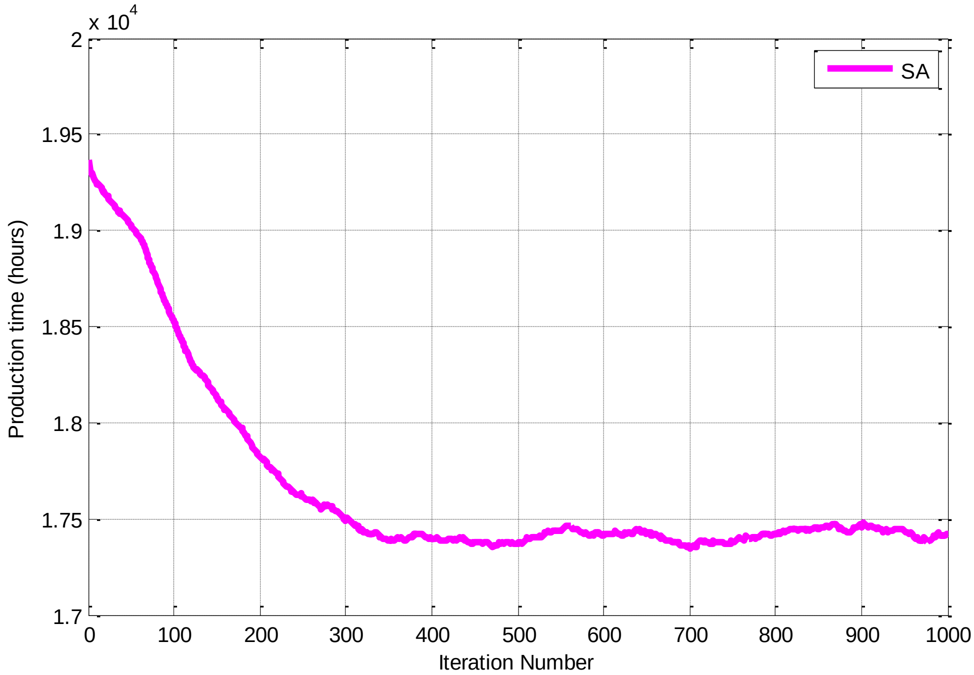

Figure 7.5: Result of SA

According to the result of the SA in Figure 7.5, which shows the convergence speed, it is faster and the performance is worse than the GA and ACO in terms of total production time. Due to the slow cooling of solid annealing, the solid can reach the equilibrium state at each temperature. Therefore, the value of the control parameters must be slowly reducing in order to ensure that SA will obtain the global optimal solution, so the number of iterations cannot determine the accuracy.

7.3.3 Ant colony optimisation (ACO)

Based on the literature, the ACO algorithm is modelled based on ant behaviour. It has been found that the information transfer between ants through a type of material which is called the pheromone which enables the transfer of information allowing ants to cooperate with each other in order to complete a complex task. Ants in the process of movement are able to deposit a pheromone in the path; other ants who are able to perceive its existence of such substances and its intensity and is guided in the direction of the pheromone. Therefore, the collective behaviour of an ant colony, which is composed of a large number of ants, demonstrates a positive feedback phenomenon: the more ants pass through a path, the greater the probability that the path will be chosen. For the JSP, the biggest difficulty is the computational complexity; then effective scheduling rules and methods can reduce the searching space and shorten the searching process. The flowchart of the ACO procedure is as shown in Figure 7.6.

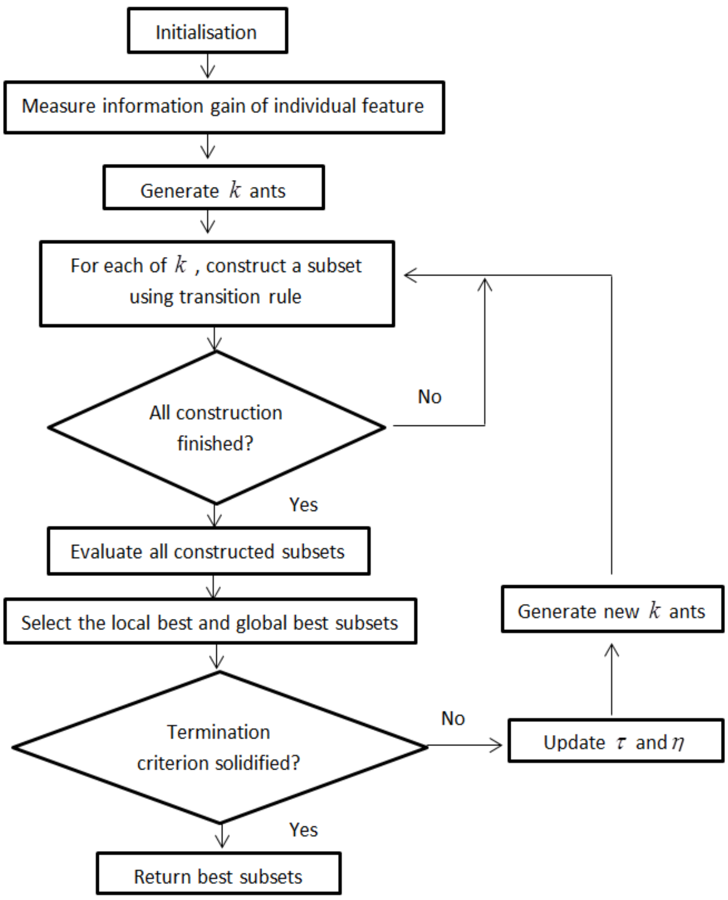

Figure 7.6: The workflow of the ACO

The main parameters used are as follows:

- *Number of iterations: 1000*
- *Population size: 20*
- *Pheromone strength index: 1*
- *Information heuristic factor: 3*
- *Expected heuristic factor: 0*
- *Pheromone lasting coefficient: 0.7*

The result of ACO is obtained as follows:

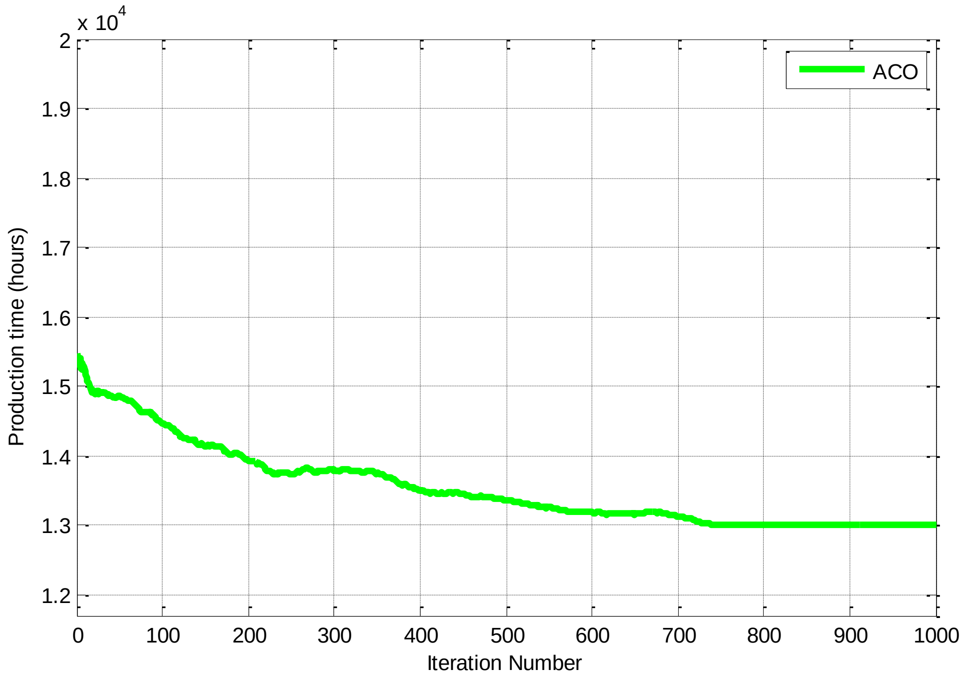

Figure 7.7: Result of ACO

As the above result, the ACO algorithm converged more slowly and reached stagnation. The result shows a better performance than the SA, and optimal production time is 13008 hours as contrasted to 17447 hours for the SA, it saving approximately 25%.

7.3.4 Improved GA

The improved GA, which is described in section 6.3, which is improved by the encoding representation. The real number is applied instead of the traditional binary encoding in the GA as it is beneficial to retrieve the special heuristic information and to improve the computational efficiency and accuracy. And also the adaptive adjustment of crossover probability and mutation probability is proposed based on chromosome condition, which is effective in order to find the most optimal results and to reduce the number of iterations. The result is obtained as follows in Figure 7.8. It demonstrates that the improved GA performs better than the traditional GA, saving approximately 5%

reduced time. In this particular result, the shortest total production time is 12277 hours and 12744 hours respectively, for the improved GA and the traditional GA.

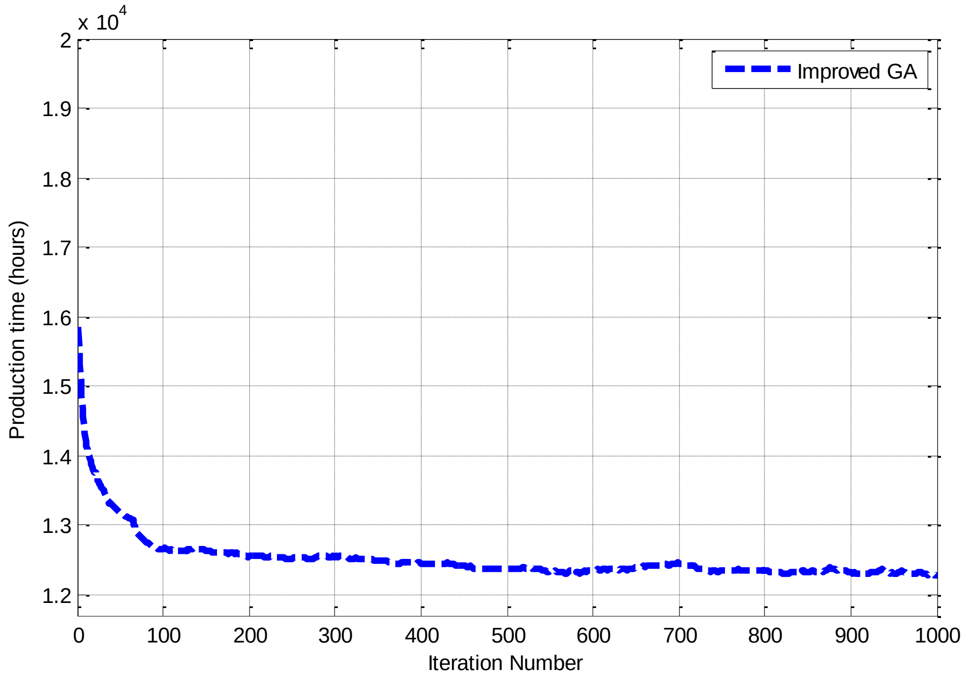

Figure 7.8: Optimised result of the improved GA

7.3.5 Improved SA

The improved SA, which is described in Section 6.4, is achieved by the improvement of generator and improvement of the acceptance probability of temperature drop function. The computational method of the jump distance is improved which can solve the problem of local minimum effectively when the population is decentralised and solve the problem of zero jump distance when the population is simplified. Also, it improves the speed of the convergence. The result is obtained as follows in Figure 7.9. The result shows that the improved SA gives better performance than the traditional SA; it saves 10% in regard to the total production time.

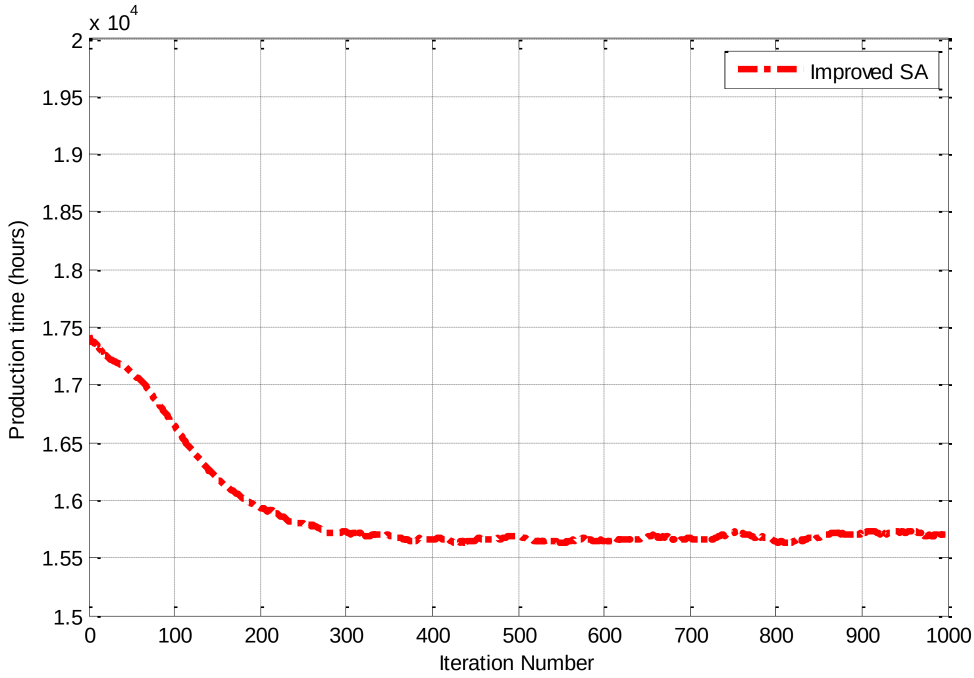

Figure 7.9: Optimised result of the improved SA

7.3.6 Hybrid algorithm of the combined improved GA and improved SA

The main parameters used are as follows:

Table 7.2: Parameters of the proposed hybrid algorithm

	Parameters	Values
1	Population size	20
2	Number of iterations	1000
3	Crossover probability	0.8
4	Mutation probability	0.2
5	Initial temperature	1000
6	Temperature drop coefficient	0.99
7	Counter	1

The approach of MATLAB/Simulink has been employed to simulate the model. The result of the change of the optimised crossover rate and mutation rate is shown in Figure

7.10:

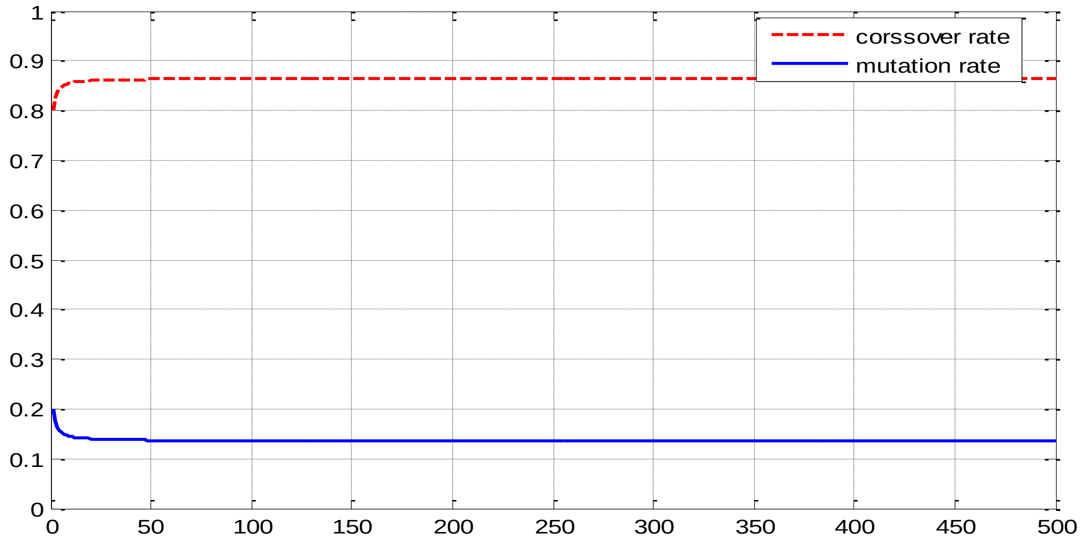

Figure 7.10: Optimised convergence of the crossover rate and mutation rate

According to the result of Figure 7.10, which illustrates the crossover rate and mutation rate patterns of convergence; at the initial phase, the distribution of both crossover and mutation rate is 0.8 and 0.2, respectively. With the optimising iteration increasing, the crossover rate is increasing and converges at approximately 0.87; the mutation rate is decreasing and converges at approximately 0.13. Therefore, the optimal crossover probability and mutation probability can be applied to optimise the production system via the hybrid algorithm.

On the basis of the iterations increasing progressively, the change of the optimised production time is presented in Figure 7.11.

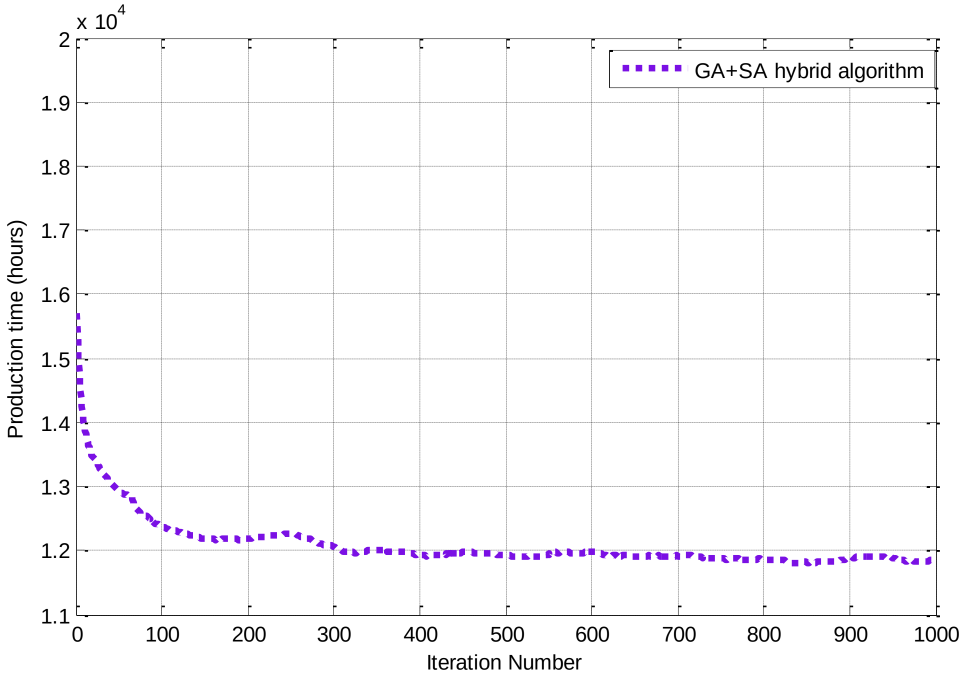

Figure 7.11: Optimised result of the proposed hybrid algorithm

It is shown that the total production time is 11851 hours of the 1000 iterations. Also, the sequence of product production in different vessel is obtained as below:

Table 7.3: Optimised sequences of production

Product types	Sequence of vessels
p_1	3, 3, 3, 3, 3, 3, 3, 3, **2**, 3, **1**, 3, 3, **2**, 3……
p_2	1, 1, 1, 1, 1, 1, 1, 1, 1, 1, **2**, 1, 1, 1, 1……
p_3	2, 2, 2, 2, 2, 2, 2, 2, **3**, 2, **3,** 2, 2, 3, 2……

According to Table 7.3, it demonstrates the change of the sequences of vessels after optimisation. Initially, there is no changeover time when p_1, p_2 and p_3 are produced in v_1, v_2 and v_3 separately. Moreover, p_1 will be changed the vessel 3 to vessel 2 for production after nine operations continuously, and then to change to vessel 3. Likewise,

p_2 will be produced in vessel 1 for ten operations continuously, then changed to vessel 2 once, and then to produce in vessel 2. Furthermore, p_3 will be produced in vessel 2 for nine operations continuously, then produce in vessel 3 once, and then schedule to produce in vessel 2. All sequences of vessel are given in Appendix B

7.3.7 Results comparison and analysis

7.3.7.1 Scenario 1

In order to validate the performance of the proposed novel method, the comparison of traditional GA, SA, Ant colony optimisation (ACO), improved GA, improved SA and the hybrid algorithm has been carried out. The 100 random raw orders for each product type which are given in Appendix A.1. The comparison of the results is presented in Figure 7.12.

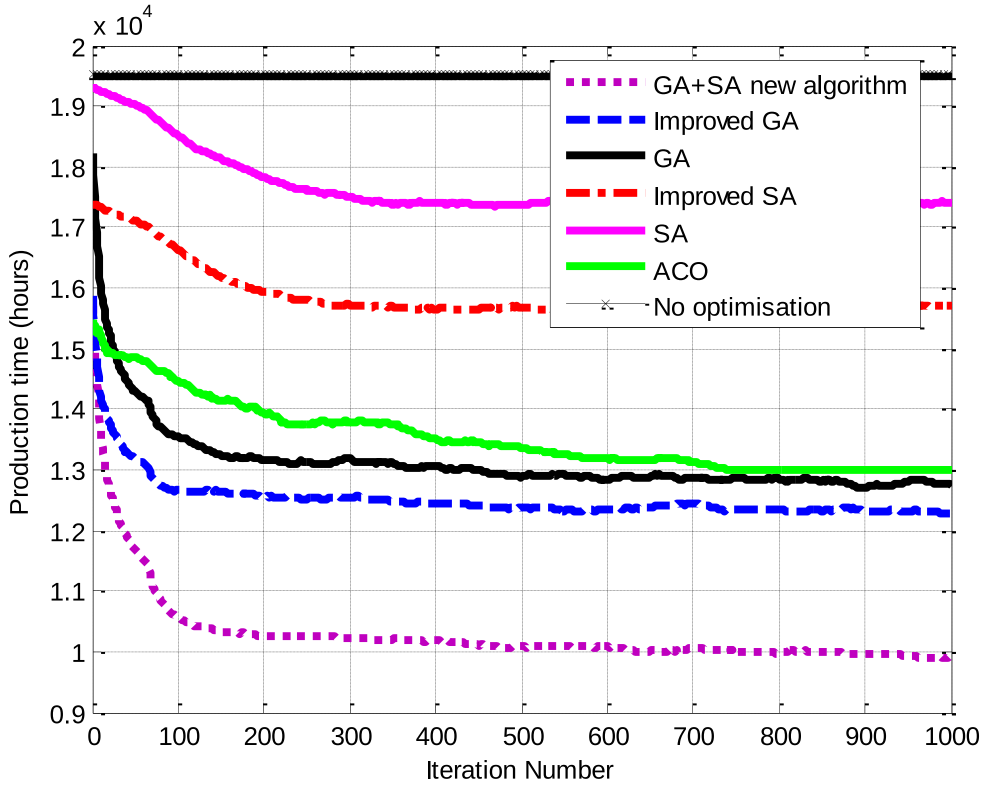

Figure 7.12: Comparison of optimisation results for different algorithms (Scenario 1)

Figure 7.12 demonstrates that the proposed novel algorithm performs better than the traditional GA, SA, ACO, improved GA and improved SA. It not only achieved the combination of the global search capability of the GA and the local search capability of the SA, but can also help the SA to take full advantage of the global information from the GA. At the early stage of the evolution, the SA temperature is higher (initial temperature value is 1000), then it can avoid the premature convergence and strength then the global convergence is according to the result in Figure 7.12. It obtained the smallest value; at a later stage of the evolution, the SA temperature is lower, then the hill climbing performance of the SA can speed up the convergence of the hybrid method. (The result is convergence in 100 iterations and much faster than others.)

The computational results are given in Table 7.4.

Table 7.4: Computational results of total production time of each algorithm (Scenario 1)

	Algorithms	Total production time (hrs)
1	Hybrid algorithm	9913
2	Improved GA	12277
3	GA	12744
4	ACO	13008
5	Improved SA	15702
6	SA	17421
7	No optimisation	19512

It is most important to notice the significant improvement in the total production time in Table 7.4. This particular result is typical and is representative, being regarding as a general observation when dealing with the micro-brewery production system. The hybrid algorithm obtained optimal total production time which is 9913 hours and it saves approximately 20%, 22%, 24%, 37% and 44%, respectively, in comparison with the improved GA, GA, ACO, improved SA and SA. It is also a saving 51% approximately of no optimisation. Hence, it is a significant advantage to the production process.

7.3.7.2 Scenario 2

Furthermore, we implemented different sets of data for the 100 raw orders for each product as shown in Appendix A.2. In order to validate the proposed hybrid algorithm as a reliable and effective approach in comparison with other algorithms, the test is repeated. The results are shown in Figure 7.13

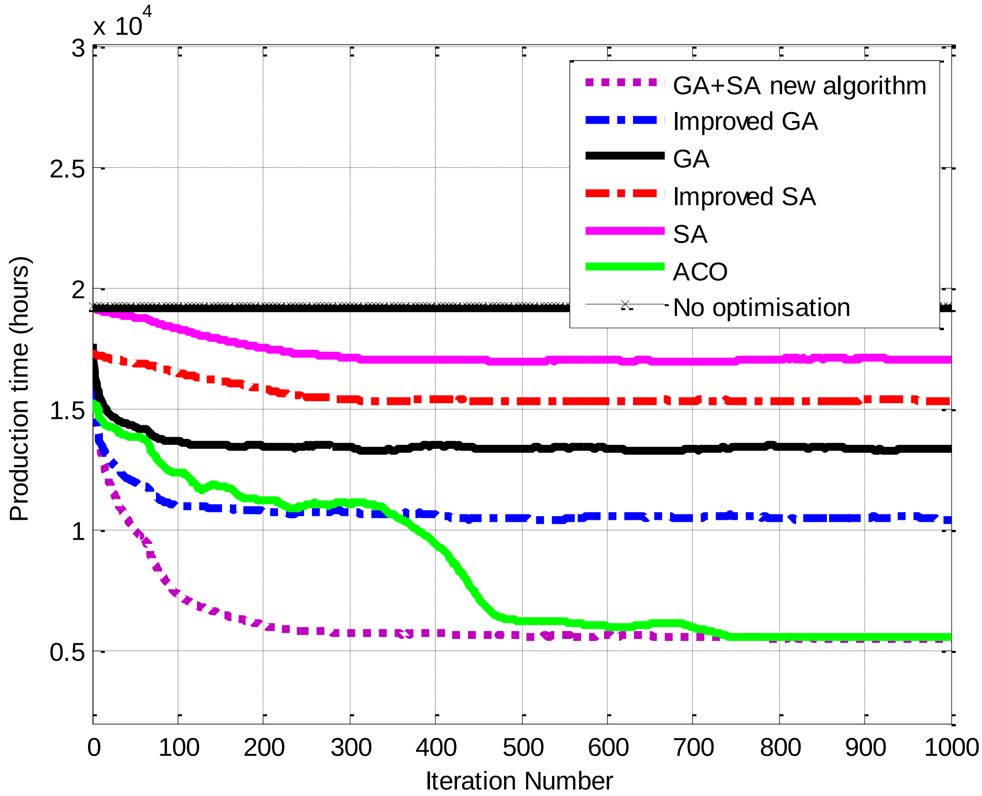

Figure 7.13: Comparison of optimisation results for different algorithms (Scenario 2)

Figure 7.13 shows the comparison of the optimisation results for the different algorithms. In this particular Scenario 2, the result of the proposed hybrid algorithm performs slightly better than SA, but the convergence is much faster than ACO. The computational results are given in Table 7.5 and the optimal sequences of vessel are given in Appendix B.

Table 7.5: Computational results of total production time of each algorithm (Scenario 2)

	Algorithms	Total production time (hrs)
1	Hybrid algorithm	5459
2	Improved GA	10388
3	GA	13324
4	ACO	5536
5	Improved SA	15326
6	SA	17023
7	No optimisation	19248

The total production time of different algorithms that we have obtained; the result of the hybrid algorithm is 5459 hours which is optimal and it saves approximately 47%, 59%, 1%, 64%, and 68%, respectively, in comparison with the improved GA, GA, ACO, improved SA and SA. Particularly, it saves approximately 72% in the case of no optimisation.

7.3.7.3 Scenario 3

Similarly, as for Scenario 2, Scenario 3 corresponds to the different sets of data for 100 orders for each product. These are shown in Appendix A.3. It is assumed that orders have higher demand for each product type. The results are shown in Figure 7.14 and the optimal sequences of vessel are given in Appendix B

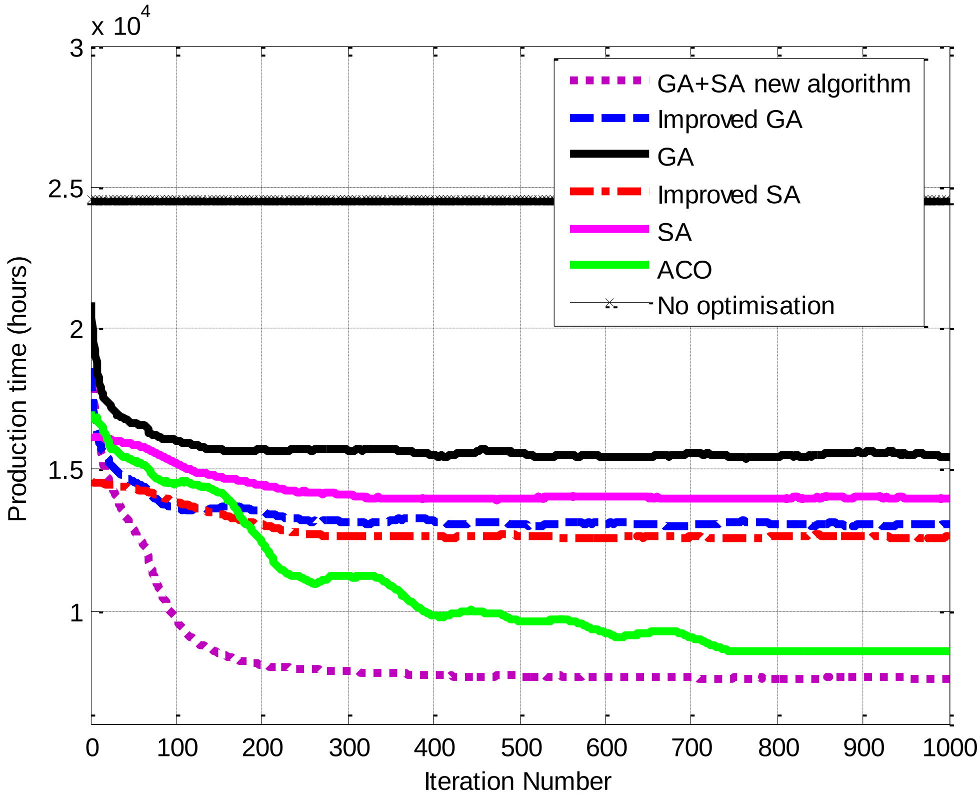

Figure 7.14: Comparison of optimisation results for different algorithms (Scenario 3)

The computational results are given in Table 7.6:

Table 7.6: Computational results of total production time of each algorithm (Scenario 3)

	Algorithms	Total production time (hrs)
1	Hybrid algorithm	7597
2	Improved GA	13020
3	GA	15449
4	ACO	8559
5	Improved SA	12595
6	SA	13998
7	No optimisation	24552

Figure 7.14, shows the optimisation results, once again the proposed hybrid algorithm performs much better than other algorithms and has a significant improvement in this particular scenario. The result of the hybrid algorithm is 7597 hours which is optimal and it saves approximately 42%, 51%, 11%, 40%, and 46%, respectively, in comparison with the improved GA, GA, ACO, improved SA and SA. Particularly, it saves approximately 69% in the case of no optimisation.

7.4 Conclusion

In this chapter, we have applied the GA, SA, ACO, improved GA and improved SA to optimise a typical micro brewery production system. Different algorithms have been compared with the hybrid algorithm as proposed in Section 6.5. The comparison results have demonstrated that the proposed algorithm gives better performance and effective ability to search optimisation solutions in order to minimise the total production time. It has not only achieved the combination of the global search capability of the GA and the local search capability of the SA, but can also help the SA to take full advantage of the global information from the GA. Therefore, the proposed method provides an optimal solution in comparison with other algorithms and it can be effectively applied to improve both the performance and efficiency of the micro brewery production systems.

Chapter 8: Conclusions and Future Work

8.1 Summary

In this chapter, the main contributions of this research and directions for future work are summarised based on the outcomes presented in the preceding chapters as given. The aims of this research were to develop standard simulation models and a generic mathematical model for optimising a micro brewery production system via different heuristic algorithms. By achieving these goals, the following have been completed:

- The scheduling problem and optimisation methods are reviewed critically in Chapter 2
- The rationales, concept, principle, applications of the meta-heuristic methods that are implemented for optimising the brewery production system have been explored in Chapter 3

- The brewery production process has been investigated in Chapter 4
- Production scheduling problems are formulated by mathematical models in Chapter 4
- The Simulink model is modelled in Chapter 5 to simulate the scenarios of the brewery production system
- The traditional GA and SA are improved in Chapter 6
- A hybrid algorithm is proposed to integrate the improved GA with the SA to optimise the brewery production system in Chapter 6
- The proposed hybrid algorithm is implemented and verified as contrasted with GA, SA, ACO, the improved GA and the improved SA, for optimising a real-life brewery production process in Chapter 7

8.2 Contributions of the research

There are four main contributions that have been made in this research:

- The generic mathematical model is formulated in Chapter 4, which is used to tackle the problems for the brewery production system. Various product types are to be produced in different vessels with limited capacity; the operation of the brewery is determined by the processing time, setting up time, changeover time of each product, and cleaning time of each vessel, as well as the due date based on consideration of the customer demand. The model needs to satisfy with the full conditions and constraints within the brewery production yet also to minimise the production time.

- The Simulink model is developed in Chapter 5 which is modelled to observe the performance of each stage of the production process for a brewery production system. It considered three product types to be produced in three parallel vessels with limited capacity. Orders of each product arrive randomly and accumulate ready for the batch production; each batch will be equivalent to the maximum capacity of each vessel. The operation of the brewery is determined by setting up time, production time, changeover time of products, and cleaning time of vessels. Management level decision making employs the dispatching rules which allocates the resources optimally in order to achieve the targets that include priority rules, due date, "first come first served", shortest processing time, changeover, etc. The Simulink model achieves these targets which include the sequence of orders, the sequence of the batch, the decision making strategies (workload of full capacity), the changeover condition of different product types which are produced in different vessels and production time. Furthermore, the parameters of the model can be easily changed at different stages. It has a feasibility which may be applied in any similar production process. For example, it can be extended to cater for more product types and more vessels, should the micro brewery expand. To the best knowledge of the author, limited to the literature survey, no other such model in Simulink exist for a micro brewery production process.
- The hybrid algorithm is proposed in Chapter 6. It integrates the improved GA with the improved SA as the hybrid method in order to optimise the brewery production system. Based on the drawbacks of the GA and the SA, the GA is improved by the real number encoding, instead of tradition binary encoding,

and adaptive adjustment of crossover probability and mutation probability in order to improve the searching capability solution. The SA is improved by the computational method to solve the problem of local minimum effectively when the population is decentralised, and solves the problem of the jump distance to prevent single population; also the acceptance probability of the temperature drop function is improved to speed up the convergence. Subsequently, the hybrid algorithm has combined the advantages of the improved GA and improved SA to achieve the optimal solution.

- The validation of the hybrid algorithm is demonstrated in Chapter 7. It has applied meta-heuristic methods to optimise a typical brewery production system in terms of the total production time. The GA, SA, ACO, improved GA, and improved SA have been implemented to compare with the proposed hybrid method. The optimal result of the hybrid method is obtained and shows better performance than other heuristic algorithms. It could save approximately 10%, 47%, 9%, 5%, and 30%, separately, as contrasted with GA, SA, ACO, improved GA, and improved SA.

In conclusion, the mathematical model of micro brewery production scheduling problem have been developed and investigated. To begin, the production scheduling methods were reviewed as preliminary knowledge engaged in exploring the rationales, concepts and problems of production scheduling. The application of optimisation methods is discussed for optimising the scheduling problem related to the brewery production system. Furthermore, the optimisation based simulation model of the brewery production scheduling problem is formulated mathematically in order to minimise the production time and maximise the performance. Moreover, the Simulink model is developed and proposed to observe the performance of each stage in the brewery production. The dispatching rules are employed to allocate resources optimally according to the real dynamical production process that some main factors have achieved in the research. This includes the sequence of orders, the sequence of batches, the decision making strategies, the occurrences of changeovers, and the production time. Subsequently, the meta-heuristic algorithms have been implemented to optimise the brewery production system that includes the GA, SA and ACO. Due to the shortfalls of these algorithms, it may result in the premature convergence, trapped in local optima, etc. In order to solve these problems, the GA is improved by the real-number encoding and adaptive adjustment of crossover probability and mutation probability that can improve the computational efficiency and obtain the global optimal solutions effectively. The SA is improved by the computational method of the jump distance and the acceptance probability of the temperature drop function that can solve the local minimum problem and speed up the convergence. Finally, the most significant contribution in this thesis is the hybrid method which has been proposed to integrate the improved GA with the improved SA. The result shows that the hybrid method outperforms the GA, SA, ACO, improved GA and improved SA. The efficiency of the proposed algorithm has been demonstrated when applied to the simulation model of a micro brewery batch production system.

8.3 Future Research

As for future research directions, the following issues need further development:

(1) The Simulink model is developed to control the entire beer production system automatically and observe the performance. Due to the limitations and facts of reality, some factors may need to be extended further in the model, such as order cancellations, machine breakdown, labour strike, etc.

(2) Although the proposed model proved its efficiency, it only considers the minimum operation time. In the actual brewery, there are many issues that need to be investigated further, such as the problem of delays in shipment, raw material supplement, inventory problems, supply chain management, etc.

(3) The future investigation would focus on the very latest algorithms to solve these problems. The proposed methodology for production scheduling requires more tests before implementation within real production environment of the micro brewery.

(4) A further phase would be to optimise alongside the current manual scheduling procedure. The development approach could be used to train new brewery staff to guide them to take the effective decisions in the everyday scheduling of the production.

Appendix A

The 100 random raw orders have been received daily in 100 days sequence for each product type, each number represented as barrels of each order in each day as follows:

A.1:

Product type A

13	22	1	10	5	3	6	11	12	17	13
21	7	27	1	21	13	17	5	6	25	30
10	21	27	27	3	2	6	27	3	13	29
16	21	10	21	26	1	23	30	23	9	24
4	14	28	9	9	4	1	21	7	8	15
2	18	5	18	21	4	13	21	13	2	17
20	16	29	18	28	5	5	25	12	5	28
11	23	22	27	19	23	11	9	27	13	29
20	19	4	29	14	18	13	8	28	18	1
19										

Product type B

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

Product type C

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

A.2:

Product type A

4	6	1	3	2	1	2	3	4	5	4
6	2	8	1	6	4	5	2	2	7	8
3	6	8	8	1	1	2	8	1	4	8
5	6	3	6	7	1	7	8	6	3	7
1	4	8	3	3	2	1	6	2	3	4
1	5	2	5	6	1	4	6	4	1	5
6	5	8	5	8	2	2	7	4	2	8
3	7	6	8	5	7	3	3	8	4	8
6	5	1	8	4	5	4	2	8	5	1
5										

Product type B

14	1	17	14	13	10	7	19	9	9	19
16	5	16	6	24	26	15	26	3	16	2
13	3	4	18	7	4	7	11	15	7	20
15	16	12	24	18	5	22	29	16	27	11
18	13	14	24	17	29	17	3	11	26	13
1	8	3	30	30	25	19	23	6	9	16
11	2	30	14	16	10	8	12	25	23	12
1	24	9	18	1	20	12	15	13	11	17
30	4	10	2	23	20	7	13	20	20	6
27										

Product type C

9	11	5	8	14	14	2	4	1	7	1
7	10	5	11	9	1	9	4	7	5	11
7	3	9	12	5	4	6	15	15	11	14
13	6	2	10	9	6	4	7	8	5	5
7	13	9	5	5	7	4	4	8	2	8
6	11	12	11	11	6	11	6	9	5	7
1	4	15	4	11	10	11	8	9	2	2
3	3	2	2	9	3	15	11	9	11	4
14	13	11	8	13	12	10	14	10	6	9
3										

A.3:

Product type A

21	37	1	16	8	5	10	18	20	27	21
35	11	44	2	34	21	28	8	10	41	49
16	35	44	45	5	2	9	44	5	22	48
27	35	16	35	42	1	38	50	38	15	40
6	23	46	15	15	7	1	34	11	14	25
3	29	8	30	35	6	21	35	21	3	27
34	26	48	30	46	7	7	41	20	9	47
18	38	37	45	32	38	18	14	45	22	49
34	32	6	48	23	29	21	12	46	29	1
31										

Product type B

14	1	17	14	13	10	7	19	9	9	19
16	5	16	6	24	26	15	26	3	16	2
13	3	4	18	7	4	7	11	15	7	20
15	16	12	24	18	5	22	29	16	27	11
18	13	14	24	17	29	17	3	11	26	13
1	8	3	30	30	25	19	23	6	9	16
11	2	30	14	16	10	8	12	25	23	12
1	24	9	18	1	20	12	15	13	11	17
30	4	10	2	23	20	7	13	20	20	6
27										

Product type C

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

Appendix B

The sequences of vessels have been obtained that include the no optimisation, GA, SA, ACO, improved GA, improved SA, and hybrid algorithm. Number 1, 2, and 3 represents vessel 1, vessel 2 and vessel 3, respectively.

B.1: No optimisation

Production sequence of vessels without the use of optimisation for product type A:

3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3										

Production sequence of vessels without the use of optimisation for product type B:

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1										

Production sequence of vessels without the use of optimisation for product type C:

2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2
2										

B.2: GA optimisation

Optimised production sequence of vessels for product type A:

Scenario 1:

3	3	3	3	3	3	3	3	3	3	3
3	3	3	2	3	2	2	2	2	3	1
3	2	1	3	3	2	2	2	2	3	2
3	3	2	2	2	2	2	1	1	2	1
1	1	1	1	1	2	3	3	2	1	1
2	3	2	1	1	1	3	1	2	1	1
1	1	2	1	1	2	1	1	3	2	1
3	2	2	1	1	2	2	2	2	2	1
2	1	1	3	1	1	1	1	1	2	1
1										

Scenario 2

3	3	3	3	3	3	3	3	2	2	2
2	3	3	2	3	2	3	2	3	3	2
3	3	3	3	2	2	2	1	1	1	2
3	2	2	1	2	1	3	1	2	2	1
1	3	3	2	2	1	1	2	1	3	2
1	3	2	3	1	2	1	1	1	1	1
3	3	2	3	2	2	1	1	3	2	1
3	2	2	1	1	2	2	2	2	2	1
2	1	1	3	1	1	1	1	1	2	1
1										

Scenario 3

3	3	3	3	3	3	3	3	3	2	2
3	3	3	2	3	3	3	2	2	2	2
3	3	3	2	2	1	2	3	2	3	3
2	3	2	2	3	3	1	3	1	2	1
2	2	2	1	2	3	2	2	2	2	3
2	2	1	3	1	3	2	3	1	1	1
2	2	3	2	2	1	2	2	2	1	2
2	1	3	2	2	2	1	3	1	2	2
1	1	1	1	1	2	1	1	1	1	1
1										

Optimised production sequence of vessels for product type B:

Scenario 1:

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	2
1	1	2	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	2	2	1	2
2	2	2	2	2	1	1	1	1	2	2
1	1	1	2	2	2	1	2	1	2	2
2	2	1	2	2	1	2	2	1	1	2
1	1	1	2	2	1	1	1	1	1	2
1	2	2	1	2	2	2	2	2	1	2
2										

Scenario 2:

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	2	2	2	1
1	1	1	2	1	2	1	2	1	1	2
2	1	1	1	1	2	2	1	2	1	1
2	1	1	1	2	1	2	2	2	2	2
1	1	1	1	1	1	2	2	1	1	2
1	1	1	2	2	1	1	1	1	1	2
1	2	2	1	2	2	2	2	2	1	2
2										

Scenario 3:

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	2	1	1	1	1	1
1	1	1	1	1	1	2	1	2	1	2
1	1	1	2	1	1	1	1	1	1	1
1	1	2	1	2	1	1	1	2	2	2
1	1	1	1	1	2	1	1	1	2	1
1	2	1	1	1	1	2	1	2	1	1
2	2	2	2	2	1	2	2	2	2	2
2										

Optimised production sequence of vessels for product type C:

Scenario 1:

2	2	2	2	2	2	2	2	2	2	2
2	2	2	3	2	3	3	3	3	2	3
2	3	3	2	2	3	3	3	3	2	3
2	2	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	2	2	3	3	3
3	2	3	3	3	3	2	3	3	3	3
3	3	3	3	3	3	3	3	2	3	3
2	3	3	3	3	3	3	3	3	3	3
3	3	3	2	3	3	3	3	3	3	3
3										

Scenario 2:

2	2	2	2	2	2	2	2	3	3	3
3	2	2	3	2	3	2	3	2	2	3
2	2	2	2	3	3	3	3	3	3	3
2	3	3	3	3	3	2	3	3	3	3
3	2	2	3	3	3	3	3	3	2	3
3	2	3	2	3	3	3	3	3	3	3
2	2	3	2	3	3	3	3	2	3	3
2	3	3	3	3	3	3	3	3	3	3
3	3	3	2	3	3	3	3	3	3	3
3										

Scenario 3:

2	2	2	2	2	2	2	2	2	3	3
2	2	2	3	2	2	2	3	3	3	3
2	2	2	3	3	3	3	2	3	2	2
3	2	3	3	2	2	3	2	3	3	3
3	3	3	3	3	2	3	3	3	3	2
3	3	3	2	3	2	3	2	3	3	3
3	3	2	3	3	3	3	3	3	3	3
3	3	2	3	3	3	3	2	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3										

B.3: SA optimisation

Optimised production sequence of vessels for product type A:

Scenario 1:

3	3	3	3	3	3	3	3	2	3	3
3	2	3	2	3	2	3	3	1	3	2
3	3	3	2	3	3	2	2	2	3	1
2	2	3	2	2	2	2	3	1	3	2
2	3	1	2	2	3	3	3	2	1	2
2	3	2	3	1	1	2	1	2	1	3
1	1	1	1	1	1	2	1	3	2	1
3	1	2	1	2	1	2	1	1	1	1
1	1	2	1	1	1	1	1	1	1	1
1										

Scenario 2:

3	3	3	3	3	3	2	3	3	3	2
3	3	3	3	3	2	2	2	2	1	3
3	3	3	3	3	1	2	2	3	3	1
2	3	3	3	2	3	2	1	3	2	2
3	2	2	1	2	2	2	2	3	2	1
1	2	2	2	2	1	1	1	1	1	1
2	1	2	2	2	1	2	1	1	2	1
2	1	1	1	3	1	1	1	1	1	1
1	2	2	1	1	1	1	1	1	1	1
1										

Scenario 3:

3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	2	3	3	3	3	3	3
2	1	3	3	3	3	3	2	3	2	2
2	3	3	2	3	1	3	3	2	3	3
1	2	2	2	2	2	2	1	2	3	2
2	1	2	1	1	3	1	3	2	2	1
2	1	1	2	2	1	2	1	2	1	2
1	2	1	1	2	1	2	2	1	1	2
2	2	1	1	1	1	2	1	1	1	1
1										

Optimised production sequence of vessels for product type B:

Scenario 1:

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	2	1	1
1	1	1	1	1	1	1	1	1	1	2
1	1	1	1	1	1	1	1	2	1	1
1	1	2	1	1	1	1	1	1	2	1
1	1	1	1	2	2	1	2	1	2	1
2	2	2	2	2	2	1	2	1	1	2
1	2	1	2	1	2	1	2	2	2	2
2	2	1	2	2	2	2	2	2	2	2
2										

Scenario 2:

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	2	1
1	1	1	1	1	2	1	1	1	1	2
1	1	1	1	1	1	1	2	1	1	1
1	1	1	2	1	1	1	1	1	1	2
2	1	1	1	1	2	2	2	2	2	2
1	2	1	1	1	2	1	2	2	1	2
1	2	2	2	1	2	2	2	2	2	2
2	1	1	2	2	2	2	2	2	2	2
2										

Scenario 3:

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	2	1	1	1	1	1	1	1	1	1
1	1	1	1	1	2	1	1	1	1	1
2	1	1	1	1	1	1	2	1	1	1
1	2	1	2	2	1	2	1	1	1	2
1	2	2	1	1	2	1	2	1	2	1
2	1	2	2	1	2	1	1	2	2	1
1	1	2	2	2	2	1	2	2	2	2
2										

Optimised production sequence of vessels for product type C:

Scenario 1:

2	2	2	2	2	2	2	2	3	2	2
2	3	2	3	2	3	2	2	3	2	3
2	2	2	3	2	2	3	3	3	2	3
3	3	2	3	3	3	3	2	3	2	3
3	2	3	3	3	2	2	2	3	3	3
3	2	3	2	3	3	3	3	3	3	2
3	3	3	3	3	3	3	3	2	3	3
2	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3										

Scenario 2:

2	2	2	2	2	2	3	2	2	2	3
2	2	2	2	2	3	3	3	3	3	2
2	2	2	2	2	3	3	3	2	2	3
3	2	2	2	3	2	3	3	2	3	3
2	3	3	3	3	3	3	3	2	3	3
3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	2	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3										

Scenario 3:

2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	3	2	2	2	2	2	2
3	3	2	2	2	2	2	3	2	3	3
3	2	2	3	2	3	2	2	3	2	2
3	3	3	3	3	3	3	3	3	2	3
3	3	3	3	3	2	3	2	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3										

B.4: ACO optimisation

Optimised production sequence of vessels for product type A:

Scenario 1:

3	3	3	3	3	3	3	3	3	3	3
3	3	3	2	2	3	2	3	3	2	2
3	2	3	2	1	3	3	2	3	3	3
1	3	3	2	2	3	1	3	2	2	3
2	2	1	3	3	2	1	2	3	1	2
2	1	3	1	1	1	2	1	1	2	2
2	1	3	2	2	1	1	1	2	3	1
1	3	2	1	2	3	2	1	2	1	1
1	1	1	2	2	1	1	3	1	1	2
1										

Scenario 2:

3	3	3	3	3	3	3	3	3	3	3
3	3	3	2	3	2	2	2	2	3	1
3	2	1	3	3	2	2	2	2	3	2
3	3	2	2	2	2	2	1	1	2	1
1	1	1	1	1	2	3	3	2	1	1
2	3	2	1	1	1	3	1	2	1	1
1	1	2	1	1	2	1	1	3	2	1
3	2	2	1	1	2	2	2	2	2	1
2	1	1	3	1	1	1	1	1	2	1
1										

Scenario 3:

3	3	3	3	2	3	3	3	3	1	3
3	3	3	3	2	3	2	2	2	2	2
3	1	2	3	3	2	3	3	3	3	3
2	3	2	2	2	3	3	3	2	2	2
2	1	2	1	2	2	2	3	2	2	2
2	3	3	1	1	1	2	1	3	2	2
2	3	1	2	1	1	2	2	1	2	1
3	1	1	2	1	1	1	3	2	3	1
2	1	1	1	2	1	1	1	1	2	2
1										

Optimised production sequence of vessels for product type B:

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	2	1	1	1	1	1	1
2	1	1	1	1	1	2	1	1	1	1
1	1	2	1	1	1	2	1	1	2	1
1	2	1	2	2	2	1	2	2	1	1
1	2	1	1	1	2	2	2	1	1	2
2	1	1	2	1	1	1	2	1	2	2
2	2	2	1	1	2	2	1	2	2	1
2										

Scenario 2:

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	2
1	1	2	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	2	2	1	2
2	2	2	2	2	1	1	1	1	2	2
1	1	1	2	2	2	1	2	1	2	2
2	2	1	2	2	1	2	2	1	1	2
1	1	1	2	2	1	1	1	1	1	2
1	2	2	1	2	2	2	2	2	1	2
2										

Scenario 3:

1	1	1	1	1	1	1	1	1	2	1
1	1	1	1	1	1	1	1	1	1	1
1	2	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	2	1	2	1	1	1	1	1	1	1
1	1	1	2	2	2	1	2	1	1	1
1	1	2	1	2	2	1	1	2	1	2
1	2	2	1	2	2	2	1	1	1	2
1	2	2	2	1	2	2	2	2	1	1
2										

Optimised production sequence of vessels for product type C:

Scenario 1:

2	2	2	2	2	2	2	2	2	2	2
2	2	2	3	3	2	3	2	2	3	3
2	3	2	3	3	2	2	3	2	2	2
3	2	2	3	3	2	3	2	3	3	2
3	3	3	2	2	3	3	3	2	3	3
3	3	2	3	3	3	3	3	3	3	3
3	3	2	3	3	3	3	3	3	2	3
3	2	3	3	3	2	3	3	3	3	3
3	3	3	3	3	3	3	2	3	3	3
3										

Scenario 2:

2	2	2	2	2	2	2	2	2	2	2
2	2	2	3	2	3	3	3	3	2	3
2	3	3	2	2	3	3	3	3	2	3
2	2	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	2	2	3	3	3
3	2	3	3	3	3	2	3	3	3	3
3	3	3	3	3	3	3	3	2	3	3
2	3	3	3	3	3	3	3	3	3	3
3	3	3	2	3	3	3	3	3	3	3
3										

Scenario 3:

2	2	2	2	3	2	2	2	2	3	2
2	2	2	2	3	3	2	3	3	3	3
2	3	3	2	2	3	2	2	2	2	2
3	2	3	3	3	2	2	2	3	3	3
3	3	3	3	3	3	3	2	3	3	3
3	2	2	3	3	3	3	3	2	3	3
3	2	3	3	3	3	3	3	3	3	3
2	3	3	3	3	3	3	2	3	2	3
3	3	3	3	3	3	3	3	3	3	3
3										

B.5: Improved GA optimisation

Optimised production sequence of vessels for product type A:

Scenario 1:

3	3	3	3	3	3	3	3	3	3	3
3	3	3	2	3	2	2	2	2	3	1
3	2	1	3	3	2	2	2	2	3	2
3	3	2	2	2	2	2	1	1	2	1
1	1	1	1	1	2	3	3	2	1	1
2	3	2	1	1	1	3	1	2	1	1
1	1	2	1	1	2	1	1	3	2	1
3	2	2	1	1	2	2	2	2	2	1
2	1	1	3	1	1	1	1	1	2	1
1										

Scenario 2:

3	3	3	2	2	3	2	3	1	3	3
3	3	3	3	2	3	3	3	2	2	3
3	1	1	2	1	2	2	3	3	2	2
2	1	2	2	2	1	3	3	1	3	2
1	2	1	3	2	1	1	2	1	3	2
3	1	1	1	1	1	3	1	2	1	3
1	3	1	1	1	1	1	2	2	3	1
3	1	3	1	2	1	1	2	1	1	2
1	1	2	1	1	2	1	1	2	1	2
1										

Scenario 3:

3	3	3	3	3	3	3	2	3	2	3
3	3	3	2	3	2	2	3	2	2	2
3	1	2	3	3	2	2	3	1	2	3
2	2	2	3	2	3	3	3	1	2	3
3	1	2	3	2	1	2	2	2	3	2
2	3	1	1	1	2	2	1	2	3	3
2	2	1	2	1	1	1	2	1	1	1
2	3	1	2	1	2	1	1	1	3	1
2	1	1	3	1	1	1	1	1	2	1
1										

Optimised production sequence of vessels for product type B:

Scenario 1:

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	2
1	1	2	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	2	2	1	2
2	2	2	2	2	1	1	1	1	2	2
1	1	1	2	2	2	1	2	1	2	2
2	2	1	2	2	1	2	2	1	1	2
1	1	1	2	2	1	1	1	1	1	2
1	2	2	1	2	2	2	2	2	1	2
2										

Scenario 2:

1	1	1	1	1	1	1	1	2	1	1
1	1	1	1	1	1	1	1	1	1	1
1	2	2	1	2	1	1	1	1	1	1
1	2	1	1	1	2	1	1	2	1	1
2	1	2	1	1	2	2	1	2	1	1
1	2	2	2	2	2	1	2	1	2	1
2	1	2	2	2	2	2	1	1	1	2
1	2	1	2	1	2	2	1	2	2	1
2	2	1	2	2	1	2	2	1	2	1
2										

Scenario 3:

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	2	1	1	1	1	1	1	2	1	1
1	1	1	1	1	1	1	1	2	1	1
1	2	1	1	1	2	1	1	1	1	1
1	1	2	2	2	1	1	2	1	1	1
1	1	2	1	2	2	2	1	2	2	2
1	1	2	1	2	1	2	2	2	1	2
1	2	2	1	2	2	2	2	2	1	2
2										

Optimised production sequence of vessels for product type C:

2	2	2	2	2	2	2	2	2	2	2
2	2	2	3	2	3	3	3	3	2	3
2	3	3	2	2	3	3	3	3	2	3
2	2	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	2	2	3	3	3
3	2	3	3	3	3	2	3	3	3	3
3	3	3	3	3	3	3	3	2	3	3
2	3	3	3	3	3	3	3	3	3	3
3	3	3	2	3	3	3	3	3	3	3
3										

Scenario 2:

2	2	2	3	3	2	3	2	3	2	2
2	2	2	2	3	2	2	2	3	3	2
2	3	3	3	3	3	3	2	2	3	3
3	3	3	3	3	3	2	2	3	2	3
3	3	3	2	3	3	3	3	3	2	3
2	3	3	3	3	3	2	3	3	3	2
3	2	3	3	3	3	3	3	3	2	3
2	3	2	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3										

Scenario 3:

2	2	2	2	2	2	2	3	2	3	2
2	2	2	3	2	3	3	2	3	3	3
2	3	3	2	2	3	3	2	3	3	2
3	3	3	2	3	2	2	2	3	3	2
2	3	3	2	3	3	3	3	3	2	3
3	2	3	3	3	3	3	3	3	2	2
3	3	3	3	3	3	3	3	3	3	3
3	2	3	3	3	3	3	3	3	2	3
3	3	3	2	3	3	3	3	3	3	3
3										

B.6: Improved SA optimisation

Optimised production sequence of vessels for product type A:

Scenario 1:

3	3	3	3	3	3	3	3	2	2	3
2	3	3	3	2	3	2	3	3	3	2
3	2	3	3	3	3	2	2	3	3	3
1	3	3	3	2	1	2	1	2	3	2
2	2	3	2	1	2	2	2	2	2	2
2	2	3	2	2	3	1	2	2	2	1
2	1	1	3	2	2	2	1	3	1	2
1	1	1	2	1	2	1	1	2	1	1
1	1	1	1	1	1	1	1	2	1	1
1										

Scenario 2:

3	2	3	3	3	3	2	2	3	3	3
3	2	3	3	3	3	3	2	3	3	3
2	3	3	3	3	2	3	3	2	3	1
2	3	2	3	2	2	2	2	3	3	3
2	2	2	2	2	2	3	3	2	1	2
2	3	2	1	1	1	3	2	1	2	3
1	2	2	2	1	1	1	2	1	1	2
1	2	1	2	2	1	1	2	1	1	1
1	1	2	2	1	2	1	1	1	1	1
1										

Scenario 3:

3	3	3	3	3	3	3	3	3	3	2
2	2	3	3	3	3	3	3	3	3	2
2	3	2	3	2	3	2	2	2	1	3
2	2	2	3	2	2	3	3	1	3	2
2	2	2	2	2	1	3	1	2	3	2
1	1	2	3	2	1	2	1	2	1	1
1	1	2	1	2	3	2	2	1	1	1
1	2	1	1	2	1	1	1	1	2	1
1	1	1	1	1	1	1	1	1	1	1
1										

Optimised production sequence of vessels for product type B:

Scenario 1:

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	2	1	2	1	1	1
1	1	1	1	2	1	1	1	1	1	1
1	1	1	1	1	1	2	1	1	1	2
1	2	2	1	1	1	1	2	1	2	1
2	2	2	1	2	1	2	2	1	2	2
2	2	2	2	2	2	2	2	1	2	2
2										

Scenario 2:

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	2
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	2	1
1	1	1	2	2	2	1	1	2	1	1
2	1	1	1	2	2	2	1	2	2	1
2	1	2	1	1	2	2	1	2	2	2
2	2	1	1	2	1	2	2	2	2	2
2										

Scenario 3:

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	2	1
1	1	1	1	1	1	1	1	2	1	1
1	1	1	1	1	2	1	2	1	1	1
2	2	1	1	1	2	1	2	1	2	2
2	2	1	2	1	1	1	1	2	2	2
2	1	2	2	1	2	2	2	2	1	2
2	2	2	2	2	2	2	2	2	2	2
2										

Optimised production sequence of vessels for product type C:

Scenario 1:

2	2	2	2	2	2	2	2	3	3	2
3	2	2	2	3	2	3	2	2	2	3
2	3	2	2	2	2	3	3	2	2	2
3	2	2	2	3	3	3	3	3	2	3
3	3	2	3	3	3	3	3	3	3	3
3	3	2	3	3	2	3	3	3	3	3
3	3	3	2	3	3	3	3	2	3	3
3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3										

Scenario 2:

2	3	2	2	2	2	3	3	2	2	2
2	3	2	2	2	2	2	3	2	2	2
3	2	2	2	2	3	2	2	3	2	3
3	2	3	2	3	3	3	3	2	2	2
3	3	3	3	3	3	2	2	3	3	3
3	2	3	3	3	3	2	3	3	3	2
3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3										

Scenario 3:

2	2	2	2	2	2	2	2	2	2	3
3	3	2	2	2	2	2	2	2	2	3
3	2	3	2	3	2	3	3	3	3	2
3	3	3	2	3	3	2	2	3	2	3
3	3	3	3	3	3	2	3	3	2	3
3	3	3	2	3	3	3	3	3	3	3
3	3	3	3	3	2	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3										

B.7: Hybrid algorithm

Optimised production sequence of vessels for product type A:

Scenario 1:

3	3	3	3	3	3	3	3	2	3	1
3	3	2	3	3	3	3	2	2	2	3
2	3	2	2	1	2	3	3	2	3	2
3	3	2	2	2	2	2	1	1	2	1
1	1	1	1	1	2	3	3	2	1	1
2	3	2	1	1	1	3	1	2	1	1
1	1	2	1	1	2	1	1	3	2	1
3	2	2	1	1	2	2	2	2	2	1
2	1	1	3	1	1	1	1	1	2	1
1										

Scenario 2:

3	3	3	3	3	2	3	3	3	3	3
2	2	3	3	3	3	2	1	3	3	3
3	3	3	1	3	2	3	3	1	3	2
2	3	2	3	3	3	3	1	1	2	2
3	2	1	1	3	3	1	1	3	2	2
2	3	2	3	2	2	2	1	1	2	1
1	1	2	2	1	3	1	2	1	2	3
3	1	3	1	2	1	1	2	1	1	2
1	1	2	1	1	2	1	1	2	1	2
1										

Scenario 3:

3	3	3	3	3	3	3	3	3	2	2
3	3	3	2	3	3	3	1	3	2	3
2	3	3	2	2	3	3	2	3	2	3
2	3	1	1	2	2	2	1	2	3	2
2	1	2	1	2	2	2	3	2	2	2
2	3	3	1	1	1	2	1	3	2	2
2	3	1	2	1	1	2	2	1	2	1
3	1	1	2	1	1	1	3	2	3	1
2	1	1	1	2	1	1	1	1	2	2
1										

Optimised production sequence of vessels for product type B:

Scenario 1:

1	1	1	1	1	1	1	1	1	1	2
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	2	1	1	1	1	1	1
1	1	1	1	1	1	1	2	2	1	2
2	2	2	2	2	1	1	1	1	2	2
1	1	1	2	2	2	1	2	1	2	2
2	2	1	2	2	1	2	2	1	1	2
1	1	1	2	2	1	1	1	1	1	2
1	2	2	1	2	2	2	2	2	1	2
2										

Scenario 2:

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	2	1	1	1
1	1	1	2	1	1	1	1	2	1	1
1	1	1	1	1	1	1	2	2	1	1
1	1	2	2	1	1	2	2	1	1	1
1	1	1	1	1	1	1	2	2	1	2
2	2	1	1	2	1	2	1	2	1	1
1	2	1	2	1	2	2	1	2	2	1
2	2	1	2	2	1	2	2	1	2	1
2										

Scenario 3:

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	2	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	2	2	1	1	1	2	1	1	1
1	2	1	2	1	1	1	1	1	1	1
1	1	1	2	2	2	1	2	1	1	1
1	1	2	1	2	2	1	1	2	1	2
1	2	2	1	2	2	2	1	1	1	2
1	2	2	2	1	2	2	2	2	1	1
2										

Optimised production sequence of vessels for product type C:

Scenario 1:

2	2	2	2	2	2	2	2	3	2	3
2	2	3	2	2	2	2	3	3	3	2
3	2	3	3	3	3	2	2	3	2	3
2	2	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	2	2	3	3	3
3	2	3	3	3	3	2	3	3	3	3
3	3	3	3	3	3	3	3	2	3	3
2	3	3	3	3	3	3	3	3	3	3
3	3	3	2	3	3	3	3	3	3	3
3										

Scenario 2:

2	2	2	2	2	3	2	2	2	2	2
3	3	2	2	2	2	3	3	2	2	2
2	2	2	3	2	3	2	2	3	2	3
3	2	3	2	2	2	2	3	3	3	3
2	3	3	3	2	2	3	3	2	3	3
3	2	3	2	3	3	3	3	3	3	3
3	3	3	3	3	2	3	3	3	3	2
2	3	2	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3
3										

Scenario 3:

2	2	2	2	2	2	2	2	2	3	3
2	2	2	3	2	2	2	3	2	3	2
3	2	2	3	3	2	2	3	2	3	2
3	2	3	3	3	3	3	3	3	2	3
3	3	3	3	3	3	3	2	3	3	3
3	2	2	3	3	3	3	3	2	3	3
3	2	3	3	3	3	3	3	3	3	3
2	3	3	3	3	3	3	2	3	2	3
3	3	3	3	3	3	3	3	3	3	3
3										

References

Abdelmaguid, T. F. (2009). Permutation-induced acyclic networks for the job shop scheduling problem. *Applied Mathematical Modelling*, *33*(3), 1560-1572.

Ackley, D. (2012). *A Connectionist Machine for Genetic Hillclimbing* (Vol. 28). Springer Science & Business Media.

Advantage Software limited (2008) [Online] Available from < http://www.advantages.co.nz/arena.asp> [2008]

Aguilar-Saven, R. S. (2004). Business process modelling: Review and framework. *International Journal of production economics*, *90*(2), 129-149.

Al-Khayyal, F., Griffin, P. M., & Smith, N. R. (2001). Solution of a large-scale two-stage decision and scheduling problem using decomposition. *European Journal of Operational Research*, *132*(2), 453-465.

Ankenbrandt A. (1994). An extension to the theory of convergence and a proof the time complexity of genetic algorithms [J] In Foundations of genetic algorithms, p53

Baker, J. E. (1985, July). Adaptive selection methods for genetic algorithms.

In*Proceedings of an International Conference on Genetic Algorithms and their applications* (pp. 101-111).

Baker, K.R. (1974) *Introduction to sequencing and scheduling*. New York: Wiley & Sons.

Banks, J., Carson, J. S., & Nelson, B. L. (2000). *DM Nicol, Discrete-Event System Simulation*. Englewood Cliffs, NJ, USA: Prentice hall.

Barros, R. C., de Carvalho, A. C., & Freitas, A. A. (2015). *Automatic Design of Decision-Tree Induction Algorithms*. Cham: Springer International Publishing.

Benedettini, O., & Tjahjono, B. (2009). Towards an improved tool to facilitate simulation modelling of complex manufacturing systems. *The International Journal of Advanced Manufacturing Technology*, *43*(1-2), 191-199.

Beasley, J. E., & Chu, P. C. (1996). A genetic algorithm for the set covering problem. *European Journal of Operational Research*, *94*(2), 392-404.

Blackstone, J., D. Phillips and G. Hogg (1982), "A state-of-the-art survey of dispatching rules for manufacturing job shop operations," *International Journal of Production Research*, 20 (1): 27-45.

Bonvin, D. (1998). Optimal operation of batch reactors—a personal view.*Journal of process control*, *8*(5), 355-368.

Bosilj-Vuksic, V., Ceric, V., & Hlupic, V. (2007). Criteria for the evaluation of business process simulation tools. *Interdisciplinary Journal of Information, Knowledge, and Management*, *2*, 73-88.

Box, M. J. (1965). A new method of constrained optimization and a comparison with other methods. *The Computer Journal*, *8*(1), 42-52.

British beer and pub association (2015). The beer story. [online] available from <

http://www.beerandpub.com/thebeerstory> [2015]

Bvack, T. (1993). Optimal M utation Rates in Genetic Search. In *Proceedings of the Fifth International Conference on Genetic Algorithms* (pp. 2-8).

Carrillo-Ureta, G. E., Roberts, P. D., & Becerra, V. M. (2001). Genetic algorithms for optimal control of beer fermentation. *The 2001 IEEE International Symposium on Intelligent Control*. Mexico city, mexico, p391-196

Chen.L., and Hu.Z.X. (1992). Research on hybrid production scheduling strategy and its application in multi-segment and multi-species batch process. *Information and Control*, 1992,21(1):6-12

Chen, R.Q. and Ma, S.H. (1999) *Production and operation management*. China: Higher education press

Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, *45*(1), 41-51.

Cerdá, J. (*c.*2006) Optimization methods for batch scheduling. Argentina: The Institute of Technological Development for the Chemical Industry, National University of the Littoral.

Cochocki, A., & Unbehauen, R. (1993). *Neural Networks for Optimization and Signal Processing*. John Wiley & Sons, Inc.

Damodaran, P., & Vélez-Gallego, M. C. (2012). A simulated annealing algorithm to minimize makespan of parallel batch processing machines with unequal job ready times. *Expert Systems with Applications*, *39*(1), 1451-1458.

Dasgupta, D., & Forrest, S. (1999, July). Artificial immune systems in industrial applications. In *Intelligent Processing and Manufacturing of Materials, 1999.*

IPMM'99. Proceedings of the Second International Conference on (Vol. 1, pp. 257-267). IEEE.

Davis, L. (1985, July). Job shop scheduling with genetic algorithms. In *Proceedings of an International Conference on Genetic Algorithms and Their Applications* (Vol. 140). Carnegie-Mellon University Pittsburgh, PA.

Dekkers, A., & Aarts, E. (1991). Global optimization and simulated annealing. *Mathematical Programming*, *50*(1-3)

Di Fonzo, T., & Marini, M. (2011). A Newton's method for benchmarking time series according to a growth rates preservation principle. *IMF Working Papers*, 1-42.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, *26*(1), 29-41.

Dorigo, M., & Gambardella, L. M. (1997). Ant colonies for the travelling salesman problem. *BioSystems*, *43*(2), 73-81.

Eiben, A. E., & Smith, J. E. (2003). Genetic algorithms. In *Introduction to Evolutionary Computing* (pp. 37-69). Springer Berlin Heidelberg.

EETimes (2001) [Online] Available from <http://www.eetimes.com/electronics-news/4197115/Matlab-Simulink-Equation-Solver> [5 July 2001]

Elsayed, E.A and Boucher, T.O. (1994) *Analysis and control of production systems*. USA: Department of Industrial Engineering College of Engineering Rutgers University, 2nd Edition

Floudas, C. A., Aggarwal, A., & Ciric, A. R. (1989). Global optimum search for nonconvex NLP and MINLP problems. *Computers & Chemical Engineering*, *13*(10), 1117-1132.

Garey, M. R., Johnson, D. S., & Sethi, R. (1976). The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research*, *1*(2), 117-129.

Gerst, M. S. (1971). Symbolic coding processes in observational learning. *Journal of Personality and Social Psychology*, *19*(1), 7.

Goldberg, D. E., & Rudnick, M. (1990). Genetic algorithms and the Variance of fitness.

Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine learning*, *3*(2), 95-99.

Grefenstette, J. J. (1986). Optimization of control parameters for genetic algorithms. *Systems, Man and Cybernetics, IEEE Transactions on*, *16*(1), 122-128.

Harjunkoski, I., & Grossmann, I. E. (2001). A decomposition approach for the scheduling of a steel plant production. *Computers & Chemical Engineering*, *25*(11), 1647-1660.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, *79*(8), 2554-2558.

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

Huang, X. W., Zhao, X. Y., & Ma, X. L. (2014). An Improved genetic algorithm for Job-shop scheduling problem with process sequence flexibility. *International Journal of Advanced Manufacturing Technology*, *28*(5-6), 532-540.

Huang.Z.Q. (2002). Beer enterprises lean production research. *Journal of Guangzhou Food Science and Technology*, 18(4): p71-77

Ierapetritou, M. G., & Floudas, C. A. (1998). Short-term scheduling: New mathematical models vs algorithmic improvements. *Computers & Chemical Engineering*, *22*, S419-S426.

Jansen-Vullers, M., & Netjes, M. (2006, October). Business process simulation–a tool survey. In *Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, Aarhus, Denmark*.

Ji, G.L. (2004) Literature Review on Genetic Algorithm. *Computer Applications and Software*, vol21, no.2.

Jong, K. D. (1980). Adaptive system design: a genetic approach. *Systems, Man and Cybernetics, IEEE Transactions on*, *10*(9), 566-574.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671-680.

Konfršt, Z. (2004). Parallel genetic algorithms: Advances, computing trends, applications and perspectives. *18th IPDPS 2004*, 162.

Kudva, G., Elkamel, A., Pekny, J. F., & Reklaitis, G. V. (1994). Heuristic algorithm for scheduling batch and semi-continuous plants with production deadlines, intermediate storage limitations and equipment changeover costs. *Computers & Chemical Engineering*, *18*(9), 859-875.

Laguna, M., & Marklund, J. (2013). *Business process modelling, simulation and design*. CRC Press.

Lei, D. (2012). Co-evolutionary genetic algorithm for fuzzy flexible job shop scheduling. *Applied Soft Computing*, *12*(8), 2237-2245.

Li, R. K., Shyu, Y. T., & Adiga, S. (1993). A heuristic rescheduling algorithm for computer-based production scheduling systems. *The International Journal of Production Research*, *31*(8), 1815-1826.

Liu, H., Xu, Z., & Abraham, A. (2005, September). Hybrid fuzzy-genetic algorithm approach for crew grouping. In *Intelligent Systems Design and Applications, 2005. ISDA'05. Proceedings. 5th International Conference on* (pp. 332-337). IEEE.

Lopez, J. and Roubellat, F. (2008) Production Scheduling. Blackwell: Wiley-Blackwell.

MacCarthy, B. L., & Liu, J. (1993). Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling. *The International Journal of Production Research*, *31*(1), 59-79.

MathWorks. (2011) Simulink: simulation and model-based design. [online] available from <http://www.mathworks.com/products/simulink/> [May 2011]

MathWorks. (2012) [online] Available from < http://www.mathworks.co.uk/ > [2012]

McBride, R. D., & O'Leary, D. E. (1993). The use of mathematical programming with artificial intelligence and expert systems. *European Journal of Operational Research*, *70*(1), 1-15.

Melin, P., & Castillo, O. (2001). *Modelling, simulation and control of non-linear dynamical systems: an intelligent approach using soft computing and fractal theory*. CRC Press.

Merkuryev.Y and Pecherska.J (2005) Discrete-Event Simulation: Methodology and Practice. [online] available from <http://pelincec.isep.pw.edu.pl/doc/Simulation_Warsaw%20Part%205.pdf> [11 Nov 2005]

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, *21*(6), 1087-1092.

Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press.

Miller, J., Potter, W. D., Gandham, R. V., & Lapena, C. N. (1993). An evaluation of local improvement operators for genetic algorithms. *Systems, Man and Cybernetics, IEEE Transactions on*, *23*(5), 1340-1351.

Mockus, L., & Reklaitis, G. V. (1999). Continuous time representation approach to batch and continuous process scheduling. 2. Computational issues. *Industrial & Engineering Chemistry Research*, *38*(1), 204-210.

Ning, N. and Guo, C. (2008) An Improved Genetic Annealing Evolutionary Algorithm. University of Electronic Science and Technolo.gy of China，Chengdu

Nof, S. Y., & Hank Grant, F. (1991). Adaptive/predictive scheduling: review and a general framework. *Production Planning & Control*, *2*(4), 298-312.

Patterson, J. H. (1984). A comparison of exact approaches for solving the multiple constrained resource, project scheduling problem. *Management Science*, *30*(7), 854-867.

Panwalkar, S. S., & Iskander, W. (1977). A survey of scheduling rules. *Operations Research*, *25*(1), 45-61.

Peng, B., Lü, Z., & Cheng, T. C. E. (2015). A tabu search/path relinking algorithm to solve the job shop scheduling problem. *Computers & Operations Research*, *53*, 154-164.

Pinedo, M. L. (2012). *Scheduling: theory, algorithms, and systems*. Springer Science & Business Media.

Pinedo, M. (2008). Scheduling: Theory Algorithms, and Systems (3rd edition), Springer

Pinto, J. M., & Grossmann, I. E. (1994). Optimal cyclic scheduling of multistage continuous multiproduct plants. *Computers & Chemical Engineering*,*18*(9), 797-

816.

ProModel (2011) Justifing simuation: why use simulation? [online] Available from <http://www.promodel.com/aboutus/Justifying%20Simulation.pdf> [19 January 2011]

Prügel-Bennett, A. (2004). When a genetic algorithm outperforms hill-climbing.*Theoretical Computer Science*, *320*(1), 135-153.

Rao, S. S., & Rao, S. S. (2009). *Engineering optimization: theory and practice*. John Wiley & Sons.

Ren Pingping (2010) The information of integration and optimisation in the beer fermentation process. Zhejiang University, China.

Rippin, D. W. T. (1983). Design and operation of multiproduct and multipurpose batch chemical plants. An analysis of problem structure. *Computers & Chemical Engineering*, *7*(4), 463-481

Rockwell (2011) [Online] Available from < http://www.rockwellautomation.com/ > [April 2011] Supersedes Publication: ARENCC-PP001G-EN-P, USA.

Rodammer, F., & White Jr, K. P. (1988). A recent survey of production scheduling. *Systems, Man and Cybernetics, IEEE Transactions on*, *18*(6), 841-851.

Sabmiller (*c.* 2012) The brewing of beer, India. [Online] Available from< http://www.sabmiller.in/know-beer_brewing.html> [*c.* 2012]

Sadeh, N., & Fox, M. S. (1996). Variable and value ordering heuristics for the job shop scheduling constraint satisfaction problem. *Artificial Intelligence*, *86*(1), 1-41.

Sandanayake, N. S., Church, N. I., Chapman, M. H., Johnson, G. J., Dhar, D. K., Amin, Z., ... & Webster, G. J. (2009). Presentation and management of post-treatment relapse in autoimmune pancreatitis/immunoglobulin G4-associated cholangitis. *Clinical Gastroenterology and Hepatology*, *7*(10), 1089-1096.

Shah, N., Pantelides, C. C., & Sargent, R. W. H. (1993). A general algorithm for short-term scheduling of batch operations—II. Computational issues.*Computers & Chemical Engineering*, *17*(2), 229-244.

Shaw, K. J., Lee, P. L., Nott, H. P., & Thompson, M. (2000). Genetic algorithms for multiobjective scheduling of combined batch/continuous process plants. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on* (Vol. 1, pp. 293-300). IEEE.

Shen, Z., Burnham, K. J., and Smalov, L. (2012). Towards formulating a business process simulation model for a brewery production system: preliminary steps. In *Proceeding of 22nd International Conference on Systems Engineering,* 11-13 September, Coventry, UK, 2012.

Shen.Z, Burnham.J.K, Samlov.L, and Amin.S. (2014) Formulating scheduling problem for a manufacturing production system. In *Proceeding of 2014 International Conference on Computer, Network Security and Communication Engineering.* DEStech Pubilications, Inc. (pp. 683-687).

Shen, Z., Burnham, K. J., & Smalov, L. (2015) Optimised Job-Shop Scheduling via Genetic Algorithm for a Manufacturing Production System. In *Progress in Systems Engineering* (pp. 89-92). Springer International Publishing.

Shen, Z., Burnham, K. J., and Smalov, L. (2015) An improved genetic algorithm for optimising a manufacturing production process. In *Proceeding of 13th Annual industrial Simulation Conference,* 1-3 June, Valencia, Spain, 2015.

Shi Bin (2006) To design auto-control system for beer saccharification process. Inner

Mongolia University of Technology, Inner Mongolia, China.

Sidnev, Alexander., Tuominen, Juha and Krassi, Boris.(2005) "Business process modeling and simulation." Industrial information technology laboratory publication, Helsinki University of Technology, ISSN: 951-22-7929-0.

Simon, D. (2013). *Evolutionary optimization algorithms*. John Wiley & Sons.

Seda, M. (2008) Mathematical models of flow shop and job shop scheduling problems. *International Journal of Applied Mathematics and Computer Sciences* 4, (4).

Šeda, M. (2007). Mathematical models of flow shop and job shop scheduling problems. *World Academy of Science, Engineering and Technology*, *1*(31), 122-127.

Song, Y. H., Wang, G. S., Wang, P. Y., & Johns, A. T. (1997, July). Environmental/economic dispatch using fuzzy logic controlled genetic algorithms. In *Generation, Transmission and Distribution, IEE Proceedings-*(Vol. 144, No. 4, pp. 377-382). IET

Song.X and Xiao.Y (2013) An Improved Adaptive Genetic Algorithm. *International Conference on Education Technology and Management Science 2013*. P816-819.

Srinvas M, and Patnaik L M. (1994) Adaptation Probabilities of Crossover and Mutation in Genetic Algorithms [J]. IEEE Trans on Systems, Man and Cybernetics, 1994; 24(4):656-667.

Starkweather, T., Whitley, D., Mathias, K., & McDaniel, S. (1992). *Sequence scheduling with genetic algorithms* (pp. 129-148). Springer Berlin Heidelberg.

Stender, J. (1993). *Parallel genetic algorithms: theory and applications* (Vol. 14). IOS press.

Subbu, R., Sanderson, A. C., & Bonissone, P. P. (1998, September). Fuzzy logic controlled genetic algorithms versus tuned genetic algorithms: an agile manufacturing application. In *Intelligent Control (ISIC), 1998. Held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Intelligent Systems and Semiotics (ISAS), Proceedings*(pp. 434-440). IEEE.

Sun, L., Cheng, X.C. and Liang, Y.C. (2010) Solving Job Shop Scheduling Problem Using Genetic Algorithm with Penalty function. *International Journal of Intelligent Information Processing* 1, (2).

Suresh, V., & Chaudhuri, D. (1993). Dynamic scheduling—a survey of research. *International Journal of Production Economics*, *32*(1), 53-63.

Suwa, H., & Sandoh, H. (2013). *Online Scheduling in Manufacturing*. Springer.

Suwa, H. and Sandoh, H. (2012) *Online Scheduling in Manufacturing: A Cumulative Delay Approach*. London: Springer.

Systems Navigator (2012) [Online] Available from <http://www.systemsnavigator.com/sn_website/?q=arena_simulation_software> [2012]

Szelke, E., & Kerr, R. M. (1994). Knowledge-based reactive scheduling.*Production Planning & Control*, *5*(2), 124-145.

Tamilarasi, A. (2010). An enhanced genetic algorithm with simulated annealing for job-shop scheduling. *International Journal of Engineering, Science and Technology*, *2*(1), 144-151.

Tang, O. (2000) *Planning and Replanning within the Material Requirements Planning Environment: A Transform Approach*. Sweden: Production-Economic Research in Linkoping.

Tao, et al. (2015) *Configurable Intelligent optimization algorithm*: chapter 2, recent advances of intelligent optimization algorithm in manufacturing. Switzerland: Springer international publishing

Thamilselvan, R., & Balasubramanie, P. (2012). Integrating Genetic Algorithm, Tabu Search and Simulated Annealing For Job Shop Scheduling Proble. *International Journal of Computer Applications*, *48*(5), 42-54.

Tomassini, M. (1995). A survey of genetic algorithms. *Annual Reviews of Computational Physics*, *3*(2), 87-118.

Tozkapan, A., Kırca, Ö., & Chung, C. S. (2003). A branch and bound algorithm to minimize the total weighted flowtime for the two-stage assembly scheduling problem. *Computers & Operations Research*, *30*(2), 309-320.

Tsai, C. F., & Tsai, C. W. (2002). A new approach for solving large traveling salesman problem using evolutionary ant rules. In *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on* (Vol. 2, pp. 1540-1545). IEEE.

Van Laarhoven, P. J., Aarts, E. H., & Lenstra, J. K. (1992). Job shop scheduling by simulated annealing. *Operations Research*, *40*(1), 113-125

Vasconcelos. J. A.， Ramirez. J. A., Takahashi. R. H. C. and Saldanha. R. R. (2001) Improvements in Genetic Algorithms. IEEE Trans. Magnetics. Vol. 37. P3414-3417

Wang, Z.Y. (2005) Study on the modelling of temperature control in beer fermentation. Jiangxi science and technology project (224), Jiangxi Province, China.

Wang, H., Ding, H.J and Li, F.L. (2005) The Application of Improved Genetic Algorithm in Function Optimization problem. College of Computer &

Information Engineering，Hohai Univ，Changzhou.

Wang, L., & Tang, D. B. (2011). An improved adaptive genetic algorithm based on hormone modulation mechanism for job-shop scheduling problem. *Expert Systems with Applications*, *38*(6), 7243-7250.

Wight, O.W., (1984). Production and inventory management in the computer age, Van Nostrand Reinhold Company, Inc., New York.

Wu, D. (1987), "An Expert Systems Approach for the Control and Scheduling of Flexible Manufacturing Systems," Ph.D. Dissertation, Pennsylvania State University.

Wu, D., & Ierapetritou, M. G. (2003). Decomposition approaches for the efficient solution of short-term scheduling problems. *Computers & Chemical Engineering*, *27*(8), 1261-1276.

Wunderlich, S., & Back, W. (2009). Overview of manufacturing beer: ingredients, processes, and quality criteria. In *Beer in Health and Disease Prevention* (pp. 3-17). Elsevier Academic Press Amsterdam.

Xiao, Jie and Zhou, Ze-kui (2004) Application of Ant Colony Algorithm to the Optimization of Beer Fermentation Control. Department of Control Science & Engineering, Zhejiang University, Hangzhou, China; 33(4), 2004.

Yan, N., Lv, B.H., Zhang, L., Luo, L.P., Jin, H.G and Liu, Z.G (2009) Research on simulation for a scenario of a beer production system in a small brewery. Dalian university of technology, Dalian university of light industry, China.

Yu, H., Fang, H., Yao, P., & Yuan, Y. (2000). A combined genetic algorithm/simulated annealing algorithm for large scale system energy integration. *Computers & Chemical Engineering*, *24*(8), 2023-2035.

Yun, Y., & Gen, M. (2003). Performance analysis of adaptive genetic algorithms with fuzzy logic and heuristics. *Fuzzy Optimization and Decision Making*, *2*(2), 161-175.

Zheng, S., Ge, M., Li, C and Xue, A (2011) Mathematical programming model for beer production scheduling and its optimization. Control Theory & Applications, Hangzhou Dianzi University, 28(4), 2011

Zheng S. (2008) Study on optimisation and automation of beer production process. Zhejiang University, China.

Zhou, M. and Sun, S.D. (1999) Genetic Algorithm Theory and Applications. Beijing: National Defence Industry Press.