MODELING AND ANALYSIS OF THE BATCH PRODUCTION SCHEDULING PROBLEM FOR PERISHABLE PRODUCTS WITH SETUP TIMES

A Dissertation
Presented to
The Academic Faculty

By
Aphiwat Charnprasitphon

In Partial Fulfillment
Of the Requirements for the Degree of
Doctor of Philosophy in the
School of Industrial and System Engineering

Georgia Institute of Technology

May 2007

MODELING AND ANALYSIS OF THE BATCH PRODUCTION SCHEDULING PROBLEM FOR PERISHABLE PRODUCTS WITH SETUP TIMES

Approved by:

Dr. Faiz Al-Khayyal, Co-advisor
School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. Paul M. Griffin, Co-advisor
School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. Earl Barnes
School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. Jye-Chyi Lu
School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. John Elton
School of Mathematics
Georgia Institute of Technology

Date Approved: January 5, 2007

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my parents for their love and support. My great gratitude goes to the Anandamahidol Foundation and the School of Industrial and Systems Engineering at Georgia Institute of Technology for sponsoring my study. Without them, none of this would have been possible.

Second, I would like to thank my advisors, Dr. Faiz Al-Khayyal and Dr. Paul M. Griffin for their guidance, encouragement, patience and many helpful suggestions during my time at Georgia Tech.

I am also grateful to Dr. Earl Barnes, Dr. Jye-Chyi Lu and Dr. John Elton for serving on my thesis committee and for their constructive comments.

I would like to thank Dr. Chen Zhou and Dr. Ronald Billings for their suggestions on manufacturing systems and their kind collaboration when I was a TA.

I owe a great debt to my girlfriend, Vatcharapa (Nuch), for her love, encouragement and supports.

I would like to thank all my friends here for their support and friendship. Thank you Vi, Nu, Kaew, Mhu, Kaulad, Anong, To, Lek, Auey, Oran, Josh, K, L, Jan, Chompoo, Tan, Kong, Koe, An, Gu, Yanwu, Kim, Hyun-suk, Erwin, Michael and Helen.

Last but not least, I would like to thank other people who have supported me, but I have not mentioned their names here. Please accept my sincere apology and thank you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The focus of this dissertation is problem of batch production scheduling for perishable products with setup times, with the main applications in answering production planning problems faced by manufacturers of perishable products, such as beers, vaccines and yoghurts. The benefits of effective production plans can help companies reduce their total costs substantially to gain competitive advantages without reduction of service level in a globalize economy.

We develop concepts and methodologies that are applied to two fundamental problems: (i) the batch production scheduling problem for perishable products with sequence-independent setup times (BPP-SI) and (ii) the batch production scheduling problem for perishable products with sequence-dependent setup times (BPP-SD).

The problem is that given a set of forecast demand for perishable products to be produced by a set of parallel machines in single stage batch production, with each product having fixed shelf-life times and each machine requiring setup times before producing a batch of product, find the master production schedule which minimizes total cost over a specified time horizon. We present the new models for both problems by formulating them as a Mixed Integer Program (MIP) in discrete time. Computational studies on BPP-SI and BPP-SD for industrial problems are presented. In order to efficiently solve the large BPP-SI problems in practice, we develop five efficient heuristics. The extensive computational results show that the developed heuristics can obtain good solutions for very large problem sizes and require a very short amount of computational time.

CHAPTER I

INTRODUCTION

This research addresses the complex set of production decisions in a single stage of batch processes for a manufacturer of fixed shelf-life products. Due to the higher flexibility in producing a wide variety of products, batch production processes have gained considerable popularity over the last two decades. Some examples of batch process used in production of perishable products include the fermentation process for beers, the mixing process for medicines, and the incubation process for vaccines. Production planning for batch production is very difficult because of large varieties of constraints, such as non-preemptive processes, intermediate storage policy, lot sizing, processing sequences, shared resources, many pieces of processing equipment with varying operational characteristics, etc.

This research is motivated to help a manufacturer of fixed shelf-life products determine an efficient Master Production Schedule (MPS) for a single batch operation stage, while incorporating several issues, such as setup times (sequence independent setup times or sequence dependent setup times), lot size (discrete or continuous), capacity of machines, fixed processing time, shelf-life of products, deterministic demand for products, and number of machines available in order to minimize total cost, comprising costs of inventory, spoilage, production, setup and penalty for unmet demand.

The resulting MPS indicates the amount of products to be produced in each period, sequencing of production of products on each of the machines, as well as, timing of setup on the machines. This MPS plan is useful for planners to efficiently allocate resources among products.

Our focus is on one batch processing unit for production of perishable products, since the batch process step typically accounts for most of the residence time of products in the system and is the bottleneck step, such as a fermentation tank used to brew beer (Virkajarvi, 2000) or an incubator for flu vaccine production (A report from the American Academy of Microbiology, 2005). Main features of this batch operation are

- Batch operation is non-preemptive.
- Each machine can process at most one product at a time**.**
- Each batch of product requires a setup whenever a new batch is released on machine.

Figure 1.1 represents three major components in the BPP problem arising in industries.

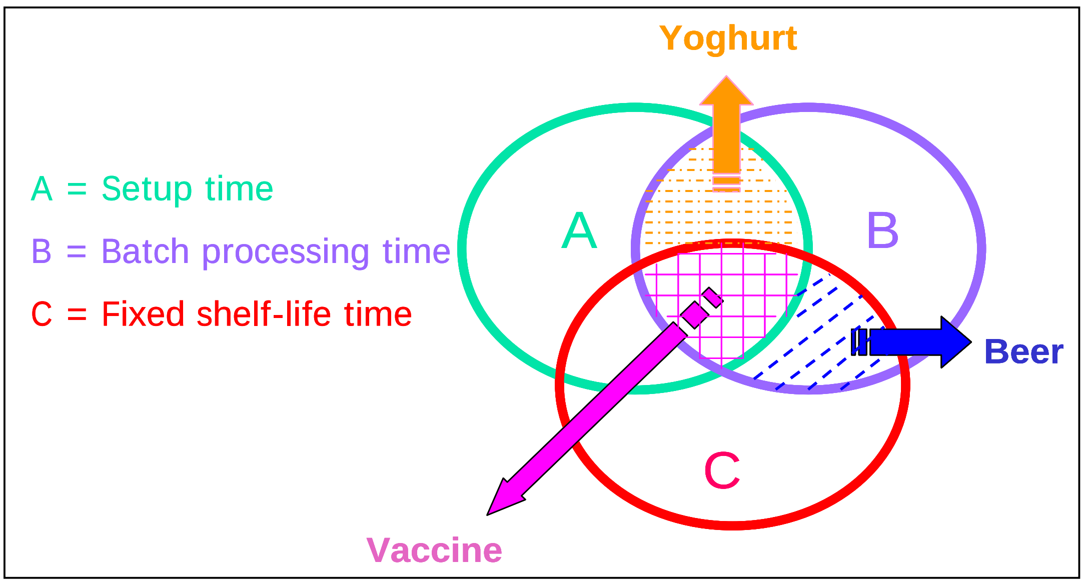


Figure 1.1 Venn Diagram for Three Major Components in the BPP Problem

This dissertation presents a new integrative approach for dealing with batch production scheduling problems for fixed shelf-life products with setup times on a single processing unit of parallel machines. This dissertation differs from previous work done under lot-sizing and scheduling problems and inventory management for perishable products in that our models incorporate several practical issues, such as the limited shelf-life of products, the change in number of available machines and the penalty for unmet demand into the models, which also include the issues of lot-sizing and setup-times. We formulate the discrete-time MIP models for the batch production scheduling problems for fixed shelf-life products for the case of sequence-independent setup times (BPP-SI), and the case of sequence-dependent setup times (BPP-SD). Furthermore, we develop five efficient heuristics for solving the batch production scheduling problems with sequence-independent setup times (BPP-SI). The extensive computational results show that the developed heuristics can obtain good solutions for very large problem sizes and require a very short amount of computational time. Moreover, we apply both optimization and heuristic approaches to solve problems in industry. We also examine factors of interest on the system performance and analyze the performance of heuristics.

This chapter provides the background for general batch production scheduling problems for perishable products (BPP) and the overview of two types of manufacturing industries in which the BPP problems usually take place. The relevant academic literature is reviewed in Chapter II. In Chapter III, a formal definition of BPP-SI and the mathematical model are provided. Chapter IV covers a formal definition of BPP-SD and the mathematical model. Chapter V presents the numerical study for BPP-SI and BPP-SD for three different configuration settings. In Chapter VI, the solution strategies for BPP-SI problems are presented. Chapter VII presents a numerical study for large BPP-SI problems in three industries including beer, vaccine and yoghurt. Chapter VIII investigates the performance of heuristics for BPP-SI problems by a computational study. The summary and future extensions are overviewed in Chapter IX. Next we briefly explain two industries in which the BPP problems usually arise.

1.1 Overview of Brewing Industry

According to the report of The Brewers Association in 2006, the overall U.S. brewing industry dollar volume was $83 billion in 2005, and total U.S. beer sale was 205.65 million barrels (1 barrel = 31 U.S. gallons). There are 1,452 U.S breweries, which consist of craft breweries (1415), large breweries (21), and regional breweries (16). Craft breweries include brewpubs (9.2%), microbreweries (10.9%), regional craft breweries (66%), and contract breweries (13.9%) with a growth rate of 9% in 2005. According to the industry data of Beer Institute, per U.S. capita consumption for 2003 is 30.6 gallons of beer per person.

The basic ingredients of beer are water, malted barley (the main source of starch and enzymes), yeast, and hops.

The process of brewing beer includes

- Mashing: Malted grains are crushed and soaked in warm water in order to create a malt extract. The mash is held at constant temperature for converting starches into fermentable sugars.

- Filtering: Water is filtered through the mash to dissolve the sugars. The darker, sugar-heavy liquid is called the "wort".
- Boiling: The wort is boiled in order to remove excess water and kill any microorganisms. Hops are added at this stage for favor enhancement.
- Fermentation: The yeast is added or pitched and the beer is left to ferment in fermentation tank. Yeast is used to convert fermentable carbohydrates into alcohol, carbon dioxide, and numerous byproducts. Fermentation depends on the composition of wort, yeast, and fermentation condition. After primary fermentation, the beer may be allowed a second fermentation for further settling of yeast.
- Down-stream processing: filtration, stabilization, and packaging. Figure 1.2 illustrates the process of brewing lager beer by Linko et al. (1998).

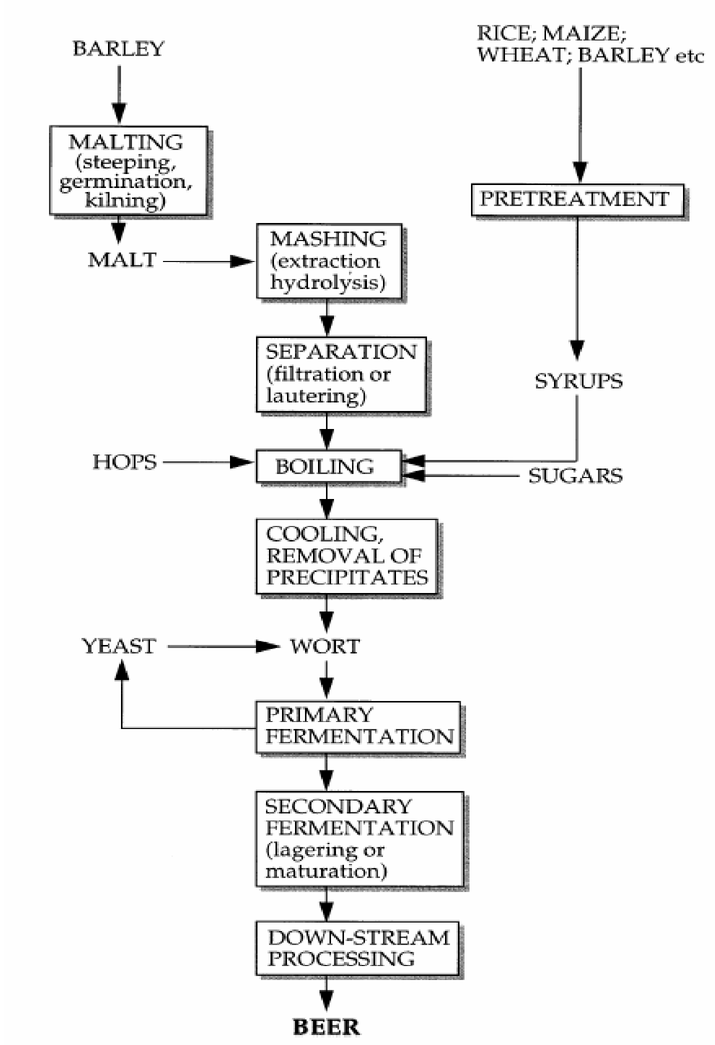


Figure 1.2: The Process of Brewing Lager Beer

Beer is perishable, since it deteriorates due to the action of bacteria, light, and air. Beer is not legally required to carry a "sell by" date. However, some companies, such as Boston Beer Company, carry a freshness date. Anheuser-Busch uses "born on" dates. Freshness period or shelf-life of beer varies with the type of beers and the storage conditions. According to the Beverage Testing Institute, the freshness period for a lager is 4 months, stronger craft-brewed ales is 5 months. High-gravity, high-strength beer varies from 6 to 12 months, if beers are properly handled and stored. Bradt, a board of directors for the Brewers

Association, said that “in general, most brewers are comfortable with a shelf life of 3-4 months for standard-strength bottles" on the news in July, 2005 by the Lawrence company.

According to Virkajarvi (2000), the fermentation is the most time consuming step in the production of beer and is a batch process. Therefore, the effective use of fermentation tanks is an important element to brewing economy. In his paper, the fermentation time for lager beers typically lasts from 2 to 4 weeks. The capacity of a fermentation tank ranges from 600 to 50,000 gallons. According to a source of Thai Asia Pacific Brewery Company, the setup times for cleaning a fermentation tank is approximately 2-3 hours, which do not significantly vary with the type of beer to be fermented, and is very relatively small compared to the fermentation time. Consequently, this setup time can be considered as sequence-independent.

In summary, the fermentation process for beer provides a good environment to demonstrate the effectiveness of our proposed model, namely, the batch production scheduling problem for a perishable product with sequence-independent setup times (BPP-SI). The result of the model is the optimal production schedule for fermenters over the planning horizon. The mathematical model is presented in Chapter III and the numerical result for simulated problems is in Chapter V.

1.2 Overview of Vaccine Industry

The aim of a vaccine is to stimulate our body's immune system to prevent illness by destroying the foreign invader or making it harmless. A vaccine contains a dead or weakened form of the organism (virus, bacterium or other organisms) that causes a particular disease. When given to a person, the vaccine stimulates his immune system to produce antibodies against the organisms. If he/she is exposed to the disease, in which he/she has been vaccinated, then the antibodies will destroy the invading germ.

The first vaccine against smallpox was used in 1798 by Edward Jenner. According to World Vaccine Congress of 2006, the global vaccine market was around $10 billion in 2005. Walsh (2003) estimated that around 500,000 adults die annually in U.S.A. from the conditions, which could have been prevented by vaccination. Vaccines have been used to prevent several diseases, such as smallpox, rubella, polio, measles, mumps, chickenpox, typhoid, etc. The World Health Organization (WHO) and the Centers for Disease Control and Prevention (CDC) recommend that all travelers be up-to-date with the routine vaccines, such as Diphtheria/tetanus/pertussis (DTP), Hepatitis B (HBV), Poliomyelitis, Haemophilus influenza type b (Hib), Measles/mumps/rubella (MMR).

CDC classifies four types of traditional vaccines:

- *Live attenuated vaccines* are live micro-organisms that have been cultivated under conditions, which disable their virulent properties. Examples are vaccines against yellow fever, measles, and mumps.

- *Killed vaccines* contain killed virulent micro-organisms by chemicals or heat.

Examples are vaccines against flu, cholera, and hepatitis A

- *Toxoid vaccines* are inactivated toxic compounds from micro-organisms

Examples of these vaccines are tetanus and diphtheria.

- *Component vaccines* contain parts of the whole bacteria or viruses. Example is

vaccine against Hepatitis B (HBV).

Innovative vaccines are Conjugate, Recombinant Vector, and DNA vaccination (See detail in Crowcroft (1999), Henahan (1997), Walsh (2003)).

The manufacture of vaccines is one of the most highly regulated and rigorously controlled manufacturing processes in order to produce safe and effective vaccines according to Good Manufacturing Practice (GMP). The following factors contributing to the safe manufacture of vaccines include: the design and layout of manufacturing facility, raw materials (such as vaccine strains, chemicals) and equipment used, manufacturing process, the training and commitment of employees relating to manufacturing operations, etc.

Because the manufacturing process for each of vaccines is different depending on the strain of vaccine, growth media, etc, we will not discuss the detail of manufacture of vaccine in this thesis. The interested reader is referred to Walsh (2003), Plotkin and Mortimer (1994), The World Health Organization (WHO).

The incubation step is one of the most time consuming processes in manufacture of vaccines. Cell culture is the culturing of cells under controlled conditions (growth media, pH, temperature). For example, the cell cultures should be incubated for 2 weeks for influenza vaccine, and 4 weeks for smallpox vaccine, according to the recommendation for the production and quality control of vaccine by WHO. Furthermore, it is the necessary step to activate the freeze-dried vaccines. For instance, the smallpox vaccine in its freeze-dried form has to be incubated at 37 degrees C for one month so as to maintain its full potency by International standard requirement.

An Incubator is an apparatus, which is used to grow and maintain cell cultures. The incubator keeps cultures at an optimal temperature and humidity. CO_2 incubators regulate the oxygen and carbon dioxide (CO_2) content. The capacity of a CO_2 incubator ranges from 14 to 170 liters (Information on incubators can be found on the websites of NuAire Inc., Voigt Global Distribution Inc., and Wolf Laboratories Limited). Due to the strict rules on the safe manufacture of vaccines, the manufacturers have to follow the cleaning, decontamination, and sanitation (CDS) procedures in order to prevent cells from contaminants. The procedure of cleaning CO_2 incubators can be found in Moody (2002).   The setup time for cleaning an incubator is four to eight hours, depending on the sequence of vaccines to be produced.

A vaccine has a limited shelf life, which is dependent on the type of vaccines, storage condition (i.e. temperature, sunlight).   For example, the measles vaccine can maintain its potency for 4 weeks at 37 degrees C, and 8 months at room temperature. However, the reconstituted vaccine remains potent for 2 days at 20-25 degrees C, and 7 hours at 37 degrees C (See Plotkin and Mortimer (1994)).

In short, the model of the batch production scheduling problem for perishable products with sequence-dependent setup times (BPP-SD) can be applied to the problem of production scheduling of incubation process for manufacture of vaccine. The mathematical model is presented in Chapter IV and the numerical results for the BPP-SD problem is in Chapter V.

CHAPTER II

REVIEW OF RELEVANT LITERATURE

This chapter reviews the literature and background, which are closely related to our research. This includes lot sizing and scheduling, batch production process, scheduling with batching and inventory management of perishable items.

2.1 Literature Review of Lot Sizing and Scheduling Problem

Lot-sizing and scheduling are dependent decisions. The lot-sizing step needs information about setup times, which is determined by item sequence and machine assignment from the result of scheduling step. Meanwhile, the scheduling step requires the production quantity as input in order to determine item sequence and machine assignment. An integrative solution approach is needed to simultaneously solve the lot-sizing and scheduling problem. Therefore, the optimal production plan is obtained.

Eppen and Martin (1987) classify lot sizing problems with finite planning periods into two models - small bucket and big bucket models. Small bucket models have relatively short periods. In the small bucket model, at most one type of item can be produced and one setup can incur on the machine during each time period. Examples of this type of model are the Discrete Lot Sizing Problem (DLSP), and Continuous Lot Sizing Problem (CSLP). In DLSP, production must be at capacity if a machine is used to produce an item. In CLSP, the amount of production can vary, but is limited by the capacity of a machine. The solution of the small bucket problem contains production sequence of items on the machine. On the other hand, big the bucket model has fewer, but longer period without restriction on the number of items or setups per period and machine. In the large bucket model, many different items can be produced on the same machine in one time period. Examples of large bucket models are the Capacitated Lot Sizing Problem (CLSP), and the General Lot sizing and Scheduling Problem (GLSP).

The big bucket model does not take into account the item sequence in a period and its solution does not contain the production schedule. With the same length of planning period, the number of periods in the small bucket model is much larger than that of the large bucket model, so the small bucket model takes more computational time. We next discuss the research on the major types of lot sizing and scheduling problems.

2.1.1 The Single-Level, Single Item, Lot-Sizing Problem

Research on lot-sizing models began with the classic Economic Order Quantity model (EOQ model). Ford W. Harris (1915) develop the simple EOQ model, in which demand is assumed to be stationary, no stock-outs are permitted, only holding and fixed order costs are present, and a single-level production has no restriction on capacity. The EOQ model is a continuous time model with an infinite planning horizon. The EOQ model can be easily extended to the case in which items are produced internally with a finite production rate. The optimal batch size can be obtained by the EOQ formula with a modified holding cost. Hadley and Whitin (1963) develop the EOQ model for resource-constrained multiple items. Examples of limited resources include budget and space. They showed that when the ratio of the item value or space consumed by the item over the holding cost is the same for all items, the solution can be obtained easily. When the ratio is different, they propose using Lagrange multiplier to solve this problem.

The Wagner-Whitin problem is an extension of EOQ model where demands are dynamic, planning horizon is finite, and capacity limits are not considered. Wagner and Whitin (1958) develop the dynamic programming algorithm in order to optimally solve the single-item, uncapacitated lot sizing problem. The authors also prove that there exists an optimal solution that satisfies the Wagner-Whitin property. Under an optimal lot-sizing policy, either the inventory carried from a pervious period to period t+1 will be zero or the production quantity in period t+1 will be zero. Federgruen and Tzur (1991), Wagelmans et al. (1992), Aggarwal and Park (1993) develop more efficient algorithms for this problem.

De Matteis (1971) and Silver and Meal (1973) develop heuristic approaches to solve the uncapacitated, single item, single-level lot-sizing problem. However, when the finite capacity of facility is incorporated into the model, this capacity constraint considerably complicates the analysis.

2.1.2 Economic Lot Scheduling Problem (ELSP)

The objective of the Economic Lot Scheduling Problem (ELSP) is to find the optimal schedule that allows to cyclic production pattern for each item produced by a single machine so that the total of inventory and setup costs is minimized and no stock-outs occur during the production cycle. The ELSP is a single-level, multi-item problem, where the capacitated, single facility is commonly used to produce several items. Like the EOQ model, ELSP is a continuous time model with an infinite planning horizon. Comprehensive research on the ELSP is found in Maxwell (1964), Elmaghraby (1978), Silver et al. (1998), and Nahmias (2005).

The underlying assumptions of the traditional ELSP model are as follow:

- Only one item can be produced at a time.
- Demand rates are deterministic, and stationary.
- Production rates are constant, and deterministic.
- Production capacity is capacitated, and sufficient to meet total demand.
- There is a setup cost and a setup time associated with producing each item.
- No backordering for any demand is allowed.
- Inventory of each item is charged at a linear time-weighted holding cost rate.

Next, we define the following notation used in the ELSP model.

i = index for item ($i = 1,\ldots,N$)

D_i = Demand rate for item i (in units of item per period)

P_i = Production rate for item I (in units of item per period)

h_i = Holding cost per unit per time for item i (in dollars/unit/period)

K_i = Setup cost of the machine to produce item i (in dollars/setup)

s_i = Set time for item i (in periods)

Q_i = lot sizes for item i (in units of item)

T = Cycle time (in periods)

T^* = Optimal cycle time when setup time is assumed to be zero (in periods)

Ensuring that the machine has sufficient capacity to satisfy the demand for all items leads to the constraints $\sum_{i=1}^{N}\left(\frac{D_i}{P_i}\right) \leq 1$. To solve the ELSP, we use a rotation cycle policy. This means that exactly one setup for each item in each cycle, and items are produced in the same sequence in each cycle. The resulting optimal cycle T^* for ELSP without setup time is obtained. $T^* = \sqrt{\frac{2\sum_{i=1}^{N} K_i}{\sum_{i=1}^{N} D_i h_i'}}$, where $h_i' = h_i\left(1-\frac{D_i}{P_i}\right)$

When the setup time for an item is incorporated, one has to ensure that the total time required for setups and production during each cycle does not exceed the cycle time T.

This condition can be expressed as $\sum_{i=1}^{N}\left(s_i + \frac{D_i T}{P_i}\right) \leq T$.

After rearranging terms, $T \geq \frac{\sum_{i=1}^{N} s_i}{\left(1-\sum_{i=1}^{N}\frac{D_i}{P_i}\right)} = T_{min}$

The cycle time T for ELSP with nonzero setup time is the larger of T_{min} and T^* $T \geq \min\{T_{min}, T^*\}$ Lot sizes for item i are given by $Q_i = D_i T$

Hsu (1983) shows that ELSP is NP-hard. There have been a number of heuristic procedures developed by Dobson (1987), Zipkin (1991) and Gallego (1994) for solving the ELSP. In the next section, we discuss the Capacitated Lot Sizing Problem (CLSP), which is a typical example of the large bucket model.

2.1.3 Capacitated Lot Sizing Problem (CLSP)

The Capacitated Lot Sizing Problem (CLSP) consists of determining the lot sizes of multi-items over a finite planning horizon in order to minimize setup and inventory holding costs. The CLSP is a single-level, multi-item problem where limited capacity is shared by items produced in each period, no backorders are permitted, and demands for items are assumed to be dynamic, and deterministic. In each period in which an item is produced, a setup cost is incurred. Unlike ELSP, CLSP assumes that several items can be produced per period, so CLSP is a large bucket problem. The planning horizon typically is less than six months. A period usually represents a time period of approximately one week. We define the "setup carry-over" as the continuation of production of an item from one period to the next without an additional setup. The fundamental assumption of the CLSP is that setup costs occur for each lot in a period. In CLSP model, setup carry-over is not allowed, i.e., setup is incurred even if the same item was produced last in period t and produced first in period t+1. Consequently, a result from CLSP model could cause a substantial setup cost. Another disadvantage of CLSP is that the optimal solution, based on aggregate data, could be more expensive than the optimal solution obtained by using disaggregated data, if we apply CLSP to the short term planning problems with small periods. It should be noted that the CLSP does not include the sequence decision in the solution.

The following notation is used to model CLSP, DLSP and CSLP

Indices:

i = index for items ($i = 1,\ldots,N$)

t = index for time periods ($t = 1,\ldots,T$)

Data:

a_i = The number of setup periods required before production of item i (in periods)

$d_{i,t}$ = Demand for item i in period t (in units of item)

h_i = Holding cost per unit per time for item i (in dollars/unit/period)

K_i = Setup cost of the machine to produce item i (in dollars/setup)

k_i = Setup cost per setup period for item I (in dollars/period)

$I_{i,0}$ = Initial inventory for item i (in units of item)

C_t = Available capacity of the machine in period t

r_i = Capacity needed to produce one unit of item i (in unit of capacity/unit of item)

Variables:

q_{it} = Production quantity of item i in period t (in units of item)

I_{it} = Amount of inventory at the end of period t of item I (in units of item)

y_{it} = Binary variable indicating whether item I is produced in period t ($y_{it} = 1$) or not ($y_{it} = 0$)

v_{it} = Binary variable indicating whether the machine is set up for item i in period t (v_{it} = 1) or not (v_{it} = 0)

Mathematically, the CLSP can be formulated as a mixed integer program model:

Minimize (Objective)	***Minimize Total Cost***		
$Z_{CLSP} = min \sum_i \sum_t \left(K_i\, y_{it} + h_i\, I_{it} \right)$	2.1 -	Sum of setup and holding costs	
Subject to:			
$I_{i,t-1} + q_{i,t} - d_{i,t} = I_{i,t}$	$\forall i,t$	2.2 -	Inventory balance for item
$\sum_i r_i\, q_{it} \leq C_t$	$\forall t$	2.3 -	Amount of production is limited by capacity
$r_i q_{it} \leq C_t y_{it}$	$\forall i,t$	2.4 -	Logical constraint on setup
$q_{it} \geq 0,\ I_{it} \geq 0,\ y_{it} \in \{0,1\}$	$\forall i,t$	2.5 -	Variable constraints

The objective function (2.1) is to minimize total inventory and setup costs. Equations (2.2) express the inventory flow balance in each period. Constraints (2.3) ensure that total production in each period does not exceed the capacity. Constraints (2.4) ensure that a setup is performed in each period in which an item is produced. Constraints (2.5) define non-negative variables, and binary variables for setup respectively.

Florian, Lenstra and Rinnooy Kan (1980) and Bitran and Yanasse (1982) show that solving the CLSP optimally is NP-hard. Many attempts have been made to solve a mixed integer program of the CLSP by using exact solution approach, such as the branch and bound technique, cut-generation technique, and variable redefinition technique. Barany et al. (1984) use cut-generation technique by adding strong valid inequalities, which are facets for the single-item uncapacitated problem. The reformulated problem results in a good approximation of the convex hull of feasible solutions to the CLSP. Then the resulting reformulated problem is solved using a branch-and-bound algorithm. Eppen and Martin (1987) use variable redefinition technique for converting the traditional CLSP formulation into a graph-based representation. The resulting reformulation has more variables and constraints, but provides tighter linear relaxation than the traditional formulation. The LP-relaxation problem is first solved and then a branch and bound algorithm is used to obtain the optimal solution. They solve the multi-item capacitated lot-sizing problem instance up to 200 items and 10 periods. Belvaux and Wolsey (2000) develop strong formulations and a specialized branch-and-cut system for practical lot-sizing problems.

Due to the complexity of the problem, it is unlikely that one can develop any efficient exact method to solve CLSP. Therefore, several efficient heuristics are proposed for the CLSP. Dixon et al. (1981), Dogramact et al. (1981) and Gunther (1987) employ a period-by-period heuristic approach, where lot sizes of items are determined by a cost saving criterion. Thizy and van Wassenhove (1985) develop a Lagrangean based heuristic for CLSP. This method includes a primal partitioning scheme with a network flow subproblem. Cattrysse et al. (1990) propose a set partitioning and column generation heuristic for multi-item, single-level capacitated dynamic lot-sizing problems.

Maes et al. (1991) show that finding a feasible solution of the multi-item capacitated lot-sizing problem with setup time (MCL) is NP-complete. Many authors have developed heuristic methods to solve MCL. Trigiero, Thomas, and McClain (1989) develop a Lagrangean heuristic for MCL. This algorithm iterates between primal and dual procedures. A smoothing heuristic is implemented after each primal step. The dual procedure employs subgradient optimization to compute dual prices for capacity in each period. The primal procedure uses dynamic programming to solve the set of uncapacitated, single-item problem, which results from the Lagrangean relaxation. The smoothing heuristic is used to modify the primal solution, seeking to eliminate overtime. The authors points out that when the capacity constraint is tight, a feasible solution is not always obtained. Diaby et al. (1992) propose a Lagrangean relaxation-based heuristic to solve very large scale MCL with limited overtime. The authors relax the capacity constraints and solve the resulting transportation formulation. Miller et al. (2000) solve the multi-item capacitated lot sizing problem with setup times using a branch-and-cut algorithm.

2.1.4 Small Bucket Models

In this section, we discuss the small bucket models, such as DLSP, CLSP, and PLSP. In small bucket models, the planner decides what is to be done in each time period. That is, he has to determine which item the machine is producing in each time period (production variable), and whether or not production has changed to a new item in this time period (setup variable). We discuss the small bucket models in detail, since our problem of interest has some similar features with this model.

2.1.4.1 Discrete Lot sizing and Scheduling Problem (DLSP)

The standard discrete lot sizing and scheduling problem (DLSP) is the problem of determining lot sizing and sequencing for a number of different items on a single machine over a discrete and finite planning horizon. The objective is to find a minimal cost production schedule such that dynamic demand is fulfilled without backlogging. In DLSP, we divide the finite macro-periods into several micro-periods. In each time period, at most one type of item can be produced. The setup on machine can occur only once in each time period. The main assumption of DLSP is "all-or-nothing production". That is, only one item can be produced per period, and if so, the full capacity is used.

To describe the setup cost structure in DLSP, Cattrysse et al. (1993) define "a batch of item i" as an uninterrupted sequence of periods in which production takes place for item i. After a machine finishes set-up periods for an item, it can be used to produce the item for an uninterrupted sequence of periods without another set-up. If the machine is idle, a setup is needed before producing an item. As a result, the DLSP does not preserve the setup state over idle periods. We next compare the main differences between CLSP and DLSP.

- Unlike CLSP, the DLSP is a small bucket problem because at most one item can be produced per period. For CLSP, the setup cost is incurred in every period in which production takes place. The periods in DLSP are relatively shorter than those in CLSP, such as hours, or shifts. Due to short period in DLSP, the setup cost is incurred only when the production of a new lot starts. The DLSP has the same objective function as CLSP, but some constraints need to be modified in order to cover the issue of a certain setup period before producing an item.

DLSP has many important practical applications. For example, Van Wassenhove and Vanderhenst (1983) describe the application of DLSP in a decision support system for production planning in a large chemical plant. Jans and Degraves (2004) consider an extension of the standard DLSP to an industrial production planning problem for a tire manufacturer.

The standard DLSP can be formulated as a mixed integer program, which was proposed by Fleischmann (1990).

Minimize (Objective)		***Minimize Total Cost***	
$Z_{DLSP} = min \sum_i \sum_t K_i v_{it} + h_i I_{it}$		2.6 -	Sum of setup and holding costs

Subject to:

$r_i q_{it} = C_t y_{it}$	$\forall i,t$	2.7 -	Amount of items produced in each period
$I_{i,t-1} + q_{i,t} - d_{i,t} = I_{i,t}$	$\forall i,t$	2.8 -	Inventory balance for item
$\sum_i y_{it} \leq 1$	$\forall t$	2.9 -	Machine can produce at most one type of item in each time period
$v_{i,t} \geq y_{it} - y_{i,t-1}$	$\forall i,t$	2.10-	Changeover requires a new setup

$q_{it} \geq 0,\ \mathrm{I}_{it} \geq 0,$ $\forall i,t$ 2.11- Nonnegative variables

$\mathrm{v}_{it}, y_{it} \in \{0,1\}$ $\forall i,t$ 2.12- Binary variables for setup and production

The objective function (2.6) is to minimize total inventory and setup costs. Constraints (2.7) ensure that the quantity produced in each period is either zero or full production capacity ("all or nothing production"). Equations (2.8) express the inventory flow balance in each period. A set of machine capacity constraints (2.9) guarantees that in each period, the machine produces at most one type of item.

A set of constraints (2.10) ensure the correct sequence of setup and production periods for items. When producing different types of items, a new setup is required.

Non-negativity constraints are defined in inequalities (2.11). Conditions (2.12) define binary variables for the setup status and production status of machine in each period respectively. Note that the following valid inequalities $v_{i,t} \leq y_{it}\ \forall i,t$ (2.13) and $v_{i,t} \leq 1 - y_{i,t-1}\ \forall i,t$ (2.14) may be added to improve the computational time. Constraints (2.13) imply that machine will produce an item i in period t if a setup for item i incurs at the beginning of period t. Constraints (2.14) imply that machine will not be setup for an item i in period t if it produces such item in previous period t-1, because DLSP allows setup carryover for the same item in consecutive periods. It should be noted that in the standard DLSP, the setup cost is included into the model, but setup time is assumed to be zero.

To account for DLSP model, in which the number of setup periods required before producing an item i (a_i) is not zero, one has to modify the standard DLSP model by replacing the objective function (2.6) with equation (2.6a) and replacing constraints (2.9-2.10) with constraints (2.9a, 2.10a, 2.10b, 2.10c). This model was proposed by Bruggemann and Jahnke (2000).

Minimize (Objective)		***Minimize Total Cost***	
$Z_{DLSP-st} = min \sum_i \sum_t k_i v_{it} + h_i I_{it}$		2.6	Sum of setup and holding costs

Subject to:

$r_i q_{it} = C_t y_{it}$	$\forall i,t$	2.7 -	Amount of items produced in each period
$I_{i,t-1} + q_{i,t} - d_{i,t} = I_{i,t}$	$\forall i,t$	2.8 -	Inventory balance for item
$\sum_i (y_{it} + v_{it}) \leq 1$	$\forall t$	2.9a -	Prevent simultaneous setup and production on the machine
$v_{i,t-a_i+\tau} \geq y_{it} - y_{i,t-1}$	$\forall i,$ $t = a_i + 1,...,T$ $\tau = 0,...,a_i - 1$	2.10a-	Setup for item with nonzero setup periods
$v_{i,t-a_i+\tau} \geq y_{it} - y_{i,t-1}$	$\forall i,t,\ a_i = 0$	2.10b-	Setup for item with zero setup period.
$y_{it} = 0$	$\forall i,\ t = 1,...,a_i$	2.10c-	Logical constraints on production
$q_{it} \geq 0,\ I_{it} \geq 0,$	$\forall i,t$	2.11-	Nonnegative variables
$v_{it}, y_{it} \in \{0,1\}$	$\forall i,t$	2.12-	Binary variables for setup & production

We then describe the new constraints in the detail. Constraints (2.9a) are used to prevent simultaneous action of setup and production on the same machine. Constraints (2.10a and 2.10b) relate the correct sequence of setup and production periods for the machine. Constraints (2.10c) enforce that there is no production of item i during periods [1,…, a_i] with no preceding setup.

We next discuss recent literature on the Discrete Lot-Sizing and Scheduling Problem (DLSP). A comprehensive overview of DLSP literature can be found in Hasse (1994), Jordon (1996), Drexl and Kimms (1997), and Quadt (2004).

Fleischmann (1990) develops a generic model for the DLSP and presents a branch and bound approach based on Lagrange relaxation of the capacity constraints. They solve the DLSP whose sizes are up to 12 items and 122 periods or 3 items and 250 periods.

Magnanti and Vachni (1990) describe a solution approach based on polyhedral methods for DLSP on a single machine with sequence independent set-up costs and zero setup times. They solve problems with 2 items and 20 periods, and 5 items and 15 periods by using cutting planes. They found that the inequalities effectively reduce the integrality gap between the value of an integer program formulation and its linear program relaxation by a factor of 94 to 100%

Solomon et al. (1991) introduce a six-field classification scheme for different DLSP variants and analyze the computational complexity of single machine, and parallel machine variants of DLSP. They show that solving the DLSP optimally is NP-hard. If either setup times or parallel machines are considered, even the feasibility problem is NP-complete. Bruggemann and Jahnke (1997) and Webster (1999) correct some proofs of Solomon's computational complexity of DLSP.

Cattrysse et al. (1993) propose a heuristic for the DLSP on a single machine with setup times. The DLSP is formulated as a Set Partitioning Problem (SPP). A column generation scheme is applied and the dual prices are computed with a dual ascent method and subgradient optimization. Further, the heuristic generates lower and upper bounds.

Computational results on the medium sized problem of 6 items and 60 periods show that the heuristic is effective, both in terms of quality of the solutions and computational time.

Van Hoesel and Kolen (1994) propose a mixed integer program formulation for DLSP and present an optimal solution procedure for the DLSP based on variable splitting.

Feischmann (1994) considers the DLSP with sequence-dependent setup costs. His heuristic is based on the transformation of the problem into a Traveling Salesman Problem with Time Windows. Problems of moderate size are solved using simple local improvement based heuristics. Lower bounds to evaluate the quality of the solutions from the heuristics are generated by Lagrangean relaxation procedures. His computational study shows that the gap between lower and upper bounds could be as large as 30% in some cases.

Salomon et al. (1997) consider DLSP on a single machine with sequence-dependent setup costs and setup times (DLSPSD), which is known to be NP-Hard. They reformulate the problem as a Travelling Salesman Problem with time windows (TSPTW). They optimally solve it using a dynamic programming algorithm, which is proposed by Dumas et al. (1995). They solve the lot sizing problems up to 10 items and 60 periods with sequence dependent setup costs and times to proven optimality.

Bruggemann and Jahnke (2000) show the proof for the NP-hardness in the strong sense for DLSP and consider an extension of DLSP with batch availability, where items only become available after the whole batch is completed. They construct a two-phase simulated annealing (SA) heuristic to solve the DLSP with batch availability. This heuristic searches for a feasible solution in phase 1, and optimizes cost in phase 2. Production schedules are generated by dividing, combining and shifting batches.

Belvaux and Wolsey (2000) discuss a specialized branch-and-cut system for a wide variety of lot sizing problems. Their software can be applied to both big bucket and small bucket models with both setup times and setup costs.

2.1.4.2 Continuous Setup Lot-sizing Problem (CSLP)

In CSLP, the lot sizes of items are allowed to be continuous under full capacity. In addition, setup carryover over idle periods is permitted. However, only one item can be produced or set up for production in each period.

In DLSP, set-up carryover is not allowed for idle periods. In the CSLP, no setup occurs between two batches of the same item if no other item has been produced during idle periods. For example, assume that a batch of item i is finished in period a, and the same item i is produced in the subsequent period b. Consider the case where the machine is idle between periods [a+1, b-1]. The setup costs for item j are incurred twice in the DLSP model, but setup costs incur once in the CSLP model, since setup costs are incurred only when producing a different type of item.

To formulate a mixed-integer program model for the standard CSLP, one simply replaces constraints (2.7) in the standard DLSP model with constraints (2.15)

$r_i q_{it} \leq C_t y_{it} \quad \forall i,t$ 2.15 - Amount of items produced in each period

This allows the production to be any continuous size between zero and full capacity. One disadvantage of the CSLP model is that, when the capacity of a period is not used in full, the remaining capacity is left unused. This problem could be addressed by the proportional lot sizing and scheduling problem (PLSP) in the subsequent section.

Continuous Setup Lot sizing Problems (CSLP) have been investigated by several researchers. Karmarkar et al. (1985) consider CSLP where each of the items has a setup period of one. They formulate the CSLP as a network problem, and present a Lagrange relaxation approach coupled with subgradient optimization to solve it. Pochet et al. (1991) solve the single level of CSLP using strong cutting planes.

To get a better understanding of the standard DLSP and CSLP models, we present and solve a small example. Example 2.1: Consider the production planning problem of 2 items, 1 machine, and 10 planning periods. Assume that the capacity of machine is 50 units in each period (C_t=50), and it takes one unit of machine to produce one unit of each item (r_i=1). Setup time is assumed to be very small, so it can be negligible. Data for demand for items, holding cost, and setup cost are given in Table 2.1. Table 2.2 represents the optimal production quantity for item in each period (q_{it}) and total cost. The optimal machine schedule for DLSP and CSLP is displayed in Figure 2.1

Table 2.1: Data of Example 2.1 for Standard DLSP and CSLP

Period	1	2	3	4	5	6	7	8	9	10	hi	si
Demand for item 1		40				40				60	2	400
Demand for item 2					30	30			40		1.5	150

Table 2.2: Optimal Solution for DLSP and CSLP for the Example 2.1

Model	Period (t)	1	2	3	4	5	6	7	8	9	10	Total optimal cost
DLSP	q_{1t}	0	50	50	0	0	0	0	0	0	50	**1720**
	q_{2t}	0	0	0	0	50	50	0	0	0	0	
CSLP	q_{1t}	0	40	0	40	0	0	0	0	10	50	**1190**
	q_{2t}	0	0	0	0	30	30	0	40	0	0	

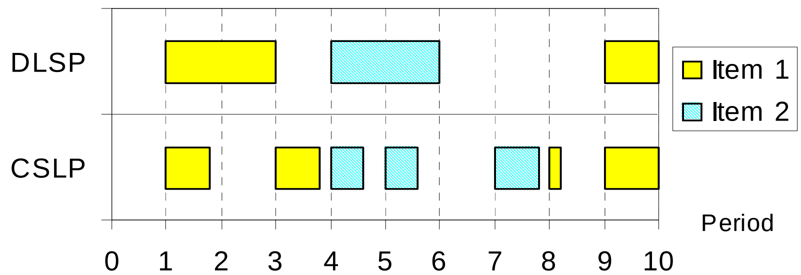


Figure 2.1: Gantt Chart for DLSP and CSLP for Example 2.1

As seen in Figure 2.1 for CSLP, no setup for item 2 is required at the beginning of period 6 after the machine is idle in at the end of period 5. It should be pointed out that the set of feasible solutions of DLSP is a subset of the set of feasible solutions of CSLP due to a restriction on production in each period. Consequently, the optimal total cost of CSLP is always no greater than that of DLSP.

We further consider the case of DLSP with setup times (DLSP-ST). The machine takes one period of time to setup for production of items 1 and 2, then solving DLSP-ST with data in example 2.1 yields the following optimal machine schedule shown in Figure 2.2. The optimal production quantities and total cost remain unchanged as in DLSP.

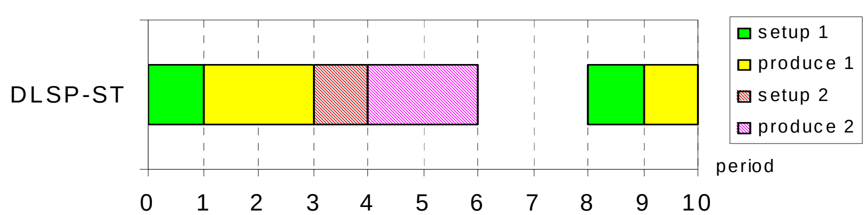


Figure 2.2: Gantt Chart for DLSP-ST for Example 2.1

2.1.4.3 Proportional Lot sizing and Scheduling Problem (PLSP)

The basic concept of the proportional lot sizing and scheduling problem (PLSP) is to use remaining capacity for production of a second item in the period in which the remaining capacity is left unused. In PLSP, a machine produces continuous lot-sizes over one or several, either adjacent or non-adjacent periods when the machine is idle. The underlying assumption of the PLSP is that at most one changeover is allowed within each period. As a result, at most two items can be produced per period. If the first item does not fully use capacity in a period, the remaining capacity can be used by the second item. Similar to the CLSP, the PLSP preserves the setup state over idle periods. PLSP can be formulated as a mixed-integer program. The details of the formulation can be found in Hasse (1994) and Drexl and Kimms (1997).

Several variants of the proportional lot sizing and scheduling problem are studied. Hasse (1994) introduces the mixed-integer program formulation for the PLSP with setup times and the PLSP with sequence dependent setup costs. Kimms (1999) develops a mixed-integer program formulation for the multi-level, multi-machine PLSP, and presents a genetic algorithm to solve PLSP.

2.1.5 Other Lot Sizing and Scheduling Models

In this section, we briefly review research of other lot sizing and scheduling models.

2.1.5.1 General Lot-sizing and Scheduling Problem (GLSP)

The GLSP is a single-level, single machine, multi-item problem where each lot of item is uniquely assigned to a position number in order to determine the sequence of items in each period. GLSP is a large bucket model. In contrast to CLSP, decisions on lot sizing and scheduling are made simultaneously in order to minimize total setup and holding costs. The underlying assumption of the GLSP is that a user arbitrarily imposes the number of lots per period. The reason for this is to reduce the computational time for a problem with a large number of periods. Note that it is possible to produce the same item at several positions in a period. If the maximum number of lots is one in every period, then GLSP is the same as CSLP. Drexl and Kimms (1997) propose mixed integer program for GLSP. However, the GLSP has not been received much attention from researchers.

2.1.5.2 Capacitated Lot-Sizing Problem with Linked lot sizes (CLSPL)

The CLSPL is a big bucket model, where multiple items can be produced by a single machine within a period, and at most one setup status for items can be carried over from one period to the next. That is, two lots of adjacent periods are linked, requiring an additional setup in the second period. The CLSPL can be formulated as a mixed integer program. Haase (1994) developed a stochastic heuristic to solve the CLSPL. Sox et al. (1999) propose a mixed integer program based on a shortest-path representation for the CLSPL without setup time. They present a Lagrangian decomposition heuristic based on subgradient optimization and dynamic programming to solve the CLSPL. Gopalakrishnan et al. (2001) develop a tabu-search heuristic for the CLSPL with sequence dependent setup times and setup costs. CLSPL instances with up to 30 items and 20 periods are solved. Suerie et al. (2003) propose a mixed integer program for CLSPL with sequence independent setup times and setup costs.

They use a branch and cut approach within a standard MIP solver to solve CLSPL instances of up to 30 items and 20 periods. They argue that this solution approach provides a better solution quality than other solution algorithms.

2.1.5.3 Multi-level Lot-sizing and Scheduling

In a practical setting, manufacturers might face multi-level lot sizing and scheduling problems with general item structure. Pochet et al. (1991) solve the multi-stage lot-sizing problem with general item structure using strong cutting plane. The authors deal with the general item structure using echelon stock. They find near-optimal solutions to problems with up to 50 components. Tempelmeier and Derstroff (1996) propose a Lagrangean relaxation-based heuristic approach for the dynamic multi-level multi-item lot-sizing problem for general item structures with multiple resources and setup times. Lower bounds and upper bounds on the minimum objective function value are derived. The problem with up to 40 items, 16 periods, and 6 resources is solved.

2.2 Literature Review of Batch Production Process

In batch production process, products are produced in batches rather than in a discrete or continuous model. Batch processes are widely used in the pharmaceutical, chemical, food, paint, and agrichemical industries, because they provide the flexibility to produce various products using the same processing facility. Compared with discrete parts manufacturing scheduling, such as those used in the electronic and automotive industries, scheduling batch processes is fairly complicated due to the large varieties of constraints, such as non-preemptive processes, intermediate storage policy (i.e. unlimited, finite), batch size (i.e. variable, fixed) , processing sequences, and shared resources (i.e. equipment, labor, utilities).

According to Orcun et al. (2001), the chemical industry has become more interested in batch production processes in the last two decades because of higher flexibility for the production of a high variety of products in small amounts, and lower investment in comparison with continuous processes. Since customers now require a wide-variety of specifications of products, demands for products are subject to uncertainty and rapid fluctuations, and demand for a brand new product is more difficult to be forecast. Therefore, batch processing has gained considerable popularity.

To obtain high flexibility from batch production processes, the planner should effectively coordinate resources, such as equipment, utility, labor, raw materials, and storage tank by determining the optimal product mix, and developing efficient production plans, as well as, operational scheduling of equipment. However, in practice, optimizing the production scheduling of batch production plants is difficult due to the large variety of processing equipment with varying operational characteristics, uncertainty in demand for products, etc.

The literature of scheduling of batch processes in chemical plants can be divided in two main groups. The first group addresses the optimal design problem of multi-product batch plants by determining the assignment of processing tasks to processing units, and sizes of equipments subject to scheduling restrictions so as to minimize total cost (See Coulman (1989), Birewar and Grossmann (1990), Voudouris and Grossmann (1996)).

The second group deals with the scheduling problems of operations of existing batch plants. A good overview of research advances in this area can be found in Floudas and Lin (2004). Typical objectives of these problems include makespan minimization, earliness and tardiness cost minimization, and profit maximization. (See Pekny et al. (1990), Kondili et al. (1993), Dessouky et al. (1999), McGraw and Dessouky (2001), Floudas and Lin (2004)). The model includes several constraints, such as non pre-emptive operation, intermediate storage policy, batch size, processing sequences, changeover, and shared resources. Most of the researchers formulate this problem as a MIP. None of these works consider the shelf-life of products and a change in the number of available machines over the planning horizon due to planed maintenance. Furthermore, all demands are assumed to be satisfied. The mathematical models for this problem are generally classified into two classes according to the type of time domain representation, namely, discrete-time, and continuous-time scheduling methods.

Discrete-time formulations divide the planning horizon into a number of time intervals of equal duration (period), and events, such as setup and production, have starting and completion times associated with the boundaries of time intervals. Although discrete-time models are able to account for many operational features, such as storage modes, resource constraints, changeovers, mass balance, they have two major drawbacks: the discrete approximation of time, and the large size of MIP problems for real industrial problems due to very large number of binary variables and constraints. Kondili et al. (1993) suggest that a time interval should be sufficiently small in order achieve a suitable approximation of the real-world problem, namely the greatest common factor of the processing times and setup times. However, this could result in a very large combinatorial problem of intractable size.

Kondili et al. (1993) present a general discrete-time MIP formulation for short-term scheduling problems of batch operations in chemical plants, which are represented using a state-task network. They consider several operational constraints such as equipment allocation, capacity limitation, inventory balance, storage capacity in order to maximize profit, which is the difference between total revenue and total cost (i.e., feedstocks, storage, and utilities).

Due to the difficulty in solving large MIP problems for a batch chemical plant based on discrete time model, several techniques have been developed in order to improve solution efficiency, including

(i) Reformulating allocation and batch sizing constraints based on variable aggregation or disaggregation by Shah et al. (1993), Sahinidis et al. (1991), and Yee et al. (1998).

(ii) Adding additional constraints (cuts), which reduce the region of integer infeasibility by Dedopoulos et al. (1995) and Yee et al. (1998).

(iii) Intervening the branch-and-bound procedure and fixing variables to values implied during branch-and-bound procedure by Dedopoulos and Shah (1995).

(iv) Using decomposition techniques, which divide a large and complex problem into smaller subproblems, by Bassett et al. (1996).

In continuous-time models, events are allowed to take place at any point in the continuous domain of time using variables related to time event. Variables are used to determine the timings of events. Floudas et al. (2004) point out that the continuous-time models could eliminate a major fraction of the inactive interval assignments. The resulting mathematical models have usually smaller sizes and require less computational time for their solutions. However, due to the variable nature of the timings of the events, one faces the difficulty in formulating the mathematical models in continuous domain of time, and the resulting models may be more complicated compared to their discrete-time ones. Continuous-time models do not account for unfulfilled orders, sequence-dependent setup costs and products with fixed shelf-life. By using continuous-time domain, scheduling problems of batch operations in chemical plants can be formulated as MIP or MINLP. Using linearization techniques can covert MINLP into MIP (See Glover (1975), Floudas (1995)).

Floudas et al. (2004) classify continuous-time models into two categories based on the type of processes, namely sequential processes and general network-represented processes. For sequential processes, most researchers use non-slot based formulations. That is, continuous variables are used to directly represent the task timings. (See Ku and Karimi (1988), Moon et al. (1996), Cerdá et al. (1997), Méndez et al. (2000), Hui et al. (2000), Orçun et al. (2001)). For general network-represented processes, a review of research in this area can be found by Zhang and Sargent (1998), Mockus and Reklaitis (1999), Schilling and Pantelides (1996), Floudas and Lin (2004).

In our research, we consider the production scheduling problem of multi- product batch plants in which products have fixed shelf-life, one stage and parallel machines of batch production processes. Given deterministic demand for products, and restriction on capacity of machines, processing time, and shelf-life of products, we formulate this problem as a MIP in a discrete time domain. The model incorporates several factors, such as sequence-dependent setup costs/times, fixed processing time of non preemptive batch operation, fixed shelf life of products, batch size (production lot for each release), and other costs associated with inventory, unmet demand, spoilage, and production.

This model can be applicable to production scheduling in many real industrial problems including fermentation processes for beers and yoghurts, incubation processes for vaccine production, and mixing processes for medicine production.

In this research, we interchangeably use the terms "product" and "item", and "machine" and "equipment". We define **"the batch size of item"** (production lot for each release) as the amount of the same type of item processed by a machine at the same time. That is, a machine can processes at most one type of item for a fixed processing time without any interruption (no preemption is allowed).

In this context, **"batch"** means that a whole of the same type of item goes into and goes out from the processing unit (e.g. fermentation tanks, reactors, incubators) at the same time. A one stage of batch production process is shown in Figure 2.3. A setup on a machine has to be performed before production of a batch.

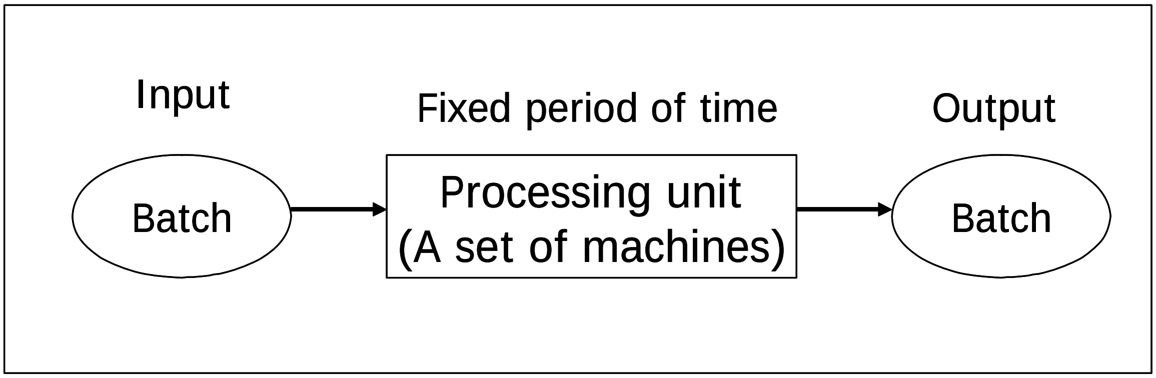


Figure 2.3: Single Stage of Batch Production Process

To illustrate our concept of a batch production process, consider the following illustrative example. A yogurt homemade producer has one fermentation tank. Suppose that the tank can be used to ferment two types of yogurt, Y1 and Y2. The setup time for production of each of yogurts takes 1 period, and the fixed process times are 2 and 3 periods for Y1 and Y2 respectively. He would like to determine a feasible schedule for the tank for a planning horizon of 12 periods. One of the feasible schedules for the tank is displayed in Figure 2.4

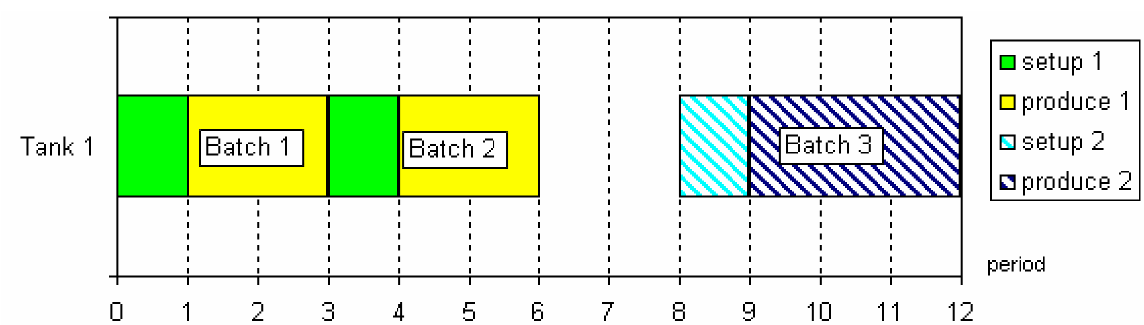


Figure 2.4: Gantt Chart for BPP-SI for the Example 2.2

The main features of the batch production process are:

- A setup carryover for the same product is not allowed in the batch production process. As seen in Figure 2.4, a new setup is required for batch #2 of yogurt Y-1 at the beginning of period 4. On the other hand, DLSP models in the previous section allow a setup carryover over period if the same type of product is produced contiguously on a machine. Hence, no setup is required for batch #2 in DLSP models.
- Each batch of product requires a setup whenever a new batch is released to the machine.
- In batch production processes, each machine can process at most one product at a time, while a batching machine, discussed in the subsequent section, is able to process more than one product at the same time.
- The batch operation is non-preemptive, i.e. once begun, an operation cannot be interrupted until it is completed.

2.3 Literature Review of Inventory Management of Perishable Items

Perishable items can be divided into two categories: fixed or random lifetime. For items where the lifetime is fixed, the utility of each unit is constant during a fixed period of time. An example of this type of item is blood, which can only be stored for a period of approximately 42 days, according to the Red Cross Organization. For items where the lifetime is random, the utility of the item gradually decreases throughout its lifetime. Examples of random lifetime items are fresh produce, and some types of volatile chemicals. An extensive literature review on inventory management of perishable items can be found in Nahmias (1982) and Silver et al. (1998). In practice, the life of a perishable product is dependent on the product's characteristics and the storage conditions, such as temperature, humidity level, and air circulation, etc. In general, cool temperatures and low humidity provide the best storage conditions.

In our research, we focus on items with a fixed lifetime period by using first-in-first-out (FIFO) as an inventory management policy. During their fixed lifetime period, the quality of products does not significantly change in taste, color, texture, or nutrient content, but the products will then be disposed of after such period.

In the next chapter, we present details of the batch production problem for perishable products with sequence-independent setup times (BPP-SI).

CHAPTER III

PROBLEM DEFINITION AND MATHEMATICAL MODEL FOR BATCH PRODUCTION SCHEDULING FOR PERISHABLE PRODUCTS WITH SEQUENCE INDEPENDENT SETUP TIMES (BPP-SI)

Our focus is on a single stage batch process, which is used to produce a variety of perishable products. These multiple products share the same production equipment, and setup times are significant. The reasons for batch production include economies of scale due to large setup costs and technological restrictions, such as the fixed size of a processing tank in a chemical process. The batch production is used in many different environments, i.e., pharmaceutical, polymer, food, specialty chemistry industries. As mentioned earlier, the key features of the batch production are the operation is non-preemptive, each machine can process at most one product at a time, and each batch of product requires a setup whenever a new batch is released to the machine. Because products are perishable, the First-In-First-Out (FIFO) policy is used to manage inventory. When the setup time on a machine is *independent of the sequence* of products produced, it can be incorporated into the fixed processing time. We focus on two cases of batch size either discrete lot size (i.e. full capacity or zero production) or continuous lot size (i.e. batch size falls between zero and full capacity)

In this chapter, we define the batch production scheduling problems for fixed shelf-life products with sequence-independent setup time (BPP-SI), develop a Mixed Integer Program (MIP) for this problem, and present a numerical study for a small sized problem of the fermentation process of beer, previously discussed in Section 1.1.

3.1 Problem Statement for BPP-SI

Master production scheduling is one of key components of short-term plans. The resulting optimal schedule indicates how to allocate the products to machines, the sequencing of processing products on each machine, and the production quantity of each product. This short-medium term plan typically has a planning horizon of less than six months. The operational decisions are made daily or weekly. In this chapter, we are interested in determining the master production scheduling (MPS) for the batch production scheduling problem for fixed shelf-life products with sequence-independent setup time (BPP-SI). The costs associated with the batch production problem include:

- Fixed setup costs
- Variable production costs
- Variable holding costs
- Variable disposal costs
- Variable costs for unmet demand

In the BPP-SI, the forecast demand for each product and capacity planning during the planning horizon are given. The decision maker is concerned with how to efficiently deal with one batch processing unit of parallel machines for production of perishable products.

The reason for choosing one major batch unit is that the batch operation step typically accounts for most of the residence time of products in the system, and is the bottleneck step in production system, such as fermentation, drying etc. Furthermore, the cost of equipment for this unit is fairly expensive, so it seems reasonable for a manufacturer not having excessive pieces of equipment so as to save investment cost. The minimum total cost is considered as our measure of system performance. In the master production scheduling, he faces the following questions:

- How large should each of production lot (batch size) of products be?
- When should a setup for a machine for each batch of product to be performed?
- What is the sequence of processing products on each machine?
- What are the amounts of inventory level, spoilage, and unmet demand for each product over the planning horizon?

In order to answer to these questions, one has to simultaneously solve lot-sizing and scheduling problems with setup times. If all model parameters are provided, decisions on lot-sizing and sequencing can be made together by solving a MIP on discrete time domain.

3.2 Model formulation

We define the BPP-SI in the following way. The planning horizon of T is divided into a number of intervals of equal duration {1,…,T}. For a batch processing unit, each of M parallel machines has a capacity of C units. Each machine can be used to produce N perishable products, each having a limited life of LT_i periods. Each machine requires a setup of STi periods before taking a fixed processing time of BTi periods to produce a batch of product i. That is, each machine requires total production time of ATi (=BTi+STi) periods, whenever it used to produce a batch of product i.

Consider the time line for a planning horizon in Figure 3.1, if the planner schedules to setup the machine at the end of period 0 (at the beginning of period 1) for production of a batch of product i, which will be finished at the end of period ATi. This batch of product i is in good condition during periods [ATi+1, ATi+LTi]. After that interval, this batch will go bad. Note that the machine is continuously reserved for one batch of product i during AT_i periods.

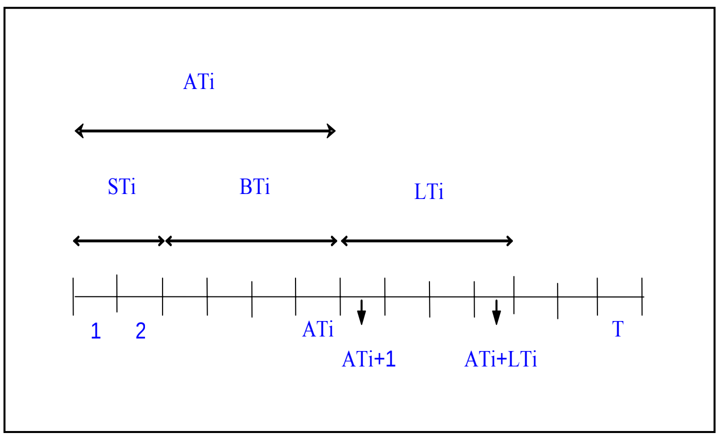


Figure 3.1: Time Line for Planning Horizon in BPP Problem

Demand for product i in each period t is given by D_{it}. Initial inventory of product i at the beginning of planning horizon is given by I_{io}. It is assumed that unit cost of disposing spoiled product is dc_i. Unmet demand for product in a period is lost with a unit penalty cost of uc_i. Leftover product carried over to the next period incurs a unit holding cost of hc_i per period. Unit production cost for product i is pc_i. Assuming that if a machine is setup for product i, the fixed setup cost of rc_i incurs, and is independent of sequence of products produced and the batch size of product.

Due to perishable characteristics, managing inventory of product is based on the FIFO. In other words, the first products produced are assumed to be the first sold.

Regarding the batch size (production lot) of product in batch production environment, we assume that if the equipment (i.e. a mixing tank in chemical process) is used for the production, the full capacity of the equipment will be employed, i.e. the batch size is equal to full capacity. If not, production is zero. This assumption is called "all or nothing production". The capacity of each of equipment might be different. For simplicity, we assume that all equipment has the same capacity of C. For example, suppose that the equipment has capacity of 100 gallons, and it takes 5 hours for setup time and batch processing time to produce a product. Therefore, 100 gallons of a product can be produced every 5 hours. Note that we do not use the production rate of 20 gallons per hour to define the term of capacity, since we cannot produce a product of 20 gallons within one hour due to the characteristics of batch process.

3.2.1 Assumptions for the BPP-SI Model

The following underlying assumptions are made for the BPP-SI model:

A0: Each machine can be used to produce at most one product in each period.

A1: If the production for a product takes place, the batch size, which is equal to full capacity, will be produced. If not, production is zero.

A2: No preemption is allowed. That is, if a machine is scheduled to produce a product in period t, the machine will be occupied by such product for next AT_i periods without any interruption. This assumption is reasonable especially for chemical and food industry, because the setup cost for production is prohibitively expensive, and the batch production of product cannot be interrupted to attain the specification of products.

A3: At most one setup can incur in each period.

A4: Demand for products is dynamic and deterministic over planning horizon.

A5: Setup time and batch-processing time are deterministic, independent of product sequence on machine and the size of production lot.

A6: Any products, which lasts over their limited shelf-lives, goes bad (spoilage).

A7: Assuming that initial inventory is brand new.

A8: Unmet demands for product are lost with a unit penalty cost.

A9: Assuming that there is sufficient amount of raw materials used to produce products. The material costs are included in the production costs of products. This assumption is reasonable to avoid the starvation problem of batch production unit.

Also, there is no restriction of the storage space for raw materials and products.

A10: Workforce restriction is not considered, since the batch production step is not labor intensive.

A11: Investing in a new machine is not an alternative way to satisfy demand, since the planning horizon is fairly short.

3.2.2 BPP-SI with Discrete Batch Size

The following notation is used for BPP-SI model.

Indices:

i = Index for products ($i = 1,\dots,N$)

j = Index for machines ($j = 1,\dots,M$)

t = Index for time periods ($t = 1,\dots,T$)

Data:

dc_i = Unit disposal cost for spoiled product i ($/unit of product)

hc_i = Holding cost per unit per time for product i ($/period/unit of product)

pc_i = Unit production cost for product i ($/unit of product)

rc_i = Fixed setup cost on machine for production of a batch of product i ($/setup)

uc_i = Unit penalty cost for unmet demand of product i ($/unit of product)

$D_{i,t}$ = Demand for product i in period t (unit of product)

$I_{i,0}$ = Initial inventory for product i (unit of product)

AT_i = Production time for product i ($AT_i = BT_i + ST_i$) (periods)

BT_i = Batch processing time for product i (periods)

ST_i = Setup time for product i (periods)

LT_i = Limited shelf life for product i (periods)

M = Total number of machines

NM_t = Number of machines available for use in period t

C = Capacity of a machine (unit of product)

Decision Variables:

P_{it} = Amount of product i obtained at the beginning of period t

I_{it} = Amount of inventory of product i at the end of period t

U_{it} = Amount of unmet demand of product i in period t

S_{it} = Amount of product i spoiled in period t

O_t = Total number of machines used in period t

$q_{i,j,t}$ = Amount of product i scheduled to released in period t to machine j

$w_{i,j,t}$ = Binary variable for machine status indicating whether machine j is occupied by product i in period t ($w_{i,j,t} = 1$) or not ($w_{i,j,t} = 0$)

$r_{i,j,t}$ = Binary variable for setup on machine indicating whether machine j is setup to produce product i at the beginning of period t, and the batch will be completed at the beginning of period $t+AT_i$ ($r_{i,j,t} = 1$) or not ($r_{i,j,t} = 0$)

The BPP-SI can be formulated as a MIP as follows:

Minimize (Objective) **3.1** ***Minimize Total Cost***

$= \sum_t \sum_i I_{i,t} hc_i$	-	Variable holding cost
$+ \sum_t \sum_i P_{i,t} pc_i$	-	Variable production cost
$+ \sum_t \sum_i U_{i,t} uc_i$	-	Variable penalty cost for unmet demand
$+ \sum_t \sum_i S_{i,t} dc_i$	-	Variable disposal cost for spoiled product
$+ \sum_t \sum_j \sum_i r_{i,j,t} rc_i$	-	Fixed setup cost of machine (releasing cost)

Subject to:

$I_{i,t} - U_{i,t} = I_{i,t-1} + P_{i,t} - S_{i,t} - D_{i,t}$	$\forall i, t$	3.2 -	Inventory balance for product
$P_{i,t} = C \sum_j r_{i,j,t-AT_i}$	$\forall i,\ AT_i + 1 \le t \le T$	3.3 -	Amount of product obtained at the beginning of period
$S_{i,t} \ge I_{i,t-1} - C \sum_{\tau=1}^{LT_i - 1} P_{i,t-\tau}$	$\forall \mathrm{i},\ \mathrm{LT_i} + 1 \le \mathrm{t} \le T$	3.4 -	Amount of product spoiled at the beginning of period
$\sum_{\tau=1}^{AT_i} r_{i,j,t+\tau-1} \le 1$	$\forall i, j,$ $1 \le t \le T - AT_i + 1$	3.5-	Batch release on machine during batch production time
$\sum_i r_{i,j,t} \le 1$	$\forall j,\ t$	3.6 -	At most one batch release on machine in each period
$\sum_i \sum_j w_{i,j,t} = O_t$	$\forall t$	3.7 -	Calculate number of machines in use
$\mathrm{O_t} \le \mathrm{NM_t}$	$\forall t$	3.8 -	Maximum number of machines

$AT_i r_{i,j,t} \le \sum_{\tau=1}^{AT_i} w_{i,j,t+\tau-1}$	$\forall i, j,$ $1 \le t \le T - AT_i + 1$	3.9 -	Machine occupied by a product during total production time
$\sum_i w_{i,j,t} \le 1$	$\forall j,t$	3.10 -	Machine occupied by a product in each period
$r_{i,j,t} = 0$ $P_{i,t} = 0$	$\forall i, \mathrm{T}\text{-}\mathrm{AT_i}+1 \le \mathrm{t} \le \mathrm{T}$ $\forall i, 1 \le \mathrm{t} \le AT_i$	3.11	Logical constraints
$\mathrm{I_{i,t}}, \mathrm{U_{i,t}}, \mathrm{S_{i,t}}, \mathrm{P_{i,t}} \ge 0$	$\forall i,t$	3.12 -	Non-negative variables
$O_t \in \{0,1,2,...\}$	$\forall t$	3.13 -	Non-negative integer variables
$\mathrm{w_{i,j,t}}, \mathrm{r_{i,j,t}} \in \{0,1\}$	$\forall i, j, t$	3.14 -	Binary variables

Objective function (3.1) is to minimize total cost over planning horizon. Total cost consists of holding cost, production cost, and penalty cost of unmet demand, disposal cost for spoilage, and setup cost. There are two main groups of constraints. The first group of constraints (3.2-3.11) are production scheduling constraints, such as inventory balance, production, spoilage, batch processing time, number of machines in use, sequence of products on machines. The second last group of constraints (3.12-3.13) involves variable constraints.

The meanings of these constraints are as follows:

- Constraints (3.2) ensure the inventory balance for product in each period.
- Constraints (3.3) ensure that the amount of product produced at the beginning of period t is obtained by releasing full batch of such product at the beginning of period t-AT_i to each machine.

- Constraints (3.4) are used to compute the amount of product spoiled at the beginning of each period. To determine the amount of product i spoiled at the beginning of period t (S_{it}), one first needs to know two things, which are the amount of product left at the end of period t-1 and total amount of product *produced* from period t-LT_i+1 to period t-1.

  The amount of spoilage can be expressed by the following equation:

  $$S_{i,t} = \max\ \{0, I_{i,t-1} - C\sum_{\tau=1}^{LT_i-1} P_{i,t-\tau}\} \qquad \forall i,t$$

  This equation can be readily converted into the following inequalities:

  $$S_{i,t} \geq I_{i,t-1} - C_i\sum_{\tau=1}^{LT_i-1} P_{i,t-\tau} \qquad \forall i,t$$

  $$S_{i,t} \geq 0 \qquad \forall i,t$$

- Constraints (3.5) ensure that a machine can be scheduled to produce the product i only once during production time of AT_i periods.
- Constraints (3.6) ensure that a machine can be scheduled to produce at most one product in each period.
- Constraints (3.7) are used to determine total number of machines used in each period.
- Constraints (3.8) ensure that in each period, total number of machines in use does not exceed total number of machines available.
- Constraints (3.9) ensure that if a machine is initially setup to produce the product i at the beginning period t, then the machine must be occupied by the product i from period t until period t+AT_i-1.
- Constraints (3.10) ensure that at most one product can occupy a machine in each period.

- Constraints (3.11) are logical constraints on batch release and production.
- Constraints (3.12) are non-negativity constraints on amounts of inventory, unmet demand, spoilage, and production of each product in every period.
- Constraints (3.13) are non-negative integer constraints on the number of machines.
- Constraints (3.14) impose on binary variables.

3.2.3 BPP-SI with Continuous Batch Size (BPP-SI-CB)

Suppose that we relax assumption A1 of all or nothing production. In other words, the production lot (batch size) of product can take on continuous values between zero and full capacity, assuming there is no restriction on the minimum production lot (batch size) for each product. While the objective function remains unchanged, we need to modify the constraints by replacing constraints (3.3-3.4) with constraints (3.15-3.16), and adding constraints (3.17-3.18). The BPP-SI-CB can be formulated as a MIP.

$P_{i,t} = \sum_{j} q_{i,j,t-AT_i}$	$\forall i,\ AT_i + 1 \le t \le T$	3.15-	Amount of product produced at the beginning of period
$S_{i,t} \ge I_{i,t-1} - \sum_{j} \sum_{\tau=1}^{LT_i-1} q_{i,j,t-AT_i-\tau}$	$\forall \mathrm{i}\ ,\ \mathrm{LT_i}+1 \le \mathrm{t} \le T$	3.16 -	Amount of product spoiled at the beginning of period
$q_{i,j,t} \le \mathrm{C}r_{i,j,t}$	$\forall i, j,\ 1 \le t \le T - AT_i + 1$	3.17-	Batch size of product on each machine in a period
$\mathrm{q_{i,j,t}} \ge 0$	$\forall i, t$	3.18-	Non-negative variables

3.2.4 Numerical Result for an Example of BPP-SI and BPP-SI-CB

In this section, we present a small example for BPP-SI and BPP-SI-CB model for the fermentation process of beer. There are 3 types of products (beers), 3 machines, and planning horizon of 10 periods. The capacity of each machine (fermentation tank) is 50 units. Assume that all of machines do not fail and are scheduled for maintenance.

Each beer has a fixed shelf life of 3 periods. The amounts of initial inventory are 40, 50, and 32 units for beer 1, 2, and 3 respectively. Data for demand for beers, costs, and production times are given in Tables 3.1, 3.2, and 3.3 respectively. The results of the optimal quantities (production, inventory, spoilage, and unmet demands) for products, machine schedule, and the optimal total cost of BPP-SI and BPP-SI-CB are shown in Tables 3.4, and 3.5 respectively.

Table 3.1: Demand Data of the Example 3.1

Product	Time Period									
	1	2	3	4	5	6	7	8	9	10
1	20	10	10	20	10	21	0	30	30	40
2	10	20	18	5	15	10	24	3	20	30
3	5	10	15	0	0	15	20	13	4	0

Table 3.2: Cost Data of the Example 3.1

Product (i)	Setup_Cost (RCi) ($/setup)	Holding_Cost (HCi) ($/unit)	Production_Cost (PCi) ($/unit)	Unmet-Dem_Cost (UCi) ($/unit)	Disposal_Cost (DCi) ($/unit)
1	200	2	20	35	3
2	300	3	24	40	4
3	400	4	30	45	5

Table 3.3 Data of Setup Time and Process Time of the Example 3.1

Product (i)	Setup Time (STi)	Process Time (BTi)	Production Time (ATi = STi+BTi)
1	1	2	3
2	1	3	4
3	2	3	5

Table 3.4: Result of BPP-SI for the Example 3.1

		Time Period (t)										Cost	**Subtotal**
		1	2	3	4	5	6	7	8	9	10	($)	**cost ($)**
Production (units)	P_{1t}				50				50	50		3000	
	P_{2t}					50				50		2400	
	P_{3t}						50					1500	**6900**
Inventory at end of period (units)	I_{1t}	20	10	0	30	20	0	0	20	40	0	280	
	I_{2t}	38	18	0	0	34	24	0	0	30	0	432	
	I_{3t}	25	15	0	0	0	33	13	0	0	0	344	**1056**
Spoilage (units)	S_{1t}											0	
	S_{2t}	2				1						12	
	S_{3t}	2					2					20	**32**
Unmet demand (units)	U_{1t}						1					35	
	U_{2t}				5				3			320	
	U_{3t}									4		180	**535**
Setup cost ($)	MC1	$200				$200						400	
	MC2	$300				$300						600	
	MC3	$400					$200					600	**1600**
Gantt chart	MC1	s1	p1			s1	p1					**Total**	**10123**
	MC2	s2	p2			s2	p2						
	MC3	s3		p3			s1	p1					

Table 3.5: Result of BPP-SI-CB for the Example 3.1

		Time Period (t)										Cost	**Subtotal**
		1	2	3	4	5	6	7	8	9	10	($)	**cost ($)**
Production (units)	P_{1t}				50				50	50		3000	
	P_{2t}					49				50		2376	
	P_{3t}						48					1440	**6816**
Inventory at end of period (units)	I_{1t}	20	10	0	30	20	0	0	20	40	0	280	
	I_{2t}	38	18	0	0	34	24	0	0	30	0	432	
	I_{3t}	25	15	0	0	0	33	13	0	0	0	344	**1056**
Spoilage (units)	S_{1t}											0	
	S_{2t}	2										8	
	S_{3t}	2										10	**18**
Unmet demand (units)	U_{1t}						1					35	
	U_{2t}				5				3			320	
	U_{3t}									4		180	**535**
Setup cost ($)	MC1	$200				$200						400	
	MC2	$300				$300						600	
	MC3	$400					$200					600	**1600**
Gantt chart	MC1	s1	p1			s1	p1					**Total**	**10025**
	MC2	s2	p2			s2	p2						
	MC3	s3		p3			s1	p1					

From the results in Table 3.4 and Table 3.5, the optimal cost of BPP-SI-CB is slightly lower than that of BPP-SI by $98, due to lower costs of spoilage, production, and inventory. As BPP-SI-CB has more flexibility of production than BPP-SI in the sense that the production lot can be continuous between zero and full capacity, the BPP-SI-CB model produces the amount of products as needed, while satisfying the capacity restriction. Therefore, there is no spoilage from overproduction, but it could have spoilage from the initial inventory. To show how to compute the amount of spoilage and unmet demand, we consider product 2 with initial inventory of 50 units and 3 shelf-life periods. Since total demand of product 2 during the first three periods is 48 units, we would rather dispose of 2 units of the product at the beginning of period 1, instead of holding it until period 4 before disposing of it. By doing this, the unnecessary holding cost can be avoid. Since a machine takes 4 periods of production time for one batch of product 2, the demand of 5 units for product 2 in period 4 will be unmet. In this example, the setup costs for two models are equal, since they have the same Gantt chart. However, it is not necessarily true that the machine schedule for both models will always be the same for a given problem. An example of this would be when the production cost is very high, and setup cost is very low. This example is presented in Chapter VI.

CHAPTER IV

PROBLEM DEFINITION AND MATHEMATICAL MODEL FOR BATCH PRODUCTION SCHEDUDING FOR PERISHABLE PRODUCTS WITH SEQUENCE DEPENDENT SETUP TIMES (BPP-SD)

In some manufacturing settings, the setup times/costs might be significant and sequence-dependent. In pharmaceutical tablet production, for example, the process of tablet coating involves setup times (i.e. cleaning time) and setup costs (chemical agents for cleaning) for switching types of solution. When switching from a lighter to a darker coating solution, a minimum of cleaning is required. However, when switching from a darker to a lighter one, the coater must be completely cleaned in order to avoid color residue and impurities. The difference in the setup time can take up to 16 hours according to Camelot IDPRO AG.

In this chapter, we consider the batch production scheduling problems for fixed shelf-life products with sequence-dependent setup time (BPP-SD), develop a Mixed Integer Program (MIP) for this problem, and present the numerical result for a small problem of the incubation process of vaccine.

4.1 Problem Statement for BPP-SD

We assume that setup times/costs, the forecast demand for each product, and capacity planning during the planning horizon are given. The decision maker is concerned with how to efficiently deal with one batch processing unit of parallel machines for production of perishable products, when the setup times/costs are sequence-dependent. The characteristics of batch production can be found in section 3.1.

He would like to determine the master production scheduling, which minimizes total cost comprising costs of inventory, spoilage, production, setup and penalty for lost sales. MPS indicates the sequencing of production of products on each machine, the production quantity of each product in each period, the beginning time and completion time of each batch. The unmet demands are assumed to be lost.

If all model parameters are provided, decisions on lot-sizing and sequencing for BPP-SD can be made together by solving a MIP on discrete time domain. Due to the complexity of sequence-dependent setup times/costs, solving this problem is very difficult even for small sized problems.

4.2 Model Formulation

Similar to BPP-SI, we divide the planning horizon of T into a number of intervals of equal duration {1,…,T}. At one batch processing unit, each of M parallel machines has a capacity of C units. Each machine can be used to produce N perishable products, each having a limited life of LT_i periods. We use the similar concept of the time line in section 3.2, except the setup times depending on the former and current products.

To deal with sequence-dependent setup times/costs, we introduce an artificial i of zero, which indicates that the machine is idle. Furthermore, we incorporate another index k, which indicates that the previous product produced or setup on a machine, for two binary variables for production and setup respectively ($Y_{k,i,j,t}$ and $V_{k,i,j,t}$). We assume the setup costs/times for switching from product 0 (machine is idle) to other product i is equal to those for switching from product i to product i. As mention in the former chapter, each batch of production requires setup times/costs for cleaning and changing tools for machines.

We illustrate the new setup times of BPP-SD. Suppose that a machine is used for a product k at the beginning of period t. In order for such machine to produce a product i, it takes ST_{ki} periods for setup, and BT_i periods for processing this batch. Hence, the batch of product i will be obtained at the beginning of period $t+ST_{ki} + BT_i$. This implies that each machine requires total production time of AT_{ki} ($=ST_{ki} + BT_i$) periods, whenever it used to produce a batch of product i. Due to the fixed shelf-life periods of LT_i for product i, this batch of product i will be in good condition during periods $[t+AT_{ki}+1, t+AT_{ki}+LT_i]$. After that interval, this batch will go bad. Note that the machine is continuously reserved for one batch of product i during a period of AT_{ki}.

Demand for product i in each period t is given by D_{it}. Initial inventory of product i at the beginning of planning horizon is given by I_{io}. It is assumed that unit cost of disposing spoiled product is dc_i. Unmet demand for product in a period is lost with a unit penalty cost of uc_i. Leftover product carried over to the next period incurs a unit holding cost of hc_i per period. Unit production cost for product i is pc_i. Assuming that if a machine is setup for a product i, the fixed setup cost of sc_{ki} incurs, and is dependent of sequence of the former product k and current product i. Managing inventory of product is based on the First-In, First-Out (FIFO). The batch size (production lot) of product is either zero or full capacity.

4.2.1 Assumptions for the BPP-SD model

Almost all of the assumptions of BPP-SD are the same as that of BPP-SD in section 3.2.1, except that the fixed setup costs of sc_{ki} and the fixed setup times of ST_{ki} for switching from product k to product i.

4.2.2 BPP-SD with Discrete Batch size

The following notation is used for the BPP-SD model

Indices:

i, k, k' = Indices for products (i, k,k' = 0,…,N) , Index 0 : no production.

j = Index for machines (j = 1,…,M)

t = Index for time periods (t = 1,…,T)

Data:

dc_i = Unit disposal cost for spoiled product i ($/unit of product)

hc_i = Holding cost per unit per time for product i ($/period/unit of product)

pc_i = Unit production cost for product i ($/unit of product)

sc_{ki} = Fixed setup cost of switching from product k to product i ($/setup).

uc_i = Unit penalty cost for unmet demand of product i ($/unit of product)

$D_{i,t}$ = Demand for product i in period t (unit of product)

$I_{i,0}$ = Initial inventory for product I (unit of product)

AT_{ki} = Production time interval between product k to product i (in periods). That is, it is sum of setup time from switching production of product k to product i, and the fixed processing time for product i ($AT_{ki} = ST_{ki} + BT_i$)

BT_i = Fixed processing time for product i (in periods)

ST_{ki} = Setup time for switching from product k to product i (in periods)

LT_i = Fixed shelf life for product i (in periods)

C = Capacity of a machine (unit of product)

Decision Variables:

P_{it} = Amount of product i produced in period t

I_{it} = Amount of inventory of product i at the end of period t

U_{it} = Amount of unmet demand of product i in period t

S_{it} = Amount of product i spoiled in period t

$V_{k,i,j,t}$ = Binary variable indicating whether machine j is switched from product k to product i at the beginning of period t, and then the product i will be obtained at the beginning of period $t+AT_{k,i}$ ($V_{k,i,j,t} = 1$) or not ($V_{k,i,j,t} = 0$)

$Y_{k,i,j,t}$ = Binary variable indicating whether the machine j produces product i at the beginning of period t, while it previously produced product k ($Y_{k,i,j,t} = 1$) or not ($Y_{k,i,j,t} = 0$)

The BPP-SD can be formulated as a MIP as follows:

Minimize (Objective)	**4.1**	***Minimize Total Cost***
$= \sum_t \sum_i hc_i I_{i,t}$	-	Variable holding cost
$+ \sum_t \sum_i pc_i P_{i,t}$	-	Variable production cost
$+ \sum_t \sum_i uc_i U_{i,t}$	-	Variable penalty cost for unmet demand
$+ \sum_t \sum_i dc_i S_{i,t}$	-	Variable disposal cost for spoiled product
$+ \sum_t \sum_j \sum_i \sum_k sc_{k,i} (Y_{k,i,j,t})$	-	Fixed setup cost of machines

Subject to:

$I_{i,t} - U_{i,t} = I_{i,t-1} + P_{i,t} - S_{i,t} - D_{i,t}$	i=1,…,N; $\forall t$	4.2	Inventory balance
$P_{i,t} = C \sum_j \sum_k Y_{k,i,j,t}$	$\forall i,t$	4.3	Amount of product produced at the beginning of period
$S_{i,t} \geq I_{i,t-1} - \sum_{\tau=1}^{LT_i - 1} P_{i,t-\tau}$	$\forall$i, $LT_i + 1 \leq t \leq T$	4.4	Amount of product spoiled at the beginning of period
$\sum_{\tau=1}^{AT_{k,i}} \sum_k Y_{k,i,j,t+\tau-1} \leq 1$	$\forall i,k,j,\ AT_{k,i} \geq 2$ $1 \leq t \leq T - AT_{k,i} + 1$	4.5	At most one production take places during setup , and processing periods
$\sum_k \sum_i Y_{k,i,j,t} \leq 1$	$\forall j,t$	4.6	At most one production takes place in a period

$\sum_k \sum_i V_{k,i,j,t} \le 1$	$\forall j,t$	4.7	At most one setup can occur in each period
$V_{k,i,j,t-AT_{k,i}} + \sum_{\tau=1}^{AT_{k,i}-1} V_{i,i,j,t-\tau} \ge AT_{k,i} Y_{k,i,j,t}$	$\forall$i,k,j, $AT_{k,i}+1 \le t \le T$	4.8a	Relate the production and setup status of machine
$Y_{k,i,j,t} \le \sum_{k'} V_{i,k',j,t}$	$\forall$i,k,j,t	4.8b	Relate production and setup status of machine
$\sum_{k'} V_{k',k,j,t-1} \ge V_{k,i,j,t}$	$\forall$i,k,j, $2 \le t \le T$	4.9	Model a setup sequence on a machine. Each machine has to be setup before processing a product.
$Y_{k,i,j,t} = 0$	$\forall i,k,j, 1 \le t \le AT_{k,i}$	4.10	Logical constraints
$I_{i,t}$, $U_{i,t}$, $S_{i,t}$, $P_{i,t} \ge 0$	$\forall i,t$	4.11	Non-negativity variables
$Y_{k,i,j,t}$, $V_{k,i,j,t} \in \{0,1\}$	$\forall k,i,j,t$	4.12	Binary variables

- Objective function (4.1) is to minimize total cost over planning horizon. Total cost consists of holding cost, production cost, and penalty cost of unmet demand, disposal cost for spoilage, and sequence-dependent setup cost.
- Constraints (4.2) ensure the inventory balance for product in each period.
- Constraints (4.3) ensure that the amount of product produced at the beginning of period t.
- Constraints (4.4) are used to compute the amount of product spoiled at the beginning of each of periods. Detail is explained in section 3.2.2.

- Constraints (4.5) ensure that a machine can produce at most one product during its sequence dependent setup period and fixed processing period.
- Constraints (4.6) ensure that a machine can produce at most one product in a period.
- Constraints (4.7) enforce that if a machine, which previously processed an product k, produces an product i at the beginning of period t, then such machine can be setup only once during setup time of $ST_{k,i}$ periods and fixed processing time of BT_i periods.
- Constraints (4.8a-4.8b) are used to relate the production and setup status of a machine by keep tracking the information of a previously processed product and the next product to be produced. In order for machine j to produce a product i in period t, such machine must be set up for a product i in period t-$AT_{ki,}$ after the machine j previously produced product k in at beginning of this period.
- Constraints (4.9) are used to determine the correct sequence of setup for product on a machine. In other words, a machine can be setup for a product i in period t after it was previously setup for product k only if the machine was setup for product k in period t-1. Without these constraints, the sequence of setup could be wrong because the previous product, which was setup on machine, is ignored. Hence, the status of production and setup of machine may not be related as desired.
- Constraints (4.10) are logical constraints on production.
- Constraints (4.11) are non-negativity constraints on amounts of inventory, unmet demand, spoilage, and production of each product in every period.
- Constraints (4.12) impose on binary variables on setup and production.

4.2.3 BPP-SD with Continuous Batch Size (BPP-SD-CB)

In this section, we allow the production lot (batch size) of product to take on continuous values between zero and full capacity. Assuming that there is no restriction on the minimum production lot (batch size) for each product. While the objective function remains unchanged, we need to modify the constraints by replacing constraints (4.3) with constraints (4.13 -4.15). The BPP-SD-CB can be formulated as a MIP.

$P_{i,t} = \sum_{j} q_{i,j,t}$	$\forall i,t$	4.13-	Amount of product produced at the beginning of period
$q_{i,j,t} \leq C\sum_{k} Y_{k,i,j,t}$	$\forall i,j,t$	4.14-	Batch size of a product on each machine in a period
$q_{i,j,t} \geq 0$	$\forall i,t$	4.15-	Non-negative variables on batch size

4.2.4 Numerical Result for an Example BPP-SD and BPP-SD-CB

In this section, we present an illustrative example for BPP-SD and BPP-SD-CB model. Example 4.1: Consider a production scheduling problem in the incubation process of vaccines. There are 2 types of products (vaccines), 2 machines (incubators), and planning horizon of 10 periods. Assume that the capacity of each incubator is 50 units. Assume that all of incubators do not fail and are scheduled for maintenance.

Each of vaccines has its fixed shelf life of 6 periods. The amounts of initial inventory are 26 and 72 units for vaccine A, and B respectively. Data for demand for vaccines, costs, and production times are given in Table 4.1, 4.2, and 4.3 respectively. The results of the optimal quantities (production, inventory, spoilage, and unmet demands) for products, machine schedule, and the optimal total cost of BPP-SD and BPP-SD-CB are shown in Table 4.4, and 4.5 respectively.

Table 4.1: Demand Data of the Example 4.1

Product	Time Period									
	1	2	3	4	5	6	7	8	9	10
A	10	10	10	5	5	5	10	10	10	3
B	15	15	20	25	15	15	10	10	47	50

Table 4.2: Cost Data of the Example 4.1

- Data of setup costs ($/setup) for the example 4.1

	To (i)	
From (k)	Product A	Product B
Product A	100	150
Product B	250	200

- Data of other costs (in $/unit of product) for the example 4.1

Product (i)	Holding_Cost (HCi)	Production_Cost (PCi)	Unmet-Dem_Cost (UCi)	Disposal_Cost (PCi)
A	2	20	35	3
B	3	24	40	4

Table 4.3: Data of Setup Time and Process Time for the Example 4.1

- Data of setup times for the example 4.1

	To (i)	
From (k)	Product A	Product B
Product A	1	2
Product B	2	1

- Data of processing times for the example 4.1

Product (i)	Process Time (BTi)
A	2
B	3

Table 4.4: Result of BPP-SD for the Example 4.1

		Time Period (t)										Cost	**Subtotal**
		1	2	3	4	5	6	7	8	9	10	($)	**cost ($)**
Production (units)	P_{At}				50							1000	
	P_{Bt}					50				100		3600	**4600**
Inventory at end of period	I_{At}	16	6	0	40	35	30	20	10	0	0	314	
	I_{Bt}	57	42	22	0	35	20	10	0	53	3	726	**1040**
Spoilage (units)	S_{At}				5							15	
	S_{Bt}											0	**15**
Unmet demand (units)	U_{At}			4							3	245	
	U_{Bt}				3							120	**365**
Setup cost	MC1	$100			$150							250	
	MC2	$200				$200						400	**650**
Gantt chart	MC1	s:0-A	p:A		s:A-B		p:B					**Total**	**6670**
	MC2	s:0-B	p:B			s:B-B	p:B						

Table 4.5: Result of BPP-SD-CB for the Example 4.1

		Time Period (t)										Cost	**Subtotal**
		1	2	3	4	5	6	7	8	9	10	($)	**cost ($)**
Production (units)	P_{At}				45							900	
	P_{Bt}					50				97		3528	**4428**
Inventory at end of period	I_{At}	16	6	0	40	35	30	20	10	0	0	314	
	I_{Bt}	57	42	22	0	35	20	10	0	50	0	708	**1022**
Spoilage (units)	S_{At}											0	
	S_{Bt}											0	**0**
Unmet demand (units)	U_{At}			4							3	245	
	U_{Bt}				3							120	**365**
Setup cost	MC1	$100			$150							250	
	MC2	$200				$200						400	**650**
Gantt chart	MC1	s:0-A	p:A		s:A-B		p:B					**Total**	**6465**
	MC2	s:0-B	p:B			s:B-B	p:B						

From the results in Table 4.4 and Table 4.5, the optimal cost of BPP-SD-CB is slightly lower than that of BPP-SD by $205, due to lower costs of spoilage, production, and inventory. In the BPP-SD-CB model, the production lot can be continuous, so we produce the amount of products as needed, while still satisfying the capacity restriction. As a result, no spoilage incurs from overproduction in case of continuous batch size. In this example, the setup costs for the two models are equal due to the same Gantt chart.

CHAPTER V

EXAMINATION OF EFFECT OF FACTORS ON THE SYSTEM PERFORMANCE FOR BPP-SI AND BPP-SD MODELS

5.1 Introduction

In this chapter, we examine the effect of several factors (i.e. type of lot size, product shelf-life) on the system performance for the batch production process for limited shelf life products with sequence-independent (BPP-SI) and sequence-dependent setup times (BPP-SD). We randomly simulated the data for problem instances based on real examples found in the literature or engineers working on those industries.

As the number of time periods, number of machines, and number of products increase, the number of variables and the number of constraints in the model grow exponentially. Consequently the computational time increases dramatically. For our experiment, we choose three different settings of the triple (N, M, T) as listed in Table 5.1.

Table 5.1: Three Configurations of the Batch Production Scheduling Problem

Problem size	N (# of products)	M (# of machines)	T (# of periods)
Small	3	4	15
Medium	4	5	20
Large	5	6	25

The output performance measure (response) is the total cost, comprising the holding cost, spoilage cost, production cost, setup cost, and penalty cost for unmet demand. The machine utilization, fill rate, and the computational time are reported.

Finally we report the numerical results by solving MIP of the generated problem instances for BPP-SI and BPP-SD on discrete time domain in the next two sections. The computational environment is performed on Pentium IV 1.6 GHz with 1 GB RAM using CPLEX 9.1 as solver in GAMS 22.0.

The initial inventory for each product is zero, and there is no demand for a product during its first batch processing time. Therefore, there is no unmet demand during these periods.

5.2 Extension of BPP-SI to the Beer Production

In this section, we extend our BPP-SI model to beer production, especially the fermentation process, which is the most-time consuming step for beer manufacturing. The overview of brewery industry can be found in section 1.1. Note that, we use the term “units” for “gallons” for the beer case, and “fermentation tanks” for “machines”. Each period is one week.

5.2.1 Parameters and Level of Factors

Table 5.2: Distribution and Value of Parameters for Beer Production Case

Symbol	Description	Unit	Distribution / Value
C	Capacity of each machine	gals	1200
PCi	Production cost	$/gal	U[2,4]
HCi	Holding cost	$/gal/week	0.10*PCi/52
DCi	Disposal cost for spoilage	$/gal	0.2*PCi
UCi	Penalty cost for unmet demand	$/gal	1.5*PCi
SCi	Setup cost	$/setup	U[200,400]
STi	Setup time	weeks	0
BTi	Batch processing time	weeks	Round (U[2,4])

The details of each of parameters are explained below:

Capacity information:

In our study, we consider that a microbrewery firm, which has annual production capacity of approximately 465,000 gallons of beer. This firm has fermentation tanks with a capacity of 1,200 gallons per tank.

Cost information:

- We estimate that the production cost is around 20-40% of selling price of beer. On average, the selling price for U.S. beer is around $10/gallon, so the estimated production cost of one gallon of beer is between $2-4. Unit production cost for each beer (in dollars/gallon of beer) is generated from a uniform [2, 4] distribution.
- The sequence-independent setup cost for production of beer i (in dollars /setup) is generated from a uniform [200,400] distribution.
- The unit penalty cost for an unmet demand (lost sale) for each beer (in dollars/gallon) is 150% of unit production cost. The reason for selecting this value higher than the unit production cost is that we would like to satisfy demand as much as possible and avoid the occurrence of lost sale.
- The disposal cost for a unit of spoilage for each beer (in dollars/gallon/week) is 20% of unit production cost.
- The holding cost of a beer is 10% of its production cost divided by 52 (Assuming that a plant operates 52 week per year, the annual interest rate is 10%)

Time information:

- Setup time can be negligible when it is compared to the batch processing time.
- Batch processing time for each product (in weeks) is generated from the rounding of uniform [2, 4] distribution.

Factors:

The input parameters (factors) of interest are demand probability, demand sizes, batch size of item, shelf life of beer. The summary of level of factors is listed in Table 5.3.

Table 5.3: Level of Factors for Beer Production Case

Factor	Description	Unit	Distribution / Value	
			Low level (-)	High level (+)
A	Demand probability		0.6	0.8
B	Demand variation	gal/week	Round (U[250, 350])	Round (U[200, 400])
C	Shelf life time	weeks	10	14
D	Batch size	gals	discrete	continuous

5.2.2 Numerical Result for Beer Production Case

We report the numerical results for solving production scheduling problems for a microbrewery with three configuration settings. By varying the levels of factors, we can observe the effect of such factors on the total cost and computational time.

Table 5.4: Number of Variables and Constraints for Beer Production Case

	Problem size (N,M,T)		
	Small (3,4,15)	Medium (4, 5, 20)	Large (5, 6, 25)
No. of total variables	900	1920	3500
No. of discrete variables	540	1200	2250
No. of constraints	896	1900	3477

Let's introduce the following two terms, which are used to evaluate the system performance:

- **Fill rate** is the proportion of demands that are met from stock.

$$\mathrm{FR_i}\text{: Fill rate for product i (\%)} = \left(1 - \frac{\sum_t U_{it}}{\sum_t D_{it}}\right) * 100$$

Where $\mathrm{U_{it}}$: Unmet demand for product i in period t

$\mathrm{D_{it}}$: Demand for product i in period t

- The **utilization** of a machine j in batch production ($\mathrm{Util_j}$) is the fraction of time in which the machine is busy over the planning period.

The numerical results for BPP-SI problems report the optimal cost, computational time, system performance (i.e. fill rate and utilization) and the breakdown of each of cost components, which are shown in table 5.5-5.7.

Table 5.5: Result Summary for the **Small** BPP-SI Problems for Beer

No	Factor				Comp. time	Optimal costs	Fill rate (%)			Utilization (%)			Costs ($)				
	A	B	C	D	(secs)	($)	Mean	Min	Max	Mean	Min	Max	Spoilage	Inventory	Production	Unmet demand	Setup
1	-	-	-	-	2.00	24626.8	89%	88%	89%	27%	20%	40%	399.6	66.7	19200	3261.0	1699.5
2	-	-	-	+	1.30	21970.6	100%	100%	100%	40%	20%	53%	0.0	45.3	19376	0.0	2549.3
3	-	-	+	-	311.16	23305.3	92%	88%	100%	27%	20%	40%	242.5	81.3	19200	2082.0	1699.5
4	-	-	+	+	47.92	21638.5	100%	100%	100%	35%	13%	60%	0.0	58.2	19367	13.5	2199.8
5	-	+	-	-	0.39	24208.3	90%	88%	91%	27%	13%	40%	320.6	67.7	19200	2920.5	1699.5
6	-	+	-	+	0.64	22137.7	100%	100%	100%	40%	33%	53%	0.0	44.4	19544	0.0	2549.3
7	-	+	+	-	465.73	23073.5	93%	88%	100%	27%	20%	33%	185.7	80.3	19200	1908.0	1699.5
8	-	+	+	+	1579.98	21809.7	100%	100%	100%	35%	27%	40%	0.0	59.9	19532	18.0	2199.8
9	+	-	-	-	3.14	26075.8	62%	42%	88%	18%	0%	40%	6.0	44.1	13200	11626.5	1199.2
10	+	-	-	+	75.86	23512.5	96%	87%	100%	35%	20%	53%	0.0	50.5	20369	828.0	2265.0
11	+	-	+	-	8892.88	24455.4	77%	58%	88%	22%	13%	40%	0.0	58.7	16800	6181.5	1415.2
12	+	-	+	+	24.95	23247.0	100%	100%	100%	35%	27%	40%	0.0	61.0	20921	0.0	2265.0
13	+	+	-	-	6.39	25557.0	63%	44%	88%	18%	0%	40%	12.8	45.0	13200	11100.0	1199.2
14	+	+	-	+	117.25	23100.3	96%	88%	100%	35%	13%	60%	0.0	54.3	20046	735.0	2265.0
15	+	+	+	-	2816.27	23881.2	78%	58%	88%	22%	0%	33%	0.0	62.0	16800	5604.0	1415.2
16	+	+	+	+	20.50	22860.0	100%	100%	100%	35%	0%	53%	0.0	59.0	20536	0.0	2265.0

Table 5.6: Result Summary for the **Medium** BPP-SI Problems for Beer

No	Factor				Comp. time	Best current total cost	Relative opt. gap	Tree size	Costs ($)				
	A	B	C	D	(secs)	($)	(%)	(MB)	Spoilage	Inventory	Production	Unmet demand	Setup
1	-	-	-	-	15888	39813.29	4.73	1870	n/a	n/a	n/a	n/a	n/a
2	-	-	-	+	19125	35964.68	0.64	1878	n/a	n/a	n/a	n/a	n/a
3	-	-	+	-	14759	38348.69	5.94	1872	n/a	n/a	n/a	n/a	n/a
4	-	-	+	+	14644	35964.68	1.27	1878	n/a	n/a	n/a	n/a	n/a
5	-	+	-	-	17193	39569.87	4.33	1873	n/a	n/a	n/a	n/a	n/a
6	-	+	-	+	17057	35975.98	1.05	1879	n/a	n/a	n/a	n/a	n/a
7	-	+	+	-	17085	38127.80	5.35	1872	n/a	n/a	n/a	n/a	n/a
8	-	+	+	+	13112	35978.44	1.29	1875	n/a	n/a	n/a	n/a	n/a
9	+	-	-	-	16678	47646.31	4.51	1868	n/a	n/a	n/a	n/a	n/a
10	+	-	-	+	5640	44541.73	0.50	467	0.0	125	40049.0	63.0	4305.2
11	+	-	+	-	15930	47648.57	5.38	1867	n/a	n/a	n/a	n/a	n/a
12	+	-	+	+	18	44544.00	0.50	1	0.0	127	40049.0	63.0	4305.2
13	+	+	-	-	15506	48005.95	5.21	1874	n/a	n/a	n/a	n/a	n/a
14	+	+	-	+	9579	44426.90	0.50	1046	0.0	123	39880.0	118.5	4305.2
15	+	+	+	-	14344	47915.00	6.42	1872	n/a	n/a	n/a	n/a	n/a
16	+	+	+	+	10	44431.50	0.50	1	0.0	128	39880.0	118.5	4305.2

Table 5.7: Result Summary for the **Large** BPP-SI Problems for Beer

No	Factor				Comp. time (secs)	Best current total cost ($)	Relative opt. gap (%)	Tree size (MB)
	A	B	C	D				
1	-	-	-	-	11743	74609.84	4.94	1883
2	-	-	-	+	13055	71534.28	1.26	1880
3	-	-	+	-	12327	74177.84	4.69	1883
4	-	-	+	+	13057	71152.49	1.19	1881
5	-	+	-	-	11997	75752.77	5.79	1881
6	-	+	-	+	13060	71851.40	0.99	1881
7	-	+	+	-	11363	75270.87	5.20	1882
8	-	+	+	+	13315	71838.73	0.91	1880
9	+	-	-	-	12182	89308.79	4.67	1882
10	+	-	-	+	13878	85890.67	0.86	1880
11	+	-	+	-	12251	88185.44	3.43	1875
12	+	-	+	+	15074	85887.26	0.84	1881
13	+	+	-	-	12714	88736.88	4.54	1883
14	+	+	-	+	12525	85604.03	1.03	1881
15	+	+	+	-	12293	87861.76	3.53	1881
16	+	+	+	+	13274	85634.16	1.05	1880

As shown in Tables 5.5-5.7, we make the following observations:

i) For small problems, it typically takes less than 500 seconds to optimally solve almost all of problem instances. However, in certain scenarios, such as when the high level of demand probability, the high level of shelf-life, and discrete batch size are used (i.e. scenario#11 and scenario#15), the computational time increases greatly up to 8900 seconds. The continuous batch size always results in the better a total cost than does the discrete batch size, while other factors remain unchanged. For example, an example of this can be seen by comparing the scenario#1 versus scenario#2, one can save total cost of $2656.2 by switching from the discrete batch size to the continuous batch size. As the shelf-life of beers increases, with other factors remaining unchanged, the total cost decreases due to the lower spoilage cost and lower cost for unmet demand. For example, when comparing scenario #1 versus scenario #3, one can save a total cost of $1321.5 by an increase in the shelf-life of 4 weeks.

ii) For medium problems, we are able to optimally solve some of problem instances, when the continuous lot size is used. To obtain the optimal solution faster, we set the value of relative optimal gap (optcr) in GAMS to be 0.5%, since the branch and bound converges very slowly to the optimal solution after it reaches around 1-5% of the relative optimal gap. When the size of the branch and bound tree becomes extremely large (around 1880 MB), the CPLEX scenarios run out of memory while solving the problem and finally terminates. Using the continuous lot size produces the solution within 1.5% of relative optimal gap, while using the discrete lot size results in the solution within 4-6% of the relative optimal gap.

iii) As the problem size increases significantly with increase in the number of periods, number of products and number of machines, we cannot optimally solve the large BPP-SI problems to their optimal solutions within a required computational time of one day, even when the continuous lot size is used. The best current solutions using discrete lot size are within 3.4-5.8% of the relative optimal gap, while the best current solutions using continuous lot size are approximately 0.8-1.3% of the relative optimal gap.

iv) When the continuous batch size is used (i.e. the production lot can take on the values, falling between zero and full capacity), there is no occurrence of spoilage.

In summary, to solve the very large BPP-SI problems, one needs an efficient heuristic in order to obtain a fairly good solution in a shorter amount of computational time. We introduce our heuristic approach for solving large BPP-SI problems in the subsequent chapter.

5.3 Extension of BPP-SD to the Vaccine Production

In this section, we extend our BPP-SD model to the vaccine production, in particular the incubation process, which is one of the most-time consuming steps for vaccine production. The overview of vaccine industry can be found in section 1.2.

Note that, we use the term "units" for "liters" for the vaccine case, and "incubators" for "machines". Each period is one day.

5.3.1 Parameters and Level of Factors

Table 5.8: Distribution and Value of Parameters for Vaccine Production Case

Symbol	Description	Unit	Distribution / Value
C	Capacity of each machine	liters	100
PCi	Production cost	$/liter	U[200,400]
HCi	Holding cost	$/liter/day	0.10*PCi/52
DCi	Disposal cost for spoilage	$/liter	0.2*PCi
UCi	Penalty cost for unmet demand	$/liter	1.5*PCi
SCki	Setup cost when swtiching from product k to product i	$/setup	U[500,1000]
STki	Setup time	days	Round (U[1,2])
BTi	Batch processing time	days	Round (U[4,7])

Table 5.9: Level of Factors for Vaccine Production Case

Factor	Description	Unit	Distribution / Value	
			Low level (-)	High level (+)
A	Demand probability		0.6	0.8
B	Demand variation	liter/day	Round (U[17, 23])	Round (U[14, 26])
C	Shelf life time	days	7	12
D	Batch size	liters	discrete	continuous

5.3.2 Numerical Result for Vaccine Production Case

In this section, we report the numerical results for solving production scheduling problems for a manufacturer of vaccines with two configuration settings (small and medium). The large configuration setting is not considered, since its computational time is extremely large exceeding 100,000 seconds.

Table 5.10: Number of Variables and Constraints for Vaccine Production Case

	Problem size (N,M,T)	
	Small (3,4,15)	**Medium (4, 5, 20)**
No. of total variables	2400	5900
No. of discrete variables	1920	5000
No. of constraints	3536	8775

Table 5.11: Result Summary for the **Small** BPP-SD Problems for Vaccine

No	Factor				Comp. time	Optimal costs	Fill rate (%)			Utilization (%)			Costs ($)				
	A	B	C	D	(secs)	($)	Mean	Min	Max	Mean	Min	Max	Spoilage	Inventory	Production	Unmet demand	Setup
1	-	-	-	-	68	147816.1	70%	60%	80%	38%	0%	60%	2082.8	354.2	89873	53715.1	1791.0
2	-	-	-	+	2034	115447.2	93%	80%	100%	62%	53%	80%	0.0	298.0	103603	8109.2	3437.0
3	-	-	+	-	5121	131751.8	78%	69%	85%	38%	0%	60%	181.5	450.9	89873	39455.4	1791.0
4	-	-	+	+	2742	115447.2	93%	80%	100%	62%	53%	80%	0.0	298.0	103603	8109.2	3437.0
5	-	+	-	-	64	142915.1	72%	62%	82%	38%	0%	60%	1831.7	343.0	89873	49076.5	1791.0
6	-	+	-	+	6432	113323.9	94%	82%	100%	62%	53%	80%	0.0	296.2	102335	7255.6	3437.0
7	-	+	+	-	1824	131295.6	79%	68%	87%	38%	0%	60%	453.9	435.4	89873	38742.4	1791.0
8	-	+	+	+	6191	113323.9	94%	82%	100%	62%	53%	80%	0.0	296.2	102335	7255.6	3437.0
9	+	-	-	-	2558	149850.2	68%	62%	72%	37%	0%	53%	389.7	317.5	88116	59335.2	1692.0
10	+	-	-	+	4607	136330.0	86%	72%	100%	60%	53%	80%	0.0	320.2	110869	22282.6	2858.0
11	+	-	+	-	4585	149865.2	68%	62%	72%	37%	0%	53%	389.7	332.5	88116	59335.2	1692.0
12	+	-	+	+	6479	136330.0	86%	72%	100%	60%	53%	80%	0.0	320.2	110869	22282.6	2858.0
13	+	+	-	-	1451	149724.6	68%	62%	72%	37%	0%	53%	874.5	291.6	88116	58750.7	1692.0
14	+	+	-	+	4094	133750.2	86%	72%	100%	60%	53%	80%	0.0	300.3	107548	23043.7	2858.0
15	+	+	+	-	4250	149758.3	68%	62%	72%	37%	0%	53%	874.5	325.2	88116	58750.7	1692.0
16	+	+	+	+	1520	133750.2	86%	72%	100%	60%	53%	80%	0.0	300.3	107548	23043.7	2858.0

Table 5.12: Result Summary for the **Medium** BPP-SD Problems for Vaccine

No	Factor				Relative opt. gap	Tree size	Comp. time	Best current total cost
	A	B	C	D	(%)	(MB)	(secs)	($)
1	-	-	-	-	14.13	451	26818	200627.46
2	-	-	-	+	1.03	138	29065	174779.85
3	-	-	+	-	13.02	290	30304	200647.72
4	-	-	+	+	1.29	175	29626	175159.46
5	-	+	-	-	16.30	771	31336	198471.09
6	-	+	-	+	1.35	244	32045	168545.60
7	-	+	+	-	15.51	636	32398	198511.40
8	-	+	+	+	1.91	257	32880	169534.69
9	+	-	-	-	9.44	69	32596	336478.79
10	+	-	-	+	2.75	39	32590	313493.97
11	+	-	+	-	9.40	342	73472	336562.81
12	+	-	+	+	2.76	73	73970	313532.50
13	+	+	-	-	9.76	379	74281	330072.76
14	+	+	-	+	2.26	182	74587	305164.98
15	+	+	+	-	9.63	191	33362	330072.76
16	+	+	+	+	2.29	130	92854	305164.98

As shown in Tables 5.11-5.12, we make the following observations:

i) For small problems, we can optimally solve all problem instances within the reasonable computational time. The largest computational time is around 6500 seconds. The continuous batch size always results in a lower total cost than does the discrete batch size, while other factors remain unchanged. Moreover, the system performance obtained by using continuous batch size is better than that of discrete batch size, i.e. higher utilization, and higher fill rate. When the shelf-life increases and other factors remain unchanged, the total cost decreases due to the lower spoilage cost and lower cost for unmet demand.

ii) For medium problems, we cannot solve any problem instances to optimality. Since the number of variables and constraints significantly increases with the problem sizes as shown in Table 5.10. The presence of sequence-dependent setup times increases the complexity of problems, so the computational time increases dramatically. There is a large difference between the solutions obtained from discrete batch size and those from continuous batch size. The difference is up to15% of relative optimal gap. The best current solutions using discrete lot size are around 9-16.5% of the relative optimal gap, while the best current solutions using continuous lot size are approximately 1-3% of the relative optimal gap.

In summary, as the problem size increases and there is the presence of sequence-dependent setup time, the computational time for solving BPP-SD problems is much larger than that for solving sequence-independent BPP-SI problems. Efficient heuristics are needed for solving the large BPP-SD problems as exact approach requires a significant amount of computational time.

CHAPTER VI

SOLUTION STRATEGY FOR BPP-SI

6.1 Introduction

In practical applications, the size of batch production scheduling problems is very large. For instance, the world's largest beer producer, Anheuser-Busch Company, has more than 12 products and more than 100 fermentation tanks in twelve U.S. breweries. However, the company might do the production plan by plant separately. When the problem size increases, the computational time for the optimal solution using the branch-and-bound technique is prohibitively large, as the results show in Chapter 5. We therefore must develop efficient heuristics, which result in good feasible solutions. It is also of interest to compute the lower bound on the objective value by solving the LP relaxation of the original problem as a means to evaluate the performance of heuristics. Here we propose five efficient heuristics for solving the batch production scheduling problems for perishable products with sequence-independent setup times (BPP-SI).

6.2 Heuristic Approach

In this section, we develop five heuristics, including three Modified Lot-For-Lot (MLFL) heuristics, Fixed Order Quantity (FOQ) heuristic, and Hybrid heuristic, in order to determine a Master Production Schedule (MPS) for BPP-SI. The amount of solution time significantly increases as the length of planning horizon increases. Finding the MPS by considering the entire planning horizon at the same time is very time-consuming and difficult to implement. We define the term "Time Window when considering a product i to produce in period t" (TW_{it}) as periods of length LT_i starting from period $t+AT_i$ to period $t+AT_i+LT_i-1$. Our heuristics are forward-looking and myopic approaches in the sense that in each decision period t, heuristics consider a part of the entire planning horizon, i.e. TW_{it} in order to identify which product to produce on which machine. Then we move on the next period until the end of planning horizon. Using the concept of time window greatly reduces the solution time. In general, the solution obtained from heuristics is not necessarily optimal.

Before implementing heuristics, we need the following inputs: demand over the planning horizon, initial inventory, the capacity and availability of machine, costs, production times (AT_i), and shelf-life times (LT_i) for each product. We discuss how to deal with the given initial inventory in an efficient way. We should use such inventory to satisfy corresponding demands as much as possible before they become spoiled. From an economic perspective, we should not dispose of the spoiled products in the period when they become spoiled, since this would incur unnecessary inventory cost for holding excess products before disposal. Hence we get rid of excess products right after they come out from machines. This implies that the excess of initial inventory should be disposed of in the first period. Hence, the amount of spoilage in the first period can be readily determined. The updated demand can be computed by subtracting the fulfilled demand by initial inventory from the original demand.

The following notations are used in the heuristics:

TW_{it} = Time window for product i of LT_i periods from period $t+AT_i$ to period $t+AT_i+LT_i-1$

D_{it} = Original demand for product i in period t

D'_{it} = Remaining demand for product i in period t

$CDem_{it}$ = Cumulative remaining demand for product i during time window (TW_{it})

$$= \sum_{k=t+AT_i}^{t+AT_i+LT_i-1} D'_{ik} \qquad \text{-----} \qquad \text{(Equation 6.1)}$$

Ben_{it} = Benefit from producing one batch of product i in period t

= Penalty cost of non-production of one batch - Total cost incurs from production of one batch of product i (i.e. sum of production cost, setup cost, holding cost and spoilage cost)

$Flag_t$ = Economic indicator in period t

$$= \begin{cases} 0 & \text{Start of batch production in period t is worthy} \\ 1 & \text{Start of batch production in period t is unworthy} \\ & \text{when none of products has positive benefit} \end{cases}$$

TF_{it} = The *first* period within time window (TW_{it}) in which the cumulative demand for product i starting from period $t+AT_i$ will be either equal to or great than the capacity of one machine (C)

ΔT_{it} = The length of periods, which makes the cumulative demand for product i starting from period $t+AT_i$ first exceed the capacity of one machine

= Delta Time = $TF_{it} - (t + AT_i) + 1$ ----- (Equation 6.2)

F_t = Set of products in period t, which has ΔT_{it} equal to the smallest value of

$$\Delta Tmin_t = \min\{\Delta T_{it}\}$$

R_{ijt} $= \begin{cases} 1 & \text{if product i is released on machine j in period t} \\ 0 & \text{otherwise} \end{cases}$

B_{jt} $= \begin{cases} 1 & \text{if the machine j is busy in period t} \\ 0 & \text{otherwise} \end{cases}$

$Nmca_t$ = Number of machines available for use in period t

= Total number of machines – Number of machines used in period t

$$= M - \sum_{j=1}^{M} B_{jt} \qquad \text{-----} \qquad \text{(Equation 6.3)}$$

T_last = The last period in which we can release a batch and obtain the batch within planning horizon

= Number of periods in horizon – Minimum of production time of products

$$= T - \min(AT_i) \qquad \text{-----} \qquad \text{(Equation 6.4)}$$

E_{it} = Earliness for product i in period t (periods). That is, if the batch of product i is released in period t and completed in period $t+AT_i$, but such batch will be carried for at least E_{it} periods to satisfy the *first positive* remaining demand of this product in period $\tilde{t}$.

$= \tilde{t}$ {s.t. $D'_{it''} > 0$ for the first time and $\tilde{t} \geq t + AT_i$ } – $(t+AT_i)$

IZE_{it} $= \begin{cases} 1 & \text{if product i has zero earliness of production in period t,} \\ & \text{i.e. } D'_{i,t+AT_i} > 0 \ (E_{it} = 0) \\ 0 & \text{otherwise} \end{cases}$

IC_{it} $= \begin{cases} 1 & \text{if } CDem_{it} \geq C \text{ (High demand)} \\ 0 & \text{otherwise} \end{cases}$

$NumZE_t$ = Number of products with zero earliness of production in period t

$Class_{it}$ = The class of product i in period t. This indicator represents the priority of product to be selected. This classification is used in the Hybrid heuristic.

- If $IC_{it} = 1$ and $IZE_{it} = 1$, then $Class_{it} = 1$
- If $IC_{it} = 0$ and $IZE_{it} = 1$, then $Class_{it} = 2$
- If $IZE_{it} = 0$, then $Class_{it} = 3$

The reason for introducing the variable "T_last" is to reduce the solution time for solving the problem. Intuitively, it is useless to release the batch of product in a period in which the batch will be beyond the horizon.

Before computing the benefit for product i in period t (Ben_{it}), we first need to quantify the amount of unmet demand from non-production. In our proposed heuristics, we assign only one machine to a product in each decision and compare $CDem_{it}$ with capacity of one machine. There are two cases to be considered.

Case 1) $CDem_{it} \geq C$: If we decide not to produce a batch size of C for product i in period t, then this will incur the unmet demand of C units for product i.

Case 2) $CDem_{it} < C$: If we decide not to produce product i with the batch size of $CDem_{it}$ in period t, then this will incur the unmet demand of $CDem_{it}$ units for product i.

Second, we need to compute the amount of spoilage in period $t+AT_i$ ($S_{i,t+ATi}$) in the case of discrete batch size. Recall that no spoilage incurs when the continuous batch size is used.

- If $CDem_{it} \geq C$, $S_{i,t+ATi} = 0$. This is because there is enough demand to satisfy the production from this batch.

- If $CDem_{it} < C$, $S_{i,t+AT_i} = C - CDem_{it}$ for $t+AT_i < T$. As mentioned earlier, the excess production should be eliminated as soon as it is obtained unless there are future demands. However, if $CDem_{it} < C$, $S_{i,t+AT_i} = 0$ for $t+AT_i = T$. The spoilage at the end of horizon is zero because the excess production, which is just obtained from the machine in the last period, is deemed as the ending inventory. Such amount could be used to satisfy the future demand, if the planning horizon is extended.

Third, we describe how to update the status of machine (B_{jt}) after knowing the value of batch release (r_{ijt}). **Fact 1: No interruption on a machine during batch production.** If r_{ijt} =1 (i.e. machine j initially takes on product i in period t), then setting values of B_{jt} to $B_{j,t+AT_i-1}$ to be 1. In other words, the machine j will be busy for period t to period $t+AT_i-1$ for this batch of product i.

To determine the batch size after knowing the value of batch release (r_{ijt}), we define $q_{i,j,t}$ as the batch size of product i released on machine j in period t.

- If $CDem_{it} \geq C$, setting $q_{i,j,t} = C$ for case of discrete or continuous batch size.

- If $CDem_{it} < C$ and the batch size is discrete, setting $q_{i,j,t} = C$

- If $CDem_{it} < C$ and the batch size is continuous, setting $q_{i,j,t} = CDem_{it}$

Next, we illustrate how to compute the cumulative inventory cost of one batch of product i, which is released in period t by the following algorithm.

Algorithm 6.1: Computing the cumulative inventory cost of one batch (CIC_{it})

Step 1: Initialization

ti = $t+AT_i$ where ti is the counter of time period

qleft = $q_{i,j,t} - S_{i,ti}$ where qleft is amount of production left to satisfy future demand

t_step = 0 where t_step is the counter for shelf-life of product

$CIC_{it} = 0$ where CIC_{it} is cumulative inventory cost of one batch

Step 2: Computing the cumulative inventory cost of one batch

While (ti ≤ T) and (t_step < LT_i) and (qleft > 0)

qleft = max{qleft- $D'_{i,ti}$, 0} "computing the amount of production left"

CIC_{it} = $InvCost_{it}$ + qleft*hc_i "computing cumulative inventory cost"

ti = ti+1

t_step = t_step +1

End (for while loop)

In this research, we consider the benefit as one factor in determining which product to produce first. The benefit for product i in period t (Ben_{it}) can be computed using the following formula:

$$Ben_{it} = \begin{cases} uc_i C - pc_i C - rc_i - CIC_{it} \text{ when } CDem_{it} \geq C \text{, disc. or var. batch size (Eq. 6.5a)} \\ uc_i CDem_{it} - pc_i C - rc_i - CIC_{it} - sc_i S_{it}\text{, when } CDem_{it} < C\text{, disc. batch size (Eq.6.5b)} \\ uc_i CDem_{it} - pc_i CDem_{it} - rc_i - CIC_{it}\text{, when } CDem_{it} < C\text{, var. batch size (Eq. 6.5c)} \end{cases}$$

After knowing the product i with batch size of Q released in period t, we keep updating the remaining demand using the following algorithm.

Algorithm 6.2: Updating the remaining demand

Step 1: Initialization

$ti = t+AT_i$ where ti is the counter of time period

$qleft = q_{i,j,t}$ where qleft is amount of production left to satisfy future demand

$t\_step = 0$ where t_step is the counter for shelf-life of product

Step 2: Updating remaining demand during time window

While ($ti \leq T$) and ($t\_step < LT_i$) and ($qleft > 0$)

$x = D'_{i,ti}$ where x = dummy to keep the remaining demand before update

$D'_{i,ti} = \max\{x\text{-}qleft, 0\}$ "updating demand"

$qleft = \max\{qleft\text{-}x, 0\}$ "computing the amount of production left"

$ti = ti+1$

$t\_step = t\_step + 1$

End (for while loop)

6.2.1 Modified Lot-For-Lot (MLFL) Heuristics

In this section, we discuss the criterion used by each of MLFL heuristics, their advantages and disadvantages. When $CDem_{it} < C$, all of three heuristics employ the same criterion, which chooses the product i^* that has the highest positive benefit (Ben_{it}) to produce first. However, when $CDem_{it} \geq C$, three heuristics use different rules to select which product to process first, which will be further explained in Algorithm 6.3.

- **MLFL-A Heuristic** In each period t, this heuristic considers the time interval of LT_i periods starting from period $t+AT_i$ to period $t+AT_i+LT_i-1$, and selects the product i^* that has the highest positive benefit (Ben_{it}) to produce first. We call this criterion as "High Benefit First" (HBF).

This heuristic is simple and easy to implement. The intuition behind this heuristic is to seek cost reduction in each decision period by selecting the product with highest positive benefit to produce first. However, this heuristic has two major drawbacks.

First, it ignores the urgency of actual demand within time window (TW_{it}), before selecting the product to produce first. Making the decision on the batch release without considering the actual demand could lead to a poor MPS. Thus the resulting total cost could be significantly high. The planner furthermore could face the problem of high inventory, since the resulting MPS might recommend the planner to release a product too earlier before needed, especially when demand for products is lumpy. The lumpy demand patterns occur frequently for several reasons: The demand pattern is dominated by large, infrequent customer orders; demand patterns may be a result of outlier or unusual conditions.

To illustrate the effect of lumpy demand on the performance of the heuristic, we consider the example 6.2 A: Finding a MPS for one product, one machine with capacity of C units, and the planning horizon of T periods. The production time is one period and its shelf-life is longer than T periods. No initial inventory for both products is given. Suppose that there is only demand of C units in period T. Obviously, the optimal plan is to release the product in period T-1 in order to have zero inventory cost. However, using the MLFL-A heuristic results in the poor production plan, which instead release the product in period 1. That is, at the beginning of period 1, the $CDem_1$ is C units, and suppose that the benefit of releasing this batch at this period is positive, so it is worth releasing the batch in period 1, even though the actual demand incurs in the last period. As a result, this plan incurs the extra inventory cost for holding C units of the product for T-2 periods.

Second, the heuristic does not take into account the production time of product while choosing the product to produce first. Consider that case in which the production time of the selected product i^* is much longer then that of others. Selecting such product i^* could decrease the number of periods of availability of machine in the future more than selecting others. Therefore, the planner could have insufficient number of machines to satisfy the future demand for other products. As a result, this could cause a huge of unmet demands, and so the total cost dramatically increases. We called this as "the problem of product-blocking on machine".

To see this problem, consider the example 6.2 B: Finding the MPS for 2 products (A and B), one machine with capacity of 50 units, and the planning horizon of 4 periods. The production times are 3 and 1 periods for A and B respectively. Their shelf-life times are 5 periods. No initial inventory for both products is given. Forecast demand for A is 50 units in period 4. Forecast demand for B is 50 units in period 2, 3 and 4. Assume that holding cost for both products is very small, so it is negligible. The setup cost for each product is $50/setup. The production cost for each product is $2/unit. The penalty for unmet demand is $3.5/units for product A and $3.49/unit for product B. Clearly, the benefit of A is higher than B. Using HBF rule selects the product A to produce first in period 1. The machine will be busy from period 1 to 3, so all of demands for product B are unmet.

The MPS from HBF rule is shown in Table below.

		Time Period (t) 1	2	3	4	Cost ($)	**Subtotal cost ($)**
Demand (units)	D_{At}	0	0	0	50		
	D_{Bt}	0	50	50	50		
Production (units)	P_{At}				50	100.0	
	P_{Bt}					-	**100.0**
Inventory at end of period	I_{At}					-	
	I_{Bt}					-	**-**
Spoilage (units)	S_{At}					-	
	S_{Bt}					-	**-**
Unmet demand (units)	U_{At}					-	
	U_{Bt}		50	50	50	523.5	**523.5**
Setup cost ($)	MC1	$50				50.0	**50.0**
Gantt chart	MC1	A	A	A		**Total**	**673.5**

In contrast to the resulting MPS from HBF rule, the optimal MPS is to produce only the product B as shown in the following table.

		Time Period (t) 1	2	3	4	Cost ($)	**Subtotal cost ($)**
Demand (units)	D_{At}	0	0	0	50		
	D_{Bt}	0	50	50	50		
Production (units)	P_{At}					-	
	P_{Bt}		50	50	50	300.0	**300.0**
Inventory at end of period	I_{At}					-	
	I_{Bt}					-	**-**
Spoilage (units)	S_{At}					-	
	S_{Bt}					-	**-**
Unmet demand (units)	U_{At}				50	175.0	
	U_{Bt}					-	**175.0**
Setup cost ($)	MC1	$50	$50	$50		150.0	**150.0**
Gantt chart	MC1	B	B	B		**Total**	**625.0**

In this example, the MPS from MLFL-A heuristic results in higher total cost of $48.5 (7.76%) than the optimal MPS plan.

In summary, using only High Benefit rule to select the product to produce first could lead to poor MPS in certain situations as previously explained. As such, we might consider the urgency of actual demand in time window (TW_{it}) and the length of batch production time as important factors in determining which product and when to produce. The details will be discussed in the next two heuristics.

- **MLFL-B Heuristic** In each period t, this heuristic considers time interval of $\Delta Tmin_t$ periods starting from period $t+AT_i$ to period $t+AT_i+\Delta Tmin_t-1$, finds the set of products in F_t, and selects the product i^* from F_t that yields the highest positive benefit (Ben_{it}) to produce first. We call this criterion as "Small Delta Time first and High Benefit second" (SDT-HB). In case where two more candidates satisfy SDT-HB criterion, the Short Production Time (SPT) is used as the tiebreaking rule.

The MLFL-B heuristic is more complicated than the MLFL-A heuristic, since MLFL-B heuristic takes into account the urgency of actual demand in time window (TW_{it}) and the benefit from producing products in order to select the product to produce first. The heuristic consists of two main steps. The key concept of the first step is to satisfy the high urgent product, whose cumulative demand is no less than capacity during the time window. We use the "Small Delta Time" to indicate the urgency of product. That is, the smaller delta time, the higher urgency. At the end of this first step, it could be more than one product with the same smallest delta time. The key concept of the second step is to reduce total cost by using High Benefit rule to select the product to produce. The advantage of this heuristic is to attempt to satisfy high urgent products, while trying to lower cost by using High Benefit rule. On the other hand, this heuristic might not work well under the following conditions.

- Demand for products is lumpy, as explained in example 6.2A. That is, this heuristic cannot delay the production of the selected product.
- The difference between benefit of product candidates with the smallest delta time is slightly small, but there is big gap between the production times of such products. In this situation, the latter factor might dominate the former factor. The problem of product-blocking on machines could therefore occur as explained in example 6.2 B.

- **MLFL-C Heuristic** In each period t, this heuristic considers time interval of $\Delta Tmin_t$ periods starting from period t+ AT_i to period t+AT_i+$\Delta Tmin_t$ -1, finds the set of products in F_t, and selects a product i* from F_t that has lowest value of production time (AT_i) to produce first. We call this criterion as "Small Delta Time first and Short Production Time second" (SDT-SPT). In case where two more candidates satisfy SDT-SPT criterion, the High Benefit (HB) is used as the tiebreaking rule.

The MLFL-C heuristic is similar to MLFL-B heuristic in that it first uses the urgency of actual demand in time window (TW_{it}) to select the set of candidate products. However, the MLFL-C heuristic then uses the short production time rule to determine the product to produce, not the high benefit. The main reason for the second step is to avoid the problem of product-blocking on machines. That is, selecting the product with shorter production time could save the availability periods of machine to be used for satisfying the future demand. However, this heuristic might not work well under the following conditions.

- Demand for products is lumpy, as explained in example 6.2A.
- The difference between the production times of candidate products with the smallest delta time is slightly small, but there is big gap between benefits of such products. In this situation, the latter factor might dominate the former factor. Hence, the gain from increase in the number of period of machine availability by using short production time rule might be less than the gain of high benefit rule.

Another common problem that occurs when using the three MLFL heuristics is that solving the BPP-SI problem using any one of three heuristics cannot guarantee the following **fact 2. "As the number of machines increases, the total cost of MPS will improve".** In certain instances, solving the BPP-SI problem with the lower number of machines using the proposed heuristics could provide the lower total cost than solving the same problem with higher number of machines, while all other parameters remain unchanged. This is inconsistent with the fact 2. This fact is obviously true. The more resource available, the better performance we could obtain (i.e. lower total cost). To get a better understanding of this problem, consider the example 6.2 C: Finding a MPS for one product (A), and the planning horizon of 3 periods. The production time is 1 period and its shelf-life is 5 periods. No initial inventory is given. There is only demand of 100 units in period 3. The estimated holding cost is \$0.5/unit/period and the setup cost is \$50/setup. The production cost is \$5/unit and the penalty for unmet demand is \$6.5/unit.

- *Suppose that there is one machine with capacity of 50 units.*

The resulting MPS from proposed heuristics is the same as the MPS from branch and bound method, as shown in table below.

		Time Period (t)			Cost	**Subtotal**
		1	2	3	($)	**cost ($)**
Demand (units)	D_{At}	0	0	100		
Production	P_{At}		50	50	500.0	**500.0**
Inventory at end of period	I_{At}		50		25.0	**25.0**
Spoilage (units)	S_{At}				-	
Unmet demand (units)	U_{At}				-	
Setup cost ($)	MC1	$50	$50		100.0	**100.0**
Gantt chart	MC1	A	A		**Total**	**625.0**

- *Suppose that there are two machines with capacity of 50 units.*

The resulting MPS from proposed heuristics is different from that of branch and bound method. The following table represents the MPS from proposed heuristics

		Time Period (t)			Cost	**Subtotal**
		1	2	3	($)	**cost ($)**
Demand (units)	D_{At}	0	0	100		
Production	P_{At}		100		500.0	**500.0**
Inventory at end of period	I_{At}		100		50.0	**50.0**
Spoilage (units)	S_{At}				-	
Unmet demand (units)	U_{At}				-	
Setup cost ($)	MC1	$50			50.0	
	MC2	$50			50.0	**100.0**
Gantt chart	MC1	A			**Total**	**650.0**
	MC2	A				

The following table represents the optimal MPS from branch and bound method.

		Time Period (t)			Cost	**Subtotal**
		1	2	3	($)	**cost ($)**
Demand (units)	D_{At}	0	0	100		
Production	P_{At}			100	500.0	**500.0**
Inventory at end of period	I_{At}				-	**-**
Spoilage (units)	S_{At}				-	
Unmet demand (units)	U_{At}				-	
Setup cost ($)	MC1		$50		50.0	
	MC2		$50		50.0	**100.0**
Gantt chart	MC1		A		**Total**	**600.0**
	MC2		A			

From the results in the tables above, as the number of machines increases from one to two, the total cost obtained by using MLFL heuristics is $50 higher than that obtained by using branch and bound method. However, using the three MLFL heuristics for solving the batch scheduling problem with two machines increases total cost by $25. The reason for increased total cost is that heuristics attempt to utilize available machines immediately when the criterion of selecting product is satisfied. As seen in the resulting MPS from MLFL heuristics, the machines are used in the early periods of planning horizon, and are idle afterward, so the overall utilization of machines could be low. This could also cause a huge inventory cost due to the early releasing batch of products before needed. Furthermore, in reality, the space for holding such inventory might be insufficient. However, the optimal MPS shows that it would be better to delay releasing the batch of products in order to avoid extra inventory cost, which could be alleviated by the next two heuristics. The flow chart of MLFL heuristics is in Appendix A1. The algorithm for MLFL heuristics is as follows:

Algorithm 6.3: MLFL Heuristics

Step 0: Initialization

Set $D'_{i,t} = D_{i,t} \quad \forall i,t$ "Remaining demand is set to be the original demand"

Set $B_{j,t} = 0 \quad \forall j,t$ "Each machine is available for use"

Compute T_last using Equation 6.4

Step 1: For time period t from 1 to T_last

Step 1.1: Compute $Nmca_t$ using Equation 6.3

Step 1.2: Set $Flag_t = 0$

Step 1.3: While ($Nmca_t$ > 0) and ($Flag_t$ = 0), do the following

a) For each product i, with batch production time AT_i and shelf-life time LT_i, compute its cumulative remaining demand during its shelf-life time from period $t+AT_i$ to period $t+AT_i+LT_i-1$ ($CDem_{it}$) using Equation 6.1.

b) Determine which product with $CDem_{it} \geq C$. Compute the total number of products with $CDem_{it} \geq C$ (Nover).

Compute the following quantities:

- Ben_{it} for the MLFL-A heuristic using Equations 6.5a-6.5c
- Ben_{it}, TF_{it}, ΔT_{it} and F_t for the MLFL-B and MLFL-C heuristics.

c) If (Nover > 0)

c1) Determine which product to produce first. Let i^* be the product selected.

Selecting the product depends on the chosen heuristic:

- **MLFL-A uses "High Benefit First" (HBF).**
- **MLFL-B uses "Small Delta Time first and High Benefit second" (SDT-HB).**
- **MLFL-C uses "Small Delta Time first and Short Production Time second" (SDT-SPT).**

c2) Determine which machine to use. Select the available machine with the lowest index to produce the product i^*. Let machine j^* be selected. Update the status of machine.

c3) Compute $Nmca_t$. Determine the batch size. Determine the spoilage, which would incur from this batch.

c4) Update the remaining demand for products $D'_{i,t}$ using Algorithm 6.2.

else (Nover = 0)

d1) Compute the benefit for each product using Equation 6.5a-6.5c

d2) Compute the total number of product with positive benefit (Nben).

d3) **If (Nben > 0) (i.e. production is worth)**

- Determine which product to produce.

Take the product that has the highest positive benefit. Let product i^* be the product is selected.

- Repeat steps c2) to c4)

else (Nben $\leq$ 0)

- Set $Flag_t = 1$, i.e. the production is unworthy.

End (while loop)

End (for loop)

Step 2: Compute the amount of production for each product in each period.

Step 3: Compute the inventory and unmet demand for each product in each period by using mass balance equation. $I_{i,t} - U_{i,t} = I_{i,t-1} + P_{i,t} - S_{i,t} - D_{i,t} \quad \forall i,t$

Step 4: Compute the setup cost for each machine and total setup cost.

Step 5: Compute total production cost, total disposal cost, total penalty cost for unmet demand, and total inventory cost. Compute total cost for production plan and the solution time.

6.2.2 Fixed Order Quantity (FOQ) Heuristic

The basic idea of this heuristic is to attempt to release a batch of full capacity, when needed, so the extra inventory cost from early production is reduced, but it does not take account into the benefit of releasing a batch in selecting the product to produce. Advantages of the heuristic are simple and easy for use. It usually takes very short amount of computational time. At time period t, this heuristic uses demand information in the period t+ATi to decide whether to release the batch of product i. That is, releasing the batch of product i with the lot size of capacity of one machine in period t, when the following three conditions hold: i) There is demand in period t+ATi. ii) There is available machine for use in period t. iii) Such batch will be finished by the end of planning horizon. However, this FOQ heuristic does not consider the benefit of each batch of product in selecting which the product to produce first, poor decisions of unworthy batch release could therefore yield the considerable total cost. Consequently, using FOQ heuristic could lead to prohibitively high inventory and spoilage costs. For example, if the demand of product i in the period $t+AT_i$ is much less than the capacity of one machine and there is no demand in the next several periods, then the large portion of this batch of product i has to be carried for several periods and could become spoiled. The flow chart of FOQ heuristic is in Appendix A2. The algorithm for FOQ heuristic is as follows:

Algorithm 6.4: FOQ Heuristic

Step 0: Initialization

Set $D'_{i,t} = D_{i,t} \quad \forall i,t$ "Remaining demand is set to be the original demand"

Set $B_{j,t} = 0 \quad \forall j,t$ "Each machine is available to use"

Compute T_last using Equation 6.4

Step 1: For period t from 1 to T

Step 1.1: Compute $Nmca_t$ using Equation 6.3

Step 1.2: Set $NumZE_t = 0$

Step 1.3: "Identify whether the product i has zero earliness in period t or not."

For each product i,

- Set $IZE_{it} = 0$

- If $D'_{i,t+AT_i} > 0$ then set IZEit = 1, $NumZE_t = NumZE_t+1$

Step 1.4: "Select the product and assign it to a machine"

While ($Nmca_t > 0$) and ($NumZE_t > 0$), do the following

a) Set Select = 0 "we have not assigned the product to a machine yet"

Set Ipdt = 1 " Ipdt is a counter of product"

Set PID = 0 " PID is a product to be selected"

While (Select = 0) and (Ipdt $\le$ N), do the following

< Select product to produce>

If IZEit = 1, then set PID = Ipdt, Select = 1.

"First product with zero earliness is selected"

Else Ipdt = Ipdt+1 "consider next product"

End (inner while loop)

b) If Select = 1

< Assign the product to a machine>

b1) Determine which machine to use. Select the available machine with the lowest index to produce the product i^*. Let machine j^* be selected. Update the status of machine.

b2) Compute $Nmca_t$. Use the batch size of capacity of one machine. Determine the spoilage, which would incur from this batch.

b3) Update the remaining demand for products $D'_{i,t}$ using Algorithm 6.2.

End (outer while loop)

End (for loop)

Step 2-5: Same as the MLFL heuristics

By using FOQ heuristic to solve the example 6.2 C, we luckily obtain the same MPS as in the optimal MPS, which is better than the results from MLFL heuristics. To clearly grasp the disadvantage of FOQ heuristic, consider the example 6.2 D: Finding a MPS for one product (A), and the planning horizon of 5 periods. The production time is 1 period and its shelf-life is 5 periods. No initial inventory is given. Forecast demands are 1 unit in period 3 and 99 units in period 5. The estimated holding cost is \$1/unit/period and the setup cost is \$50/setup. The production cost is \$5/unit and the penalty for unmet demand is \$7/unit. Suppose that there is one machine with capacity of 50 units.

The resulting MPS by using FOQ heuristic is shown in the following table.

		Time Period (t) 1	2	3	4	5	Cost ($)	**Subtotal cost ($)**
Demand (units)	D_{At}	0	0	1	0	99		
Production	P_{At}			100		100	1,000	**1,000**
Inventory at end of period	I_{At}					1	1	**1**
Spoilage (units)	S_{At}			99			99	**99**
Unmet demand (units)	U_{At}						-	**-**
Setup cost ($)	MC1		$50				50	**50**
Gantt chart	MC1		A				**Total**	**1150**

As seen in Table above, 99 units of spoilage incur in period 3 since the benefit of releasing is not considered in selecting the product to produce for the FOQ heuristic. This problem can be solved by using the Hybrid heuristic.

6.2.3 Hybrid Heuristic (Zero Earliness, High Demand and High Benefit Heuristic)

This heuristic uses three measures including zero earliness, high demand, and high benefit to determine which product to produce first. More specifically, we first select the product with the zero earliness of production by releasing the batch of such product when needed to avoid unnecessary holding cost from early production. If there are more than one product candidates, then selecting the product whose cumulative demand during time window is no less than the capacity of one machine to produce first yields zero spoilage cost for this batch. Finally, using the high benefit determines the product to produce.

To identify the priority of product, we categorize products into one of three classes (1, 2 and 3) by using the factors of earliness of production (E_{it}) and the cumulative demand during time window, which is compared with the capacity of one machine. The priority of the classes from high to low is 1, 2 and 3. That is, selecting a product in class 1 before class 2, and class 2 before class 3. Let $Class_{it}$ be the class of product i in period t.

- If $CDem_{it} \geq C$ and $D'_{i,t+AT_i} > 0$ $(E_{it} = 0)$, then $Class_{it} = 1$

- If $CDem_{it} < C$ and $D'_{i,t+AT_i} > 0$ $(E_{it} = 0)$, then $Class_{it} = 2$

- If $D'_{i,t+AT_i} = 0$ $(E_{it} > 0)$, then $Class_{it} = 3$

The flow chart of Hybrid heuristic is in Appendix A3. The algorithm for Hybrid heuristic is as follows:

Algorithm 6.5: Hybrid Heuristic

Step 0: Initialization

Set $D'_{i,t} = D_{i,t}$ $\forall i,t$ "Remaining demand is set to be the original demand"

Set $B_{j,t} = 0$ $\forall j,t$ "Each machine is available for use"

Compute T_last using Equation 6.4

Step 1: For time period t from 1 to T

Step 1.1: Compute $Nmca_t$ using Equation 6.3

Step 1.2: Set $Flag_t = 0$

Step 1.3: Set $NumZE_t = 0$

Step 1.4: For each product i,

- If $D'_{i,t+AT_i} > 0$ $(E_{it} = 0)$, then $NumZE_t = NumZE_t + 1$

Step 1.5: While ($Nmca_t > 0$) and ($Flag_t = 0$) and ($NumZE_t > 0$), do the following

a) < Determine the class of a product by using IZE_{it} and IC_{it} >

Let c be index for the class of product {1, 2, 3}

Let NP_c be the number of products in class c

- Set $NumZE_t = 0$

- Set $NP_c = 0$

For each product i,

- Set $IZE_{it} = 0$

- Set $IC_{it} = 0$

- If $D'_{i,t+AT_i} > 0\ (E_{it} = 0)$, then

  $IZE_{it} = 1$

  $NumZE_t = NumZE_t+1$

- Compute $CDem_{it}$ using Equation 6.1.

- If $CDem_{it} \geq C$, then set $IC_{it} = 1$

- If $IC_{it} = 1$ and $D'_{i,t+AT_i} > 0\ (E_{it} = 0)$, then

  $Class_{it} = 1$

  $NP_1 = NP_1+1$

- If $IC_{it} = 0$ and $D'_{i,t+AT_i} > 0\ (E_{it} = 0)$, then

  $Class_{it} = 2$

  $NP_2 = NP_2+1$

- If $D'_{i,t+AT_i} = 0\ (E_{it} > 0)$, then

  $Class_{it} = 3$

  $NP_3 = NP_3+1$

b) < Select the class of product to produce according to the priority rule>

Let cid^* be the class of product to be selected.

Iclass = 1

While (NP_{Iclass} = 0) and (Iclass $\leq$ 3), do the following

Iclass = Iclass + 1

End (inner while loop)

cid^* = Iclass

c) < Compute batch size, amount of unmet demand for each product, and associated costs and benefit>

For each product i,

- If discrete batch is used, $batch_i = C$
- If continuous batch is used and $IC_{it} = 1$, $batch_i = C$
- If continuous batch is used and $IC_{it} = 0$, $batch_i = CDem_{it}$
- If $IC_{it} = 1$, $unmet_i = C$
- If $IC_{it} = 0$, $unmet_i = CDem_{it}$
- Compute unmet demand cost, production cost, setup cost.
- Compute spoilage cost (Cspo)

$Cspo_i = Max\ (batch_i - CDem_{it}, 0)$

- Compute cumulative holding cost using Algorithm 6.1
- Compute the benefit of batch releasing of each product (Ben_{it})

d) Select the product with highest positive benefit from the chosen class in b

Let pid^* be the product with highest positive benefit to be selected.

If $cid^* \neq 3$ and $Ben_{cid*,t} > 0$ "Release this batch" d1) Determine which machine to use. Select the available machine with the lowest index to produce the product pid^*. Let machine j^* be selected. Update the status of machine.

d2) Compute $Nmca_t$. Set batch size = $batch_i$. Determine the spoilage, which would incur from this batch.

d3) Update the remaining demand for products $D'_{i,t}$ using Algorithm 6.2.

Else "Not release this batch due to nonzero earliness or unworthy"

- Set Flag = 1

End (outer while loop)

End (for loop)

Step 2-5: Same as the MLFL heuristics

The Hybrid heuristic solves the problem of releasing an unworthy batch, which could incur from the FOQ heuristic, but it is not always true that Hybrid heuristic outperforms the FOQ heuristics in terms of costs. In some instances like the example 6.2 D, Hybrid heuristic yields the lower total cost of MPS than FOQ heuristic by $592.

		Time Period (t)					Cost	**Subtotal**
		1	2	3	4	5	($)	**cost ($)**
Demand (units)	D_{At}	0	0	1	0	99		
Production	P_{At}					100	500	**500**
Inventory at end of period	I_{At}					1	1	**1**
Spoilage (units)	S_{At}						-	**-**
Unmet demand (units)	U_{At}			1			7	**7**
Setup cost ($)	MC1				$50		50	**50**
Gantt chart	MC1				A		**Total**	**558**

However, the Hybrid heuristic could lead to higher total cost than the FOQ heuristic in the small example in the next section. In addition, the Hybrid heuristic is more difficult to implement than the FOQ heuristic due to several factors used in determining the product to produce. It should be noted that the MLFL heuristics seek to utilize machine immediately whenever their conditions are satisfied, even though such batch releasing could cause extra inventory cost from early production. In contrast, FOQ and Hybrid heuristics employ the earliness of production to release a batch of product. That is, releasing a batch of product occurs only when its earliness of production is zero. The resulting MPS from both FOQ and Hybrid heuristics tends to have more idle periods of machines due to the delay of batch release. The planner might face the problem of running short of machines for use in order to satisfy the future demand, if he currently decides not to release a batch of the product because of the rule of releasing a batch of produce by the earliness of production with which he complies, so the resulting MPS might have a large amount of unmet demand. When the increased cost of unmet demand is higher than the benefit from holding cost reduction, the both heuristics do not perform well.

To see this problem, consider the situation in which demand for products is lumpy and number of machines is limited. Suppose that there is only one demand of a single product in the last period of 11 and there is one machine with capacity of 10 units. Assume that such demand is 100 units, which is much more than the capacity of machine. The production time is one period. FOQ and Hybrid heuristics result in the same MPS, which has only one batch release in period 10, so the demand of 90 units in period 11 will be unmet. In case that the shelf-life of product is 10 periods and holding cost is very small, it is worth releasing a batch of 10 units from period 1 to 10 to satisfy all of demand in period 11. Such MPS could be obtained by using MLFL heuristics.

6.3 Small Example

To illustrate the basic concepts of the heuristics, consider a simplified small example. When resources are scarce, finding an efficient scheduling is very important for manufacturers to greatly reduce total cost. In the example, there is only one machine, but it does not have enough capacity to fulfill all of the demands. Using each criterion for a batch release could result in different MPS and total cost.

Consider the BPP-SI problem for 2 products (A and B), 1 machine and the planning horizon of 7 periods. Production times (AT_i) for products A and B are 1 and 2 periods respectively. Shelf-life times for each product are 6 periods. A machine has capacity of 50 units. There is no initial inventory of products at the beginning of planning horizon. Data for demand and costs for products are given in Tables 6.1 and 6.2.

Table 6.1: Demand Data of the Small Example

Product	Time Period						
	1	2	3	4	5	6	7
A	0	0	50	52	51	50	45
B	0	0	0	0	20	40	41

Table 6.2: Cost Data of the Small Example

Product (i)	Setup_Cost (rc_i) ($/setup)	Holding_Cost (hc_i) ($/unit)	Production_Cost (pc_i) ($/unit)	Unmet-Dem_Cost (uc_i) ($/unit)	Disposal_Cost (dc_i) ($/unit)
A	40	0.2	3	4.5	0.4
B	50	0.3	4	6	0.5

Two cases for batch size to be considered are discrete and continuous.

6.3.1 Discrete Batch Size

In this section, we present the optimal solution from the branch and bound method and the numerical result from each heuristic in Tables 6.3-6.8. Details of implementing each of the heuristics for solving the small example are explained in Appendix A4-A8.

Table 6.3: The Optimal MPS for the Small Example with Discrete Batch Size

		Time Period (t)							Cost	**Subtotal**
		1	2	3	4	5	6	7	($)	**cost ($)**
Demand (units)	D_{At}	0	0	50	52	51	50	45		
	D_{Bt}	0	0	0	0	20	40	41		
Production (units)	P_{At}			50	50	50	50	50	750.0	
	P_{Bt}								-	**750.0**
Inventory at end of period	I_{At}							5	1.0	
	I_{Bt}								-	**1.0**
Spoilage (units)	S_{At}								-	
	S_{Bt}								-	**-**
Unmet demand (units)	U_{At}				2	1			13.5	
	U_{Bt}					20	40	41	606.0	**619.5**
Setup cost ($)	MC1		$40	$40	$40	$40	$40		200.0	**200.0**
Gantt chart	MC1		A	A	A	A	A		**Total**	**1570.5**

Table 6.4: MPS from MLFL-A Heuristic

		Time Period (t)							Cost	**Subtotal**
		1	2	3	4	5	6	7	($)	**cost ($)**
Demand	D_{At}	0	0	50	52	51	50	45		
(units)	D_{Bt}	0	0	0	0	20	40	41		
Production	P_{At}		50					50	300.0	
(units)	P_{Bt}				50		50		400.0	**700.0**
Inventory at	I_{At}		50					5	11.0	
end of period	I_{Bt}				50	30	40		36.0	**47.0**
Spoilage	S_{At}								-	
(units)	S_{Bt}								-	**-**
Unmet demand	U_{At}				52	51	50		688.5	
(units)	U_{Bt}							1	6.0	**694.5**
Setup cost ($)	MC1	$40	$50		$50		$40		180.0	**180.0**
Gantt chart	MC1	A	B		B		A		**Total**	**1621.5**

Table 6.5: MPS from MLFL-B Heuristic

		Time Period (t)							Cost	**Subtotal**
		1	2	3	4	5	6	7	($)	**cost ($)**
Demand	D_{At}	0	0	50	52	51	50	45		
(units)	D_{Bt}	0	0	0	0	20	40	41		
Production	P_{At}		50	50			50	50	600.0	
(units)	P_{Bt}					50			200.0	**800.0**
Inventory at	I_{At}		50	50				5	21.0	
end of period	I_{Bt}					30			9.0	**30.0**
Spoilage	S_{At}								-	
(units)	S_{Bt}								-	**-**
Unmet demand	U_{At}				2	51			238.5	
(units)	U_{Bt}						10	41	306.0	**544.5**
Setup cost ($)	MC1	$40	$40	$50		$40	$40		210.0	**210.0**
Gantt chart	MC1	A	A	B		A	A		**Total**	**1584.5**

Table 6.6: MPS from MLFL-C Heuristic

		Time Period (t)							Cost	**Subtotal**
		1	2	3	4	5	6	7	($)	**cost ($)**
Demand	D_{At}	0	0	50	52	51	50	45		
(units)	D_{Bt}	0	0	0	0	20	40	41		
Production	P_{At}		50	50	50	50	50		750.0	
(units)	P_{Bt}								-	**750.0**
Inventory at	I_{At}		50	50	48	47	45		48.0	
end of period	I_{Bt}								-	**48.0**
Spoilage	S_{At}						2		0.8	
(units)	S_{Bt}								-	**0.8**
Unmet demand	U_{At}								-	
(units)	U_{Bt}					20	40	41	606.0	**606.0**
Setup cost ($)	MC1	$40	$40	$40	$40	$40			200.0	**200.0**
Gantt chart	MC1	A	A	A	A	A			**Total**	**1604.8**

Table 6.7: The MPS from FOQ Heuristic

		Time Period (t)							Cost	**Subtotal**
		1	2	3	4	5	6	7	($)	**cost ($)**
Demand	D_{At}	0	0	50	52	51	50	45		
(units)	D_{Bt}	0	0	0	0	20	40	41		
Production	P_{At}			50	50	50	50	50	750.0	
(units)	P_{Bt}								-	**750.0**
Inventory at	I_{At}							5	1.0	
end of period	I_{Bt}								-	**1.0**
Spoilage	S_{At}								-	
(units)	S_{Bt}								-	**-**
Unmet demand	U_{At}				2	1			13.5	
(units)	U_{Bt}					20	40	41	606.0	**619.5**
Setup cost ($)	MC1		$40	$40	$40	$40	$40		200.0	**200.0**
Gantt chart	MC1		A	A	A	A	A		**Total**	**1570.5**

Table 6.8: The MPS from Hybrid Heuristic

		Time Period (t) 1	2	3	4	5	6	7	Cost ($)	**Subtotal cost ($)**
Demand (units)	D_{At}	0	0	50	52	51	50	45		
	D_{Bt}	0	0	0	0	20	40	41		
Production (units)	P_{At}			50			50	50	450.0	
	P_{Bt}					50			200.0	**650.0**
Inventory at end of period	I_{At}							5	1.0	
	I_{Bt}					30			9.0	**10.0**
Spoilage (units)	S_{At}								-	
	S_{Bt}								-	**-**
Unmet demand (units)	U_{At}				52	51			463.5	
	U_{Bt}						10	41	306.0	**769.5**
Setup cost ($)	MC1		$40	$50		$40	$40		170.0	**170.0**
Gantt chart	MC1		A	B	B	A	A		**Total**	**1599.5**

6.3.2 Continuous Batch Size

In this section, we present the optimal solution from the branch and bound method and the numerical result from each heuristic in Tables 6.9-6.14. In Appendix A-9, we explain how to apply the MLFL-A heuristic for solving the small example when the batch size is continuous. Our main focus is on the step which leads to the different production plans between discrete and continuous batch sizes. This situation occurs only when all of the products have $CDem_{it}$ less than C and it is worth releasing a batch in period t (i.e. there exist positive benefit for some products). If that is the case, the batch size is set to be $CDem_{it}$ of product whose benefit is highest in case of the continuous batch size, rather than the full capacity in case of the discrete batch size.

By using similar logic, we can apply MLFL-B and MLFL-C heuristics for solving the small example with continuous batch size. Since the FOQ heuristic only allows the batch size to be zero or full capacity, the resulting MPS is the same for discrete and continuous batch size. The detail of implementing the Hybrid heuristic is omitted.

Table 6.9: Optimal MPS for the Small Example with Continuous Batch Size

		Time Period (t)							Cost	**Subtotal**
		1	2	3	4	5	6	7	($)	**cost ($)**
Demand	D_{At}	0	0	50	52	51	50	45		
(units)	D_{Bt}	0	0	0	0	20	40	41		
Production	P_{At}			50	50	50	50	45	735.0	
(units)	P_{Bt}								-	**735.0**
Inventory at	I_{At}								-	
end of period	I_{Bt}								-	**-**
Spoilage	S_{At}								-	
(units)	S_{Bt}								-	**-**
Unmet demand	U_{At}				2	1			13.5	
(units)	U_{Bt}					20	40	41	606.0	**619.5**
Setup cost ($)	MC1		$40	$40	$40	$40	$40		200.0	**200.0**
Gantt chart	MC1		A	A	A	A	A		**Total**	**1554.5**

Table 6.10: The MPS from MLFL-A Heuristic

		Time Period (t)							Cost	**Subtotal**
		1	2	3	4	5	6	7	($)	**cost ($)**
Demand	D_{At}	0	0	50	52	51	50	45		
(units)	D_{Bt}	0	0	0	0	20	40	41		
Production	P_{At}		50					45	285.0	
(units)	P_{Bt}				50		50		400.0	**685.0**
Inventory at	I_{At}		50						10.0	
end of period	I_{Bt}				50	30	40		36.0	**46.0**
Spoilage	S_{At}								-	
(units)	S_{Bt}								-	**-**
Unmet demand	U_{At}				52	51	50		688.5	
(units)	U_{Bt}							1	6.0	**694.5**
Setup cost ($)	MC1	$40	$50		$50		$40		180.0	**180.0**
Gantt chart	MC1	A	B	B	B	B	A		**Total**	**1605.5**

Table 6.11: The MPS from MLFL-B Heuristic

		Time Period (t)							Cost	**Subtotal**
		1	2	3	4	5	6	7	($)	**cost ($)**
Demand	D_{At}	0	0	50	52	51	50	45		
(units)	D_{Bt}	0	0	0	0	20	40	41		
Production	P_{At}		50	50			50	45	585.0	
(units)	P_{Bt}					50			200.0	**785.0**
Inventory at	I_{At}		50	50					20.0	
end of period	I_{Bt}					30			9.0	**29.0**
Spoilage	S_{At}								-	
(units)	S_{Bt}								-	**-**
Unmet demand	U_{At}				2	51			238.5	
(units)	U_{Bt}						10	41	306.0	**544.5**
Setup cost ($)	MC1	$40	$40	$50		$40	$40		210.0	**210.0**
Gantt chart	MC1	A	A	B		A	A		**Total**	**1568.5**

Table 6.12: The MPS from MLFL-C Heuristic

		Time Period (t)							Cost	**Subtotal**
		1	2	3	4	5	6	7	($)	**cost ($)**
Demand	D_{At}	0	0	50	52	51	50	45		
(units)	D_{Bt}	0	0	0	0	20	40	41		
Production	P_{At}		50	50	50	50			600.0	
(units)	P_{Bt}							41	164.0	**764.0**
Inventory at	I_{At}		50	50	48	47			39.0	
end of period	I_{Bt}								-	**39.0**
Spoilage	S_{At}								-	
(units)	S_{Bt}								-	**-**
Unmet demand	U_{At}						3	45	216.0	
(units)	U_{Bt}					20	40		360.0	**576.0**
Setup cost ($)	MC1	$40	$40	$40	$40	$50			210.0	**210.0**
Gantt chart	MC1	A	A	A	A	B			**Total**	**1589.0**

Table 6.13: The MPS from FOQ Heuristic

		Time Period (t) 1	2	3	4	5	6	7	Cost ($)	**Subtotal cost ($)**
Demand (units)	D_{At}	0	0	50	52	51	50	45		
	D_{Bt}	0	0	0	0	20	40	41		
Production (units)	P_{At}			50	50	50	50	50	750.0	
	P_{Bt}								-	**750.0**
Inventory at end of period	I_{At}							5	1.0	
	I_{Bt}								-	**1.0**
Spoilage (units)	S_{At}								-	
	S_{Bt}								-	**-**
Unmet demand (units)	U_{At}				2	1			13.5	
	U_{Bt}					20	40	41	606.0	**619.5**
Setup cost ($)	MC1		$40	$40	$40	$40	$40		200.0	**200.0**
Gantt chart	MC1		A	A	A	A	A		**Total**	**1570.5**

Table 6.14: The MPS from Hybrid Heuristic

		Time Period (t) 1	2	3	4	5	6	7	Cost ($)	**Subtotal cost ($)**
Demand (units)	D_{At}	0	0	50	52	51	50	45		
	D_{Bt}	0	0	0	0	20	40	41		
Production (units)	P_{At}			50			50	45	435.0	
	P_{Bt}					50			200.0	**635.0**
Inventory at end of period	I_{At}								-	
	I_{Bt}					30			9.0	**9.0**
Spoilage (units)	S_{At}								-	
	S_{Bt}								-	**-**
Unmet demand (units)	U_{At}				52	51			463.5	
	U_{Bt}						10	41	306.0	**769.5**
Setup cost ($)	MC1		$40	$50		$40	$40		170.0	**170.0**
Gantt chart	MC1		A	B	B	A	A		**Total**	**1583.5**

After solving the small example using the branch and bound method, the five heuristics, and solving relaxation of MIP, we summarize the MPS result in Table 6.15. which displays total cost, solution time, and optimality gap of each heuristic. Optimality gap illustrates the solution quality of the heuristics, compared to the optimal solution from the branch and bound method. The resulting Gantt Chart is shown in Figure 6.1.

Table 6.15: Total Cost, Solution Time and Optimality Gap for the Small Example

	Discrete Batch Size			**Continuous Batch Size**		
Method	Total cost ($)	Sol. time (sec)	Optimality gap (%)	Total cost ($)	Sol. time (sec)	Optimality gap (%)
Branch and bound	***1570.50***	***0.03***	***0.00***	***1554.50***	***0.03***	***0.00***
Relaxation MIP	1528.10	0.01	2.70	1528.10	0.01	1.70
MLFL-A heuristic	1621.50	0.10	3.25	1605.50	0.08	3.28
MLFL-B heuristic	1584.50	0.02	0.89	1568.50	0.01	0.90
MLFL-C heuristic	1604.80	0.01	2.18	1589.00	0.01	2.22
FOQ heuristic	1570.50	0.01	0.00	1570.50	0.01	1.03
Hybrid heuristic	1599.50	0.03	1.85	1583.50	0.02	1.87

Figure 6.1: Gantt Chart for the Small Example

	Discrete Batch Size							**Continuous Batch Size**						
Method	**Time Period (t)**							**Time Period (t)**						
	1	**2**	**3**	**4**	**5**	**6**	**7**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Branch & Bound		A	A	A	A	A			A	A	A	A	A	
MLFL-A	A	B		B		A		A	B		B		A	
MLFL-B	A	A	B		A	A		A	A	B		A	A	
MLFL-C	A	A	A	A	A			A	A	A	A	B		
FOQ		A	A	A	A	A			A	A	A	A	A	
Hybrid		A	B		A	A			A	B		A	A	

In this example, all heuristics yield the solution within 4% of the optimality gap. The MLFL-A heuristic takes the longest amount of solution time to solve the problem since it needs to compute the benefit of all products in each decision. The Hybrid heuristic takes the second longest amount of solution time since it takes into account several factors, such as, earliness of production, cumulative demand during the shelf-life time, and the benefit of production of each batch to determine which product to be selected first. The solution time of MLFL-B, MLFL-C, and FOQ heuristics is relatively shorter, since these heuristics do not necessarily require that the benefit of all of products be computed. Overall, the solution time for each heuristic is relatively short. Among five heuristics, the FOQ heuristic provides the lowest total cost when discrete lot size is used, while MLFL-B heuristic yields the lowest total cost when continuous batch size is used for this small problem. However, the FOQ heuristic usually results in a very poor result for the large BPP-SI problem, which is discussed in the subsequent chapter.

6.4 Summary

In summary, our five heuristics are simple, forward-looking methods to help the planner solve practical large BPP-SI problems much faster than the branch and bound method. The heuristics require very short amount of solution time around one second with small relative gap when compared to the lower bound of the MIP relaxation. The relative gap obtained from the heuristics can vary depending on the values of parameters. Also we point out the conditions under which each heuristic might not perform well. In the following chapter, we present a numerical study of applying heuristics to large BPP-SI problems in several industries.

CHAPTER VII

COMPUTATIONAL STUDY FOR INDUSTRIAL LARGE BPP-SI PROBLEMS

7.1 Introduction

After developing five heuristics for BPP-SI problems in the previous chapter, we apply these heuristics to solve some large problems from several industries such as beer, vaccine, and yoghurt. We investigate the effect of several factors including demand probability, demand variation, length of shelf-life and type of lot size on the total cost of the batch production process, and compare the performance of each heuristic by performing the statistical analysis. Also the lower bound on the objective value is calculated by solving the relaxation of the original MIP as a means to evaluate the performance of heuristics. In this study, the data for problem instances are randomly simulated on the basis of real examples found in the literature or engineers working on those industries.

Next we present the computational results of large BPP-SI problems using each heuristic. The computational environment is performed on Pentium IV 1.6 GHz with 1 GB RAM. The relaxation of MIP is solved using CPLEX 9.1 as the primary solver in GAMS 22.0. The BPP-SI problem is solved by each heuristic using Matlab 7.0.4.

7.2 Computational Study for a Beer Manufacturer

In this section, we apply heuristics to solve a large BPP-SI problem for a beer manufacturer in Thailand. The manufacturer has 10 different types of beer and 20 fermentation tanks with equal capacity of 50,000 gallons. The manufacturer wishes to determine the MPS over the next 26 weeks and evaluate the effect of several factors on the system performance.

7.2.1 Parameters and Level of Factors

We summarize the distribution and values of parameters for the large problem of beer production in Table 7.1 as well as the two levels of four factors of interest on the system performance in Table 7.2. Therefore, the number of scenarios is 16. For each scenario, we randomly generate 100 data instances to evaluate the overall performance of heuristics.

Table 7.1: Distribution and Values of Parameters for Beer Production Problem

Symbol	Description	Unit	Distribution / Value
C	Capacity of each fermentation tank	Gals	50000
PCi	Production cost	$/gal	U[2,4]
HCi	Holding cost	$/gal/week	0.10*PCi/52
DCi	Disposal cost for spoilage	$/gal	0.2*PCi
UCi	Penalty cost for unmet demand	$/gal	1.5*PCi
SCi	Setup cost	$/setup	U[1000,2000]
ATi	Batch production time	Weeks	Round (U[2,4])

Table 7.2: Level of Factors for Beer Production Problem

Factor	Description	Unit	Distribution / Value	
			Low level (-)	High level (+)
A	Demand probability		0.6	0.8
B	Demand variation	kgal/week	Round (U[25, 35])	Round (U[20, 40])
C	Shelf life time	weeks	12	16
D	Lot size	gals	Discrete	Continuous

7.2.2. Numerical Result

We present the numerical results of the large BPP-SI problem for beer production obtained by using five heuristics in terms of average and standard deviation of total cost and the average lower bound on the total cost of the LP Relaxation of MIP (RMIP) in Table 7.3. The relative gap of each heuristic compared to the lower bound and the average solution time for each scenario are shown in Table 7.4. Furthermore, the system performance including utilization and fill rate for each scenario is summarized in Table 7.5. The statistical analysis for the effect of factors on the solution quality is shown in Table 7.6. The confidence interval of the multiple comparisons of average total cost between heuristics using Tukey's procedure displays in Table 7.7 in order to compare the performance of heuristics.

Table 7.3: Average and Standard Deviation of Total Cost for Each Heuristic and Average Lower Bound on Total Cost for Beer Production Problem

No	Factor				Average total cost ($)						Std. Dev. of total cost ($)				
	A	B	C	D	RMIP	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid
1	-	-	-	-	14,188,270	15,206,454	14,443,332	15,266,633	31,372,866	14,416,155	1,761,170	1,613,948	1,735,338	3,558,573	1,624,603
2	-	-	-	+	14,188,270	15,037,193	14,271,616	15,092,643	31,372,866	14,246,475	1,746,637	1,602,862	1,723,760	3,558,573	1,608,316
3	-	-	+	-	14,188,270	15,376,762	14,442,161	15,563,767	30,172,834	14,416,155	1,821,636	1,619,231	1,786,434	3,413,594	1,624,603
4	-	-	+	+	14,188,270	15,211,088	14,271,506	15,388,593	30,172,834	14,246,475	1,801,488	1,603,366	1,768,200	3,413,594	1,608,316
5	-	+	-	-	14,202,079	15,226,553	14,463,849	15,275,806	31,396,038	14,438,134	1,806,363	1,651,786	1,780,146	3,566,993	1,660,447
6	-	+	-	+	14,202,079	15,054,687	14,289,812	15,109,721	31,396,038	14,265,686	1,789,062	1,636,877	1,765,850	3,566,993	1,642,520
7	-	+	+	-	14,202,079	15,399,860	14,463,107	15,586,854	30,196,902	14,438,134	1,859,970	1,655,122	1,830,027	3,428,323	1,660,447
8	-	+	+	+	14,202,079	15,230,688	14,289,732	15,418,391	30,196,902	14,265,686	1,839,736	1,637,237	1,810,898	3,428,323	1,642,520
9	+	-	-	-	14,328,842	15,374,889	14,581,245	15,445,561	31,503,690	14,565,245	1,595,524	1,462,299	1,565,791	3,508,915	1,466,879
10	+	-	-	+	14,328,842	15,201,770	14,406,679	15,268,598	31,503,690	14,389,362	1,578,074	1,449,337	1,549,265	3,508,915	1,454,860
11	+	-	+	-	14,328,842	15,555,134	14,581,245	15,738,840	30,309,412	14,565,245	1,652,748	1,462,299	1,610,830	3,369,505	1,466,879
12	+	-	+	+	14,328,842	15,383,678	14,406,679	15,564,247	30,309,412	14,389,362	1,633,037	1,449,337	1,593,706	3,369,505	1,454,860
13	+	+	-	-	14,331,996	15,372,577	14,583,987	15,440,540	31,507,538	14,566,758	1,615,495	1,478,257	1,595,006	3,501,768	1,482,502
14	+	+	-	+	14,331,996	15,207,056	14,415,185	15,272,423	31,507,538	14,395,383	1,602,942	1,468,405	1,578,992	3,501,768	1,471,966
15	+	+	+	-	14,331,996	15,559,788	14,583,987	15,743,426	30,312,958	14,566,758	1,671,265	1,478,257	1,636,278	3,367,762	1,482,502
16	+	+	+	+	14,331,996	15,387,381	14,415,185	15,577,895	30,312,958	14,395,383	1,655,141	1,468,405	1,621,241	3,367,762	1,471,966

Table 7.4: Relative Gap and Average Solution time for Each Heuristic for Beer Production Problem

No	Factor				Relative Gap compared with RMIP (%)					Average solution time (secs)				
	A	B	C	D	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid
1	-	-	-	-	7.18	1.80	7.60	121.12	1.61	0.30	0.32	0.23	0.21	0.27
2	-	-	-	+	5.98	0.59	6.37	121.12	0.41	0.22	0.26	0.23	0.21	0.31
3	-	-	+	-	8.38	1.79	9.69	112.66	1.61	0.32	0.25	0.23	0.20	0.27
4	-	-	+	+	7.21	0.59	8.46	112.66	0.41	0.22	0.24	0.23	0.20	0.27
5	-	+	-	-	7.21	1.84	7.56	121.07	1.66	0.22	0.24	0.24	0.21	0.28
6	-	+	-	+	6.00	0.62	6.39	121.07	0.45	0.22	0.25	0.26	0.21	0.27
7	-	+	+	-	8.43	1.84	9.75	112.62	1.66	0.22	0.27	0.24	0.21	0.27
8	-	+	+	+	7.24	0.62	8.56	112.62	0.45	0.22	0.25	0.24	0.20	0.29
9	+	-	-	-	7.30	1.76	7.79	119.86	1.65	0.23	0.24	0.24	0.23	0.27
10	+	-	-	+	6.09	0.54	6.56	119.86	0.42	0.22	0.28	0.24	0.21	0.27
11	+	-	+	-	8.56	1.76	9.84	111.53	1.65	0.26	0.28	0.24	0.20	0.27
12	+	-	+	+	7.36	0.54	8.62	111.53	0.42	0.23	0.24	0.23	0.21	0.27
13	+	+	-	-	7.26	1.76	7.73	119.84	1.64	0.23	0.25	0.23	0.21	0.27
14	+	+	-	+	6.11	0.58	6.56	119.84	0.44	0.22	0.24	0.25	0.21	0.27
15	+	+	+	-	8.57	1.76	9.85	111.51	1.64	0.24	0.24	0.23	0.21	0.27
16	+	+	+	+	7.36	0.58	8.69	111.51	0.44	0.22	0.25	0.24	0.20	0.27
Average					7.27	1.19	8.13	116.28	1.03	0.24	0.26	0.24	0.21	0.27

Table 7.5: Average Utilization and Fill Rate for Each Heuristic for Beer Production Problem

No	Factor				Utilization (%)					Fill rate (%)				
	A	B	C	D	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid
1	-	-	-	-	63.57	76.52	64.81	93.81	76.22	81.42	97.80	84.86	34.25	97.45
2	-	-	-	+	67.71	80.46	68.91	93.81	79.90	83.47	99.80	86.95	34.25	99.29
3	-	-	+	-	61.68	76.52	60.68	94.54	76.22	79.04	97.81	80.56	39.84	97.45
4	-	-	+	+	65.70	80.41	64.65	94.54	79.90	81.03	99.80	82.57	39.84	99.29
5	-	+	-	-	63.60	76.46	64.84	93.81	76.15	81.42	97.69	84.82	34.25	97.29
6	-	+	-	+	67.71	80.52	69.10	93.81	79.93	83.44	99.73	86.89	34.25	99.14
7	-	+	+	-	61.67	76.46	60.47	94.54	76.15	78.96	97.70	80.29	39.84	97.29
8	-	+	+	+	65.62	80.48	64.56	94.54	79.93	80.95	99.73	82.35	39.84	99.14
9	+	-	-	-	63.80	77.15	64.98	93.77	76.83	80.94	97.79	84.36	34.21	97.38
10	+	-	-	+	68.02	81.16	69.12	93.77	80.59	83.06	99.84	86.50	34.21	99.25
11	+	-	+	-	61.86	77.15	60.78	94.50	76.83	78.49	97.79	79.95	39.79	97.38
12	+	-	+	+	66.04	81.16	64.97	94.50	80.59	80.55	99.84	82.06	39.79	99.25
13	+	+	-	-	63.90	77.05	65.07	93.77	76.72	81.02	97.72	84.44	34.18	97.29
14	+	+	-	+	67.99	81.13	69.21	93.77	80.59	82.99	99.74	86.48	34.18	99.17
15	+	+	+	-	61.85	77.05	60.72	94.50	76.72	78.46	97.72	79.92	39.77	97.29
16	+	+	+	+	65.99	81.13	64.75	94.50	80.59	80.54	99.74	81.91	39.77	99.17

As shown in Tables 7.3-7.5, we make the following observations:

- The averages of total cost from the Hybrid and MLFL-B heuristics are very close to the lower bound of average total cost in every scenario, which ranges from 0.41% to 1.84%. The averages of total cost from the MLFL-A and MLFL-C heuristics result in a fairly good solution with the range of their relative gaps between 6% and 10%.

However, the FOQ heuristic results in very poor result for the large problem. Its relative gap is extremely high up to 121%. The main reason for this is that the FOQ heuristic does not take account to the benefit of releasing the batch of product in selecting the product to produce first. As a result, unworthy production possibly occurs.

- While the branch and bound method cannot optimally solve for this large problem instance within required time limit of one day or 86,400 seconds, the solution time for each heuristic is very small of 0.32 seconds or less for the large BPP-SI problems for beer production for every scenario. Among heuristics, the Hybrid generally takes the longest amount of computational time in almost every scenario due to more information needed to be computed for decision making at each period.
- On average, using continuous lot size slightly reduces the total cost by 1.20% for every heuristic except the FOQ heuristic. As the FOQ heuristic employs the lot size as zero or full capacity for both discrete and continuous cases, there is no difference of total cost.
- Compared to other heuristics, the Hybrid and MLFL-B heuristics result in very high average fill rate around 97-99% and low utilization of 60-70%. Accordingly, the both heuristic provides the good result in terms of total cost. On the other hand, FOQ heuristic results in extremely low average fill rate around 30-40%, so its total cost is extremely higher than that of other heuristics.

Next, we examine how factors of interest may affect the solution quality. The statistical analysis for testing whether the effect of factors on total cost is significant by using analysis of variance with alpha of 0.05 is shown in Table 7.6

Table 7.6: Statistical output for analysis of the effect of factors on total cost of beer production

Analysis of Variance for total cost for a beer manufacturer using MLFL-A

Source	DF	Seq SS	Adj SS	Adj MS	F	P
DemProb	1	1.05E+11	1.05E+11	1.05E+11	19055.67	0.000
DemVar	1	524661930	524661930	524661930	94.80	0.000
ShelfLife	1	1.27E+11	1.27E+11	1.27E+11	22874.19	0.000
LotSize	1	1.15E+11	1.15E+11	1.15E+11	20840.97	0.000
DemProb*DemVar	1	297217600	297217600	297217600	53.70	0.001
DemProb*ShelfLife	1	81802980	81802980	81802980	14.78	0.012
DemProb*LotSize	1	2665056	2665056	2665056	0.48	0.519
DemVar*ShelfLife	1	6874884	6874884	6874884	1.24	0.316
DemVar*LotSize	1	18496	18496	18496	0.00	0.956
ShelfLife*LotSize	1	69960	69960	69960	0.01	0.915
Error	5	2.77E+07	27671719	5534344		
Total	15	3.48E+11				

Analysis of Variance for total cost for a beer manufacturer using MLFL-B

Source	DF	Seq SS	Adj SS	Adj MS	F	P
DemProb	1	1.05E+11	1.05E+11	1.05E+11	19055.67	0.000
DemVar	1	524661930	524661930	524661930	94.80	0.000
ShelfLife	1	1.27E+11	1.27E+11	1.27E+11	22874.19	0.000
LotSize	1	1.15E+11	1.15E+11	1.15E+11	20840.97	0.000
DemProb*DemVar	1	297217600	297217600	297217600	53.70	0.001
DemProb*ShelfLife	1	81802980	81802980	81802980	14.78	0.012
DemProb*LotSize	1	2665056	2665056	2665056	0.48	0.519
DemVar*ShelfLife	1	6874884	6874884	6874884	1.24	0.316
DemVar*LotSize	1	18496	18496	18496	0.00	0.956
ShelfLife*LotSize	1	69960	69960	69960	0.01	0.915
Error	5	2.77E+07	27671719	5534344		
Total	15	3.48E+11				

Analysis of Variance for total cost for a beer manufacturer using MLFL-C

Source	DF	Seq SS	Adj SS	Adj MS	F	P
DemProb	1	1.14E+11	1.14E+11	1.14E+11	65834.26	0.000
DemVar	1	578089892	578089892	578089892	334.55	0.000
ShelfLife	1	3.63E+11	3.63E+11	3.63E+11	210094.8	0.000
LotSize	1	1.17E+11	1.17E+11	1.17E+11	67780.24	0.000
DemProb*DemVar	1	241010100	241010100	241010100	139.48	0.000
DemProb*ShelfLife	1	15046641	15046641	15046641	8.71	0.032
DemProb*LotSize	1	139129	139129	139129	0.08	0.788
DemVar*ShelfLife	1	132618256	132618256	132618256	76.75	0.000
DemVar*LotSize	1	66113161	66113161	66113161	38.26	0.002
ShelfLife*LotSize	1	121452	121452	121452	0.07	0.802
Error	5	8.64E+06	8639737	1727947		
Total	15	5.95E+11				

Analysis of Variance for total cost for a beer manufacturer using FOQ

Source	DF	Seq SS	Adj SS	Adj MS	F	P
DemProb	1	6.1246E+10	6.1246E+10	6.1246E+10	853479.44	0.000
DemVar	1	746218489	746218489	746218489	10398.78	0.000
ShelfLife	1	5.73E+12	5.73E+12	5.73E+12	79867367.2	0.000
LotSize	1	0	0	0	0	1.000
DemProb*DemVar	1	396925929	396925929	396925929	5531.28	0.000
DemProb*ShelfLife	1	26574025	26574025	26574025	370.32	0.000
DemProb*LotSize	1	0	0	0	0	1.000
DemVar*ShelfLife	1	88209	88209	88209	1.23	0.318
DemVar*LotSize	1	0	0	0	0	1.000
ShelfLife*LotSize	1	0	0	0	0	1.00
Error	5	3.59E+05	358801	71760		
Total	15	5.79E+12				

Analysis of Variance for total cost for a beer manufacturer using Hybrid

Source	DF	Seq SS	Adj SS	Adj MS	F	P
DemProb	1	7.5707E+10	7.5707E+10	7.5707E+10	28600.95	0.000
DemVar	1	593507044	593507044	593507044	224.22	0.000
ShelfLife	1	0	0	0	0	1.000
LotSize	1	1.19E+11	1.19E+11	1.19E+11	44885.86	0.000
DemProb*DemVar	1	283181584	283181584	283181584	106.98	0.000
DemProb*ShelfLife	1	0	0	0	0	1.000
DemProb*LotSize	1	6579225	6579225	6579225	2.49	0.176
DemVar*ShelfLife	1	0	0	0	0	1.000
DemVar*LotSize	1	756900	756900	756900	0.29	0.616
ShelfLife*LotSize	1	0	0	0	0	1
Error	5	1.32E+07	13235044	2647009		
Total	15	1.95E+11				

From the statistical analyses in Table 7.6, the following conclusions can be made about the effects of factors on total cost of BPP-SI. Both demand probability and demand variation are significant factors affecting total cost for every heuristic. The type of lot size is also the significant factor affecting total cost for every heuristic except the FOQ. As mentioned earlier, the FOQ heuristic employs the lot size as zero or full capacity for both discrete and continuous. There is evidence of shelf-life effect on total cost for MLFL-A, MLFL-C, and FOQ heuristics.

Next, we present the statistical analyses of the pairwise comparisons of the average total cost between heuristics.

Table 7.7: Pairwise Confidence Intervals for Differences of the Average Total Cost between Heuristics with the Confidence Level of 95% for Beer Production Problem

No	Factor				$C_{Hybrid} - C_{MLFL-B}$		$C_{MLFL-B} - C_{MLFL-A}$		$C_{Hybrid} - C_{MLFL-A}$		$C_{MLFL-A} - C_{MLFL-C}$		$C_{MLFL-C} - C_{FOQ}$		$C_{MLFL-A} - C_{FOQ}$	
	A	B	C	D	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB
1	-	-	-	-	-873,226	818,872	-1,609,170	82,927	-1,636,347	55,750	-906,228	785,869	-16,952,281	-15,260,184	-17,012,461	-15,320,364
2	-	-	-	+	-871,189	820,908	-1,611,626	80,471	-1,636,766	55,331	-901,499	790,598	-17,126,271	-15,434,174	-17,181,721	-15,489,624
3	-	-	+	-	-872,055	820,042	-1,780,649	-88,552	-1,806,656	-114,559	-1,033,054	659,043	-15,455,115	-13,763,018	-15,642,120	-13,950,023
4	-	-	+	+	-871,079	821,018	-1,785,631	-93,534	-1,810,661	-118,564	-1,023,553	668,544	-15,630,290	-13,938,193	-15,807,794	-14,115,697
5	-	+	-	-	-871,764	820,334	-1,608,752	83,345	-1,634,467	57,630	-895,302	796,795	-16,966,280	-15,274,183	-17,015,534	-15,323,437
6	-	+	-	+	-870,175	821,923	-1,610,924	81,173	-1,635,050	57,047	-901,083	791,014	-17,132,366	-15,440,268	-17,187,400	-15,495,303
7	-	+	+	-	-871,022	821,075	-1,782,801	-90,704	-1,807,774	-115,677	-1,033,043	659,054	-15,456,096	-13,763,999	-15,643,091	-13,950,993
8	-	+	+	+	-870,095	822,002	-1,787,004	-94,907	-1,811,051	-118,954	-1,033,752	658,345	-15,624,559	-13,932,462	-15,812,262	-14,120,165
9	+	-	-	-	-862,048	830,049	-1,639,693	52,404	-1,655,692	36,405	-916,720	775,377	-16,904,178	-15,212,081	-16,974,850	-15,282,752
10	+	-	-	+	-863,366	828,732	-1,641,140	50,957	-1,658,457	33,640	-912,876	779,221	-17,081,140	-15,389,043	-17,147,968	-15,455,871
11	+	-	+	-	-862,048	830,049	-1,819,938	-127,841	-1,835,938	-143,841	-1,029,755	662,343	-15,416,620	-13,724,523	-15,600,326	-13,908,229
12	+	-	+	+	-863,366	828,732	-1,823,048	-130,950	-1,840,365	-148,267	-1,026,617	665,480	-15,591,214	-13,899,117	-15,771,783	-14,079,686
13	+	+	-	-	-863,278	828,819	-1,634,638	57,459	-1,651,867	40,230	-914,012	778,085	-16,913,046	-15,220,948	-16,981,009	-15,288,912
14	+	+	-	+	-865,850	826,247	-1,637,920	54,177	-1,657,721	34,376	-911,415	780,682	-17,081,163	-15,389,066	-17,146,530	-15,454,433
15	+	+	+	-	-863,278	828,819	-1,821,850	-129,752	-1,839,079	-146,982	-1,029,686	662,411	-15,415,581	-13,723,483	-15,599,218	-13,907,121
16	+	+	+	+	-865,850	826,247	-1,818,245	-126,147	-1,838,046	-145,949	-1,036,563	655,534	-15,581,111	-13,889,014	-15,771,626	-14,079,529

According to Table 7.7, we can conclude that there is no evidence that average total cost obtained by Hybrid and MLFL-B heuristics are any different since its interval includes zero. However, Hybrid provides slightly lower average total cost than MLFL-B by approximately 0.2%. The statistics indicate that when the factor of shelf-life is at low level, MLFL-B and MLFL-A heuristics provide different average total cost, and Hybrid and MLFL-A heuristics result in the different average total cost. There is no evidence that average total costs obtained by MLFL-A and MLFL-C heuristics are any different, but

MLFL-A yields very slightly lower average total cost than MLFL-C by 0.4-1.4%. The experiment indicates that the MLFL-C and FOQ heuristics provide different average total costs. The MLFL-C heuristic results in the average total cost lower than FOQ heuristic. There is evidence that average total costs obtained by the FOQ and MLFL-A heuristics are different because its interval does not include zero. The MLFL-A heuristic results in the average total cost lower than FOQ heuristic. Ranking heuristics from best to worst is Hybrid, MLFL-B, MLFL-A, MLFL-C and FOQ.

7.3 Computational Study for a Vaccine Manufacturer

In this section, five heuristics are applied to solve a large BPP-SI problem for a vaccine manufacturer. The manufacturer has 8 different types of vaccine and 16 incubators with equal capacity of 120 liters. The manufacturer is interested in finding the efficient MPS over the next 26 weeks and determining key factors, which have significant effect on the system performance.

7.3.1 Parameters and Level of Factors

We summarize the distribution and values of parameters for large problem of vaccine production in Table 7.8 as well as the two levels of four factors of interest on the system performance in Table 7.9. Hence, the total number of scenarios is 16. For each scenario, we randomly generate 100 data instances to evaluate the overall performance of heuristics.

Table 7.8: Distribution and Values of Parameters for Vaccine Production Problem

Symbol	Description	Unit	Distribution / Value
C	Capacity of each machine	Liter	120
PCi	Production cost	$/liter	U[200,500]
HCi	Holding cost	$/liter/week	0.15*PCi/52
DCi	Disposal cost for spoilage	$/liter	0.25*PCi
UCi	Penalty cost for unmet demand	$/liter	1.5*PCi
SCi	Setup cost	$/setup	U[500,1000]
ATi	Batch production time	Weeks	Round (U[2,4])

Table 7.9: Level of Factors for Vaccine Production Problem

Factor	Description	Unit	Distribution / Value	
			Low level (-)	High level (+)
A	Demand probability		0.6	0.8
B	Demand variation	liter/week	Round (U[50, 70])	Round (U[40, 80])
C	Shelf life time	weeks	4	8
D	Lot size	liter	Discrete	Continuous

7.3.2. Numerical result

We present the numerical results of the large BPP-SI problem for vaccine production obtained by using five heuristics. The system performance is evaluated in terms of average and standard deviation of total cost and the average lower bound on the total cost of the LP Relaxation of MIP (RMIP) in Table 7.10. The relative gap of each heuristic compared to the lower bound and the average solution time for each scenario are shown in Table 7.11. Furthermore, other measures of system performance including utilization and fill rate for each scenario are summarized in Table 7.12. The statistical analysis for the effect of factors on the average total cost is shown in Table 7.13. The confidence interval of the multiple comparisons of average total cost between heuristics displays in Table 7.14.

Table 7.10: Average and Standard Deviation of Total Cost for Each Heuristic and Average Lower Bound on Total Cost for Vaccine Production Problem

No	Factor				Average total cost ($)						Std. Dev. of total cost ($)				
	A	B	C	D	RMIP	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid
1	-	-	-	-	3,928,044	4,095,774	4,095,940	4,095,774	10,389,181	3,998,682	400,710	400,688	400,710	1,538,191	394,717
2	-	-	-	+	3,928,044	3,974,269	3,974,266	3,974,269	10,389,181	3,941,802	393,269	393,271	393,269	1,538,191	392,931
3	-	-	+	-	3,928,044	4,164,793	4,048,849	4,145,418	9,704,874	3,997,317	421,289	399,894	418,199	982,682	398,211
4	-	-	+	+	3,928,044	4,077,890	3,978,180	4,057,186	9,704,874	3,941,786	415,614	394,444	409,869	982,682	392,928
5	-	+	-	-	3,926,428	4,198,762	4,198,816	4,198,762	10,388,342	3,999,157	419,219	419,277	419,219	1,542,400	399,186
6	-	+	-	+	3,926,428	3,972,646	3,972,578	3,972,646	10,388,342	3,940,456	402,475	402,404	402,475	1,542,400	402,377
7	-	+	+	-	3,926,428	4,190,599	4,059,113	4,160,789	9,706,111	3,996,779	430,757	405,549	422,752	985,347	405,295
8	-	+	+	+	3,926,428	4,077,126	3,976,471	4,055,628	9,706,111	3,940,435	425,870	403,430	418,939	985,347	402,433
9	+	-	-	-	3,974,919	4,142,918	4,142,918	4,142,918	10,475,819	4,046,642	360,279	360,279	360,279	1,512,470	351,208
10	+	-	-	+	3,974,919	4,020,755	4,020,755	4,020,755	10,475,819	3,988,565	346,560	346,560	346,560	1,512,470	345,984
11	+	-	+	-	3,974,919	4,217,992	4,092,878	4,195,292	9,780,234	4,046,601	374,253	354,966	368,912	967,062	351,313
12	+	-	+	+	3,974,919	4,132,207	4,024,021	4,109,092	9,780,234	3,988,589	365,016	347,968	360,817	967,062	345,975
13	+	+	-	-	3,977,495	4,249,141	4,248,513	4,249,141	10,476,536	4,047,681	378,800	378,372	378,800	1,521,997	356,269
14	+	+	-	+	3,977,495	4,023,255	4,023,095	4,023,255	10,476,536	3,991,255	352,901	352,892	352,901	1,521,997	352,562
15	+	+	+	-	3,977,495	4,244,892	4,102,886	4,211,701	9,784,065	4,047,314	380,307	359,533	373,555	973,782	357,147
16	+	+	+	+	3,977,495	4,135,656	4,026,424	4,111,955	9,784,065	3,991,262	372,023	354,193	367,405	973,782	352,545

Table 7.11: Relative Gap and Average Solution time for Each Heuristic for Vaccine Production Problem

No	Factor				Relative Gap compared with RMIP (%)					Average solution time (secs)				
	A	B	C	D	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid
1	-	-	-	-	4.27	4.27	4.27	164.49	1.80	0.22	0.21	0.19	0.17	0.22
2	-	-	-	+	1.18	1.18	1.18	164.49	0.35	0.17	0.18	0.20	0.15	0.23
3	-	-	+	-	6.03	3.08	5.53	147.07	1.76	0.18	0.18	0.17	0.14	0.18
4	-	-	+	+	3.81	1.28	3.29	147.07	0.35	0.16	0.16	0.16	0.14	0.19
5	-	+	-	-	6.94	6.94	6.94	164.57	1.85	0.15	0.16	0.16	0.14	0.18
6	-	+	-	+	1.18	1.18	1.18	164.57	0.36	0.16	0.16	0.17	0.14	0.19
7	-	+	+	-	6.73	3.38	5.97	147.20	1.79	0.15	0.16	0.16	0.16	0.19
8	-	+	+	+	3.84	1.27	3.29	147.20	0.36	0.16	0.16	0.16	0.14	0.19
9	+	-	-	-	4.23	4.23	4.23	163.55	1.80	0.15	0.16	0.16	0.14	0.18
10	+	-	-	+	1.15	1.15	1.15	163.55	0.34	0.16	0.16	0.16	0.14	0.18
11	+	-	+	-	6.12	2.97	5.54	146.05	1.80	0.15	0.16	0.16	0.15	0.20
12	+	-	+	+	3.96	1.24	3.38	146.05	0.34	0.15	0.17	0.16	0.14	0.20
13	+	+	-	-	6.83	6.81	6.83	163.40	1.76	0.18	0.16	0.16	0.14	0.19
14	+	+	-	+	1.15	1.15	1.15	163.40	0.35	0.16	0.16	0.16	0.14	0.19
15	+	+	+	-	6.72	3.15	5.89	145.99	1.76	0.15	0.16	0.16	0.14	0.18
16	+	+	+	+	3.98	1.23	3.38	145.99	0.35	0.15	0.16	0.16	0.14	0.18
Average					4.26	2.78	3.95	155.29	1.07	0.16	0.17	0.16	0.15	0.19

Table 7.12: Average Utilization and Fill Rate for Each Heuristic for Vaccine Production Problem

No	Factor				Utilization (%)					Fill rate (%)				
	A	B	C	D	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid
1	-	-	-	-	64.37	64.37	64.37	93.35	63.70	97.19	97.19	97.19	21.39	97.66
2	-	-	-	+	82.92	82.92	82.92	93.35	67.62	99.99	99.99	99.99	21.39	99.97
3	-	-	+	-	59.42	63.68	60.07	93.62	63.74	90.76	97.43	92.59	33.88	97.75
4	-	-	+	+	67.90	69.66	68.13	93.62	67.56	93.56	99.99	95.71	33.88	99.97
5	-	+	-	-	65.60	65.61	65.60	93.35	63.54	97.11	97.11	97.11	21.39	97.56
6	-	+	-	+	82.95	82.94	82.95	93.35	67.59	99.99	99.99	99.99	21.39	99.95
7	-	+	+	-	59.66	63.59	60.55	93.62	63.54	90.70	97.28	92.91	33.88	97.63
8	-	+	+	+	67.97	69.74	68.18	93.62	67.51	93.51	99.99	95.68	33.88	99.95
9	+	-	-	-	65.04	65.04	65.04	93.31	64.35	97.15	97.15	97.15	20.79	97.57
10	+	-	-	+	83.22	83.22	83.22	93.31	68.36	99.99	99.99	99.99	20.79	99.98
11	+	-	+	-	59.85	64.30	60.58	93.62	64.35	90.29	97.30	92.31	33.58	97.57
12	+	-	+	+	68.07	70.16	68.39	93.62	68.36	93.06	99.99	95.41	33.58	99.98
13	+	+	-	-	66.55	66.56	66.55	93.31	64.45	97.28	97.29	97.28	20.79	97.68
14	+	+	-	+	83.23	83.21	83.23	93.31	68.39	99.97	99.98	99.97	20.79	99.97
15	+	+	+	-	60.23	64.44	61.14	93.62	64.44	90.33	97.37	92.61	33.56	97.67
16	+	+	+	+	68.08	70.19	68.50	93.62	68.39	93.01	99.99	95.38	33.56	99.97

As illustrated in Tables 7.9-7.11, we make the following observations:

- The averages of total cost from the Hybrid heuristic are very close to the lower bound of average total cost in every scenario, which ranges from 0.34% to 1.85%. The averages of total cost from the MLFL-A, MLFL-B and MLFL-C heuristics result in a fairly good solution with the range of their relative gaps between 1% and 7%. However, the FOQ heuristic results in very poor result for the large problem. Its relative gap is extremely high up to 165%. As the FOQ heuristic does not consider the benefit of releasing the batch of product in selecting the product to produce first, unworthy production possibly occurs.
- Similar to the case for the beer production, the branch and bound method cannot optimally solve for this large problem instance within required time limit of one day or 86,400 seconds, the solution time for each heuristic is very small of 0.23 seconds or less for the large BPP-SI problems for vaccine production for every scenario.

- On average, using continuous lot size slightly reduces the total cost by around 1.41-5.67% for every heuristic except the FOQ heuristic. As the FOQ heuristic employs the lot size as zero or full capacity for both discrete and continuous cases, there is no difference of total cost.
- Among heuristics, the Hybrid, MLFL-A, and MLFL-B, MLFL-C heuristics result in fairly high average fill rate over 90%. The Hybrid heuristic in general has the lowest utilization of 63-69%. Consequently, the Hybrid heuristic provides the lowest average total cost. On the other hand, the FOQ heuristic results in extremely low average fill rate around 20-34%, so its average total cost is considerably higher than that of other heuristics.

Next, we examine how factors of interest may affect the solution quality. The analysis of variance with alpha of 0.05 is performed to test whether the effect of factors on total cost is significant in Table 7.13

Table 7.13: Statistical output for analysis of the effect of factors on total cost of vaccine production

Analysis of Variance for total cost for a vaccine manufacturer using MLFL-A

Source	DF	Seq SS	Adj SS	Adj MS	F	P
DemProb	1	1.08E+10	1.08E+10	1.08E+10	34.27	0.002
DemVar	1	4404943715	4404943715	4404943715	14.03	0.013
ShelfLife	1	1.99E+10	1.99E+10	1.99E+10	63.23	0.001
LotSize	1	7.44E+10	7.44E+10	7.44E+10	236.93	0.000
DemProb*DemVar	1	10025139	10025139	10025139	0.03	0.865
DemProb*ShelfLife	1	41348115	41348115	41348115	0.13	0.732
DemProb*LotSize	1	1517208	1517208	1517208	0.00	0.947
DemVar*ShelfLife	1	1495697613	1495697613	1495697613	4.76	0.081
DemVar*LotSize	1	4171706627	4171706627	4171706627	13.28	0.015
ShelfLife*LotSize	1	5635242158	5635242158	5635242158	17.95	0.008
Error	5	1.57E+09	1570105567	314021113		
Total	15	1.22E+11				

Analysis of Variance for total cost for a vaccine manufacturer using MLFL-B

Source	DF	Seq SS	Adj SS	Adj MS	F	P
DemProb	1	8896120921	8896120921	8896120921	20.02	0.007
DemVar	1	3308809245	3308809245	3308809245	7.45	0.041
ShelfLife	1	8466714218	8466714218	8466714218	19.06	0.007
LotSize	1	6.18E+10	6.18E+10	6.18E+10	139.02	0.000
DemProb*DemVar	1	7026476	7026476	7026476	0.02	0.905
DemProb*ShelfLife	1	6356702	6356702	6356702	0.01	0.909
DemProb*LotSize	1	4329521	4329521	4329521	0.01	0.925
DemVar*ShelfLife	1	2212691041	2212691041	2212691041	4.98	0.076
DemVar*LotSize	1	3231837226	3231837226	3231837226	7.27	0.043
ShelfLife*LotSize	1	9843765048	9843765048	9843765048	22.16	0.005
Error	5	2.22E+09	2221456793	444291359		
Total	15	9.9967E+10				

Analysis of Variance for total cost for a vaccine manufacturer using MLFL-C

Source	DF	Seq SS	Adj SS	Adj MS	F	P
DemProb	1	1.02E+10	1.02E+10	1.02E+10	25.68	0.004
DemVar	1	3695819246	3695819246	3695819246	9.32	0.028
ShelfLife	1	8.54E+09	8.54E+09	8.54E+09	21.52	0.006
LotSize	1	7.22E+10	7.22E+10	7.22E+10	182.14	0.000
DemProb*DemVar	1	10267218	10267218	10267218	0.03	0.878
DemProb*ShelfLife	1	12961800	12961800	12961800	0.03	0.864
DemProb*LotSize	1	3079148	3079148	3079148	0.01	0.933
DemVar*ShelfLife	1	1958128876	1958128876	1958128876	4.94	0.077
DemVar*LotSize	1	3564358655	3564358655	3564358655	8.99	0.030
ShelfLife*LotSize	1	6254081348	6254081348	6254081348	15.77	0.011
Error	5	1.98E+09	1982744511	396548902		
Total	15	1.08E+11				

Analysis of Variance for total cost for a vaccine manufacturer using FOQ

Source	DF	Seq SS	Adj SS	Adj MS	F	P
DemProb	1	2.692E+10	2.692E+10	2.692E+10	499700	0.000
DemVar	1	6115729	6115729	6115729	114	0.000
ShelfLife	1	1.90E+12	1.90E+12	1.90E+12	35211984	0.000
LotSize	1	0	0	0	0	1.000
DemProb*DemVar	1	4305625	4305625	4305625	80	0.000
DemProb*ShelfLife	1	115756081	115756081	115756081	2149	0.000
DemProb*LotSize	1	0	0	0	0	1.000
DemVar*ShelfLife	1	6734025	6734025	6734025	125	0.000
DemVar*LotSize	1	0	0	0	0	1.000
ShelfLife*LotSize	1	0	0	0	0	1.000
Error	5	2.69E+05	269361	53872		
Total	15	1.92E+12				

Analysis of Variance for total cost for a vaccine manufacturer using Hybrid

Source	DF	Seq SS	Adj SS	Adj MS	F	P
DemProb	1	9579270939	9579270939	9579270939	1463.24	0.000
DemVar	1	1185377	1185377	1185377	1.81	0.024
ShelfLife	1	1080041	1080041	1080041	1.65	0.026
LotSize	1	1.30E+10	1.30E+10	1.30E+10	1986.41	0.000
DemProb*DemVar	1	6094727	6094727	6094727	9.31	0.028
DemProb*ShelfLife	1	723776	723776	723776	1.11	0.341
DemProb*LotSize	1	77145	77145	77145	0.12	0.745
DemVar*ShelfLife	1	115770	115770	115770	0.18	0.692
DemVar*LotSize	1	59658	59658	59658	0.09	0.775
ShelfLife*LotSize	1	1073814	1073814	1073814	1.64	0.256
Error	5	3.27E+06	3272002	654400		
Total	15	555893240				

From the statistical analyses in Table 7.13, the following conclusions can be made about the effects of factors on total cost of BPP-SI for vaccine production. Demand probability, demand variation and shelf-life are significant factors affecting total cost for every heuristic. The type of lot size is also the significant factor affecting total cost for every heuristic except the FOQ heuristic. As noted earlier, the FOQ heuristic employs the lot size as zero or full capacity for both discrete and continuous cases.

Next, we show the statistical analyses of the pairwise comparisons of the average total cost between heuristics.

Table 7.14: Pairwise Confidence Intervals for Differences of the Average Total Cost between Heuristics with the Confidence Level of 95% for Vaccine Production Problem

No	Factor				$C_{Hybrid} - C_{MLFL\text{-}B}$		$C_{MLFL\text{-}B} - C_{MLFL\text{-}A}$		$C_{Hybrid} - C_{MLFL\text{-}A}$		$C_{MLFL\text{-}A} - C_{MLFL\text{-}C}$		$C_{MLFL\text{-}C} - C_{FOQ}$		$C_{MLFL\text{-}A} - C_{FOQ}$	
	A	B	C	D	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB
1	-	-	-	-	-396,428	201,911	-299,003	299,336	-396,262	202,078	-299,170	299,170	-6,592,577	-5,994,238	-6,592,577	-5,994,238
2	-	-	-	+	-331,635	266,705	-299,172	299,167	-331,637	266,703	-299,170	299,170	-6,714,082	-6,115,743	-6,714,082	-6,115,743
3	-	-	+	-	-350,702	247,638	-415,113	183,226	-466,645	131,694	-279,795	318,545	-5,858,626	-5,260,286	-5,839,251	-5,240,911
4	-	-	+	+	-335,563	262,776	-398,880	199,460	-435,273	163,066	-278,466	319,873	-5,946,858	-5,348,518	-5,926,154	-5,327,814
5	-	+	-	-	-498,829	99,510	-299,115	299,224	-498,775	99,565	-299,170	299,170	-6,488,750	-5,890,410	-6,488,750	-5,890,410
6	-	+	-	+	-331,292	267,048	-299,238	299,102	-331,360	266,980	-299,170	299,170	-6,714,866	-6,116,527	-6,714,866	-6,116,527
7	-	+	+	-	-361,503	236,836	-430,656	167,684	-492,989	105,350	-269,360	328,980	-5,844,492	-5,246,152	-5,814,682	-5,216,342
8	-	+	+	+	-335,206	263,133	-399,824	198,515	-435,861	162,479	-277,672	320,667	-5,949,652	-5,351,313	-5,928,155	-5,329,815
9	+	-	-	-	-395,445	202,894	-299,170	299,170	-395,445	202,894	-299,170	299,170	-6,632,071	-6,033,731	-6,632,071	-6,033,731
10	+	-	-	+	-331,359	266,980	-299,170	299,170	-331,359	266,980	-299,170	299,170	-6,754,234	-6,155,894	-6,754,234	-6,155,894
11	+	-	+	-	-345,446	252,893	-424,284	174,055	-470,561	127,779	-276,470	321,870	-5,884,111	-5,285,772	-5,861,411	-5,263,072
12	+	-	+	+	-334,602	263,738	-407,356	190,984	-442,788	155,552	-276,055	322,285	-5,970,312	-5,371,972	-5,947,197	-5,348,857
13	+	+	-	-	-500,002	98,338	-299,798	298,542	-500,630	97,710	-299,170	299,170	-6,526,565	-5,928,225	-6,526,565	-5,928,225
14	+	+	-	+	-331,010	267,330	-299,330	299,010	-331,170	267,170	-299,170	299,170	-6,752,451	-6,154,111	-6,752,451	-6,154,111
15	+	+	+	-	-354,741	243,598	-441,177	157,163	-496,748	101,591	-265,978	332,362	-5,871,534	-5,273,195	-5,838,343	-5,240,003
16	+	+	+	+	-334,331	264,009	-408,402	189,938	-443,563	154,777	-275,469	322,870	-5,971,280	-5,372,940	-5,947,579	-5,349,240

According to Table 7.10 and Table 7.14, we can conclude that there is no evidence that average total costs obtained by the Hybrid and MLFL-A, MLFL-B and MLFL-C heuristics are any different since its interval includes zero. The Hybrid heuristic provides the lowest average total cost and the smallest standard deviation of average total cost for all scenarios, while the FOQ heuristic provides the highest average total cost.

7.4 Computational Study for a Yoghurt Manufacturer

In this section, we apply five heuristics to solve a large BPP-SI problem for a yoghurt manufacturer. The manufacturer has 10 different types of yoghurt and 8 incubators with equal capacity of 10000 liters. The manufacturer is interested in finding the efficient MPS over the next 30 days and determining which factors have significant effect on the system performance.

7.4.1 Parameters and Level of Factors

We summarize the distribution and values of parameters for large problem of yoghurt production in Table 7.15 and the two levels of four factors of interest on the system performance in Table 7.16. Hence, the total number of scenarios is 16. For each scenario, we randomly generate 100 data instances to evaluate the overall performance of heuristics.

Table 7.15: Distribution and Values of Parameters for Yoghurt Production Problem

Symbol	**Description**	**Unit**	**Distribution / Value**
C	Capacity of each machine	Liter	10000
PCi	Production cost	$/liter	U[0.4,0.6]
HCi	Holding cost	$/liter/day	0.0004*PCi
DCi	Disposal cost for spoilage	$/liter	0.15*PCi
UCi	Penalty cost for unmet demand	$/liter	1.4*PCi
SCi	Setup cost	$/setup	U[400,800]
ATi	Batch production time	Day	1

Table 7.16: Level of Factors for Yoghurt Production Problem

Factor	Description	Unit	Distribution / Value	
			Low level (-)	**High level (+)**
A	Demand probability		0.6	0.8
B	Demand variation	kliter/day	Round (U[10, 14])	Round (U[8, 16])
C	Shelf life time	Days	10	20
D	Lot size	Liter	Discrete	Continuous

7.4.2 Numerical result

We present the numerical results of the large BPP-SI problem for yoghurt production obtained by using the five heuristics. The system performance is evaluated in terms of average and standard deviation of total cost and the average lower bound on the total cost of the LP Relaxation of the MIP (RMIP) in Table 7.17. The relative gap of each heuristic compared to the lower bound and the average solution time for each scenario are shown in Table 7.18. Furthermore, other performance measures including utilization and fill rate for each scenario are summarized in Table 7.19. The statistical analysis for the effect of factors on the average total cost is shown in Table 7.20. The confidence interval of the multiple comparisons of average total cost between heuristics displays in Table 7.21.

Table 7.17: Average and Standard Deviation of Total Cost for Each Heuristic and Average Lower Bound on Total Cost for Yoghurt Production Problem

No	Factor				Average total cost ($)						Std. Dev. of total cost ($)				
	A	B	C	D	RMIP	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid
1	-	-	-	-	2,083,295	2,208,125	2,145,382	2,205,207	2,378,672	2,138,752	133,006	131,801	133,519	126,294	131,069
2	-	-	-	+	2,083,295	2,203,016	2,145,352	2,199,669	2,378,672	2,136,007	132,719	132,040	133,277	126,294	131,237
3	-	-	+	-	2,083,295	2,224,042	2,145,382	2,221,159	2,378,378	2,138,752	130,358	131,801	131,452	128,021	131,069
4	-	-	+	+	2,083,295	2,218,862	2,145,352	2,216,038	2,378,378	2,136,007	130,222	132,040	131,332	128,021	131,237
5	-	+	-	-	2,084,956	2,208,032	2,162,312	2,204,813	2,380,447	2,139,289	134,812	136,145	135,164	125,780	132,923
6	-	+	-	+	2,084,956	2,202,927	2,161,293	2,199,292	2,380,447	2,136,120	134,504	136,879	135,047	125,780	132,984
7	-	+	+	-	2,084,956	2,224,230	2,162,312	2,221,491	2,379,690	2,139,289	132,344	136,145	133,489	130,798	132,923
8	-	+	+	+	2,084,956	2,219,036	2,161,293	2,216,267	2,379,690	2,136,120	131,938	136,879	133,089	130,798	132,984
9	+	-	-	-	2,093,253	2,221,326	2,159,046	2,217,879	2,389,694	2,152,053	87,007	85,254	86,389	90,862	85,562
10	+	-	-	+	2,093,253	2,216,080	2,159,046	2,212,659	2,389,694	2,149,509	86,833	85,254	86,153	90,862	85,403
11	+	-	+	-	2,093,253	2,237,153	2,159,046	2,234,175	2,389,496	2,152,053	85,309	85,254	86,593	92,122	85,562
12	+	-	+	+	2,093,253	2,231,958	2,159,046	2,228,893	2,389,496	2,149,509	84,975	85,254	86,320	92,122	85,403
13	+	+	-	-	2,094,433	2,208,032	2,162,312	2,204,813	2,380,447	2,139,289	134,812	136,145	135,164	125,780	132,923
14	+	+	-	+	2,094,433	2,202,927	2,161,293	2,199,292	2,380,447	2,136,120	134,504	136,879	135,047	125,780	132,984
15	+	+	+	-	2,094,433	2,224,230	2,162,312	2,221,491	2,379,690	2,139,289	132,344	136,145	133,489	130,798	132,923
16	+	+	+	+	2,094,433	2,219,036	2,161,293	2,216,267	2,379,690	2,136,120	131,938	136,879	133,089	130,798	132,984

Table 7.18: Relative Gap and Average Solution time for Each Heuristic for Yoghurt Production Problem

No	Factor				Relative Gap compared with RMIP (%)					Average solution time (secs)				
	A	B	C	D	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid
1	-	-	-	-	5.99	2.98	5.85	14.18	2.66	0.36	0.42	0.30	0.25	0.34
2	-	-	-	+	5.75	2.98	5.59	14.18	2.53	0.35	0.41	0.29	0.24	0.38
3	-	-	+	-	6.76	2.98	6.62	14.16	2.66	0.26	0.33	0.28	0.23	0.32
4	-	-	+	+	6.51	2.98	6.37	14.16	2.53	0.26	0.30	0.28	0.23	0.32
5	-	+	-	-	5.90	3.71	5.75	14.17	2.61	0.26	0.29	0.28	0.23	0.34
6	-	+	-	+	5.66	3.66	5.48	14.17	2.45	0.27	0.29	0.32	0.23	0.32
7	-	+	+	-	6.68	3.71	6.55	14.14	2.61	0.26	0.29	0.30	0.23	0.33
8	-	+	+	+	6.43	3.66	6.30	14.14	2.45	0.27	0.29	0.29	0.23	0.35
9	+	-	-	-	6.12	3.14	5.95	14.16	2.81	0.44	0.54	0.39	0.32	0.43
10	+	-	-	+	5.87	3.14	5.70	14.16	2.69	0.34	0.38	0.59	0.45	0.64
11	+	-	+	-	6.87	3.14	6.73	14.15	2.81	0.45	0.59	0.57	0.45	0.64
12	+	-	+	+	6.63	3.14	6.48	14.15	2.69	0.52	0.57	0.54	0.39	0.65
13	+	+	-	-	5.42	3.24	5.27	13.66	2.14	0.54	0.56	0.54	0.45	0.50
14	+	+	-	+	5.18	3.19	5.01	13.66	1.99	0.27	0.36	0.48	0.51	0.65
15	+	+	+	-	6.20	3.24	6.07	13.62	2.14	0.55	0.57	0.55	0.44	0.63
16	+	+	+	+	5.95	3.19	5.82	13.62	1.99	0.53	0.57	0.44	0.45	0.67
Average					6.12	3.26	5.97	14.03	2.49	0.37	0.42	0.40	0.33	0.47

Table 7.19: Average Utilization and Fill Rate for Each Heuristic for Yoghurt Production Problem

No	Factor				Utilization (%)					Fill rate (%)				
	A	B	C	D	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid
1	-	-	-	-	84.94	96.56	89.10	96.57	95.76	59.09	67.27	61.97	59.85	66.64
2	-	-	-	+	87.29	96.57	91.42	96.57	96.54	59.94	67.28	62.85	59.85	66.97
3	-	-	+	-	82.60	96.56	83.21	96.57	95.76	57.44	67.27	57.87	59.87	66.64
4	-	-	+	+	84.95	96.57	85.52	96.57	96.54	58.29	67.28	58.71	59.87	66.97
5	-	+	-	-	84.87	96.45	89.10	96.60	95.78	59.06	67.08	61.96	59.85	66.62
6	-	+	-	+	87.29	96.57	91.51	96.60	96.53	59.94	67.14	62.86	59.85	66.93
7	-	+	+	-	82.45	96.45	83.04	96.57	95.78	57.36	67.08	57.77	59.87	66.62
8	-	+	+	+	84.85	96.57	85.45	96.57	96.53	58.24	67.14	58.65	59.87	66.93
9	+	-	-	-	85.24	96.67	89.49	96.67	95.93	58.87	66.85	61.80	59.56	66.28
10	+	-	-	+	87.51	96.67	91.67	96.67	96.65	59.70	66.85	62.61	59.56	66.57
11	+	-	+	-	82.89	96.67	83.55	96.67	95.93	57.24	66.85	57.70	59.57	66.28
12	+	-	+	+	85.32	96.67	85.94	96.67	96.65	58.10	66.85	58.56	59.57	66.57
13	+	+	-	-	84.87	96.45	89.10	96.60	95.78	59.06	67.08	61.96	59.85	66.62
14	+	+	-	+	87.29	96.57	91.51	96.60	96.53	59.94	67.14	62.86	59.85	66.93
15	+	+	+	-	82.45	96.45	83.04	96.57	95.78	57.36	67.08	57.77	59.87	66.62
16	+	+	+	+	84.85	96.57	85.45	96.57	96.53	58.24	67.14	58.65	59.87	66.93

As illustrated in Tables 7.17-7.19, we make the following observations:

- The averages of total cost from the Hybrid heuristic are very close to the lower bound of average total cost in every scenario, which ranges from 1.99% to 2.81%. The averages of total cost from the MLFL-A, MLFL-B and MLFL-C heuristics result in a fairly good solution with the range of their relative gaps around 3-7%. However, the FOQ heuristic results in poor result for the large problem. Its relative gap is approximately 13-14%. As the FOQ heuristic does not consider the benefit of releasing the batch of product in selecting the product to produce first, unworthy production possibly occurs.
- The branch and bound method cannot optimally solve for this large problem instance within required time limit of one day or 86,400 seconds, the solution time for each heuristic is very small of 0.47 seconds or less for the large BPP-SI problems for yoghurt production for every scenario.

- On average, using continuous lot size slightly reduces the total cost up to 0.27% for every heuristic except the FOQ heuristic. As mentioned previously, the FOQ heuristic employs the lot size as zero or full capacity for both discrete and continuous cases, so there is no difference on total cost between discrete and continuous lot size cases.
- The Hybrid and MLFL-B heuristics result in the average fill rate around 66-68%, while other heuristics yields the average fill rate around 57-62%. The average utilization rate obtained by every heuristic is over 82%. The FOQ, Hybrid and MLFL-B heuristics have an average utilization of 96%, while the average utilization rate for the MLFL-A and MLFL-C heuristics is relatively smaller around 82-92%. Consequently, the Hybrid and MLFL-B heuristics outperform others in terms of average total cost.

Next, we examine how factors of interest may affect the solution quality. The analysis of variance with alpha of 0.05 is performed to test whether the effect of factors on total cost is significant in Table 7.20

Table 7.20: Statistical output for analysis of the effect of factors on total cost of yoghurt production

Analysis of Variance for total cost for a yoghurt manufacturer using MLFL-A

Source	DF	Seq SS	Adj SS	Adj MS	F	P
DemProb	1	1.72E+08	1.72E+08	1.72E+08	169538.84	0.000
DemVar	1	169728784	169728784	169728784	167220.48	0.000
ShelfLife	1	1.03E+09	1.03E+09	1.03E+09	1010160.02	0.000
LotSize	1	1.07E+08	1.07E+08	1.07E+08	105172.63	0.000
DemProb*DemVar	1	172081924	172081924	172081924	169538.84	0.000
DemProb*ShelfLife	1	210	210	210	0.21	0.668
DemProb*LotSize	1	1444	1444	1444	1.42	0.286
DemVar*ShelfLife	1	82082	82082	82082	80.87	0.000
DemVar*LotSize	1	1089	1089	1089	1.07	0.348
ShelfLife*LotSize	1	2450	2450	2450	2.41	0.181
Error	5	5.08E+03	5075	1015		
Total	15	1.65E+09				

Analysis of Variance for total cost for a yoghurt manufacturer using MLFL-B

Source	DF	Seq SS	Adj SS	Adj MS	F	P
DemProb	1	187115041	187115041	187115041	4158112.02	0.000
DemVar	1	368332864	368332864	368332864	8185174.76	0.000
ShelfLife	1	0	0	0	0.00	1.000
LotSize	1	1.07E+06	1.07E+06	1.07E+06	23759.02	0.000
DemProb*DemVar	1	187115041	187115041	187115041	4158112.02	0.000
DemProb*ShelfLife	1	0	0	0	0.00	1.000
DemProb*LotSize	1	225	225	225	5.00	0.076
DemVar*ShelfLife	1	0	0	0	0.00	1.000
DemVar*LotSize	1	1008016	1008016	1008016	22400.36	0.000
ShelfLife*LotSize	1	0	0	0	0	1
Error	5	2.25E+02	225	45		
Total	15	744640568				

Analysis of Variance for total cost for a yoghurt manufacturer using MLFL-C

Source	DF	Seq SS	Adj SS	Adj MS	F	P
DemProb	1	1.66E+08	1.66E+08	1.66E+08	22723.77	0.000
DemVar	1	168694638	168694638	168694638	23095.69	0.000
ShelfLife	1	1.09E+09	1.09E+09	1.09E+09	149447.94	0.000
LotSize	1	1.14E+08	1.14E+08	1.14E+08	15565.68	0.000
DemProb*DemVar	1	165978131	165978131	165978131	22723.77	0.000
DemProb*ShelfLife	1	2730	2730	2730	0.37	0.568
DemProb*LotSize	1	1541	1541	1541	0.21	0.665
DemVar*ShelfLife	1	376689	376689	376689	51.57	0.001
DemVar*LotSize	1	6765	6765	6765	0.93	0.380
ShelfLife*LotSize	1	56288	56288	56288	7.71	0.039
Error	5	3.65E+04	36521	7304		
Total	15	1706417710				

Analysis of Variance for total cost for a yoghurt manufacturer using FOQ

Source	DF	Seq SS	Adj SS	Adj MS	F	P
DemProb	1	122544900	122544900	122544900	265939.45	0.000
DemVar	1	63728289	63728289	63728289	138299.24	0.000
ShelfLife	1	1.01E+06	1.01E+06	1.01E+06	2183.18	0.000
LotSize	1	0	0	0	0	1.000
DemProb*DemVar	1	122544900	122544900	122544900	265939.45	0.000
DemProb*ShelfLife	1	2304	2304	2304	5	0.076
DemProb*LotSize	1	0	0	0	0	1.000
DemVar*ShelfLife	1	261121	261121	261121	566.67	0.000
DemVar*LotSize	1	0	0	0	0	1.000
ShelfLife*LotSize	1	0	0	0	0	1.00
Error	5	2.30E+03	2304	461		
Total	15	31008982				

Analysis of Variance for total cost for a yoghurt manufacturer using Hybrid

Source	DF	Seq SS	Adj SS	Adj MS	F	P
DemProb	1	179600202	179600202	179600202	88908.79	0.000
DemVar	1	162600752	162600752	162600752	80493.43	0.000
ShelfLife	1	0	0	0	0	1.000
LotSize	1	3.38E+07	3.38E+07	3.38E+07	16730.67	0.000
DemProb*DemVar	1	179600202	179600202	179600202	88908.79	0.000
DemProb*ShelfLife	1	0	0	0	0	1.000
DemProb*LotSize	1	10100	10100	10100	5	0.076
DemVar*ShelfLife	1	0	0	0	0	1.000
DemVar*LotSize	1	275100	275100	275100	136.18	0.000
ShelfLife*LotSize	1	0	0	0	0	1
Error	5	1.01E+04	10100	2020		
Total	15	555893240				

From the statistical analyses in Table 7.20, the following conclusions can be made about the effects of factors on total cost of BPP-SI for yoghurt production. Demand probability and demand variation are significant factors affecting total cost for every heuristic. Shelf-life is a significant factor affecting total cost for MLFL-A, MLFL-C and FOQ. The type of lot size is a significant factor affecting total cost for every heuristic except the FOQ heuristic. Next, we present the statistical analyses of the pairwise comparisons of the average total cost between heuristics.

Table 7.21: Pairwise Confidence Intervals for Differences of the Average Total Cost between Heuristics with the Confidence Level of 95% for Yoghurt Production Problem

No	Factor				C_{Hybrid} - $C_{MLFL\text{-}B}$		C_{Hybrid} - $C_{MLFL\text{-}A}$		$C_{MLFL\text{-}A}$ - $C_{MLFL\text{-}C}$		$C_{MLFL\text{-}C}$ - C_{FOQ}		$C_{MLFL\text{-}A}$ - C_{FOQ}	
	A	B	C	D	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB
1	-	-	-	-	-57,259	43,998	-120,002	-18,744	-47,711	53,547	-224,094	-122,837	-221,176	-119,919
2	-	-	-	+	-59,974	41,284	-117,639	-16,381	-47,281	53,976	-229,633	-128,375	-226,285	-125,027
3	-	-	+	-	-57,259	43,998	-135,919	-34,661	-47,746	53,512	-207,848	-106,590	-204,965	-103,707
4	-	-	+	+	-59,974	41,284	-133,484	-32,226	-47,805	53,453	-212,969	-111,711	-210,145	-108,887
5	-	+	-	-	-73,652	27,606	-119,372	-18,114	-47,410	53,847	-226,262	-125,005	-223,044	-121,786
6	-	+	-	+	-75,802	25,456	-117,436	-16,178	-46,994	54,264	-231,784	-130,526	-228,148	-126,891
7	-	+	+	-	-73,652	27,606	-135,570	-34,313	-47,890	53,368	-208,827	-107,569	-206,088	-104,831
8	-	+	+	+	-75,802	25,456	-133,545	-32,287	-47,860	53,398	-214,051	-112,793	-211,282	-110,024
9	+	-	-	-	-57,622	43,636	-119,902	-18,644	-47,182	54,076	-222,444	-121,186	-218,997	-117,739
10	+	-	-	+	-60,166	41,092	-117,200	-15,942	-47,208	54,050	-227,664	-126,407	-224,243	-122,985
11	+	-	+	-	-57,622	43,636	-135,728	-34,471	-47,651	53,607	-205,951	-104,693	-202,972	-101,715
12	+	-	+	+	-60,166	41,092	-133,078	-31,820	-47,564	53,694	-211,232	-109,975	-208,167	-106,910
13	+	+	-	-	-73,652	27,606	-119,372	-18,114	-47,410	53,847	-226,262	-125,005	-223,044	-121,786
14	+	+	-	+	-75,802	25,456	-117,436	-16,178	-46,994	54,264	-231,784	-130,526	-228,148	-126,891
15	+	+	+	-	-73,652	27,606	-135,570	-34,313	-47,890	53,368	-208,827	-107,569	-206,088	-104,831
16	+	+	+	+	-75,802	25,456	-133,545	-32,287	-47,860	53,398	-214,051	-112,793	-211,282	-110,024

According to Table 7.21, we can conclude that there is no evidence that average total cost obtained by the Hybrid and MLFL-B heuristics are any different since its interval includes zero. Moreover, it is clear that average total costs obtained by the Hybrid and MLFL-A heuristics are significantly different since its interval excludes zero. For every scenario, the Hybrid heuristic results in a lower average total cost than the MLFL-A heuristic. Furthermore, the statistics indicate that there is no evidence that average total costs obtained by the MLFL-A and MLFL-C heuristics are different. However, there is evidence that average total costs obtained by the MLFL-A and FOQ heuristics are different. The MLFL-A heuristic usually yields lower average total cost than the FOQ heuristic. The experiment also indicates that the MLFL-C and FOQ heuristics provide different average total costs. MLFL-C heuristic typically results in the slightly lower average total cost than the FOQ heuristic. Ranking heuristics from best to worst is Hybrid, MLFL-B, MLFL-A, MLFL-C and FOQ.

CHAPTER VIII

PERFORMANCE OF HEURISTICS FOR BPP-SI PROBLEMS

This chapter investigates the performance of heuristics for BPP-SI problems by a computational study. In the preceding chapter, it is very interesting that some of the heuristics, such as Hybrid and MLFL-B turn out to work very well for large BPP-SI problem instances due to a very small relative gap of around 2-3% with respect to the lower bound on total cost obtained by solving the relaxation model. This implies that solutions from heuristics are very close to the optimal solutions for those manufacturing settings. To see how well the heuristics perform under other conditions, we further examine the effect of changing parameters, such as the penalty of unmet demand, setup cost and the number of machines on the total cost. We expect that there are certain manufacturing settings in which these heuristics might not work well. For example, when the number of machines is limited or insufficient to be used for satisfying all demands, the planner might have difficulty in finding an efficient MPS. If that is the case, using the heuristics could lead to decisions on production scheduling that are farther from optimum. To evaluate the performance of heuristics, we seek to solve a large BPP-SI problem to optimality by the branch and bound method to obtain the optimal benchmark solution, solve the problem by our heuristics, and then compute the optimality gap for each heuristic with respect to the optimal solution. However, due to the significantly large number of binary variables over 5,000, a huge number of constraints over 8,000 in the large BPP problem and the complexity of the problem characteristics, we cannot optimally solve the problem within the target time of one day. Instead of computing the optimality gap, we use the best cut, which is the best fractional solution obtained by branch and bound method, as the benchmark of lower bound on the total cost, so the relative gap for each of heuristic with respect to this lower bound is computed.

The computational result of the effect of parameter control on the total cost for a vaccine manufacturer is given in section 8.1. The performance of heuristics for a large BPP-SI problem in vaccine production is shown in section 8.2. Recall that the manufacturer has 8 products and the length of planning horizon is 26 periods.

8.1 Computational Result of the Effect of Parameters on the Total Cost for a Vaccine Manufacturer

The difficulty of a lot-sizing and scheduling problem may depend on several criteria. Therefore, it is interesting to explore the performance of our heuristic approach applied to BPP-SI problems with different values of parameters. In the preceding chapter, we examine the effect of several factors including demand probability, demand variation, length of shelf-life and type of lot size on the total cost. In this section, we investigate the effect of following control parameters, such as the penalty of unmet demand, setup cost and the number of machines on the total cost for a vaccine manufacturer. Also we compare the performance of each heuristic by performing the statistical analysis.

8.1.1 Parameters and Level of Factors

In this study, we consider the same BPP-SI problem for a vaccine manufacturer in section 7.3, focus on the case where all of the low levels for each of four factors in Table 7.9 are selected, and use the same values of parameters listed in Table 7.8 except parameters, which are given in the following Table 8.1.

Table 8.1: Values of Parameters for the Large Problem of Vaccine Production

Symbol	Description	Unit	Low	Medium	High	Very high
UCi	Penalty cost for unmet demand	$/liter	1000	3000	6000	-
SCi	Setup cost	$/setup	U[5k,10k]	U[10k,20k]	U[20k,40k]	-
M	Number of machines		4	8	12	16

8.1.2 Numerical result

We present the numerical results of the large BPP-SI problem for vaccine production for 36 scenarios using five heuristics. For each scenario, we randomly generate 10 data instances to compute the average performance measure. The percentage of average relative gap for each heuristic with respect to the lower bound on the total cost of the Relaxation of MIP (RMIP) is shown in Table 8.2. The percentage of average utilization result is summarized in Table 8.3. The percentage of average fill rate is shown in Table 8.4. Table 8.5 shows the average percentage of the number of binary variables w and r, whose values fall between 0.3 and 0.7 from the solution of RMIP.

For simplicity, we define the set of scenarios as follows:

- Case 8A: Scenarios where penalty cost of unmet demand is $1000/liter, but setup cost and the number of machines vary.
- Case 8B: Scenarios where setup cost is uniformly distributed on (5000, 10000), but penalty cost of unmet demand and the number of machines vary.
- Case 8C: Scenarios where the number of machines is 4, but penalty cost of unmet demand and the number of machines vary.
- Case 8D: Scenarios where the number of machines is 16, but penalty cost of unmet demand and the number of machines vary.

Table 8.2: Percentage of Average Relative Gap for Each Heuristic for the Large Problem of Vaccine Production

Penalty cost of unmet demand ($/liter)	# MC	Setup cost ($/setup)														
		U(5k,10k)					U(10k,20k)					U(20k,40k)				
		MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid
1000	4	79.82	83.69	84.06	132.45	80.14	61.47	65.27	65.39	107.40	61.93	36.18	39.41	39.22	72.66	37.23
1000	8	35.39	37.67	36.08	165.62	35.18	27.77	29.15	28.22	141.07	27.33	16.65	17.94	17.18	107.02	16.62
1000	12	12.45	10.86	12.29	201.41	7.71	10.61	9.41	10.50	177.03	6.35	6.58	6.19	6.59	143.23	4.39
1000	16	29.33	29.33	29.33	238.21	3.95	21.88	21.88	21.88	213.89	3.51	9.53	9.53	9.53	180.15	2.82
3000	4	369.94	374.54	379.38	520.21	370.49	314.16	319.94	322.86	445.56	314.84	238.51	243.42	244.37	341.93	241.87
3000	8	162.77	162.59	162.69	547.13	160.26	140.60	137.27	139.64	473.61	136.50	110.01	107.11	110.29	371.62	107.83
3000	12	29.31	22.63	29.32	580.62	23.45	26.63	20.70	26.63	507.57	20.28	22.41	17.78	22.39	406.24	15.92
3000	16	41.39	41.39	41.39	617.16	5.57	40.52	40.52	40.52	544.19	5.38	39.08	39.08	39.08	442.98	4.97
6000	4	803.07	808.29	819.41	1098.95	803.83	691.31	699.97	706.59	950.60	692.28	540.14	547.41	550.26	744.43	547.11
6000	8	351.57	347.77	350.30	1119.39	347.35	307.81	297.12	304.27	972.41	299.42	247.27	237.37	246.94	768.53	244.20
6000	12	48.80	34.03	48.96	1149.44	44.67	43.88	31.56	44.00	1003.37	39.38	36.59	26.69	36.50	800.76	31.06
6000	16	41.82	41.82	41.82	1185.59	6.01	40.85	40.85	40.85	1039.65	5.82	39.68	39.68	39.68	837.22	5.57

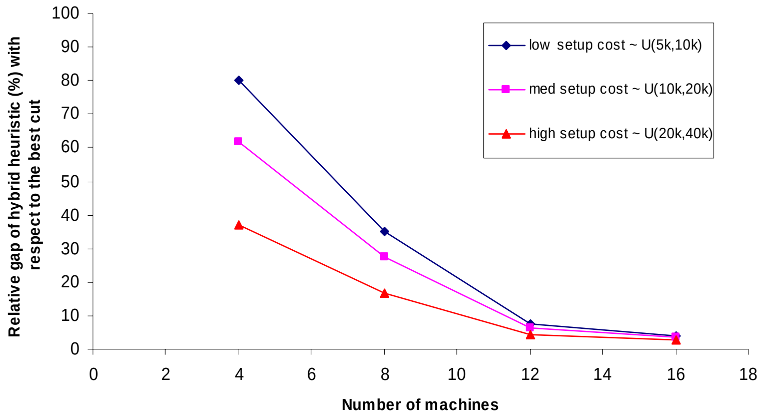


Figure 8.1: Relative Gap for Hybrid Heuristic for Case 8A

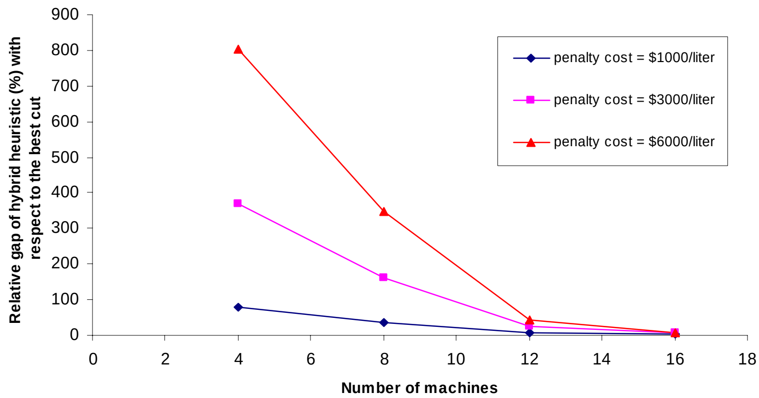


Figure 8.2: Relative Gap for Hybrid Heuristic for Case 8B

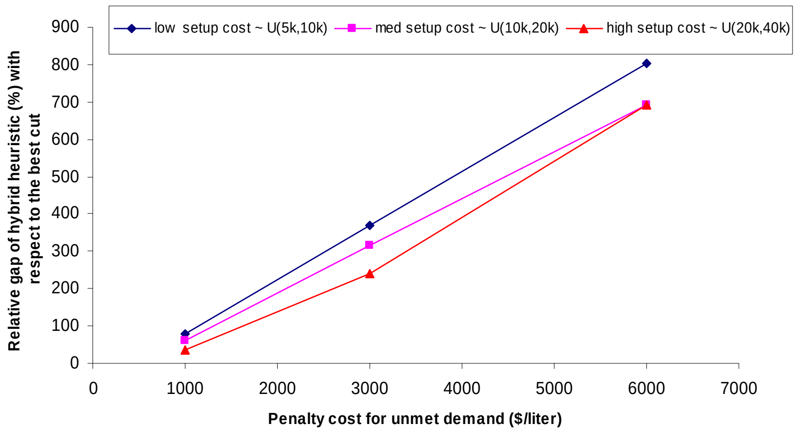


Figure 8.3: Relative Gap for Hybrid Heuristic for Case 8C

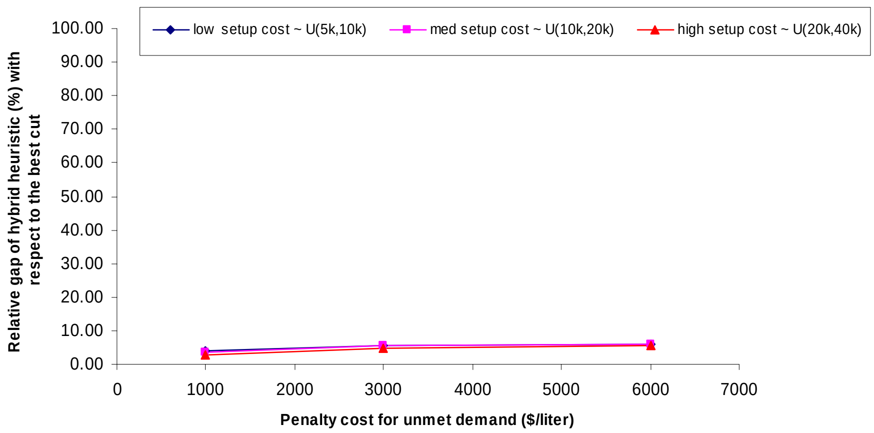


Figure 8.4: Relative Gap for Hybrid Heuristic for Case 8D

Table 8.3: Percentage of Average Utilization for Each Heuristic for the Large Problem of Vaccine Production

Penalty cost of unmet demand	# MC	Setup cost ($/setup)														
		U(5k,10k)					U(10k,20k)					U(20k,40k)				
($/liter)		MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid
1000	4	91.63	91.63	91.63	93.46	91.83	90.96	91.54	91.44	93.46	91.25	89.62	90.38	89.71	93.46	89.81
1000	8	87.79	88.94	88.37	93.46	87.50	86.78	88.70	87.64	93.46	87.50	85.77	86.54	85.96	93.46	85.58
1000	12	81.22	81.70	81.47	93.46	77.76	80.51	81.47	80.77	93.46	77.24	78.14	78.91	78.17	93.46	76.06
1000	16	74.16	74.16	74.16	93.46	60.29	70.46	70.46	70.46	93.46	60.12	62.38	62.38	62.38	93.46	59.13
3000	4	93.27	92.88	92.31	93.46	92.50	92.88	92.69	92.60	93.46	92.40	92.98	92.50	92.88	93.46	92.31
3000	8	90.72	91.15	90.77	93.46	89.66	90.48	91.30	90.96	93.46	89.81	90.38	90.67	90.19	93.46	89.04
3000	12	85.10	85.58	85.00	93.46	80.74	85.03	85.19	84.94	93.46	80.35	84.68	85.45	84.58	93.46	80.03
3000	16	80.00	80.00	80.00	93.46	62.16	80.05	80.05	80.05	93.46	61.90	80.00	80.00	80.00	93.46	61.66
6000	4	93.27	93.46	92.31	93.46	92.50	93.27	93.08	92.79	93.46	92.60	93.17	92.88	93.27	93.46	92.69
6000	8	91.06	91.30	91.39	93.46	89.66	91.06	91.68	91.44	93.46	89.90	90.91	91.01	91.11	93.46	88.99
6000	12	85.38	85.96	85.29	93.46	81.19	85.38	85.83	85.29	93.46	81.19	85.03	85.77	85.03	93.46	81.06
6000	16	79.95	79.95	79.95	93.46	62.57	80.02	80.02	80.02	93.46	62.57	80.19	80.19	80.19	93.46	62.50

Table 8.4: Percentage of Average Fill Rate for Each Heuristic for the Large Problem of Vaccine Production

Penalty cost of unmet demand ($/liter)	# MC	Setup cost ($/setup)														
		U(5k,10k)					U(10k,20k)					U(20k,40k)				
		MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid	MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid
1000	4	40.25	39.70	39.34	19.71	40.33	40.20	39.50	39.26	19.71	40.42	39.66	39.01	38.73	19.71	39.20
1000	8	73.96	74.24	73.97	21.85	74.11	73.25	74.42	73.54	21.85	73.99	72.05	72.93	71.98	21.85	72.20
1000	12	95.96	96.85	96.15	22.28	95.90	95.79	96.91	95.97	22.28	95.59	94.83	95.83	94.84	22.28	94.55
1000	16	98.48	98.48	98.48	22.33	98.79	98.19	98.19	98.19	22.33	98.63	96.88	96.88	96.88	22.33	97.53
3000	4	40.69	40.09	39.52	19.71	40.53	40.77	39.88	39.54	19.71	40.63	40.48	39.56	39.52	19.71	39.74
3000	8	74.74	74.98	74.71	21.85	74.96	74.36	75.23	74.59	21.85	74.96	73.69	74.60	73.59	21.85	73.89
3000	12	97.27	98.27	97.26	22.28	97.09	97.24	98.18	97.23	22.28	97.06	97.19	98.27	97.18	22.28	97.06
3000	16	99.97	99.97	99.97	22.33	99.84	99.97	99.97	99.97	22.33	99.77	99.94	99.94	99.94	22.33	99.70
6000	4	40.70	40.14	39.60	19.71	40.53	40.81	39.92	39.64	19.71	40.65	40.49	39.66	39.56	19.71	39.77
6000	8	74.79	75.08	74.80	21.85	74.96	74.43	75.36	74.67	21.85	74.98	73.76	74.78	73.73	21.85	73.84
6000	12	97.38	98.39	97.37	22.28	97.17	97.35	98.32	97.34	22.28	97.16	97.30	98.32	97.32	22.28	97.20
6000	16	99.98	99.98	99.98	22.33	99.93	99.97	99.97	99.97	22.33	99.93	99.98	99.98	99.98	22.33	99.92

Table 8.5: Average Percentage of the Number of Binary Variables w and r, whose values fall between 0.3 and 0.7 from the solution of RMIP

		w			r		
Penalty cost of unmet demand ($/gal)	# MC	Setup cost ($/setup)			Setup cost ($/setup)		
		U(5k,10k)	U(10k,20k)	U(20k,40k)	U(5k,10k)	U(10k,20k)	U(20k,40k)
1000	4	35.37	37.65	35.13	19.46	19.75	15.72
1000	8	24.01	23.25	23.40	29.79	30.33	22.36
1000	12	18.20	18.03	17.14	31.82	32.14	25.02
1000	16	14.85	12.60	12.92	35.03	36.57	27.51
3000	4	13.49	13.67	12.80	36.13	37.69	36.50
3000	8	17.60	16.65	18.24	31.95	33.13	31.28
3000	12	24.10	23.32	21.91	29.09	29.32	29.72
3000	16	36.79	35.59	34.80	19.15	18.91	19.66
6000	4	13.16	13.66	14.01	36.22	36.22	37.20
6000	8	17.14	18.23	18.00	32.18	32.38	30.79
6000	12	23.22	23.79	23.35	28.62	30.12	29.35
6000	16	35.75	35.74	34.04	19.82	19.55	19.74

According to the numerical results in Tables 8.3-8.5 and Figure 8.1-8.4, we make the following observations:

- Overall, the performance of all of heuristics in terms of the relative gap is greatly affected by the change in the penalty cost of unmet demand, the number of machines, and the setup cost.
- As the penalty cost of unmet demand increases and other factors remain unchanged, the relative gap for each heuristic tends to increase. The main reason for this is that when the number of machine is scare and the penalty cost of unmet demand is very high, selecting the wrong product to release on machine in improper time could cause substantial penalty cost of unmet demand. Therefore, the total cost obtained by heuristics could become bigger, so does the relative gap.

- As the number of machines increases and other factors remain unchanged, the relative gap for all of heuristics except the FOQ heuristic tends to be smaller until the average percentage of fill rate reaches around 97% by using 12 machines. Suppose that we are solving the problem to its optimal solution. Due to the fact that increase in the number of machines always raises the overall capacity of the plant, thereby improving the fill rate and lowering the amount of unmet demand. As a result the total cost becomes lower. As expected, increasing the number of machines from 12 to 16 improves the average fill rate to be almost 100%, but this makes the relative gap for the three MLFL heuristics become larger. The main reason for this is that the MLFL heuristics do not guarantee the fact that as the number of machines increases, the total cost of the MPS will improve, which is discussed in detail in section 6.2.1. Increasing the number of machines more than needed for the MLFL heuristics could result in the production earlier than needed. Thereby increasing the extra inventory cost and total cost. As a result, the relative gap for these three MLFL heuristics could become larger. On the other hand, when the number of machines is largely sufficient to satisfy almost all of demand, the Hybrid heuristic by far outperforms other heuristics. For example, let's consider the case in which the number of machines is 16, which results in the average fill rate of over 99%. The relative gap of the Hybrid heuristic is very small around 2.8-6%, while the minimum relative gap of other heuristics is 6%. Recall that the FOQ heuristic does not take account to the benefit of releasing the batch of product in selecting the product to produce first. As a result, unworthy production possibly occurs. It is possible that the relative gap of the FOQ heuristic becomes larger as the number of machines increases.

- As the setup cost increases and other factors remain unchanged, the relative gap for each heuristic tends to decrease. When setup cost is relatively higher than other costs, it is trivial to obtain good solutions by using heuristics. These solutions become closer to the lower bound from RMIP, so the relative gap becomes smaller.
- Overall, solving the RMIP of BPP-SI problems produces around 25% of total number of binary variables w and r falling between 0.3 and 0.7. Such proportion is quite huge, so rounding these values to the nearest integer does not necessarily guarantee the feasibility and the better solution. Also we find out that the proportion of binary variables falling between 0.3 and 0.7 is insensitive to the value of setup cost, while other parameters remain unchanged.

Next, we present the statistical analyses of the pairwise comparisons of the average total cost between heuristics.

Table 8.6: Pairwise Confidence Intervals for Differences of the Average Total Cost between Heuristics with the Confidence Level of 95%.

Penalty cost of unmet demand ($/liter)	Setup cost ($/setup)	# MCs	$C_{Hybrid} - C_{MLFL\text{-}B}$		$C_{MLFL\text{-}B} - C_{MLFL\text{-}A}$		$C_{Hybrid} - C_{MLFL\text{-}A}$		$C_{MLFL\text{-}A} - C_{MLFL\text{-}C}$		$C_{MLFL\text{-}C} - C_{FOQ}$		$C_{MLFL\text{-}A} - C_{FOQ}$	
			LB	UB	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB
1000	U(5k,10k)	4	-757,861	459,965	-446,847	770,979	-595,795	622,031	-786,182	431,644	-2,635,135	-1,417,308	-2,812,403	-1,594,577
1000	U(5k,10k)	8	-713,085	504,741	-513,500	704,326	-617,672	600,154	-637,863	579,963	-6,030,519	-4,812,693	-6,059,469	-4,841,643
1000	U(5k,10k)	12	-740,537	477,289	-675,454	542,372	-807,078	410,748	-602,482	615,345	-8,523,497	-7,305,671	-8,517,065	-7,299,239
1000	U(5k,10k)	16	-1,671,124	-453,298	-608,913	608,913	-1,671,124	-453,298	-608,913	608,913	-9,350,805	-8,132,979	-9,350,805	-8,132,979
1000	U(10k,20k)	4	-769,587	448,239	-426,149	791,677	-586,823	631,003	-797,374	420,452	-2,626,942	-1,409,116	-2,815,403	-1,597,577
1000	U(10k,20k)	8	-696,173	521,653	-542,673	675,153	-629,933	587,893	-630,861	586,966	-6,027,298	-4,809,471	-6,049,245	-4,831,419
1000	U(10k,20k)	12	-755,846	461,980	-666,471	551,355	-813,405	404,422	-603,608	614,218	-8,605,009	-7,387,183	-8,599,704	-7,381,878
1000	U(10k,20k)	16	-1,491,158	-273,331	-608,913	608,913	-1,491,158	-273,331	-608,913	608,913	-9,827,836	-8,610,009	-9,827,836	-8,610,009
1000	U(20k,40k)	4	-740,100	477,726	-413,822	804,005	-545,009	672,817	-792,936	424,890	-2,626,737	-1,408,911	-2,810,760	-1,592,934
1000	U(20k,40k)	8	-688,597	529,230	-530,918	686,908	-610,602	607,225	-641,058	576,768	-6,029,415	-4,811,588	-6,061,560	-4,843,733
1000	U(20k,40k)	12	-717,668	500,159	-632,144	585,683	-740,898	476,928	-610,026	607,800	-8,853,434	-7,635,608	-8,854,547	-7,636,721
1000	U(20k,40k)	16	-1,013,324	204,503	-608,913	608,913	-1,013,324	204,503	-608,913	608,913	-10,904,358	-9,686,532	-10,904,358	-9,686,532
3000	U(5k,10k)	4	-2,025,916	1,686,321	-1,663,441	2,048,796	-1,833,238	1,878,999	-2,251,812	1,460,425	-7,764,891	-4,052,655	-8,160,585	-4,448,348
3000	U(5k,10k)	8	-1,953,914	1,758,323	-1,863,460	1,848,776	-1,961,256	1,750,981	-1,852,979	1,859,257	-17,945,285	-14,233,048	-17,942,145	-14,229,909
3000	U(5k,10k)	12	-1,821,802	1,890,435	-2,135,693	1,576,544	-2,101,376	1,610,861	-1,856,560	1,855,677	-24,928,857	-21,216,620	-24,929,298	-21,217,062
3000	U(5k,10k)	16	-3,355,185	357,051	-1,856,118	1,856,118	-3,355,185	357,051	-1,856,118	1,856,118	-25,952,897	-22,240,660	-25,952,897	-22,240,660
3000	U(10k,20k)	4	-2,101,487	1,610,749	-1,577,849	2,134,388	-1,823,218	1,889,019	-2,275,012	1,437,224	-7,760,057	-4,047,821	-8,178,951	-4,466,715
3000	U(10k,20k)	8	-1,893,102	1,819,134	-2,016,088	1,696,149	-2,053,072	1,659,165	-1,809,945	1,902,292	-17,891,447	-14,179,210	-17,845,273	-14,133,037
3000	U(10k,20k)	12	-1,876,497	1,835,739	-2,140,952	1,571,285	-2,161,331	1,550,906	-1,855,847	1,856,389	-24,948,403	-21,236,166	-24,948,132	-21,235,895
3000	U(10k,20k)	16	-3,543,555	168,682	-1,856,118	1,856,118	-3,543,555	168,682	-1,856,118	1,856,118	-26,039,645	-22,327,408	-26,039,645	-22,327,408
3000	U(20k,40k)	4	-1,949,548	1,762,689	-1,559,731	2,152,506	-1,653,161	2,059,076	-2,209,910	1,502,327	-7,752,526	-4,040,290	-8,106,318	-4,394,081
3000	U(20k,40k)	8	-1,812,361	1,899,876	-2,031,316	1,680,920	-1,987,558	1,724,678	-1,873,219	1,839,018	-17,624,895	-13,912,658	-17,641,995	-13,929,759
3000	U(20k,40k)	12	-1,968,193	1,744,043	-2,135,713	1,576,523	-2,247,788	1,464,448	-1,854,883	1,857,354	-25,018,007	-21,305,771	-25,016,772	-21,304,535
3000	U(20k,40k)	16	-3,914,451	-202,214	-1,856,118	1,856,118	-3,914,451	-202,214	-1,856,118	1,856,118	-26,227,997	-22,515,760	-26,227,997	-22,515,760
6000	U(5k,10k)	4	-3,928,689	3,553,204	-3,521,186	3,960,707	-3,708,929	3,772,964	-4,428,658	3,053,235	-15,504,239	-8,022,346	-16,191,950	-8,710,057
6000	U(5k,10k)	8	-3,758,477	3,723,416	-3,900,240	3,581,653	-3,917,771	3,564,122	-3,687,777	3,794,116	-35,928,356	-28,446,463	-35,875,187	-28,393,294
6000	U(5k,10k)	12	-3,295,588	4,186,305	-4,359,166	3,122,727	-3,913,807	3,568,086	-3,747,388	3,734,505	-49,797,554	-42,315,661	-49,803,995	-42,322,102
6000	U(5k,10k)	16	-5,239,555	2,242,338	-3,740,946	3,740,946	-5,239,555	2,242,338	-3,740,946	3,740,946	-51,609,348	-44,127,455	-51,609,348	-44,127,455
6000	U(10k,20k)	4	-4,112,165	3,369,728	-3,323,228	4,158,665	-3,694,446	3,787,447	-4,478,122	3,003,771	-15,512,162	-8,030,269	-16,249,337	-8,767,444
6000	U(10k,20k)	8	-3,630,651	3,851,242	-4,253,924	3,227,969	-4,143,628	3,338,265	-3,570,865	3,911,028	-35,821,729	-28,339,836	-35,651,647	-28,169,754
6000	U(10k,20k)	12	-3,365,433	4,116,460	-4,332,577	3,149,316	-3,957,063	3,524,830	-3,746,675	3,735,217	-49,805,188	-42,323,295	-49,810,917	-42,329,024
6000	U(10k,20k)	16	-5,422,464	2,059,429	-3,740,946	3,740,946	-5,422,464	2,059,429	-3,740,946	3,740,946	-51,698,458	-44,216,565	-51,698,458	-44,216,565
6000	U(20k,40k)	4	-3,758,701	3,723,192	-3,300,535	4,181,358	-3,318,290	4,163,603	-4,354,036	3,127,857	-15,500,692	-8,018,799	-16,113,781	-8,631,888
6000	U(20k,40k)	8	-3,328,927	4,152,966	-4,338,462	3,143,431	-3,926,443	3,555,450	-3,720,679	3,761,214	-35,214,316	-27,732,423	-35,194,048	-27,712,155
6000	U(20k,40k)	12	-3,476,920	4,004,973	-4,338,623	3,143,270	-4,074,597	3,407,296	-3,735,336	3,746,557	-49,857,575	-42,375,682	-49,851,965	-42,370,072
6000	U(20k,40k)	16	-5,799,040	1,682,853	-3,740,946	3,740,946	-5,799,040	1,682,853	-3,740,946	3,740,946	-51,865,816	-44,383,923	-51,865,816	-44,383,923

According to Table 8.6, we can conclude that in general there is no evidence that average total cost obtained by the Hybrid and the three MLFL heuristics are different since its interval includes zero, except the following three scenarios:

1) Penalty cost = \$1000/liter, the number of machines = 16, and setup cost = uniformly distributed on (5000, 10000)
2) Penalty cost = \$1000/liter, the number of machines = 16, and setup cost = uniformly distributed on (10000, 20000)
3) Penalty cost = \$3000/liter, the number of machines = 16, and setup cost = uniformly distributed on (20000, 40000)

There is evidence that average total cost obtained by Hybrid heuristic is lower than that obtained by the three MLFL heuristics. However, the experiment indicates that the three MLFL heuristics provide insignificant difference on average total cost. For every scenario, the FOQ heuristic performs worst as it produces the highest average total cost.

8.2 Computational Result of Performance of Heuristics for Large BPP-SI Problems in Vaccine Production

In this section, we solve one large BPP-SI problem for each of 12 scenarios, each having different values of setup cost and number of machines by using optimization and heuristic approaches. As the problem is extremely large and complicated, the branch and bound method cannot optimally solve for this large problem instance within required time limit of one day, but we can readily obtain the first integer solution, lower bound from RMIP, the best integer solution and best cut, which is the best lower bound in fractional value obtained from the branch and bound method. To assess the performance of the heuristics, we calculate the relative gap of solution from heuristic with respect to the best cut. On the other hand, each heuristic requires less than one second for solving the large BPP-SI problems.

8.2.1 Parameters and Level of Factors

In this study, we consider the same BPP-SI problem for a vaccine manufacturer in section 7.3, focus on the case where all of the low levels for each of four factors in Table 7.9 are selected, and use the same values of parameters listed in Table 7.8 except control parameters, which are defined as in the Table 8.1, but the penalty cost of unmet demand is assumed to be $3000/liter.

8.2.2 Numerical Result

We present the numerical result of solving the large BPP-SI problem for vaccine production for 12 scenarios using the branch and bound method and five heuristics. The total cost and the relative gap for each heuristic with respect to the best lower bound (best cut) are shown in Table 8.7.

Table 8.7: Total Cost and Percentage of Average Relative Gap for Each Heuristic when setup cost and number of machines vary

			Total costs ($)									Percentage of relative gap wrt. Best cut				
No	Setup cost ($/setup)	# MCs	Lower Bound from relaxation	First Integer	Best cut (fraction)	Best Integer	Heuristic					MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid
							MLFL-A	MLFL-B	MLFL-C	FOQ	Hybrid					
1	U(5k,10k)	4	2,487,165	16,981,000	6,678,606	6,894,152	7,870,840	7,890,924	8,072,512	16,322,707	8,095,837	17.85	18.15	20.87	144.40	21.22
2	U(5k,10k)	8	2,487,165	18,819,000	2,488,491	3,781,338	4,093,035	4,152,104	4,114,910	16,350,781	3,477,878	64.48	66.85	65.36	557.06	39.76
3	U(5k,10k)	12	2,487,165	18,819,000	2,497,732	2,887,723	4,443,650	4,443,650	4,443,650	18,013,759	2,905,285	77.91	77.91	77.91	621.20	16.32
4	U(5k,10k)	16	2,487,165	18,819,000	2,488,962	2,887,663	4,906,834	4,906,834	4,906,834	19,676,737	2,905,285	97.14	97.14	97.14	690.56	16.73
5	U(10k,20k)	4	2,843,010	17,281,000	6,981,105	7,315,867	8,117,067	8,135,781	8,314,405	16,510,720	8,351,480	16.27	16.54	19.10	136.51	19.63
6	U(10k,20k)	8	2,843,010	17,750,000	2,844,512	4,134,230	4,457,940	4,525,644	4,566,625	16,726,808	3,851,578	56.72	59.10	60.54	488.04	35.40
7	U(10k,20k)	12	2,843,010	17,566,000	2,845,624	3,329,745	5,022,414	5,022,414	5,022,414	18,577,799	3,305,997	76.50	76.50	76.50	552.86	16.18
8	U(10k,20k)	16	2,843,010	17,428,000	2,844,795	3,276,186	5,546,291	5,546,291	5,546,291	20,428,791	3,305,997	94.96	94.96	94.96	618.11	16.21
9	U(20k,40k)	4	3,554,700	34,362,000	7,498,820	7,740,459	8,873,329	8,792,654	8,767,181	16,886,748	9,381,157	18.33	17.25	16.91	125.19	25.10
10	U(20k,40k)	8	3,554,700	35,351,000	3,556,322	4,625,059	5,336,004	5,420,978	5,342,781	17,478,863	4,598,978	50.04	52.43	50.23	391.49	29.32
11	U(20k,40k)	12	3,554,700	34,995,000	3,556,861	4,053,155	6,235,962	6,235,962	6,235,962	19,705,882	4,107,421	75.32	75.32	75.32	454.02	15.48
12	U(20k,40k)	16	3,554,700	34,678,000	3,556,525	4,054,387	6,825,206	6,825,206	6,825,206	21,932,901	4,107,421	91.91	91.91	91.91	516.69	15.49

According to Table 8.7, we found that no unique heuristic dominates others for all of twelve scenarios. The relative gaps for the three MLFL heuristics are not significantly different. In scenarios where the number of machines is very low like 4, the MLFL heuristics outperform the Hybrid heuristic, while in other scenarios, the Hybrid heuristic outperform others. The FOQ heuristic performs worst for all scenarios. As heuristics require a very short amount of computational time, the planner should use all five heuristics in order to select the best solution out of the five production plans. Overall, the best heuristic selected for each scenario produces the relative gap of 35.4% or less. More importantly, the Hybrid heuristic is very efficient in the sense that it could result in better solutions than a truncated branch and bound method in certain scenarios. For example, when the number of machines is 8, total cost from the Hybrid heuristic is much less than the best integer solution from the branch and bound method, when it terminates at a target computational time of one day.

We also make the following observations. First, the number of machines is one of influential factors affecting the quality of solution from heuristics. As the number of machines increases while the setup cost remains constant, the relative gap with respect to the best cut tends to greatly increase. Second, the setup cost has slightly affected the relative gap. The setup cost increases while the number of machines remains constant, the relative gap with respect to the best cut tends to slightly decrease.

CHAPTER IX

CONCLUSIONS AND EXTENTSIONS

9.1 Summary and Conclusions

This dissertation addresses the complex set of production scheduling decisions for a manufacturer of fixed shelf-life products in a single stage of batch production process. Some examples of the batch process used in production of perishable products include fermentation process for beers, and incubation process for vaccines. Batch production planning is very difficult due to the varieties of constraints, such as non-preemptive processes, lot sizing, processing sequences, setup times (sequence independent or sequence dependent), and shelf-life of products. The batch production planning problem involves finding the master production schedule (MPS) for a single stage of batch process for perishable products in order to minimize total cost, which consist of costs of inventory, spoilage, production, setup and penalty for unmet demand.

In this dissertation, we formulate the new mathematical models for representing the batch production planning problem for perishable products with an emphasis on the operational decisions, develop the tractable, efficient five heuristics for solving the large BPP-SI problems, apply these heuristic to solve large problems in industry, examine factors of interest on the system performance and analyze the performance of heuristics.

In Chapter III, we define the batch production scheduling problems for fixed shelf-life products with sequence-independent setup time (BPP-SI), develop a Mixed Integer Program (MIP) for representing the BPP-SI problem, and present a numerical result for a small example for the fermentation process of beer.

In Chapter IV, we describe the batch production scheduling problems for fixed shelf-life products with sequence-dependent setup time (BPP-SD), develop a MIP for representing the BPP-SD problem, and present a numerical result for a small example for the incubation process of vaccine.

In Chapter V, we examine the effect of factors of interest, such as type of lot size, shelf-life, demand variation and demand probability on the system performance for BPP-SI and BPP-SD for three different configuration settings depending on the size of problem defined by number of products, number of machines, and length of planning horizon. In practice, the size of industrial problem is very large, so the branch and bound method cannot be executed to the optimality within the reasonable computational time. This motivates us to develop the efficient heuristics for solving the large problems.

In Chapter VI, we develop five efficient heuristics for solving the batch production scheduling problems with sequence-independent setup times (BPP-SI). Five heuristics include Modified Lot For Lot-A (MLFL-A), MLFL-B, MLFL-C, Fixed Order Quantity (FOQ) and Hybrid. Each heuristic uses different rule in selecting the product to produce first. For example, the MLFL-A heuristic employs the benefit of production one batch of product as the decision rule. The Hybrid heuristic considers the zero earliness of production, the cumulative demand during the shelf-life of product, and benefit of production of each batch as the decision rule. Through numerical analyses from small problems, the MPS could be obtained by heuristics within a very short amount of solution time around one second with the small relative gap, which is compared to the lower bound of the MIP relaxation. However, the relative gap obtained from heuristics can be varying depending on the value of parameters selected.

In Chapter VII, we investigate the effect of several factors including demand probability, demand variation, length of shelf-life and type of lot size on the total cost of the batch production process, and compare the performance of each heuristic by performing the statistical analysis. Through numerical analyses from large problems, the

Hybrid heuristic generally provides very good solution with its relative gap of 2% or less. Each of the three MLFL heuristics usually result in fairly good solution with its relative gap of 10% or less, while the FOQ heuristic produces very poor solutions with its relative gap of 100% or more. The results show the significant improvement in computational time for the large BPP-SI problems by using our heuristics developed. Overall the computational time for each heuristic is very small around 0.3 seconds even for the large problems with 10 products, 20 machines and 26 periods. Therefore, our heuristics are very efficient for solving large problems.

Chapter VIII investigates the performance of heuristics for BPP-SI problems by a computational study. To achieve this goal, we consider the effect of change in value of parameters, such as the penalty of unmet demand, setup cost and the number of machines on the total cost and compute the relative gap for each of heuristics with respect to this lower bound on total cost when solving the very large BPP-SI problems. Through numerical analyses, there is evidence that the average total cost obtained by the Hybrid heuristic is lower than that obtained by the three MLFL heuristics. However, the experiment indicates that the three MLFL heuristics provide insignificant difference on average total cost. For every scenario, the FOQ heuristic performs worst since it produces the highest average total cost.

Overall, this dissertation presents a new integrative approach for dealing with batch production scheduling problems for fixed shelf-life products with setup times on a single processing unit of parallel machines. This dissertation differs from previous work done under lot-sizing and scheduling problems and inventory management for perishable products in that our models incorporate several practical issues, such as limited shelf-life of products, a change in the number of available machines and a penalty for unmet demand into the models, which also include the issues of lot-sizing and setup-times. We formulate the discrete-time MIP models for the batch production scheduling problems for fixed shelf-life products for the case of sequence-independent setup times (BPP-SI), and the case of sequence-dependent setup times (BPP-SD). Furthermore, we develop the five efficient heuristics for solving the batch production scheduling problems with sequence-independent setup times (BPP-SI). The extensive computational results show that the developed heuristics can obtain good solutions for very large problem sizes and require a very short amount of computational time, which is the major contribution of our research on significant improvement in computational time for solving the large BPP-SI problems. In particular, the Hybrid heuristic produces very good results whose relative gap is usually less than 10% when the number of machines is enough to satisfy almost all of demand for products during the planning horizon.

9.2 Future Extensions

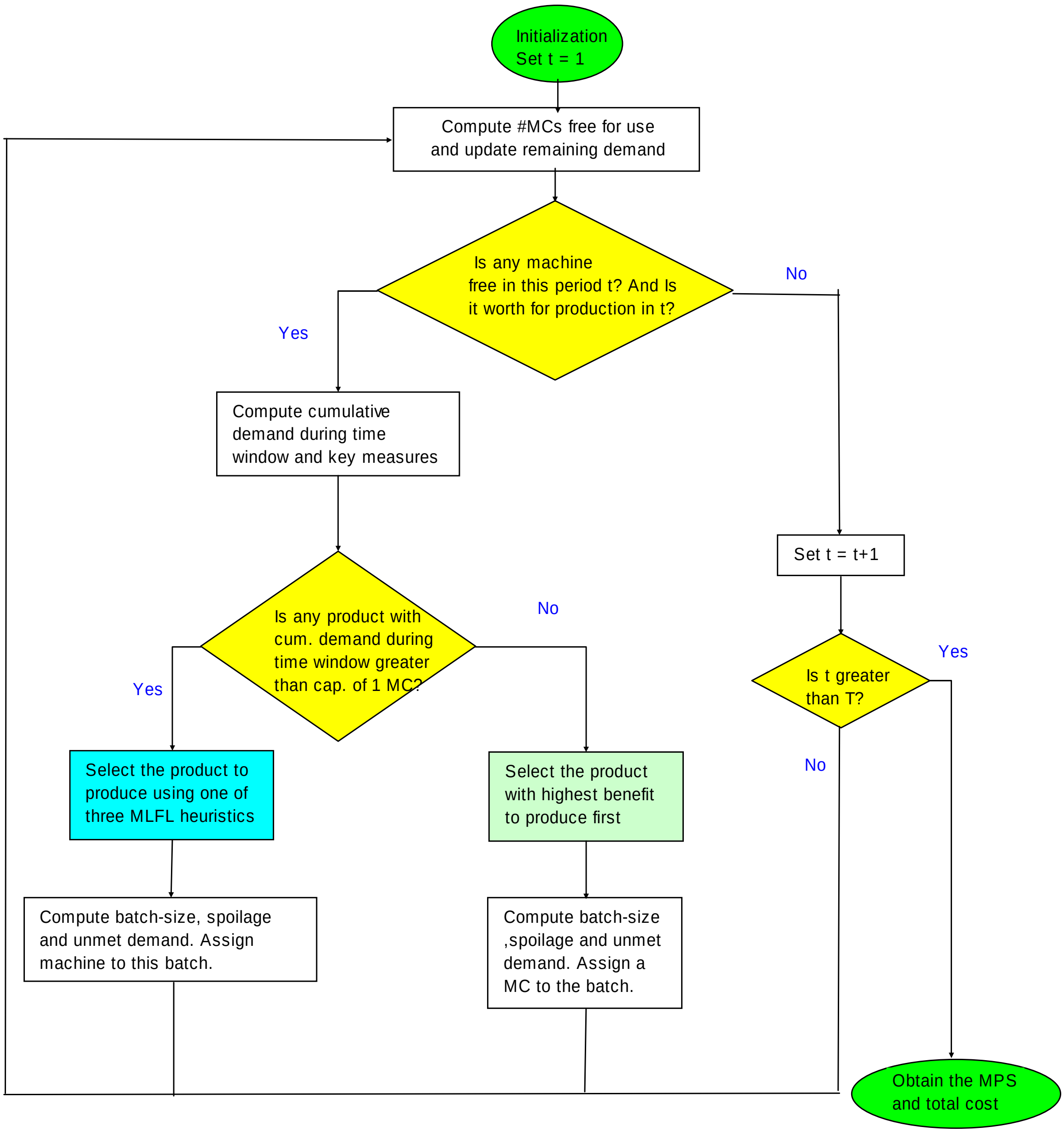
Results from this dissertation raise several potential directions for future research. This dissertation developed the mathematical model for BPP-SI and BPP-SD under the assumption that demand for products is deterministic and only one stage of batch production is considered. However, in certain situations, demand for products is stochastic and the plant might consist of multi-stages of batch production. Future work can extend the models to include both issues. The additional complexity will impact the capability of the current approach to solve large BPP problems. Future work should focus on developing efficient heuristics for solving the large BPP-SD problems as the optimization approach requires a significant amount of computational time. Developing new optimization approaches for the BPP problems is another way of expanding this research problem. Future work should focus on the fact that the BPP must be solved in a rolling horizon environment. In other words, it is a priori known that only the first part of a solution will be implemented.

APPENDIX A

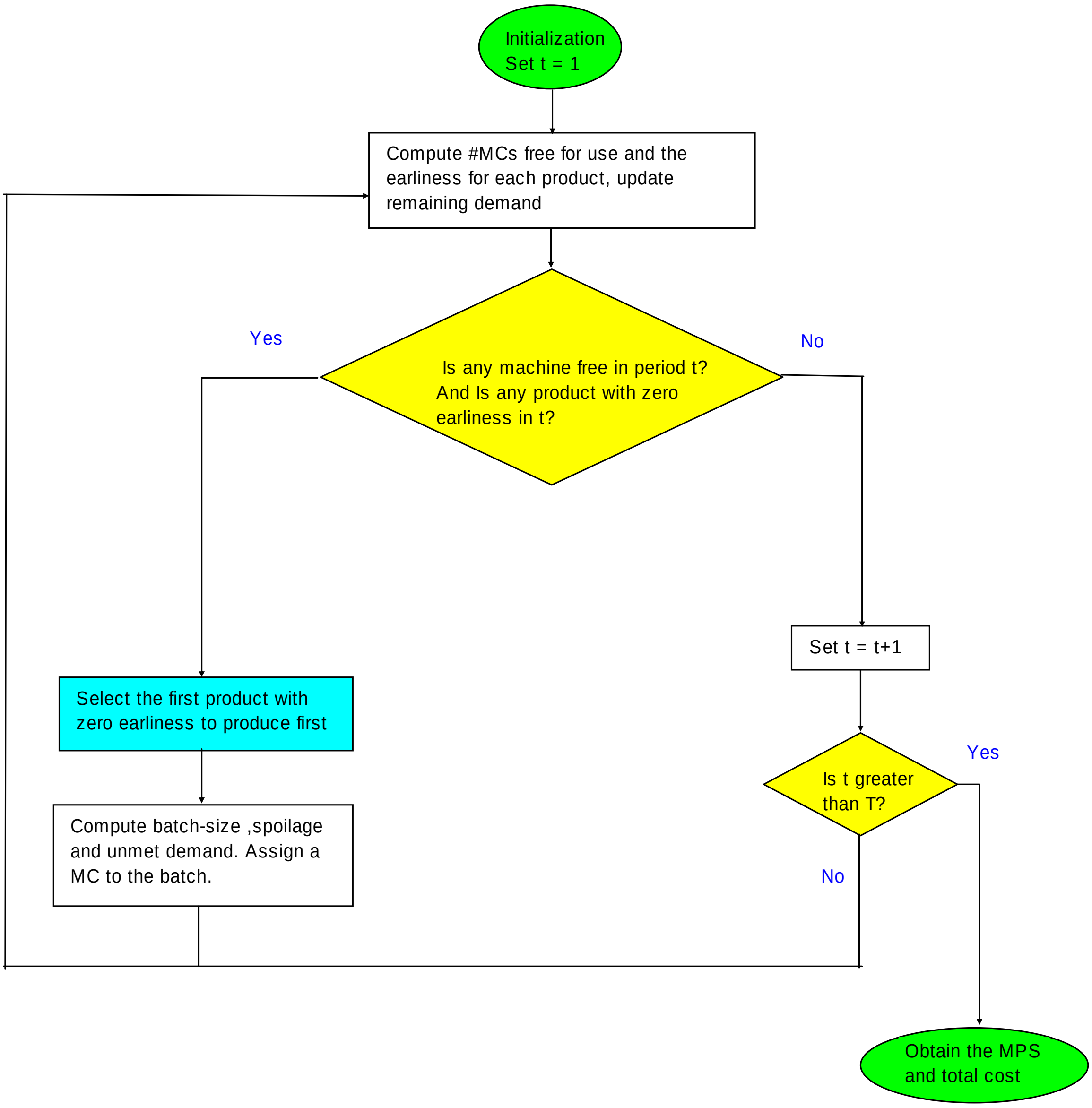
SUPPLEMENTARY FOR HEURISTICS

FOR SOLVING BPP-SI PROBLEMS

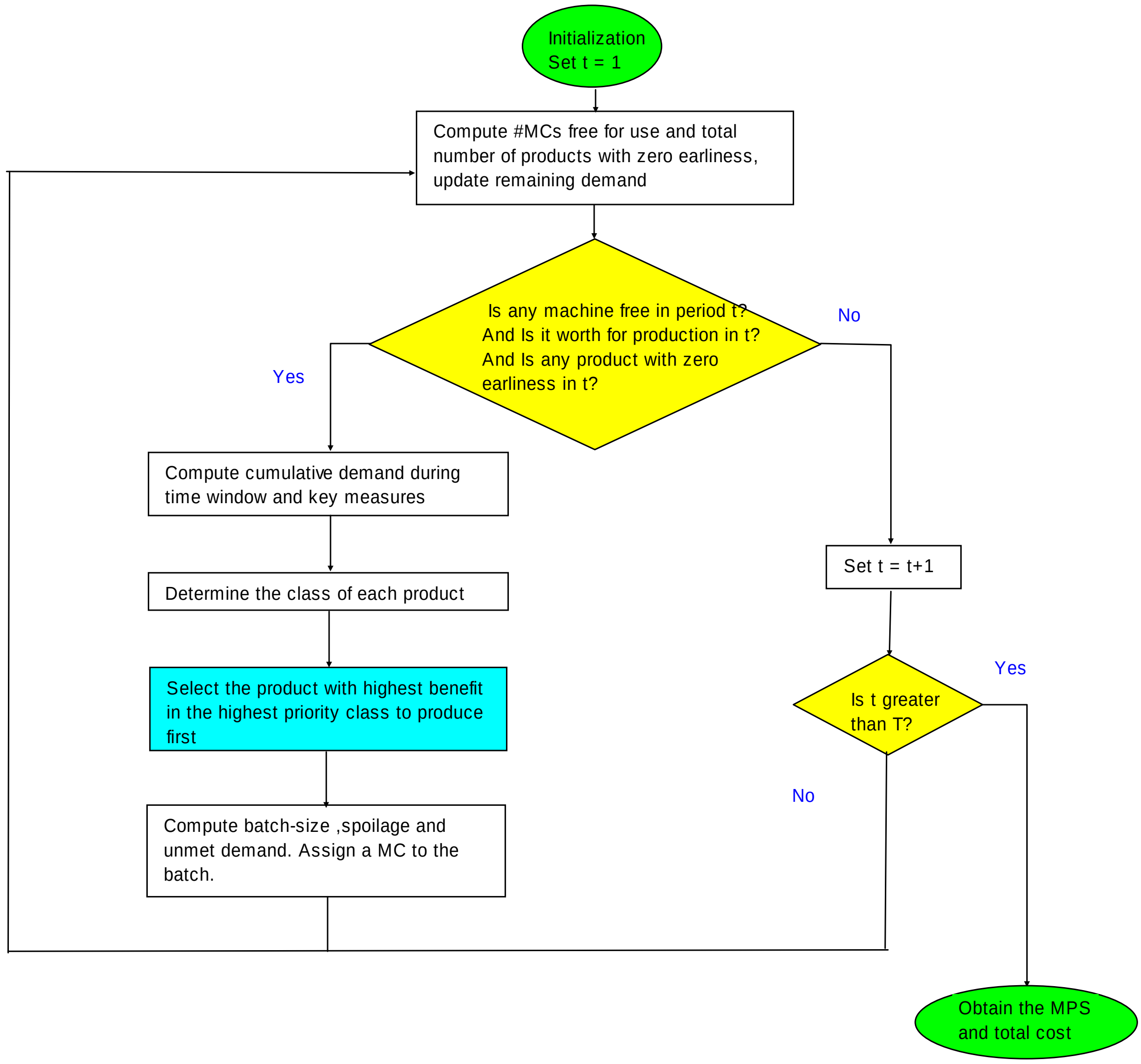
A.1 Flowchart of the MLFL Heuristics



A.2 Flowchart of the FOQ Heuristic



A.3 Flowchart of the Hybrid Heuristic



A.4 Implementation of the MLFL-A Heuristic for Solving BPP-SI Problem with Discrete Batch Size

In this section, we illustrate how to use MLFL-A Heuristic to solve the small BPP-SI example with discrete batch size in Chapter VI.

Step 0: Initialization:

Set $D'_{i,t} = D_{i,t} \quad \forall i,t$

Set $B_{j,t} = 0 \quad \forall j,t$

Compute T_last: T_last = 7 – min (1, 2) = 6

<u>Iteration 1</u>

Step 1: Start with t = 1

Step 1.1: Compute $Nmca_1$. $Nmca_1 = 1 - \sum_{j=1}^{1} B_{j1} = 1 - 0 = 1$.

This is because the machine has not been assigned to any product yet.

Step 1.2: Set $Flag_1 = 0$.

Step 1.3: Check whether the condition of ($Nmca_1 > 0$) and ($Flag_1 = 0$) for while loop is satisfied or not. We go inside to the loop, since the values of parameter satisfy such condition.

a) Compute $CDem_{it}$ from period $t+AT_i$ to period $t+AT_i+LT_i-1$ for each product:

$CDem_{A1}$ = 50+52+51+50+45 =248

$CDem_{B1}$ = 20+40+41 = 101

b) Both products have $CDem_{it}$ more than the capacity (C) of 50, so the total number of products with $CDem_{it} \geq C$ (Nover) is 2. Compute the benefit for each product.

Ben_{A1} = 50(4.5) - 40 - 50(3) - 10 = \$25

$Ben_{B1} = 50(6) - 50 - 50(4)-39 = \$11$

c) Check whether the Nover > 0. True (2>0).

c1) Determine which product to produce first using High Benefit First (HBF). Choose product A since it has the higher benefit (25>11).

c2) Determine which machine to use. Only one machine is available, so the machine is assigned to the product A. Since it takes 1 period to produce a batch of product A, such machine will be busy in period 1 for this batch, i.e. $r_{A11} = 1$, and setting B_{11} to be 1.

c3) Compute $Nmca_1$. $Nmca_1 = 1 - \sum_{j=1}^{1} B_{j1} = 1 - 1 = 0$. Since cumulative remaining demand is no less than capacity, setting the batch size to be 50. Determine the amount of spoilage, which would incur from this batch. Since this batch will be obtained at the beginning of period 2 and be used up to fulfill the demand for product A, no spoilage from this batch incurs. Hence S_{A2} is zero.

c4) Update the remaining demand for products ($D'_{i,t}$) using algorithm 6.2. The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining demands (units)	$D'_{A,t}$	0	0	0	52	51	50	45
	$D'_{B,t}$	0	0	0	0	20	40	41

- Go back to the beginning of while loop to check whether the condition of ($Nmca_1 > 0$) and ($Flag_1 = 0$) is still held.

No, since $Nmca_1 = 0$. Move out of while loop and increase time period t by 1.

Iteration 2

Step 1: Time period t = 2

Step 1.1: Compute $Nmca_2$. $Nmca_2 = 1 - \sum_{j=1}^{1} B_{j1} = 1 - 0 = 1.$

Step 1.2: Set $Flag_2 = 0$.

Step 1.3: Satisfy the condition of ($Nmca_2 > 0$) and ($Flag_2 = 0$) for while loop

Go inside the while loop.

c) Compute $CDem_{it}$ from period $t+AT_i$ to period $t+AT_i+LT_i-1$ for each product:

$CDem_{A2} = 52+51+50+45 = 198$

$CDem_{B2} = 20+40+41 = 101$

d) Both products have $CDem_{it}$ more than the capacity (C) of 50, so the total number of products with $CDem_{it} \geq C$ (Nover) is 2. Compute the benefit for each product.

$Ben_{A2} = 50(4.5) - 40 - 50(3) - 10 = \$25$

$Ben_{B2} = 50(6) - 50 - 50(4) - 24 = \$26$

c) Check whether the Nover > 0. True (2>0).

c1) Determine which product to produce first using High Benefit First (HBF). Choose product B since it has the higher benefit (26>25).

c2) Determine which machine to use. Only one machine is available, so the machine is assigned to the product B. Since it takes 2 periods to produce a batch of product B, such machine will be busy in period 2 and 3 for this batch, i.e. $r_{B12} = 1$, and setting B_{12} and B_{13} to be 1.

c3) Compute $Nmca_2$. $Nmca_2 = 1 - 1 = 0$. Since cumulative remaining demand is no less than capacity, setting the batch size to be 50.

Determine the amount of spoilage, which would incur from this batch. Since this batch will be obtained at the beginning of period 4 and be used up to fulfill the demand for product B, no spoilage from this batch incurs. Hence S_{B4} is zero.

c4) Update the remaining demand for products ($D'_{i,t}$) using algorithm 6.2. The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining demands (units)	$D'_{A,t}$	0	0	0	52	51	50	45
	$D'_{B,t}$	0	0	0	0	0	10	41

- Go back to the beginning of while loop to check whether the condition of ($Nmca_2 > 0$) and ($Flag_2 = 0$) is still held.

No, since $Nmca_2 = 0$. Move out of while loop and increase time period t by 1.

Iteration 3

Step 1: Time period t = 3

Step 1.1: Compute $Nmca_3$. $Nmca_3 = 1\text{-}1 = 0$

Step 1.2: Set $Flag_3 = 0$.

Step 1.3: Fail to satisfy the condition of ($Nmca_3 > 0$) and ($Flag_3 = 0$) for while loop since $Nmca_3 = 0$. Move out of while loop and increase time period t by 1.

Iteration 4

Step 1: Time period t = 4

Step 1.1: Compute $Nmca_4$. $Nmca_4 = 1\text{-}0 = 1$

Step 1.2: Set $Flag_4 = 1$.

Step 1.3: Satisfy the condition of ($Nmca_4 > 0$) and ($Flag_4 = 0$) for while loop

Go inside the while loop.

a) Compute $CDem_{it}$

$CDem_{A4} = 52+51+50+45 = 198$

$CDem_{B4} = 10+41 = 51$

b) Both products have $CDem_{it}$ more than the capacity (C) of 50, so the total number of products with $CDem_{it} \geq C$ (Nover) is 2. Compute the benefit for each product (Ben_{it}).

$Ben_{A4} = 50(4.5) - 40 - 50(3)-0 = \$35$

$Ben_{B4} = 50(6) - 50 - 50(4) -12 = \$38$

c) Check whether the Nover > 0. True (2>0).

c1) Determine which product to produce, using High Benefit First (HBF). Choose product B since it has the higher benefit (38>35).

c2) Determine which machine to use. Only one machine is available, so the machine is assigned to the product B. That is $r_{B14} = 1$, and setting B_{14} and B_{15} to be 1.

c3) Compute $Nmca_4$. $Nmca_4 = 1-1 = 0$. Since cumulative remaining demand is no less than capacity, setting the batch size to be 50. Determine the amount of spoilage, which would incur from this batch. S_{B6} is zero.

c4) Update the remaining demand for products ($D'_{i,t}$). The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining demands (units)	$D'_{A,t}$	0	0	0	52	51	50	45
	$D'_{B,t}$	0	0	0	0	0	0	1

- Go back to the beginning of while loop to check whether the condition of ($Nmca_4 > 0$) and ($Flag_4 = 0$) is still held.

No. since $Nmca_4 = 0$. Move out of while loop and increase time period t by 1.

Iteration 5

Step 1: Time period t = 5

Step 1.1: Compute $Nmca_5$. $Nmca_5 = 1\text{-}1 = 0$

Step 1.2: Set $Flag_5 = 0$.

Step 1.3: Fail to satisfy the condition of ($Nmca_5 > 0$) and ($Flag_5 = 0$) for while loop since $Nmca_5 = 0$. Move out of while loop and increase time period t by 1.

Iteration 6

Step 1: t = 6

Step 1.1: Compute $Nmca_6$. At the beginning of period 5, the machine is free. Therefore, $Nmca_6 = 1$.

Step 1.2: Set $Flag_6 = 0$.

Step 1.3: Satisfy the condition of ($Nmca_6 > 0$) and ($Flag_6 = 0$) for while loop

Go inside the while loop.

a) Compute $CDem_{it}$: $CDem_{A6} = 45$, $CDem_{B6} = 0$

b) Neither product A nor B has $CDem_{it}$ more than the capacity (C) of 50, so the total number of products with $CDem_{it} \geq C$ (Nover) is 0.

c) Check whether the Nover > 0. False

d1) Compute the benefit for each product (Ben_{it}).

$Ben_{A6} = 45(4.5) - 40 - 50(3)-5(0.4)-1 = 12.5$

$Ben_{B6} = 0$

d2) Compute the total number of product with positive benefit (Nben) Nben =1.

d3) Check whether the Nben > 0. True (1>0).

- Determine which product to produce first using High Benefit First (HBF). Choose product A, since there is only one candidate.
- Determine which machine to use. Only one machine is available, so the machine is assigned to the product A, i.e. r_{A16} = 1, and setting B_{16} to be 1.
- Compute $Nmca_6$. $Nmca_6 = 1 - 1 = 0$. The cumulative remaining demand is less than capacity, and the batch size is discrete, but it is worth releasing this batch. Therefore, the batch size equals 50. Determine the amount of spoilage, which would incur from this batch. This batch will be obtained at the end of horizon (period 7). As we assume that the excess production in the last period is deemed as the ending inventory, no spoilage from this batch incurs. Hence S_{A7} is zero.
- Update the remaining demand for products ($D'_{i,t}$). The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining demands (units)	$D'_{A,t}$	0	0	0	52	51	50	0
	$D'_{B,t}$	0	0	0	0	0	0	1

- Go back to the beginning of while loop to check whether the condition of ($Nmca_6 > 0$) and ($Flag_6 = 0$) is still held.

  No, since $Nmca_6 = 0$. Move out of while loop and increase time period t by 1. The resulting t will be 7, which is greater than T_last (6). Exit the for loop.

At the end of step 1, we obtain the batch release plan of each product over the horizon. In this example, the production plan is to release a batch of 50 units of product A in periods 1 and 6, and to release a batch of 50 units of product B in periods 2 and 4 as shown in the MPS in Table 6.4. Because steps 2-5 are straightforward, the details of calculation are omitted.

A.5 Implementation of the MLFL-B Heuristic for Solving BPP-SI Problem with Discrete Batch Size

In this section, we illustrate how to use MLFL-B Heuristic to solve the small BPP-SI example with discrete batch size in Chapter VI.

Step 0: Initialization:

Set $D'_{i,t} = D_{i,t} \quad \forall i,t$

Set $B_{j,t} = 0 \quad \forall j,t$

Compute T_last: T_last = 7 – min (1, 2) = 6

Iteration 1

Step 1: Start with t = 1

Step 1.1: Compute $Nmca_1$. $Nmca_1 = 1 - \sum_{j=1}^{1} B_{j1} = 1 - 0 = 1$.

Step 1.2: Set $Flag_1 = 0$.

Step 1.3: Check whether the condition of ($Nmca_1 > 0$) and ($Flag_1 = 0$) for while loop is satisfied or not. We go inside to the loop, since the values of parameter satisfy such condition.

a) Compute $CDem_{it}$

$CDem_{A1}$ = 50+52+51+50+45 =248

$CDem_{B1}$ = 20+40+41 = 101

b) Both products have $CDem_{it}$ more than the capacity (C) of 50, so the total number of products with $CDem_{it} \geq C$ (Nover) is 2.

Compute the benefit for each product (Ben_{it})

Ben_{A1} = 50(4.5) - 40 - 50(3)-10= \$25

Ben_{B1} = 50(6) - 50 - 50(4) -39 = \$11

Compute TF_{it}, ΔT_{it} and F_t

$TF_{A1} = 3$, $\Delta T_{A1} = 3-(1+1)+1 = 2$

$TF_{B1} = 6$, $\Delta T_{B1} = 6-(1+2)+1 = 4$

$F_1 = \{ A \}$, since $\Delta Tmin_1 = \min \{2, 4\} = 2$.

c) Check whether the Nover > 0. True (2>0).

c1) Determine which product to produce first using "Small Delta Time and High Benefit First" (SDT-HB). Choose product A since there is only one candidate.

c2) Determine which machine to use. Only one machine is available, so the machine is assigned to the product A, i.e. $r_{A11} = 1$, and setting B_{11} to be 1.

c3) Compute $Nmca_1$. $Nmca_1 = 1 - 1 = 0$. Since cumulative remaining demand is no less than capacity, setting the batch size to be 50. Determine the amount of spoilage, which would incur from this batch. Since this batch will be obtained at the beginning of period 2 and be used up to fulfill the demand for product A, no spoilage from this batch incurs. Hence S_{A2} is zero.

c4) Update the remaining demand for products ($D'_{i,t}$) using algorithm 6.2. The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining demands (units)	$D'_{A,t}$	0	0	0	52	51	50	45
	$D'_{B,t}$	0	0	0	0	20	40	41

- Go back to the beginning of while loop to check whether the condition of ($Nmca_1 > 0$) and ($Flag_1 = 0$) is still held.

No, since $Nmca_1 = 0$. Move out of while loop and increase time period t by 1.

Iteration 2

Step 1: Time period t = 2

Step 1.1: Compute $Nmca_2$. $Nmca_2 = 1 - 1 = 0$.

Step 1.2: Set $Flag_2 = 0$.

Step 1.3: Satisfy the condition of ($Nmca_2 > 0$) and ($Flag_2 = 0$) for while loop

Go inside the while loop.

a) Compute $CDem_{it}$

$CDem_{A2} = 52+51+50+45 = 198$

$CDem_{B2} = 20+40+41 = 101$

b) Both products have $CDem_{it}$ more than the capacity (C) of 50, so the total number of products with $CDem_{it} \geq C$ (Nover) is 2.

Compute the benefit for each product (Ben_{it})

$Ben_{A2} = 50(4.5) - 40 - 50(3)-10 = \$25$

$Ben_{B2} = 50(6) - 50 - 50(4)-24 = \$26$

Compute TF_{it}, ΔT_{it} and F_t

$TF_{A2} = 4$, $\Delta T_{A2} = 4-(2+1)+1 = 2$

$TF_{B2} = 6$, $\Delta T_{B2} = 6-(2+2)+1 = 3$

$F_2 = \{A\}$, since $\Delta Tmin_2 = \min\{2, 3\} = 2$.

c) Check whether the Nover > 0. True (2>0).

c1) Determine which product to produce first using “Small Delta Time and High Benefit First” (SDT-HB). Choose the product A.

c2) Determine which machine to use. Only one machine is available, so the machine is assigned to the product A, i.e. $r_{A12} = 1$, and setting B_{12} to be 1.

c3) Compute $Nmca_2$. $Nmca_2 = 1 - 1 = 0$. Since cumulative remaining demand is no less than capacity, setting the batch size to be 50. Determine the amount of spoilage, which would incur from this batch. As a result, $S_{A3} = 0$.

c4) Update the remaining demand for products ($D'_{i,t}$). The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining demands (units)	$D'_{A,t}$	0	0	0	2	51	50	45
	$D'_{B,t}$	0	0	0	0	20	40	41

- Go back to the beginning of while loop to check whether the condition of ($Nmca_2 > 0$) and ($Flag_2 = 0$) is still held.

  No, since $Nmca_2 = 0$. Move out of while loop and increase time period t by 1.

Iteration 3

Step 1: $t = 3$

Step 1.1: Compute $Nmca_3$. $Nmca_3 = 1 - 0 = 1$.

Step 1.2: Set $Flag_3 = 0$.

Step 1.3: Satisfy the condition of ($Nmca_3 > 0$) and ($Flag_3 = 0$) for while loop

Go inside the while loop.

a) Compute $CDem_{it}$

$CDem_{A3} = 2+51+50+45 = 148$

$CDem_{B3} = 20+40+41 = 101$

b) Both products have $CDem_{it}$ more than the capacity (C) of 50, so the total number of products with $CDem_{it} \geq C$ (Nover) is 2.

Compute the benefit

$Ben_{A3} = 50(4.5) - 40 - 50(3) – 9.6 = \$25.4$

$Ben_{B3} = 50(6) - 50 - 50(4) – 9 = \$41$

Compute TF_{it}, ΔT_{it} and F_t

$TF_{A3} = 5$, $\Delta T_{A3} = 5-(3+1)+1 = 2$

$TF_{B3} = 6$, $\Delta T_{B3} = 6-(3+2)+1 = 2$

$F_3 = \{ A, B \}$, since $\Delta Tmin_3 = \min \{2, 2\} = 2$.

c) Check whether the Nover > 0. True (2>0).

c1) Determine which product to produce first using "Small Delta Time and High Benefit First" (SDT-HB). Choose product B, since it has the higher profit.

c2) Determine which machine to use. Only one machine is available, so the machine is assigned to the product B, i.e. $r_{B13} = 1$, and setting B_{13} and B_{14} to be 1.

c3) Compute $Nmca_3$. $Nmca_3 = 1-1 = 0$. Since cumulative remaining demand is no less than capacity, setting the batch size to be 50. Determine the amount of spoilage, which would incur from this batch. As a result, $S_{B5} = 0$.

c4) Update the remaining demand for products ($D'_{i,t}$). The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining	$D'_{A,t}$	0	0	0	2	51	50	45
demands (units)	$D'_{B,t}$	0	0	0	0	0	10	41

- Go back to the beginning of while loop to check whether the condition of ($Nmca_3 > 0$) and ($Flag_3 = 0$) is still held.

No, since $Nmca_3 = 0$. Move out of while loop and increase time period t by 1.

Iteration 4

Step 1: Time period t = 4

Step 1.1: Compute $Nmca_4$. $Nmca_4 = 1\text{-}1 = 0$

Step 1.2: Set $Flag_4 = 0$.

Step 1.3: Fail to satisfy the condition of ($Nmca_4 > 0$) and ($Flag_4 = 0$) for while loop since $Nmca_4 = 0$. Move out of while loop and increase time period t by 1.

Iteration 5

Step 1: t = 5

Step 1.1: Compute $Nmca_5$. $Nmca_5 = 1\text{-}0=1$.

Step 1.2: Set $Flag_5 = 0$.

Step 1.3: Satisfy the condition of ($Nmca_5 > 0$) and ($Flag_5 = 0$) for while loop

Go inside the while loop.

a) Compute $CDem_{it}$: $CDem_{A5} = 50+45=95$, $CDem_{B5} = 41$

b) Only product A has $CDem_{it}$ more than the capacity (C) of 50, so the total number of products with $CDem_{it} \geq C$ (Nover) is 1. Compute benefit for each product (Ben_{it}). $Ben_{A5} = 50(4.5) - 40 - 50(3) - 0 = \$35$

Compute TF_{it}, ΔT_{it} and F_t

$TF_{A5} = 6$, $\Delta T_{A5} = 6-(5+1)+1 = 1$

$F_t = \{ A \}$ $\Delta Tmin_t = 1$.

c) Check whether the Nover > 0. True (1>0).

c1) Determine which product to produce first using "Small Delta Time and High Benefit First" (SDT-HB). Choose product A.

c2) Determine which machine to use. Only one machine is available, so the machine is assigned to the product A, i.e. r_{A15} = 1, and setting B_{15} to be 1.

c3) Compute $Nmca_3$. $Nmca_5$ = 1-1 = 0. Since cumulative remaining demand is no less than capacity, setting the batch size to be 50. Determine the amount of spoilage, which would incur from this batch. As a result, $S_{A6} = 0$.

c4) Update the remaining demand for products ($D'_{i,t}$). The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining demands (units)	$D'_{A,t}$	0	0	0	2	51	0	45
	$D'_{B,t}$	0	0	0	0	0	10	41

- Go back to the beginning of while loop to check whether the condition of ($Nmca_5 > 0$) and ($Flag_5 = 0$) is still held.

No, since $Nmca_5 = 0$. Move out of while loop and increase time period t by 1.

Iteration 6

Step 1: t = 6

Step 1.1: Compute $Nmca_6$. $Nmca_6 = 1$.

Step 1.2: Set $Flag_6 = 0$

Step 1.3: Satisfy the condition of ($Nmca_6 > 0$) and ($Flag_6= 0$) for while loop

Go inside the while loop.

a) Compute $CDem_{it}$: $CDem_{A6} = 45$, $CDem_{B6} = 0$

b) $CDem_{it}$ of both products is less than the capacity (C) of 50, so the total number of products with $CDem_{it} \geq C$ (Nover) is 0.

c) Check whether the Nover > 0. False

d1) Compute the benefit for each product (Ben_{it}).

$Ben_{A6} = 45(4.5) - 40 - 50(3)-2-1 = 9.5$

$Ben_{B6} = 0$

d2) Compute the total number of product with positive benefit (Nben) Nben =1.

d3) Check whether the Nben > 0. True (1>0).

- Determine which product to produce first using "Small Delta Time and High Benefit First" (SDT-HB). Choose product A, since it is only the candidate.
- Determine which machine to use. Only one machine is available, so the machine is assigned to the product A, i.e. $R_{A16} = 1$, and setting B_{16} to be 1.
- Compute $Nmca_6$. $Nmca_6 = 1 - 1 = 0$. The cumulative remaining demand is less than capacity, and the batch size is discrete, but it is worth releasing this batch. Therefore, the batch size equals 50. Determine the amount of spoilage, which would incur from this batch. This batch will be obtained at the end of horizon (period 7). As we assume that the excess production in the last period is deemed as the ending inventory, no spoilage from this batch incurs. Hence S_{A7} is zero.
- Update the remaining demand for products ($D'_{i,t}$). The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining demands (units)	$D'_{A,t}$	0	0	0	2	51	0	0
	$D'_{B,t}$	0	0	0	0	0	10	41

- Go back to the beginning of while loop to check whether the condition of ($Nmca_6 > 0$) and ($Flag_6 = 0$) is still held.

  No, since $Nmca_6 = 0$. Move out of while loop and increase time period t by 1. The resulting t will be 7, which is greater than T_last (6). Exit the for loop.

At the end of step 1, we obtain the batch release plan of each product over the horizon. In this example, the production plan is to release a batch of 50 units of product A in periods 1, 2, 5 and 6, and to release a batch of 50 units of product B in period 3 as shown in the MPS in Table 6.5. Because steps 2-5 are straightforward, the details of calculation are omitted.

A.6 Implementation of the MLFL-C Heuristic for Solving BPP-SI Problem with Discrete Batch Size

In this section, we illustrate how to use MLFL-C Heuristic to solve the small BPP-SI example with discrete batch size in Chapter VI.

Step 0: Initialization:

Set $D'_{i,t} = D_{i,t} \quad \forall i,t$

Set $B_{j,t} = 0 \quad \forall j,t$

Compute T_last: T_last = 7 – min (1, 2) = 6

Iteration 1

Step 1: Start with t = 1

Step 1.1: Compute $Nmca_1$. $Nmca_1 = 1 - \sum_{j=1}^{1} B_{j1} = 1 - 0 = 1$.

Step 1.2: Set $Flag_1 = 0$.

Step 1.3: Satisfy the condition of ($Nmca_1 > 0$) and ($Flag_1 = 0$) for while loop.

a) Compute $CDem_{it}$

$CDem_{A1} = 50+52+51+50+45 = 248$

$CDem_{B1} = 20+40+41 = 101$

b) Both products have $CDem_{it}$ more than the capacity (C) of 50, so the total number of products with $CDem_{it} \geq C$ (Nover) is 2.

Compute the benefit for each product (Ben_{it})

$Ben_{A1} = 50(4.5) - 40 - 50(3) - 10 = \$25$

$Ben_{B1} = 50(6) - 50 - 50(4) = \$11$

Compute TF_{it}, ΔT_{it} and F_t

$TF_{A1} = 3, \quad \Delta T_{A1} = 3 - (1+1) + 1 = 2$

$TF_{B1} = 6$, $\Delta T_{B1} = 6-(1+2)+1 = 4$

$F_1 = \{ A \}$, since $\Delta Tmin_1 = \min \{2, 4\} = 2$.

c) Check whether the Nover > 0. True (2>0).

c1) Determine which product to produce first using "Small Delta Time and Short Production Time Second" (SDT-SPT). Choose product A since there is only one candidate.

c2) Determine which machine to use. Only one machine is available, so the machine is assigned to the product A, i.e. $r_{A11} = 1$, and setting B_{11} to be 1.

c3) Compute $Nmca_1$. $Nmca_1 = 1 - \sum_{j=1}^{1} B_{j1} = 1 - 1 = 0$. Since cumulative remaining demand is no less than capacity, setting the batch size to be 50. Determine the amount of spoilage, which would incur from this batch. No spoilage from this batch incurs. Hence S_{A2} is zero.

c4) Update the remaining demand for products $(D'_{i,t})$. The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining	$D'_{A,t}$	0	0	0	52	51	50	45
demands (units)	$D'_{B,t}$	0	0	0	0	20	40	41

- Go back to the beginning of while loop to check whether the condition of ($Nmca_1 > 0$) and ($Flag_1 = 0$) is still held.

No, since $Nmca_1 = 0$. Move out of while loop and increase time period t by 1.

Iteration 2

Step 1: Time period t = 2

Step 1.1: Compute $Nmca_2$. $Nmca_2 = 1 - \sum_{j=1}^{1} B_{j2} = 1 - 1 = 0$.

Step 1.2: Set $Flag_2 = 0$.

Step 1.3: Satisfy the condition of ($Nmca_2 > 0$) and ($Flag_2 = 0$) for while loop

Go inside the while loop.

a) Compute $CDem_{it}$ from period $t+AT_i$ to period $t+AT_i+LT_i-1$ for each product:

$CDem_{A2} = 52+51+50+45 = 198$

$CDem_{B2} = 20+40+41 = 101$

b) Both products have $CDem_{it}$ more than the capacity (C) of 50, so the total number of products with $CDem_{it} \geq C$ (Nover) is 2.

Compute the benefit for each product (Ben_{it})

$Ben_{A2} = 50(4.5) - 40 - 50(3)-10 = \$25$

$Ben_{B2} = 50(6) - 50 - 50(4)-24=\$56$

Compute TF_{it}, ΔT_{it} and F_t

$TF_{A2} = 4$, $\Delta T_{A2} = 4-(2+1)+1 = 2$

$TF_{B2} = 6$, $\Delta T_{B2} = 6-(2+2)+1 = 3$

$F_2 = \{ A \}$, since $\Delta Tmin_2 = \min \{2, 3\} = 2$.

c) Check whether the Nover > 0. True (2>0).

c1) Determine which product to produce first using "Small Delta Time and Short Production Time Second" (SDT-SPT). Choose product A since there is only one candidate.

c2) Determine which machine to use. Only one machine is available, so the machine is assigned to the product A, i.e. $r_{A12} = 1$, and setting B_{12} to be 1.

c3) Compute $Nmca_2$. $Nmca_2 = 1 - \sum_{j=1}^{1} B_{j2} = 1 - 1 = 0$. Since cumulative remaining demand is no less than capacity, setting the batch size to be 50. Determine the amount of spoilage, which would incur from this batch. As a result, $S_{A3} = 0$.

c4) Update the remaining demand for products ($D'_{i,t}$). The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining demands (units)	$D'_{A,t}$	0	0	0	2	51	50	45
	$D'_{B,t}$	0	0	0	0	20	40	41

- Go back to the beginning of while loop to check whether the condition of ($Nmca_2 > 0$) and ($Flag_2 = 0$) is still held.

  No, since $Nmca_2 = 0$. Move out of while loop and increase time period t by 1.

Iteration 3

Step 1: $t = 3$

Step 1.1: Compute $Nmca_3$. $Nmca_3 = 1 - \sum_{j=1}^{1} B_{j3} = 1 - 0 = 1$.

Step 1.2: Set $Flag_3 = 0$.

Step 1.3: Satisfy the condition of ($Nmca_3 > 0$) and ($Flag_3 = 0$) for while loop

Go inside the while loop.

a) Compute $CDem_{it}$

$CDem_{A3} = 2+51+50+45 = 148$

$CDem_{B3} = 20+40+41 \quad = 101$

b) Both products have $CDem_{it}$ more than the capacity (C) of 50, so the total number of products with $CDem_{it} \geq C$ (Nover) is 2.

Compute the benefit

$Ben_{A3} = 50(4.5) - 40 - 50(3) - 9.6 = \$25.4$

$Ben_{B3} = 50(6) - 50 - 50(4) - 9 = \$41$

Compute TF_{it}, ΔT_{it} and F_t

$TF_{A3} = 5, \qquad \Delta T_{A3} = 5-(3+1)+1 = 2$

$TF_{B3} = 6, \qquad \Delta T_{B3} = 6-(3+2)+1 = 2$

$F_3 = \{ A, B \}$, since $\Delta Tmin_3 = \min \{2, 2\} = 2$.

c) Check whether the Nover > 0. True (2>0).

c1) Determine which product to produce first using "Small Delta Time and Short Production Time Second" (SDT-SPT). Choose product A since it has shorter production time.

c2) Determine which machine to use. Only one machine is available, so the machine is assigned to the product A, i.e. $r_{A13} = 1$, and setting B_{13} to be 1.

c3) Compute $Nmca_3$. $Nmca_3 = 1-1 = 0$. Since cumulative remaining demand is no less than capacity, setting the batch size to be 50. Determine the amount of spoilage, which would incur from this batch. As a result, $S_{A4} = 0$.

c4) Update the remaining demand for products ($D'_{i,t}$). The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining	$D'_{A,t}$	0	0	0	0	3	50	45
demands (units)	$D'_{B,t}$	0	0	0	0	20	40	41

- Go back to the beginning of while loop to check whether the

condition of ($Nmca_3 > 0$) and ($Flag_3 = 0$) is still held.

No, since $Nmca_3 = 0$. Move out of while loop and increase time period t by 1.

Iteration 4

Step 1: Time period t = 4

Step 1.1: Compute $Nmca_4$. $Nmca_4 = 1\text{-}0 = 1$

Step 1.2: Set $Flag_4 = 0$.

Step 1.3: Satisfy the condition of ($Nmca_4 > 0$) and ($Flag_4 = 0$) for while loop.

Go inside the while loop.

a) Compute $CDem_{it}$

$CDem_{A4} = 3+50+45 = 98$

$CDem_{B4} = 40+41 \quad = 81$

b) Both products have $CDem_{it}$ more than the capacity (C) of 50, so the total number of products with $CDem_{it} \geq C$ (Nover) is 2.

Compute the benefit

$Ben_{A4} = 50(4.5) - 40 - 50(3)-9.4 = \$25.6$

$Ben_{B4} = 50(6) - 50 - 50(4)-3=\$47$

Compute TF_{it}, ΔT_{it} and F_t

$TF_{A4} = 6, \qquad \Delta T_{A4} = 6-(4+1)+1 = 2$

$TF_{B4} = 7, \qquad \Delta T_{B4} = 7-(4+2)+1 = 2$

$F_4 = \{ A, B \}$, since $\Delta Tmin_4 = \min \{2, 2\} = 2$.

c) Check whether the Nover > 0. True (2>0).

c1) Determine which product to produce first using "Small Delta Time and Short Production Time Second" (SDT-SPT). Choose product A since it has shorter production time.

c2) Determine which machine to use. Only one machine is available, so the machine is assigned to the product A, i.e. $r_{A14} = 1$, and setting B_{14} to be 1.

c3) Compute $Nmca_3$. $Nmca_3 = 1\text{-}1 = 0$. Since cumulative remaining demand is no less than capacity, setting the batch size to be 50. Determine the amount of spoilage, which would incur from this batch. As a result, $S_{A5} = 0$.

c4) Update the remaining demand for products ($D'_{i,t}$). The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining	$D'_{A,t}$	0	0	0	0	0	3	45
demands (units)	$D'_{B,t}$	0	0	0	0	20	40	41

- Go back to the beginning of while loop to check whether the condition of ($Nmca_4 > 0$) and ($Flag_4 = 0$) is still held.

No, since $Nmca_4 = 0$. Move out of while loop and increase time period t by 1.

Iteration 5

Step 1: t = 5

Step 1.1: Compute $Nmca_5$. $Nmca_5 = 1\text{-}0=1$.

Step 1.2: Set $Flag_5 = 0$.

Step 1.3: Satisfy the condition of ($Nmca_5 > 0$) and ($Flag_5 = 0$) for while loop

Go inside the while loop.

a) Compute $CDem_{it}$: $CDem_{A5} = 3+45=48$, $CDem_{B5} = 41$

b) Neither product A nor B has $CDem_{it}$ more than the capacity (C) of 50, so the total number of products with $CDem_{it} \geq C$ (Nover) is 0. Compute benefit for each product (Ben_{it}). $Ben_{A5} = 50(4.5) – 40 – 50(3) = \$35$

c) Check whether the Nover > 0. False

d1) Compute the benefit for each product (Ben_{it}).

$Ben_{A5} = 48(4.5) - 40 - 50(3) - 0.8 - 9.8 = 15.4$

$Ben_{B5} = 41(6) - 50 - 50(4) - 4.5 - 2.7 < 0$

d2) Compute the total number of product with positive benefit (Nben) Nben =1.

d3) Check whether the Nben > 0. True (1>0).

- Determine which product to produce first using “Small Delta Time and Short Production Time Second” (SDT-SPT). Choose product A, since it is only the candidate.
- Determine which machine to use. Only one machine is available, so the machine is assigned to the product A, i.e. $r_{A15} = 1$, and setting B_{15} to be 1.
- Compute $Nmca_5$. $Nmca_5 = 1 - 1 = 0$. The cumulative remaining demand is less than capacity, and the batch size is discrete, but it is worth releasing this batch. Therefore, the batch size equals 50. Determine the amount of spoilage, which would incur from this batch. $S_{A6} = 50 – 48 = 2$.
- Update the remaining demand for products ($D'_{i,t}$). The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining demands (units)	$D'_{A,t}$	0	0	0	0	0	0	0
	$D'_{B,t}$	0	0	0	0	20	40	41

- Go back to the beginning of while loop to check whether the condition of ($Nmca_5 > 0$) and ($Flag_5 = 0$) is still held.

  No, since $Nmca_5 = 0$. Move out of while loop and increase time period t by 1.

Iteration 6

Step 1: t = 6

Step 1.1: Compute $Nmca_6$. $Nmca_6 = 1$.

Step 1.2: Set $Flag_6 = 0$

Step 1.3: Satisfy the condition of ($Nmca_6 > 0$) and ($Flag_6 = 0$) for while loop

Go inside the while loop.

a) Compute $CDem_{it}$: $CDem_{A6} = 0$, $CDem_{B6} = 0$

b) Both products have $CDem_{it}$ of 0, so Nover is 0.

c) Check whether the Nover > 0. False

d1) Compute the benefit for each product (Ben_{it}).

$Ben_{A6} = 0$, $Ben_{B6} = 0$

d2) Compute the total number of product with positive benefit (Nben)

Nben =0.

d3) Check whether the Nben > 0. False.

- Set Flag = 1, i.e. the production is unworth.
- Go to the beginning of while loop to check whether the condition of ($Nmca_6 > 0$) and ($Flag_6 = 0$) is still held.

No, since $Flag_6 = 1$. Move out of while loop and increase time period t by 1. The resulting t will be 7, which is greater than T_last (6). Exit the for loop.

At the end of step 1, we obtain the batch release plan of each product over the horizon. In this example, the production plan is to release a batch of 50 units of product A in periods 1, 2, 3, 4 and 5 as shown in the MPS in Table 6.6. The details of calculation from step 2 to step 5 are omitted.

A.7 Implementation of the FOQ Heuristic for Solving BPP-SI Problem with Discrete Batch Size

In this section, we illustrate how to use FOQ Heuristic to solve the small BPP-SI example with discrete batch size in Chapter VI.

Step 0: Initialization:

Set $D'_{i,t} = D_{i,t} \quad \forall i,t$

Set $B_{j,t} = 0 \quad \forall j,t$

Compute T_last: T_last = 7 – min (1, 2) = 6

Iteration 1

Step 1: Start with t = 1

Step 1.1: Compute $Nmca_1$. $Nmca_1 = 1 - \sum_{j=1}^{1} B_{j1} = 1 - 0 = 1$.

Step 1.2: Set $NumZE_1 = 0$

Step 1.3 Both products have nonzero earliness, so $NumZE_1 = 0$

Step 1.4: Fail to satisfy the condition of ($Nmca_1 > 0$) and ($NumZE_1 > 0$).

Go out of while loop and increase time period t by 1.

Iteration 2

Step 1: t = 2

Step 1.1: Compute $Nmca_2$. $Nmca_2 = 1$.

Step 1.2: Set $NumZE_2 = 0$

Step 1.3 Compute $IZE_{A2} = 1$ and $IZE_{B2} = 0$, so $NumZE_2 = 1$

Step 1.4: Satisfy the condition of ($Nmca_2 > 0$) and ($NumZE_2 > 0$).

a) Set Select = 0, Ipdt = 1, PID = 0

Satisfy the condition of (Select = 0) and (Ipdt $\leq$ 2)

- Find the first product with zero earliness and select it

Since only product A is a candidate, Ipdt = 1 and Select = 1

b) If Select = 1

b1) Determine which machine to use. Only one machine is available, so the machine is assigned to the product A, i.e. r_{A12} = 1, and setting B_{12} to be 1.

b2) Compute $Nmca_2$. $Nmca_2$ = 0. Batch size = 50. Determine the amount of spoilage, which would incur from this batch. No spoilage from this batch incurs. Hence S_{A2} is 0.

b3) Update the remaining demand for products ($D'_{i,t}$) using algorithm 6.2. The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining demands (units)	$D'_{A,t}$	0	0	0	52	51	50	45
	$D'_{B,t}$	0	0	0	0	20	40	41

- Go back to the beginning of while loop to check whether the condition of ($Nmca_2 > 0$) and ($NumZE_2 > 0$) is still held.

No, since $Nmca_2 = 0$. Move out of while loop and increase time period t by 1.

Iteration 3

Step 1: t = 3

Step 1.1: Compute $Nmca_3$. $Nmca_3 = 1$.

Step 1.2: Set $NumZE_3 = 0$

Step 1.3 Compute $IZE_{A3} = 1$ and $IZE_{B3} = 1$, so $NumZE_3 = 2$

Step 1.4: Satisfy the condition of ($Nmca_3 > 0$) and ($NumZE_3 > 0$).

a) Set Select = 0, Ipdt = 1, PID = 0

Satisfy the condition of (Select = 0) and (Ipdt $\leq$ 2)

- Find the first product with zero earliness and select it

Select product A, so Ipdt = 1 and Select = 1

b) If Select = 1

b1) Determine which machine to use. Only one machine is available, so the machine is assigned to the product A, i.e. $r_{A13} = 1$, and setting B_{13} to be 1.

b2) Compute $Nmca_3$. $Nmca_3 = 0$. Batch size = 50. Determine the amount of spoilage, which would incur from this batch. No spoilage from this batch incurs. Hence S_{A3} is 0.

b3) Update the remaining demand for products ($D'_{i,t}$) using algorithm 6.1. The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining demands (units)	$D'_{A,t}$	0	0	0	2	51	50	45
	$D'_{B,t}$	0	0	0	0	20	40	41

- Go back to the beginning of while loop to check whether the condition of ($Nmca_3 > 0$) and ($NumZE_3 > 0$) is still held.

No, since $Nmca_3 = 0$. Move out of while loop and increase time period t by 1.

Iteration 4

Step 1: t = 4

Step 1.1: Compute $Nmca_t$. $Nmca_t = 1$.

Step 1.2: Set $NumZE_t = 0$

Step 1.3 Compute $IZE_{At} = 1$ and $IZE_{Bt} = 1$, so $NumZE_t = 2$

Step 1.4: Satisfy the condition of ($Nmca_t > 0$) and ($NumZE_t > 0$).

a) Set Select = 0, Ipdt = 1, PID = 0

Satisfy the condition of (Select = 0) and (Ipdt $\leq$ 2)

- Find the first product with zero earliness and select it

Select product A, so Ipdt = 1 and Select = 1

b) If Select = 1

b1) Determine which machine to use. Only one machine is available, so the machine is assigned to the product A, i.e. r_{A1t} = 1, and setting B_{1t} to be 1.

b2) Compute $Nmca_t$. $Nmca_t$ = 0. Batch size = 50. S_{At} is 0.

b3) Update the remaining demand for products ($D'_{i,t}$) using algorithm 6.2. The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining demands (units)	$D'_{A,t}$	0	0	0	0	3	50	45
	$D'_{B,t}$	0	0	0	0	20	40	41

- Go back to the beginning of while loop to check whether the condition of ($Nmca_t$ > 0) and ($NumZE_t$ > 0) is still held.

No, since $Nmca_t$ = 0. Move out of while loop and increase time period t by 1.

<u>Iteration 5</u>

Step 1: t = 5

Step 1.1: Compute $Nmca_t$. $Nmca_t$ = 1.

Step 1.2: Set $NumZE_t$ = 0

Step 1.3 Compute IZE_{At} = 1 and IZE_{Bt} = 1, so $NumZE_t$ = 2

Step 1.4: Satisfy the condition of ($Nmca_t$ > 0) and ($NumZE_t$ > 0).

a) Set Select = 0, Ipdt = 1, PID = 0

Satisfy the condition of (Select = 0) and (Ipdt $\leq$ 2)

- Find the first product with zero earliness and select it

Select product A, so Ipdt = 1 and Select = 1

b) If Select = 1

b1) Determine which machine to use. Only one machine is available, so the machine is assigned to the product A, i.e. r_{A1t} = 1, and setting B_{1t} to be 1.

b2) Compute $Nmca_t$. $Nmca_t$ = 0. Batch size = 50. S_{At} is 0.

b3) Update the remaining demand for products ($D'_{i,t}$) using algorithm 6.2. The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining	$D'_{A,t}$	0	0	0	0	0	3	45
demands (units)	$D'_{B,t}$	0	0	0	0	20	40	41

- Go back to the beginning of while loop to check whether the condition of ($Nmca_t$ > 0) and ($NumZE_t$ > 0) is still held.

No, since $Nmca_t$ = 0. Move out of while loop and increase time period t by 1.

<u>Iteration 6</u>

Step 1: t = 6

Step 1.1: Compute $Nmca_t$. $Nmca_t$ = 1.

Step 1.2: Set $NumZE_t$ = 0

Step 1.3 Compute IZE_{At} = 1 and IZE_{Bt} = 0, so $NumZE_t$ = 1

Step 1.4: Satisfy the condition of ($Nmca_t$ > 0) and ($NumZE_t$ > 0).

a) Set Select = 0, Ipdt = 1, PID = 0

Satisfy the condition of (Select = 0) and (Ipdt $\leq$ 2)

- Find the first product with zero earliness and select it

Select product A, so Ipdt = 1 and Select = 1

b) If Select = 1

b1) Determine which machine to use. Only one machine is available, so the machine is assigned to the product A, i.e. r_{A1t} = 1, and setting B_{1t} to be 1.

b2) Compute $Nmca_t$. $Nmca_t = 0$. Batch size = 50. S_{At} is 0.

b3) Update the remaining demand for products ($D'_{i,t}$) using algorithm 6.2. The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining demands (units)	$D'_{A,t}$	0	0	0	0	0	0	0
	$D'_{B,t}$	0	0	0	0	20	40	41

- Go back to the beginning of while loop to check whether the condition of ($Nmca_t > 0$) and ($NumZE_t > 0$) is still held.

No, since $Nmca_t = 0$. Move out of while loop and increase time period t by 1. The resulting t will be 7, which is greater than T_last (6). Exit the for loop.

At the end of step 1, we obtain the batch release plan of each product over the horizon. In this example, the production plan is to release a batch of 50 units of product A in periods 2,3,4,5 and 6 in Table 6.7.

A.8 Implementation of the Hybrid Heuristic for Solving BPP-SI Problem with Discrete Batch Size

In this section, we illustrate how to use Hybrid Heuristic to solve the small BPP-SI example with discrete batch size in Chapter VI.

Step 0: Initialization:

Set $D'_{i,t} = D_{i,t} \quad \forall i,t$

Set $B_{j,t} = 0 \quad \forall j,t$

Compute T_last: T_last = 7 – min (1, 2) = 6

<u>Iteration 1</u>

Step 1: Start with t = 1

Step 1.1: Compute $Nmca_t$. $Nmca_t$= 1 - 0 = 1.

Step 1.2: Set $Flag_t = 0$

Step 1.3: Set $NumZE_t = 0$

Step 1.4: Both products have nonzero earliness, so $NumZE_t = 0$

Step 1.5: Fail the condition of ($Nmca_t > 0$) and ($Flag_t = 0$) and ($NumZE_t > 0$).

Go out of while loop and increase time period t by 1.

<u>Iteration 2</u>

Step 1: t = 2

Step 1.1: Compute $Nmca_t$. $Nmca_t$= 1 - 0 = 1.

Step 1.2: Set $Flag_t = 0$

Step 1.3: Set $NumZE_t = 0$

Step 1.4: Compute $IZE_{At} = 1$ and $IZE_{Bt} = 0$, so $NumZE_2 = 1$

Step 1.5: Satisfy the condition of ($Nmca_t > 0$) and ($Flag_t = 0$) and ($NumZE_t > 0$).

a) Find the class for each product

- $IZE_{At} = 1$, $IC_{At} = 1$, so $Class_{At} = 1$

- $IZE_{Bt} = 0$, $IC_{Bt} = 1$, so $Class_{Bt} = 3$

- $NC_1 = 1$, $NC_2 = 0$, $NC_3 = 1$

b) Select the class of product to produce according to the priority rule>

- $cid^* = 1$ since $NC_1 = 1$

c) Compute batch size, amount of unmet demand for each product, and the associated costs and benefit.

Product	Unmet demand (units)	Batch size (units)
A	50	50
B	50	50

Product	Unmet cost ($)	Prod. cost ($)	Setup cost($)	Spoil cost($)	Inv. cost ($)	Benefit ($)
A	225	150	40	0	0	35
B	300	200	50	0	24	26

d) Select the product and assign to a machine

Since $cid^* = 1$ and $Ben_{cid^*,t} > 0$, we release the batch of product A.

d1) Determine which machine to use. Only one machine is available, so the machine is assigned to the product A, i.e. $r_{A1t} = 1$, and setting B_{1t} to be 1.

d2) Compute $Nmca_t$. $Nmca_t = 0$. Set batch size = $batch_A$. Determine the amount of spoilage, which would incur from this batch. No spoilage from this batch incurs. Hence S_{A2} is 0.

d3) Update the remaining demand for products ($D'_{i,t}$) using algorithm 6.2. The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining demands (units)	$D'_{A,t}$	0	0	0	52	51	50	45
	$D'_{B,t}$	0	0	0	0	20	40	41

- Go back to the beginning of while loop to check whether the condition of ($Nmca_t > 0$) and ($Flag_t = 0$) and ($NumZE_t > 0$) is still held. No, since $Nmca_t = 0$. Move out of while loop and increase time period t by 1.

Iteration 3

Step 1: t = 3

Step 1.1: Compute $Nmca_t$. $Nmca_t= 1 - 0 = 1$.

Step 1.2: Set $Flag_t = 0$

Step 1.3: Set $NumZE_t = 0$

Step 1.4: Compute $IZE_{At} = 1$ and $IZE_{Bt} = 1$, so $NumZE_2 = 2$

Step 1.5: Satisfy the condition of ($Nmca_t > 0$) and ($Flag_t = 0$) and ($NumZE_t > 0$).

a) Find the class for each product

- $IZE_{At} = 1$, $IC_{At} = 1$, so $Class_{At} = 1$

- $IZE_{Bt} = 1$, $IC_{Bt} = 1$, so $Class_{Bt} = 1$

- $NC_1= 2$, $NC_2=0$, $NC_3=0$

b) Select the class of product to produce according to the priority rule>

- $cid^* = 1$ since $NC_1= 1$

c) Compute batch size, amount of unmet demand for each product, and the associated costs and benefit.

Product	Unmet demand (units)	Batch size (units)
A	50	50
B	50	50

Product	Unmet cost ($)	Prod. cost ($)	Setup cost($)	Spoil cost($)	Inv. cost ($)	Benefit ($)
A	225	150	40	0	0	35
B	300	200	50	0	9	41

d) Select the product and assign to a machine

From b) and c) $cid^* = 1$ and $Ben_{cid^*,t} > 0$, we select to release the batch of product B, since it has the higher benefit than product A.

d1) Determine which machine to use. Only one machine is available, so the machine is assigned to the product B, i.e. $r_{B1t} = 1$, and setting B_{1t} and $B_{1,t+1}$ to be 1, since it takes 2 periods for producing product B.

d2) Compute $Nmca_t$. $Nmca_t = 0$. Set batch size = $batch_B$. Determine the amount of spoilage, which would incur from this batch. No spoilage from this batch incurs. Hence S_{B3} is 0.

d3) Update the remaining demand for products ($D'_{i,t}$) using algorithm 6.2. The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining	$D'_{A,t}$	0	0	0	52	51	50	45
demands (units)	$D'_{B,t}$	0	0	0	0	0	10	41

- Go back to the beginning of while loop to check whether the condition of ($Nmca_t > 0$) and ($Flag_t = 0$) and ($NumZE_t > 0$) is still held. No, since $Nmca_t = 0$. Move out of while loop and increase time period t by 1.

Iteration 4

Step 1: Start with t = 4

Step 1.1: Compute $Nmca_t$. $Nmca_t = 1 - 1 = 0$.

Step 1.2: Set $Flag_t = 0$

Step 1.3: Set $NumZE_t = 0$

Step 1.4: Both products have nonzero earliness, so $NumZE_t = 0$

Step 1.5: Fail the condition of ($Nmca_t > 0$) and ($Flag_t = 0$) and ($NumZE_t > 0$).

Go out of while loop and increase time period t by 1.

Iteration 5

Step 1: $t = 5$

Step 1.1: Compute $Nmca_t$. $Nmca_t = 1 - 0 = 1$.

Step 1.2: Set $Flag_t = 0$

Step 1.3: Set $NumZE_t = 0$

Step 1.4: Compute $IZE_{At} = 1$ and $IZE_{Bt} = 1$, so $NumZE_2 = 2$

Step 1.5: Satisfy the condition of ($Nmca_t > 0$) and ($Flag_t = 0$) and ($NumZE_t > 0$).

a) Find the class for each product

- $IZE_{At} = 1$, $IC_{At} = 1$, so $Class_{At} = 1$
- $IZE_{Bt} = 1$, $IC_{Bt} = 0$, so $Class_{Bt} = 2$
- $NC_1 = 1$, $NC_2 = 1$, $NC_3 = 0$

b) Select the class of product to produce according to the priority rule>

- $cid^* = 1$ since $NC_1 = 1$

c) Compute batch size, amount of unmet demand for each product, and the associated costs and benefit.

Product	Unmet demand (units)	Batch size (units)
A	41	50
B	50	50

Product	Unmet cost ($)	Prod. cost ($)	Setup cost($)	Spoil cost($)	Inv. cost ($)	Benefit ($)
A	225	150	40	0	0	35
B	246	200	50	4.5	0	-8.5

d) Select the product and assign to a machine

From b) and c) $cid^* = 1$ and $Ben_{cid^*,t} > 0$, we select to release the batch of product A, since the class 1 contains only product A, with positive benefit.

d1) Determine which machine to use. Only one machine is available, so the machine is assigned to the product A, i.e. $r_{A1t} = 1$, and setting B_{At} to be 1, since it takes one period for producing product A.

d2) Compute $Nmca_t$. $Nmca_t = 0$. Set batch size = $batch_A$. Determine the amount of spoilage, which would incur from this batch. No spoilage from this batch incurs. Hence S_{At} is 0.

d3) Update the remaining demand for products ($D'_{i,t}$) using algorithm 6.2. The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining demands (units)	$D'_{A,t}$	0	0	0	52	51	0	45
	$D'_{B,t}$	0	0	0	0	0	10	41

- Go back to the beginning of while loop to check whether the condition of ($Nmca_t > 0$) and ($Flag_t = 0$) and ($NumZE_t > 0$) is still held. No, since $Nmca_t = 0$. Move out of while loop and increase time period t by 1.

<u>Iteration 6</u>

Step 1: $t = 6$

Step 1.1: Compute $Nmca_t$. $Nmca_t = 1 - 0 = 1$.

Step 1.2: Set $Flag_t = 0$

Step 1.3: Set $NumZE_t = 0$

Step 1.4: Compute $IZE_{At} = 1$ and $IZE_{Bt} = 0$, so $NumZE_2 = 1$

Step 1.5: Satisfy the condition of ($Nmca_t > 0$) and ($Flag_t = 0$) and ($NumZE_t > 0$).

a) Find the class for each product

- $IZE_{At} = 1$, $IC_{At} = 0$, so $Class_{At} = 2$
- $IZE_{Bt} = 0$, $IC_{Bt} = 0$, so $Class_{Bt} = 3$
- $NC_1 = 0$, $NC_2 = 1$, $NC_3 = 1$

b) Select the class of product to produce according to the priority rule>

- $cid^* = 2$ since $NC_1 = 0$ and $NC_2 = 0$

c) Compute batch size, amount of unmet demand for each product, and the associated costs and benefit.

Product	Unmet demand (units)	Batch size (units)
A	45	50
B	0	50

Product	Unmet cost ($)	Prod. cost ($)	Setup cost($)	Spoil cost($)	Inv. cost ($)	Benefit ($)
A	202.5	150	40	2	0	10.5
B	0	200	50	25	0	-275

d) Select the product and assign to a machine

From b) and c) $cid^* = 2$ and $Ben_{cid*,t} > 0$, we select to release the batch of product A, since the class 1 contains only product A, with positive benefit.

d1) Determine which machine to use. Only one machine is available, so the machine is assigned to the product A, i.e. $r_{A1t} = 1$, and setting B_{At} to be 1.

d2) Compute $Nmca_t$. $Nmca_t = 0$. Set batch size = $batch_A$. Determine the amount of spoilage, which would incur from this batch. No spoilage from this batch incurs. Hence S_{At} is 0.

d3) Update the remaining demand for products ($D'_{i,t}$)The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining demands (units)	$D'_{A,t}$	0	0	0	52	51	0	0
	$D'_{B,t}$	0	0	0	0	0	10	41

- Go back to the beginning of while loop to check whether the condition of ($Nmca_t > 0$) and ($Flag_t = 0$) and ($NumZE_t > 0$) is still held. No, since $Nmca_t = 0$. Move out of while loop and increase time period t by 1. The resulting t will be 7, which is greater than T_last (6). Exit the for loop.

At the end of step 1, we obtain the batch release plan of each product over the horizon. In this example, the production plan is to release a batch of 50 units of product A in periods 2, 5 and 6, and to release a batch of 50 units of product B in period 3 as shown in the MPS in Table 6.8.

A.9 Implementation of the MLFL-A Heuristic for Solving BPP-SI Problem with Continuous Batch Size

In this section, we illustrate how to use MLFL-A Heuristic to solve the small BPP-SI example with discrete batch size in Chapter VI.

If we apply this heuristic to the problem, the computational result from period 1 to period 5 will be the same as that from the discrete batch size. The batch size during these periods is full capacity since there is at least one product which $CDem_{it}$ is no less than C in such periods. According to the result from discrete batch size, the remaining demand for products ($D'_{i,t}$) at the end of iteration 5 is shown in Table below. Recall that the machine will be free at the beginning of period 6. Notice that in period 6, $CDem_{it}$ for both products is less than C. We next have to identify their benefits are positive.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining demands (units)	$D'_{A,t}$	0	0	50	52	51	0	45
	$D'_{B,t}$	0	0	0	0	0	0	1

After performing the calculation in the 6^{th} iteration, we obtain the following result.

<u>Iteration 6</u>

Step 1: t = 6

Step 1.1: Compute $Nmca_6$. $Nmca_6 = 1\text{-}0 = 1$.

Step 1.2: Set $Flag_6 = 0$

Step 1.3: Satisfy the condition of ($Nmca_6 > 0$) and ($Flag_6 = 0$) for while loop

Go inside the while loop.

a) Compute $CDem_{it}$: $CDem_{A6} = 45$, $CDem_{B6} = 0$

b) $CDem_{it}$ of both products is less than the capacity (C) of 50, so the total number of products with $CDem_{it} \geq C$ (Nover) is 0.

c) Check whether the Nover > 0. False

d1) Compute the benefit for each product (Ben_{it}).

$Ben_{A6} = 45(4.5) – 40 – 50(3) = 12.5$

$Ben_{B6} = 0(6) – 50 – 50(4) < 0$

d2) Compute the total number of product with positive benefit (Nben) Nben =1.

d3) Check whether the Nben > 0. True (1>0).

- Determine which product to produce first using High Benefit First (HBF). Choose product A.
- Determine which machine to use. Only one machine is available, so the machine is assigned to the product A, i.e. $r_{A16} = 1$, and setting B_{16} to be 1.
- Compute $Nmca_6$. $Nmca_6 = 1 - 1 = 0$. The cumulative remaining demand is less than capacity, and the batch size is continuous, but it is worth releasing this batch. Therefore, the batch size equals $CDem_{A6}$ of 45. Determine the amount of spoilage, which would incur from this batch. Due to the no spoilage from in case of continuous batch size, S_{A7} is zero.
- Update the remaining demand for products ($D'_{i,t}$). The following table shows the remaining demand.

		Time Period (t)						
		1	2	3	4	5	6	7
Remaining demands (units)	$D'_{A,t}$	0	0	50	52	51	0	0
	$D'_{B,t}$	0	0	0	0	0	0	1

- Go back to the beginning of while loop to check whether the condition of ($Nmca_6 > 0$) and ($Flag_6 = 0$) is still held.

No, since $Nmca_6 = 0$. Move out of while loop and increase time period t by 1. The resulting t will be 7, which is greater than T_last (6). Exit the for loop.

At the end of step 1, we obtain the batch release plan of each product over the horizon. In this example, the production plan is to release a batch of 50 units of product B in periods 2 and 4, and to release a batch of 50 units of product A in period 1 and a batch of 45 units of product A in period 6 as shown in the MPS in Table 6.10.

REFERENCES

Aggarwal, A. and Park, J. K., 1993, Improved algorithms for economic lot size problems, Operations Research 41, 549-571.

Barany, I., van Roy, T. J., Wolsey, L. A., 1984, Strong formulations for multi-item capacitated lotsizing, Management Science 30, 1255-1261.

Bassett, M. H., Pekny, J. F., Reklaitis, G. V., 1996, Decomposition techniques for the solution of large-scale scheduling problems. AIChE Journal 42, 3373-3387.

Beer Institute, Industry data, http://www.beerinstitute.org/statistics.asp?bid=199 (Accessed March 5, 2006).

Belvaux, G. and Wolsey, L. A., 2000, bc-prod: A specialized branch-and-cut system for lot-sizing problems. Management Science 46, 724-738.

Berretta, R., França, P. M., Armentano, V. A., 2005, Metaheuristic approaches for the multilevel resource-constrained lot-sizing, Asia - Pacific Journal of Operational Research 22, p. 261.

Birewar, D. B., Grossmann, I. E., 1990, Simultaneous synthesis, sizing and scheduling of multiproduct batch plants. Industrial and Engineering Chemistry Research 29, 2242-2251.

Bitran, G. R., Yanasse, H. H., 1982, Computational complexity of the capacitated lot size problem, Management Science 28, 1174-1186.

Brucker P., Gladky A., Hoogeveen J. A., Kovalyov M.Y., Potts C. N., Tautenhahn S.L., Velde van de, 1998, Scheduling a batching machine, Journal of Scheduling 1.

Bruggemann, W. and Jahnke, H., 1997, Remarks on: Some extensions of the discrete lotsizing and scheduling problem. Management Science 43(1), 122.

Bruggemann, W. and Jahnke, H., 2000, The discrete lot-sizing and scheduling problem: complexity and modifcation for batch availability. European Journal of Operational Research 124, 511-528.

Cattrysse, D., Maes, J., 1990, Set partitioning and column generation heuristics for capacitated dynamic lotsizing, European Journal of Operational Research 46, 38-47.

Cattrysse, D., Salomon, M., Kuik, R., van Wassenhove, L.N., 1993, A dual ascent and column generation heuristic for the discrete lotsizing and scheduling problem with setup-times, Management Science 39, 477-486.

Camelot IDPRO AG, Setup Matrix Builder, Characteristic-based and System-Supported Setup Time Generation, http://www.camelot-idpro.de/fileadmin/content_admin/downloads_de/Setup_Matrix_Builder.pdf (Accessed May 1, 2006).

Cerdá, J., Henning, G. P., Grossmann, I. E. , 1997, A mixed-integer linear programming model for short-term scheduling of single-stage multiproduct batch plants with parallel lines. Industrial and Engineering Chemistry Research 36, 1695-1707.

Coulman, G. A., 1989, Algorithm for optimal scheduling and a revised formulation of batch plant design. Industrial and Engineering Chemistry Research 28, 553-556.

Crowcroft, N. S. ,1999, Conjugate vaccines, The Pharmaceutical Journal 263.

De Matteis, J. J., 1971, An economic lot sizing technique, the part period algorithm, IBM Systems Journal 7, 30-38.

Dessouky, M., Kijowski, B, Verma S., 1999, Simultaneous batching and scheduling for chemical processing with earliness and tardiness penalties, Production and Operations Management 8 (4), 433-444.

Diaby, M., Bahl, H. C., Karwan, M. H., Zionts, S., 1992, A Lagrangean relaxation approach for very-large-scale capacitated lot-sizing, Management Science 38, 1329-1340.

Dixon, P., and E.A. Silver, 1981, A heuristic solution procedure for the multi-item, single-level, limited capacity, lot-sizing, Journal of Operations Management 2.

Dobson, G., 1987, The economic lot scheduling problem: achieving feasibility using time varying lot sizes, Operations Research 35, 764-771.

Drexl A, and Kimms A, 1997, Lot sizing and scheduling - survey and extensions, European Journal of Operational Research 99, 221-235.

Dogramaci A., Panayiotopoulos, J. C. and Adam, N. R., 1981, The dynamic lot-sizing problem for multiple items under limited capacity, AIIE: Transactions, 13(4), 294-303.

Dumas, Y., Desrosiers, J., Gelinas, E., and Solomon, M.M., 1995, An optimal algorithm for the Travelling salesman problem with time windows, Operations Research 43(2), 367-371.

Dzielinski, B. P. and Gomory, R. E., 1965, Optimal programming of lot sizes, inventory, and labor allocations, Management Science 11 (9).

Elmaghraby, S. E., 1978, The economic lot scheduling problem (ELSP): review and extensions, Management Science 24, 587-598.

Eppen, G. D., Martin, R. K., 1987, Solving multi-item capacitated lot-sizing problems using variable redefinition, Operations Research 35, 832-848.

Federgruen, A. and Tzur, M. ,1991, A simple forward algorithm to solve general dynamic lot sizing models with *n* periods in $O(n \log n)$ or $O(n)$ time. Management Science 37, 990-925.

Fleischmann, B., 1990, The discrete lot-sizing and scheduling problem. European Journal of Operational Research 44, 337–348.

Fleischmann, B., 1994, The discrete lot-sizing and scheduling problem with sequence dependent set-up costs. European Journal of Operational Research 75, 395–404.

Florian, M., Lenstra, J. K., Rinnooy Kan, A.H.G., 1980, Deterministic production planning: algorithms and complexity, Management Science 26, 669-679.

Floudas, C. A., 1995, Nonlinear and mixed-integer optimization. Oxford University Press.

Floudas, C. A., Lin, X., 2004, Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review, Computers and Chemical Engineering, 28, 2109–2129.

Gallego, G., Joneja, D., 1994, Economic lot scheduling problem with raw material considerations, Operations Research 42, 92-101.

Glover, E., 1975, Improved linear integer programming formulations of nonlinear integer problems. Management Science 22, 455–460.

Gopalakrishnan M., K. Ding, J-M. Bourjolly, S. Mohan, 2001, A tabu search heuristic for the capacitated lot-sizing problem with set-up carryover, Management Science 47, p. 851

Gunther, H. O., 1987, Planning lot sizes and capacity requirements in a single stage production system, European Journal of Operational Research 31 (2).

Hasse K, 1994, Lot-sizing and Scheduling for production planning, Lecture notes in economics and mathematical systems 408.

Henahan, S., 1997, DNA Vaccine OutLook, The national health museum.

Hsu, W.L., 1983, On the general feasibility test of scheduling lot sizes for several items on one machine, Management Science 29, 93-105.

Hui, C., Gupta, A., van der Meulen, H. A. J., 2000, A novel MIP formulation for short-term scheduling of multi-stage multi-product batch plants with sequence-dependent constraints. Computers and Chemical Engineering 24, 2705–2717.

Jans, R., Degraeve, Z., 2004, An industrial extension of the discrete lot-sizing and scheduling problem, IIE Transactions 36 (1), 47-58

Jordan, C., 1996, Batching and scheduling: models and methods for several problem classes, Lecture notes in economics and mathematical systems 437.

Karmarkar, Uday S., 1985, The Deterministic Dynamic Product Cycling Problem, Operations Research 33 (2), 326-345.

Kondili, E., Pantelides, C. C. and Sargent, R. W. H., 1993, A general algorithm for short-term scheduling of batch operations I. MIP formulation. Computers and Chemical Engineering 17, 211-227.

Kimms A., 1999, A genetic algorithm for multi-level, multi-machine lot sizing and scheduling, Computers & Operations Research 26 (8), p. 829

Ku, H., Karimi, L. A., 1988, Scheduling in serial multiproduct batch processes with finite interstage storage: A mixed integer linear programming formulation. Industrial and Engineering Chemistry Research 27, 1840–1848.

Lawrence company, 2005, American beer month offers occasion to reflect on beer's joyous role in our lives, http://www.lawrence.com/news/2005/jul/12/, (Accessed March 12, 2006).

Linko M., Haikara A., Ritala A., Penttila M., 1998, Recent advances in the malting and brewing industry, Journal of Biotechnology 65, 85-98.

Manne, A. S., 1958, Programming of economic lot sizes, Management Science 4, 115-135.

Maes, J., McClain, J. O., van Wassenhove, L. N., 1991, Multilevel, capacitated lotsizing complexity and LP-based heuristics, European Journal of Operational Research 53, 131-148.

Maxwell, W. L., 1964, The scheduling of economic lot sizes, Naval Research Logistics Quaterly 11, 89-124.

McGraw, K. E., Dessouky, M. M., 2001, Sequence-Dependent Batch Chemical Scheduling with Earliness and Tardiness Penalties, International Journal of Production Research 39, 3085-3107.

Méndez, C. A., Henning, G. P., Cerdá, J., 2000, Optimal scheduling of batch plants satisfying multiple product orders with different due-dates. Computers and Chemical Engineering 24, 2223–2245.

Miller, A., Nemhauser, G., Savelsbergh, M., 2000, Solving multi-item capacitated lot-sizing problems with setup times by branch-and-cut, CORE Discussion paper 2000/39.

Mockus, L., Reklaitis, G. V., 1999, Continuous time representation approach to batch and continuous process scheduling, Part 1 MINLP formulation. Industrial and Engineering Chemistry Research 38, 197–203.

Moody B., 2002, Eliminating contamination in CO2 incubators through effective cleaning, American Biotechnology Laboratory.

Moon, S., Park, S., Lee, W., 1996, New MIP models for scheduling of multiproduct batch plants under zero-wait policy. Industrial and Engineering Chemistry Research 35, 3458–3469.

Nahmias, S., 1982, Perishable Inventory Theory: A Review, Operations Research 30 (4), 680-708.

Nahmias, S., 2005, Production and Operations Analysis, McGraw Hill / Irwin, 5th Ed.

Pochet, Y., Wolsey, L. A., 1991, Solving multi-item lot-sizing problems using strong cutting planes, Management Science 37, 53-67.

Orçun, S., Altinel, I. K., Hortacsu, Ö., 2001, General continuous time models for production planning and scheduling of batch processing plants: Mixed integer linear program formulations and computational issues. Computers and Chemical Engineering 25, 371–389.

Patsidou, E. P. and Kantor J. C, 1991, Scheduling of a multipurpose batch plant using a graphically derived mixed-integer linear program model, Industrial and Engineering Chemistry Research 30.

Pekny, J. F., Miller, D. L. and McRae, G. J., 1990, An exact parallel algorithm for scheduling when production costs depend on consecutive system states. Computers and Chemical Engineering 14, 1009- 1023.

Plotkin, S. A., Mortimer E. A., 1994, Vaccines, W.B. Saunders Company, 2nd ed.

Potts, and Kovalyov, 2000, Scheduling with batching: a review, European Journal of Operational Research 120, 228–249.

Quadt, D., 2004, Lot-sizing and scheduling for flexible flow lines, Lecture notes in economics and mathematical systems 546.

Rogers, J., 1958, A computational approach to the economic lot scheduling problem, Management Science 4, 264-291.

Sahinidis, N. V., Grossmann, I. E., 1991, Reformulation of multiperiod MIP models for planning and scheduling of chemical processes. Computers & Chemical Engineering 15, 255-272.

Salomon, M., Kroon, L. G., Kuik, R., and van Wassenhove, L.N., 1991, Some extensions of the discrete lot sizing and scheduling problem, Management Science 37, 801-812.

Salomon, Marc, Solomon, Marius M, Van Wassenhove, Luk N, Dumas, Yvan, Dauzere-Peres, Stephan, 1997, Solving the discrete lotsizing and scheduling problem with sequence dependent set-up costs and set-up times using the Travelling Salesman Problem with time windows, European Journal of Operational Research 100 (3), 494-513.

Silver, S., Pyke, P., and Peterson, P., 1998, Inventory Management and Production Planning and Scheduling, John Wiley & Sons, 3rd edition.

Silver, E. A., and Meal, H.C. 1973. A heuristic for selecting lot size quantities for the case of deterministic time varying demand rate and discrete opportunity for replenishment, Production and Inventory Management 14, 64-77.

Sox, R. C., and Gao, Y., 1999, The capacitated lot sizing problem with setup carryover, IIE Transactions 31, 173.

Suerie, C., Stadtler, H., 2003, The capacitated lot-sizing problem with linked lot sizes, Management Science 49, 1039-1054.

Tempelmeier, H., Derstroff, M., 1996, A Lagrangean-based heuristic for dynamic multi-level multi-item constrained lot sizing with setup times, Management Science 42.

The American Academy of Microbiology, 2005, Vaccine development: current status and future needs, http://www.asm.org/ASM/files/ccLibraryFiles/Filename/000000001840/VaccineDev.pdf (Accessed May 1, 2006).

The Beverage Testing Institute, http://www.tastings.com/bti/about_bti.html (Accessed March 12, 2006).

The Brewers Association, http://www.beertown.org/craftbrewing/statistics.html (Accessed March 12, 2006).

The World Health Organization (WHO), Technical report series: vaccines, http://www.who.int/biologicals/publications/trs/areas/vaccines (Accessed May 2, 2006).

Thizy, J. M and Van Wassenhove, L. N., 1985, Lagrangean relaxation for the multi-item capacitated lot-sizing problem: a heuristic Implementation, IIE Trans 17, 308-313

Thizy, J. M, 1991, Analysis of Lagrangian decomposition for the multi-item capacitated lot-sizing problem, INFORM 29 (4).

Trigiero W. W., L. J. Thomas, and J. O. McClain, 1989, Capacitated lot-sizing with setup times, Management Science 35.

Van Hoesel, S. and Kolen, A., 1994, A linear description of the discrete lot-sizing and scheduling problem. European Journal of Operational Research 75, 342–353.

Van Wassenhove, L.N. and Vanderhenst, P., 1983, Planning production in a bottleneck department, European Journal of Operational Research 12, 127-137.

Virkajarvi, I., 2000, Feasibility of continuous main fermentation of beer using immobilized yeast, VTT publications 430.

Voudouris, V. T., Grossmann, I. E., 1996, MIP model for scheduling and design of a special class of multi-purpose batch plants. Computers and Chemical Engineering, 20, 1335–1360.

Wagelmans, A., van Hoesel, S., and Kolen, A.,1992, Economic lot sizing:An $O(n \log n)$ algorithm that scenarios in linear time in the Wagner-Whitin case. Operations Research 40.

Wagner, H. M., Whitin, T.M., 1958, Dynamic version of the economic lot size model, Management Science 5, 89-96.

Walsh, G., 2003, Biopharmaceuticals: Biochemistry and Biotechnology, Wiley, 2nd ed.

Webster, S., 1999, Remarks on: Some extensions of the discrete lotsizing and scheduling problem. Management Science 45, 768–769.

Webster, S.T., Baker, K. R., 1995, Scheduling groups of jobs on a single machine Operations Research 43 (4).

Yee, K. L., Shah, N., 1998, Improving the efficiency of discrete time scheduling formulation. Computers and Chemical Engineering 22, S403–S410.

Zipkin, E., 1991, Computing optimal lot sizes in the economic lot scheduling problem, Operations Research 39, 56-63.